Rend. Sem. Mat. Univ. Pol. Torino
Vol. xx, x (xxxx)

S. De Marchi*

ON OPTIMAL CENTER LOCATIONS FOR RADIAL BASIS FUNCTION INTERPOLATION: COMPUTATIONAL ASPECTS

Abstract. The problem of choosing "good" nodes is a central one in polynomial interpolation. Made curious from this problem, in this work we present some results concerning the computation of optimal point sets for interpolation by radial basis functions. Two algorithms for the construction of near-optimal set of points are considered. The first, that depends on the radial function, compute optimal point by adding one of the maxima of the *power function* with respect to the preceding set. The second, which is independent of the radial function, is shown to generate near-optimal sets which correspond to *Leja extremal points.* Both algorithms produce point sets almost similar, in the sense of their mutual *separation distances.* We then compare the interpolation errors and the growth of the Lebesgue constants for both point sets.

1. Introduction

First some introductory material and definitions concerning the interpolation problem with radial basis functions. Take a set $X = \{x_1, \ldots, x_N\} \subseteq \Omega \subseteq \mathbb{R}^d$ of N distinct points coming from a compact subset Ω of $\mathbb{R}^d$. The points $\{x_i\}$ are usually referred as the *data sites* and the set X as the *data set.* Suppose further that N data values $f_1, \ldots, f_N$ should be interpolated at the data sites. Fix then a basis function $\phi : [0, \infty) \to \mathbb{R}$, a simple way to define an interpolant $s_{f,X}$ to f at X is by linear combinations of the form

$$s_{f,X}(x) = \sum_{j=1}^{N} \alpha_j \phi(\|x - x_j\|) \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm, and the coefficients $\{\alpha_j\}$ are uniquely determined by the interpolation conditions

$$s_{f,X}(x_i) = f_i, \quad i = 1, \ldots, N \tag{2}$$

*This work has been done with the support of ex-60% funds of the University of Verona, year 2002.

if the interpolation matrix $A_{\phi,X} := (\phi(\|x_i - x_j\|)_{1\le i,j\le N}$ is invertible. Furthermore, for various reasons it is sometimes necessary to add the space $\mathbb{P}_m^d$ of polynomials of degree $\le m$ in $\mathbb{R}^d$ to the interpolanting functions. Interpolation is then uniquely possible with the further requirement: if $p \in \mathbb{P}_m^d$ satisfies

$$p(x_i) = 0, \text{ for all } x_i \in X \;\Rightarrow\; p = 0$$

and if ϕ is *conditionally positive definite* (shortly CPD) of order m on Ω (cf. e.g. [16]) . If $A_{\phi,X}$ is *positive definite* $\forall X \subseteq \Omega$, then ϕ is said *positive definite* (shortly PD), that is conditionally positive definite of *order m=0.* Instead of ϕ, we can consider the symmetric kernel function $\Phi(x, y) = \phi(\|x - y\|)$, so that $\Phi : \Omega \times \Omega \to \mathbb{R}$, which is the notation used later on in the paper.

In the paper we mainly focus to the case of *positive definiteness*, since every *CPD kernel* has an associated *normalized PD* kernel (cf. e.g. [2, 17]).

The problem of finding good interpolation points for RBF interpolations has been addressed only recently (cf. [3, 4, 8]). In particular, in [4] the authors showed how difficult is the problem just in the one dimensional setting because one has to globally minimize a highly nonlinear function of Nd unknowns which is usually a *hard* problem.

In our previous paper [7] we have already discussed the problem of finding good or near-optimal interpolation points for radial basis function interpolation essentially by minimizing the *power function* associated to the symmetric kernel Φ. The main result there was that those points are *asymptotically uniformly distributed* in the Euclidean norm. That is why we called them *near-optimal points.*

The paper is organized as follows. In section 2 we essentially describe what we consider *near-optimal points* for radial basis function interpolation and we introduce the tools we shall use in the rest of the paper. In section 3 after presenting two algorithms for computing near-optimal points, one depending on Φ and one independent, i.e. *data-independent*, we investigate on some computational aspects and consequences related to the problem presenting in particular for the dimension $d = 2$, the connection between these near-optimal points and *Leja extremal sequences.* In section 4 we present numerical results: in particular we show the interpolation errors when interpolants are built on near-optimal point sets, and the corresponding Lebesgue constants. In section 5 we conclude noticing that the most reliable near-optimal points are the ones connected to the proper Φ even if the data-independent ones are proved to be competitive.

2. Interpolation error, power function and Lebesgue constant

Given $\Phi : \Omega \times \Omega \to \mathbb{R}$, a positive definite kernel, the recovery of functions from function values $f(x_j)$ on the set $X = \{x_1, ..., x_N\} \subset \Omega$ of N different data sites, can be done via interpolants of the form

$$s_{f,X} = \sum_{j=1}^{N} \alpha_j \Phi(\cdot, x_j) \ . \tag{3}$$

This interpolant, as in classical polynomial interpolation, can also be written in terms of *cardinal functions* $u_j \in V_X = \text{span}\{\Phi(\cdot, x) : x \in X\}$ such that $u_j(x_k) = \delta_{j,k}$. Then, the interpolant (3) takes the usual Lagrangian form

$$s_{f,X} = \sum_{j=1}^{N} f(x_j) u_j . \tag{4}$$

It is well-known that local error estimates for interpolation by radial basis functions have the form (cf. e.g. [15])

$$|f(x) - s_{f,X}(x)| \le \kappa \ P_{\Phi,X}(x) \tag{5}$$

with κ a positive constant depending only on f and $P_{\Phi,X}$ being the *power function* that takes the explicit form

$$P^2_{\Phi,X}(x) = \Phi(x,x) - 2\sum_{j=1}^{N} u_j(x)\Phi(x,x_j) + \sum_{j,k=1}^{N} u_j(x)u_k(x)\Phi(x_j,x_k).$$

Moreover, letting $\mathbf{u} = (-1, u_1(x), \ldots , u_N(x))$ we have the alternative representation

$$P^2_{\Phi,X}(x) = \mathbf{u}\, A_{\Phi,Y}\, \mathbf{u}^T \ , \tag{6}$$

as a quadratic form, where $Y = X \cup \{x\}$ and $A_{\Phi,Y}$ is the interpolation matrix corresponding to the set Y. This representation says immediately that the power function is non-negative since the vector $\mathbf{u}$ annihilates all polynomials $\mathbb{P}^d_m$ due to the polynomial reproduction property.

For *positive definite* kernels, given the set X where the numbering of its points is fixed, for a second ordered set $Y = \{y_1, ..., y_N\}$ we consider the matrix $A_{\Phi,X}(y_1, ..., y_N) = (\Phi(y_i, x_j))_{1\le i,j\le N}$. We note that this matrix is symmetric and has determinant that is independent of the order of the points in X. Moreover, since Φ is positive definite, the matrix is positive definite and has positive determinant that we denote by $\det_{\Phi,X}(y_1, ..., y_N) = \det\left(\Phi(y_i, x_j)\right)_{1\le i,j\le N}$. Thus, the *cardinal functions* have the useful representation

$$u_k(x) = \frac{\det_{\Phi,X}(x_1, ..., x_{k-1}, x, x_{k+1}, ..., x_N)}{\det_{\Phi,X}(x_1, ..., x_N)} \ , \tag{7}$$

which reminds the determinantal form of the elementary Lagrange polynomials in polynomial interpolation. Moreover, from the representations (6) and (7), the power function can also be rewritten as

$$P^2_{\Phi,X}(x) = \frac{\det_{\Phi,Y}(x, x_1, ..., x_N)}{\det_{\Phi,X}(x_1, ..., x_N)} \ . \tag{8}$$

In other words, the power function is nothing but the norm of the pointwise error functional, and it can be numerically evaluated from the Lagrange basis.

Typically, error estimates and convergence rates lead to the problem of bounding the power function in terms of the *fill distance*,

$$h_{X,\Omega} = \sup_{x\in\Omega} \min_{x_j\in X} \|x - x_j\|_2 .$$

We will not discuss the details here: the interested reader can refer to [19]. Instead, we remark that this minimization property has another consequence. Letting X and Y the point sets above defined, then the associated power functions must necessarily satisfy

$$P^2_{\Phi,X}(x) \ge P^2_{\Phi,Y}(x), \qquad x \in \Omega,$$

due to the *maximality property* of the power function and the fact that the $P_{\Phi,X}$ vanishes only at the points of X (cf. [15, §4]) and this inequality holds pointwise and everywhere in Ω. The above inequality will be an important ingredient for the Algorithm 1 to be presented in the next section.

Also the *separation distance*

$$q_X = \frac{1}{2} \min_{\substack{x_i,\, x_j \,\in X \\ x_i \,\ne\, x_j}} \|x_i - x_j\| ,$$

plays a role in finding good points for radial basis function interpolation. In fact, in [8], the author studied point sets $X \subset \Omega$ which maximize the *uniformity*

$$\rho_{X,\Omega} = \frac{q_X}{h_{X,\Omega}} = \sup_{Y\in\mathcal{X}_\Omega} \rho_{Y,\Omega} \ ,$$

among all point sets $Y \in \mathcal{X}_\Omega$, $\mathcal{X}_\Omega$ consisting of Voronoi vertices used to decompose $\mathbb{R}^d$ into Voronoi tiles. The result there was that point sets that optimally balance $h_{X,\Omega}$ against q_X, are optimally distributed in the domain Ω.

Finally, our last tool is the *Lebesgue constant.* As in the (univariate) polynomial case, from the representation (4) we consider the Lebesgue function $\lambda_N(x) := \sum_{j=1}^N |u_j(x)|$. Its maximum value,

$$\Lambda_N := \max_{x\in\Omega} \lambda_N(x) = \max_{x\in\Omega} \sum_{j=1}^N |u_j(x)| \ , \tag{9}$$

is referred to as the associated Lebesgue constant and gives the norm of the interpolating projector $\mathcal{P}_n : \mathcal{C}(\Omega) \to V_\Omega$, with $V_\Omega = \mathrm{span}\{\Phi(\cdot, x) : x \in \Omega\}$, both spaces equipped with the sup-norm. As well-known in the polynomial case, optimal points are not known explicitly, therefore in applications we can restrict to *near-optimal* points, that is, roughly speaking, points whose Lebesgue constant grows asymptotically like the optimal one. Therefore, near-optimal points should be found among the ones that minimize Λ_N. In the framework of interpolation by polynomials, points that minimize the Lebesgue constant by maximizing the *Vandermonde determinant*, are known as *Fekete points.* Fekete points are well-known and widely studied for polynomial interpolation also in the multi-dimensional setting. For radial basis functions, only recently and only in the univariate case there were some attempts to find Fekete-like points [4]. The main conclusion of that paper was that, surprisingly w.r.t. the polynomial case in which Fekete points have the arccosine distribution, *optimal points for radial basis function interpolation are asymptotically equidistributed.* Actually, a similar conclusion for 2-dimensional domains was also obtained in the paper [8]. Iske considered perturbations of the data sites in order to improve the performance of the interpolation process, showing that good points realize a balance between the quantities q_X and $h_{X,\Omega}$. Moreover, the same author in [9] has shown that the Lebesgue constant Λ_N for interpolation by polyharmonic splines is indeed the *condition number* w.r.t. the sup-norm of the interpolation operator and that this constant is invariant under uniform scalings, rotations and translations of the domain.

On the basis of these arguments, using the representation by cardinal functions u_k of the interpolant $s_{f,X}$, we can try to minimize the Lebesgue constant by maximizing the denominator of each function u_k in (7). Unfortunately these Vandermonde-like matrices, which depend on Φ, are not always well-conditioned.

Hence, to find near-optimal points for radial basis function interpolation we can proceed along the following lines:

- by minimizing the power function, which depends on Φ, in order to minimize the error in (5);

- by finding a representation of the u_k by well-conditioned matrices (for instance using some kind of stable orthogonal expansions) and maximizing the corresponding Vandermonde matrix, like for Fekete points, in order to minimize the Lebesgue constant of the interpolating operator.

In this paper we have explored the first instance and in the next section we present two methods that allow to compute near-optimal interpolation points: the first minimizes the power function associated to the kernel Φ; the second, based on geometric considerations, is completely independent on Φ and related to *Leja extremal sequences.*

3. On computing near-optimal point locations and Leja sequences

In the recent paper [7] we presented a numerical method that produces well–distributed point sets based on a *greedy algorithm* that generates larger and larger point sets by adding at each step one of the point where the power function attains its maxima with respect to the preceding set. The algorithm steps are as follows.

Algorithm 1

1. Initial step: $X_1 = \{x_1\}$ for some $x_1 \in \Omega$ arbitrary chosen.
2. Iterative step:

$$X_j := X_{j-1} \cup \{x_j\} \text{ with } P_{\Phi,X_{j-1}}(x_j) = \|P_{\Phi,X_{j-1}}\|_{L_\infty(\Omega)},\ j \geq 2. \tag{10}$$

Note that practically, we maximized over some very large discrete set $X \subset \Omega$ instead of maximizing on Ω. Letting $P_j := P_{\Phi,X_j}$, this algorithm converges the sense that $\lim_{j\to\infty} \|P_j\|_{L_\infty(\Omega)} = 0$. In fact, since the point x_{j+1} is such that $P_j(x_{j+1}) = \|P_j\|_{L_\infty(\Omega)}$ and since $X_j \subseteq X_{j+1}$, we have $P_j(x) \geq P_{j+1}(x) \geq 0$ for all $x \in \Omega$.

The convergence and the speed of convergence of the Algorithm 1 are stated in the following Theorem.

THEOREM 1. (cf. [7, §4]) *Suppose $\Omega \subseteq \mathbb{R}^d$ is compact and satisfies an interior cone condition. Suppose further that $\Phi \in \mathcal{C}^2(\Omega_1 \times \Omega_1)$ is a positive definite kernel defined on a convex and compact region $\Omega_1 \supseteq \Omega$. Then, the greedy algorithm defined in (10) converges at least like*

$$\|P_j\|_{L_\infty(\Omega)} \leq C\, j^{-1/d}$$

with a constant $C > 0$.

Remarks. The Theorem holds for positive definite kernels: this is not a big restriction since, as already pointed out, every CPD kernel has an associated NPD kernel (cf. Introduction). We also observe that the positive constant C is independent of j and that the power function depress to zero quite slowly, as will appear clearer from Examples.

3.1. A geometric greedy method

From experiments we have noted that Algorithm 1, that minimizes the power function $P_{\Phi,X}$, practically fills the currently largest hole in the data by placing a new data point close to the center of that hole and as a surprise, *independently* of the function Φ. Therefore, this observation suggested a new algorithm that we termed *geometric greedy algorithm* since the construction of *optimal* points is simply based on geometric considerations.

Algorithm 2

1. Let Ω be a compact set in $\mathbb{R}^d$, and consider $X_0 = \{x_0\}$ where x_0 belongs to the boundary of Ω.

2. If $X_n \subset \Omega$ is finite and consisting of n points, choose $x_{n+1} \in \Omega \setminus X_n$ so that its distance to X_n is maximal. Thus, $X_{n+1} := X_j \cup \{x_{n+1}\}$.

Remarks. As before, for numerical purposes we should consider a discretization of Ω that is a finite set, say Ω_N, with *cardinality* N. Then, each step of the algorithm can be carried out in $\mathcal{O}(N)$ operations, since for each $x \in \Omega_N \setminus X_n$ we should compute the distance to its nearest neighbor within X_n. To update this array of length N, it requires firstly computing the $N-n$ values $\|x - x_i\|_2, \ \ i = 1, ..., n$ and then taking the componentwise minimum within the i-th array of distances. The next point x_{n+1} is then easily found by picking the maximum of the array of the minima.

Defining the separation distance for points in X_n by

$$q_n := \frac{1}{2} \min_{\substack{x,\, y \,\in\, X_n \\ x \,\neq\, y}} \|x - y\|_2$$

and the corresponding fill distance

$$h_n := \max_{x \in \Omega} \min_{y \in X_n} \|x - y\|_2 = \min_{y \in X_n} \|x_{n+1} - y\|_2 = h_{X_n, \Omega} \ .$$

Proposition 1. *Algorithm 2 produces point sets which are* **quasi-uniform** *in the Euclidean distance, that is*

$$h_n \geq q_n \geq \frac{1}{2} h_{n-1} \geq \frac{1}{2} h_n \ , \ \ \forall n \geq 2.$$

Proof. The left–hand and right–hand sides are obvious. The remaining inequalities can be settled by induction. Indeed, for X_2 we have

$$q_2 = \frac{1}{2} \|x_2 - x_1\|_2 = \frac{1}{2} \min_{y \in X_1} \|x_2 - y\|_2 = \frac{1}{2} h_1.$$

Assuming that $q_n \geq \frac{1}{2} h_{n-1}$, then

$$q_{n+1} = \min \left\{ q_n, \frac{1}{2} \min_{x \in X_j} \|x_{n+1} - x\|_2 \right\} = \min \left\{ q_n, \frac{1}{2} h_n \right\} \ ,$$

we get $q_{n+1} \geq \min \left\{ \frac{1}{2} h_{n-1}, \frac{1}{2} h_n \right\} \geq \frac{1}{2} h_n$. $\square$

Remarks.

- The above Algorithm 2 turns out to work quite well when it comes to finding subsets of Ω of cardinality n with *small* fill distance $h_{X,\Omega}$ and *large* separation distance q_X.
- The construction technique proposed in the Algorithm 2 is *independent of the Euclidean metric*. In fact, the proof does not depend on the fact that q_n and h_n are expressed by using the Euclidean metric. Hence, if μ is any metric on Ω, the Algorithm 2 can be used to compute points asymptotically equidistributed in the metric μ.

3.2. Leja sequences

Leja extremal sequences were introduced by F. Leja in his interesting paper (cf. [10]) and recently have attracted the attention of researchers for their important properties and applications (cf. e.g. [13, 1, 6]).

DEFINITION 1. Let λ_1 be arbitrarily chosen in $[a,b]$. The points $\lambda_s \in [a,b]$, $s = 2, ..., N$, such that

$$\prod_{k=1}^{s-1} |\lambda_s - \lambda_k| = \max_{x\in[a,b]} \prod_{k=1}^{s-1} |x - \lambda_k| \quad . \tag{11}$$

are called a *Leja sequence* for the interval $[a,b]$ (cf. [10]).

We recall that Leja points, in the one-dimensional case, are computationally effective for polynomial interpolation in Newton form since they provide an increasing sequence of points and they stabilize the computation of divided differences. Moreover, they can be extracted from a discretization of $[a,b]$ in a fast way (the so-called *fast Leja points*) and, like Chebyshev points, Fekete points and zeros of Jacobi orthogonal polynomials, they have the *arccosine distribution* (cf. [13, 1]).

Unfortunately the multivariate equivalent of Leja points are not yet completely explored, as it is the case for the study of near-optimal points for multivariate interpolation (cf. [14, 6, 5]). For $d = 2$ something has been done.

DEFINITION 2. *Let Ω be a compact subset of $\mathbb{C} \approx \mathbb{R}^2$ and $w : E \to \mathbb{R}_+$ a real positive function on Ω called* weight function. *Let $z_0 \in \Omega$ be such that*

$$w(z_0)\|z_0\| = \max_{z\in E} w(z)\|z\| \,, \tag{12}$$

and

$$w(z_n) \prod_{k=0}^{n-1} \|z_n - z_k\| = \max_{z\in E} w(z) \prod_{k=0}^{n-1} \|z - z_k\| \,,\ z_n \in \Omega \,. \tag{13}$$

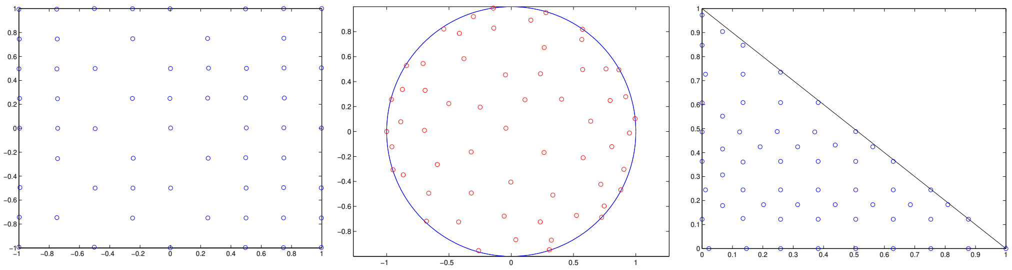

Figure 1.1: 60 Leja points on the square, the unit circle and the right triangle all discretized by 60^3 random points, computed by using (14)

where $\|\cdot\|$ *is any norm of* $\mathbb{R}^2$ *and* $z = x + iy$, $z = (x, y)$ *and* $z_n = (x_n, y_n)$, $n = 1, 2, \ldots$. *The sequence* $\{z_n\}$ *not-unique that satisfies (12) and (13) is called a* **sequence of Leja points** *for* Ω.

The distribution of Leja points so defined, depends on the choice of the weight function w. Indeed, when $w \equiv 1$, for the maximum principle of analytic functions Leja points distribute only on the boundary of Ω while for $w \neq 1$ they lye also in the interior (cf. [14] and for more examples see [6]). A conceptually similar construction of Leja points which is *independent of the weight* w, was suggested by L. Bos (private communication to the author). The idea behind the construction is simple: *find a sequence of points that maximize a function of distances from already computed points.* The proposed distance was simply the *Euclidean distance.*

Definition 3. *Let* Ω_N *be a discretization of a compact domain* $\Omega \subset \mathbb{C} \equiv \mathbb{R}^2$ *and let* z_0 *be arbitrarily chosen in* Ω_N. *The points* z_n, $n = 1, 2, \ldots$.

$$z_n := \max_{z \in \Omega_N \setminus \{z_0, \ldots, z_{n-1}\}} \min_{0 \le k \le n-1} \|z - z_k\|_2 \ . \tag{14}$$

are a set of **Leja points** *for* Ω_N.

In Figure 1.1 we show 60 Leja points on three classical domains computed by means of (14).

Moreover, supported by numerical experiments, L. Bos proposed that the following claim should be true.

Claim 1. If $z_0 = \max_{z \in \Omega} \|z\|_2$, then the Leja points defined by (14) are asymptotically equidistributed w.r.t. the Euclidean metric.

Remarks. The previous Definition 3 and the successive claim, firstly stated in the framework of Leja sequences, reveal the connection with near-optimal points computed by Algorithm 2. From Proposition 1, we now know that the points constructed by (14) are indeed the data–independent ones. Therefore, to prove the previous Claim we simply resort to the proof of Proposition 1.

$\diamond\diamond$

Thus, by Algorithms 1 and 2 (or equivalently by (14)) we have two sets of near–optimal points for radial basis function interpolation. How close are these point sets? Which point set is “better” for interpolation purposes? These are some of the questions that we want to answer by numerical experiments in the next section.

4. Numerical results

In this section we present some examples of distribution of points as computed by Algorithms 1 and 2 in the bidimensional setting. We considered the square $\Omega = [-1,1] \times [-1,1]$ on which we picked 10000 random points. We have run the Algorithm 1 until the norm of the power function went below some fixed threshold $\tilde{\eta}$. As for Algorithm 2, we computed *once and for all* the necessary points up to a given number extracting them from a discretization of Ω. We have computed 406 points extracted from a discretization of 406^3 points of Ω. The number 406 corresponds to the dimension of the bivariate polynomials of degree ≤ 27 and the reason why we have extracted N points from N^3, comes from the theory of Leja sequences, as explained in the book [14]. Moreover, we stopped to 406 because of RAM limitations of the machine where computations where done. Consider that representing 406^3 reals in double precisions requires 510Mb of RAM. But, this was not a big problem, since in the following examples the points computed by Algorithm 1 were always less than 406.

In Figures 1.2-1.4 we show the distributions of the points computed both with the greedy method, Algorithm 1, and the geometric greedy method, Algorithm 2. On each figure we also show the *separation distances* among these points, making visually clearer that Algorithm 2 generates points nearly equidistributed in the Euclidean metric (as stated in Proposition 1).

By means of the Algorithm 1 applied to the Gaussian with scale 1, to reduce the power function below $\tilde{\eta} = 2 \cdot 10^{-7}$, we computed 65 points. For the Wendland's compactly supported function with scale 1, to reduce the power function below $\tilde{\eta} = 0.1$ we computed 80 optimal points and for the inverse multiquadrics with scale 1, we computed 90 points to depress the power function to $\tilde{\eta} = 2 \cdot 10^{-5}$. The choice of different $\tilde{\eta}$ depends on the decreasing rates of the associated power functions. Note that, for a given N, i.e. the number of optimal points we wish to find, so far we are not able to determine $\tilde{\eta}_\Phi(N)$ corresponding to a particular Φ.

Furthermore, given Φ_1, let X_1 be the optimal point set computed by minimizing the associated power function, say P_{Φ_1,X_1}, using Algorithm 1. Are these points optimal also for another $\Phi_2 \neq \Phi_1$? If not, are the points computed by the Algorithm 2 optimal for any given Φ, instead? In what follows, we will try to give qualitative answers to these "obvious" questions, showing in particular that the points computed by Algorithm 2 are good enough for almost all radial basis function interpolation problems.

We labeled by *g-gauss-65*, *gg-65*, *g-wend-80*, *gg-80*, *g-invm-90* and *gg-90* the point sets computed by Algorithm 1 and 2, where the prefix '*g*' recalls the word *greedy* while '*gg*' the words *geometric greedy*. The labels *gauss, wend, invm* recall instead the type of the radial function used in the minimization process. The '*gg*' point sets do not need to recall the radial function since they are independent of it.

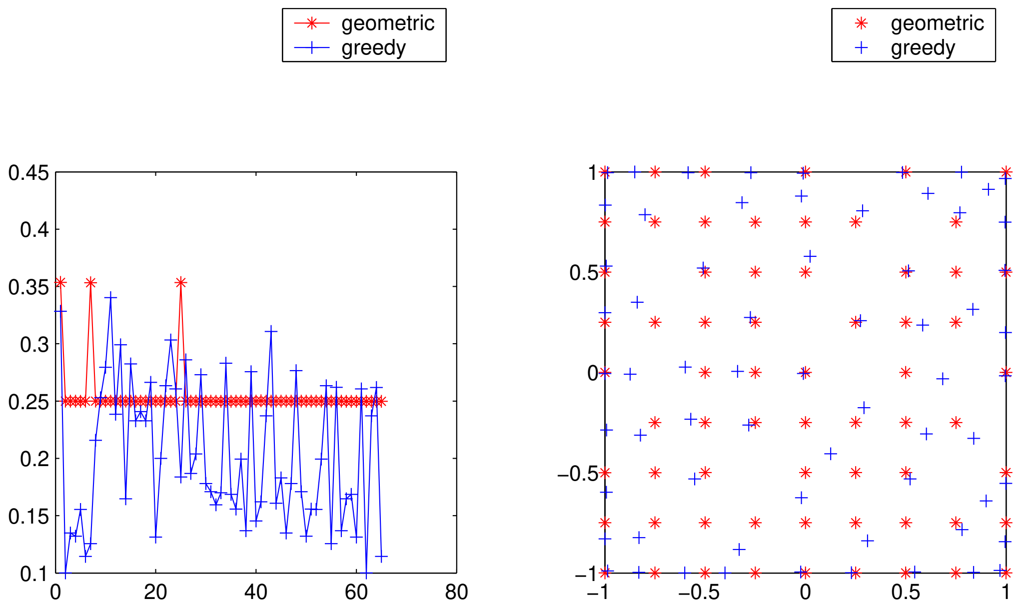

Figure 1.2: 65 optimal points for the Gaussian with scale 1. Right: the points as computed by the geometric greedy algorithm (*) and the greedy algorithm (+). Left: the separation distances among them.

As for interpolation, we have considered two test functions: $f_1(x,y) = \mathrm{e}^{-8(x^2+y^2)}$ and $f_2(x,y) = \sqrt{x^2+y^2-xy}$. The first is $\mathcal{C}^\infty$, while the second has discontinuity of the gradient. In Tables 1.1-1.3, we show the interpolation errors in the L_2-norm when the interpolant is constructed by means of the Gaussian, Wendland's and inverse multiquadrics, respectively. Each columns has an heading that recalls the set of points on which interpolation took place. The errors have been computed by sampling the functions on a regular grid of 30×30 points. While errors for the Gaussian are meaningless except in some

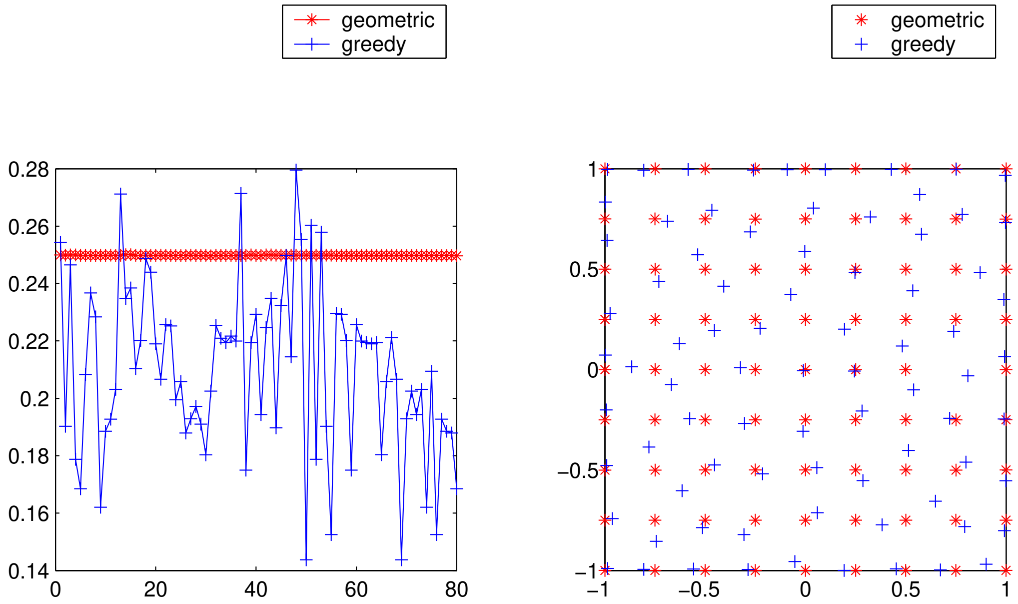

Figure 1.3: 80 optimal points for the Wendland's function. Right: the points as computed by the geometric greedy algorithm (*) and the greedy algorithm (+). Left: the separation distances among them.

cases, essentially due to errors occurring along boundaries, the interpolation errors for Wendland's and the inverse multiquadrics confirm, once again, that the points computed by Algorithm 2 are as good as the points computed by Algorithm 1.

	g-gauss-65	gg-65	g-wend-80	gg-80	g-invm-90	gg-90
f_1	$5.5\,10^{-1}$	$**$	$5.6\,10^{-1}$	$**$	$4.9\,10^{-1}$	$**$
f_2	$7.3\,10^{-1}$	$**$	$**$	$**$	$**$	$**$

Table 1.1: Errors in L_2-norm for interpolation by the Gaussian. When errors are > 1.0 we put $**$.

We have also computed, and plotted in Figures 1.5-1.7, the *Lebesgue constants* associated to these near-optimal point sets. The abscissas represent the *polynomial degree* and run till the maximum polynomial degree representable with the number of points in the sets. With the 65 points computed with the Gaussian the maximum degree is 9; for the 80 points for the Wendland's function and the 90 points computed for the inverse multiquadrics the maximum polynomial degree is 11. The computations of Lebesgue constants by means of (9) were done by discretizing the square $[-1, 1]^2$ with a grid of 40×40 points

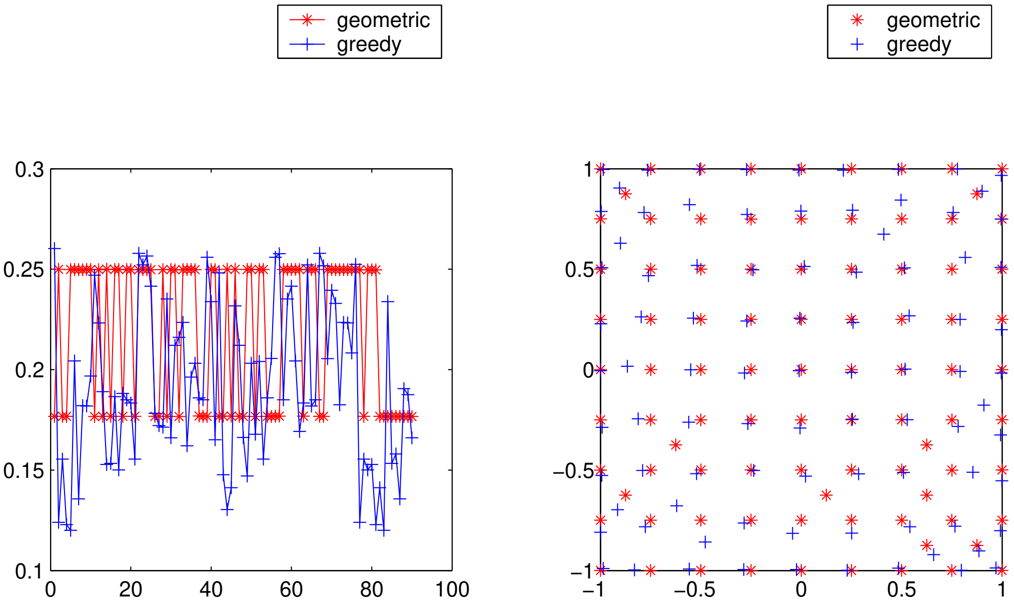

Figure 1.4: 90 optimal points for the inverse multiquadrics with scale 1. Right: the points as computed by the geometric greedy algorithm (*) and the greedy algorithm (+). Left: the separation distances among them.

	g-gauss-65	gg-65	g-wend-80	gg-80	g-invm-90	gg-90
f_1	$2.1\,10^{-1}$	$1.6\,10^{-1}$	$1.3\,10^{-1}$	$1.1\,10^{-1}$	$1.4\,10^{-1}$	$1.0\,10^{-1}$
f_2	$6.1\,10^{-1}$	$8.7\,10^{-1}$	$6.1\,10^{-1}$	$9.7\,10^{-1}$	$4.6\,10^{-1}$	$5.8\,10^{-1}$

Table 1.2: Errors in L_2-norm for interpolation by the Wendland's function.

where we sampled the cardinal functions u_k. The graphs show that, *except for the Gaussian*, the Lebesgue constants of the optimal points computed by the greedy method grow slower than the ones of the data-independent points. Moreover, in all cases they grow approximately linearly in the polynomial degree (modulo some constants). This explains once more why the errors computed with the Gaussian are meaningless. Of course, for a complete understanding of the asymptotic behavior of the Lebesgue constants we should go further in the computations, but we were not able due to hardware restrictions.

Concerning the computational efforts of both algorithms, we show in Table 1.4 the CPU time in seconds of Algorithm 1 for computing the optimal points for a given threshold. These computational costs were determined by the Matlab function `cputime`.

Algorithm 2 was run *once and for all* to compute at once the 406 points

	g-gauss-65	gg-65	g-wend-80	gg-80	g-invm-90	gg-90
f_1	$2.3\,10^{-1}$	$2.3\,10^{-1}$	$4.0\,10^{-2}$	$3.1\,10^{-2}$	$3.5\,10^{-2}$	$2.5\,10^{-2}$
f_2	$5.9\,10^{-1}$	$6.0\,10^{-1}$	$3.8\,10^{-1}$	$4.6\,10^{-1}$	$3.7\,10^{-1}$	$3.6\,10^{-1}$

Table 1.3: Errors in L_2-norm for interpolation by the inverse multiquadrics.

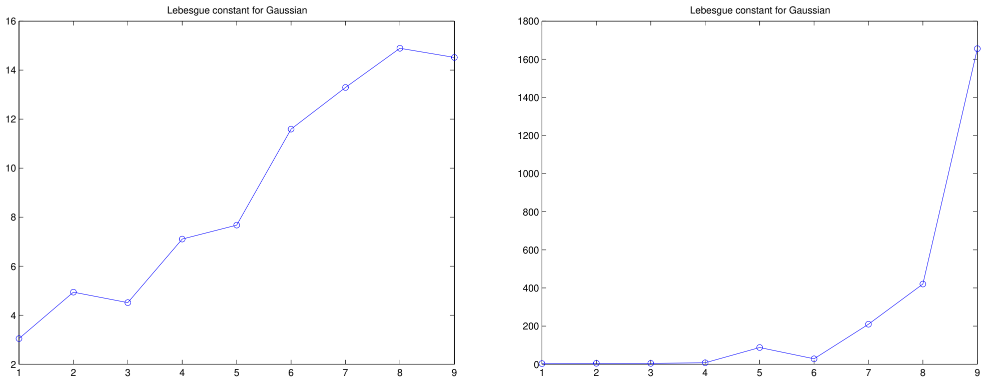

Figure 1.5: Lebesgue constants of the optimal points for the Gaussian (left) and the data-independent points (right).

by means of (14). These computations were done in about 5 minutes of CPU time on a PC with 900MHz Athlon processor and in this case the program was written in Fortran 77. The coordinates of the points were stored in a file and used later on with the same Matlab program that we wrote for making comparison plots, computing separation and fill distances as well as Lebesgue constants with respect to the points computed by Algorithm 1.

Gaussian scale 1, $\tilde{\eta} = 2 \cdot 10^{-7}$, 65 points, 51 sec.
Gaussian scale 2, $\tilde{\eta} = 2 \cdot 10^{-7}$, 32 points, 18 sec.
Wendland scale 1, $\tilde{\eta} = 0.1$, 80 points, 76 sec.
Wendland scale 15, $\tilde{\eta} = 2 \cdot 10^{-5}$, 100 points, 105 sec.
inverse multiquadrics scale 1, $\tilde{\eta} = 2 \cdot 10^{-5}$, 90 points, 110 sec.
inverse multiquadrics scale 2, $\tilde{\eta} = 2 \cdot 10^{-5}$, 34 points, 26 sec.

Table 1.4: Computational costs (`cputime` in seconds) of optimal points as computed by Algorithm 1

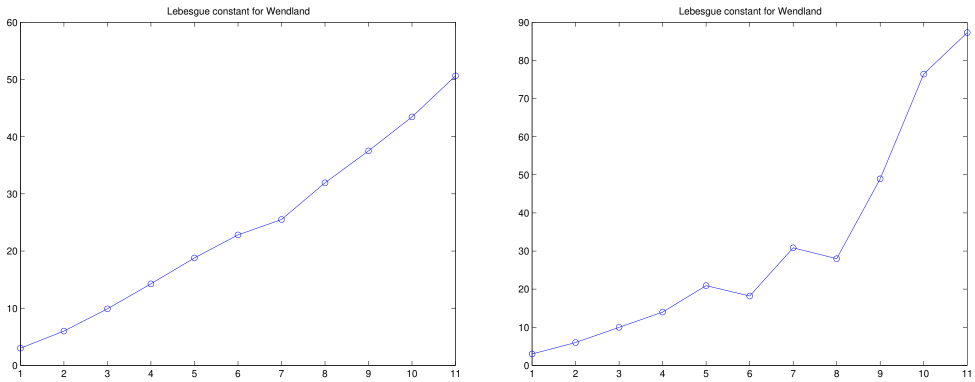

Figure 1.6: Lebesgue constants of the optimal points for the Wendland's function (left) and the data-independent points (right).

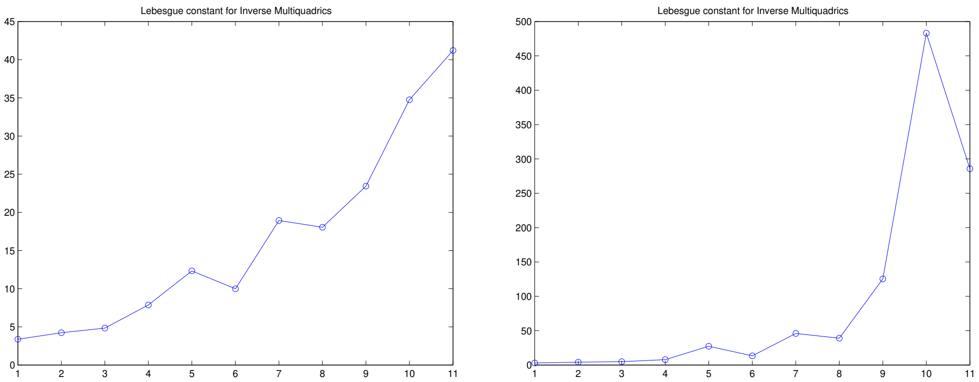

Figure 1.7: Lebesgue constants of the optimal points for the inverse multiquadrics (left) and the data-independent points (right).

5. Conclusions

The paper essentially presented two main results.

- Optimal points for radial basis function interpolation can be computed independently of the radial function and *once for all*. These points, in the two-dimensional case, correspond to Leja points in the Euclidean metric. They are asymptotically equidistributed with respect to the Euclidean metric (that is why we called *near-optimal*). Moreover, Algorithm 2 can be used with any metric, producing point sets asymptotically equidistributed with respect to that metric.

- From the Lebesgue constants behavior, we can conclude that data-independent points have Lebesgue constants that grow faster than data-dependet ones. Experiments on the growth of the Lebesgue constants on different nodal sets for bivariate polynomial interpolation are currently in progress and will be presented in the forthcoming paper [5]. From the results in that paper, here we only observe that quasi-uniformity is only a *necessary* condition for near-optimality of a point set. Therefore, generally speaking, the study of the growth of the Lebesgue constant of a point set is not a general criterion to investigate on the goodness of a point set. In the univariate setting for polynomial interpolation on bounded intervals, a similar conclusion was obtained in the paper [11]. Hence, we can confirm that data-independent points should be used in radial basis function interpolation because of their general and effective computational technique and their interpolation errors which are of the same order of the near-optimal points computed by minimizing the power function.

Acknowledgments. I would like to express my thanks to Prof. Len Bos of the University of Calgary who pointed out to me the possibility to compute Leja-like sequences by means of (14). Thanks also to an anonymous referee who made punctual suggestions which essentially improved the results presented in the paper.

References

[1] Baglama, J., Calvetti, D., Reichel, L. *Fast Leja points*, Electron. Trans. Numer. Anal. **7** (1998), 124-140.

[2] Beatson, R. K., Light, W. A., and Billings, S. *Fast solution of the radial basis function interpolation equations: Domain decomposition methods*, SIAM J. Sci. Comput. **22** (2000), 1717–1740.

[3] Beyer, A. *Optimale Centerverteilung bei Interpolation mit radialen Basisfunktionen*, Diplomarbeit, University of Göttingen, 1994.

[4] Bos, L. P., and Maier, U. *On the asymptotics of points which maximize determinants of the form* $det(g(|x_i - x_j|))$, In *Advances in Multivariate Approximation* (Berlin, 1999), W. Haussmann, K. Jetter, and M. Reimer, Eds., vol. 107 of Math. Res., Wiley-VCH., pp. 1–22.

[5] Caliari M., De Marchi, S., Vianello, M. *Bivariate Polynomial Interpolation at new nodal sets*, In preparation, 2003.

[6] De Marchi, S. *On Leja sequences: some results and applications*, In press on *Appl. Math. Comput.*, 2003.

[7] De Marchi, S., Schaback, R. and Wendland, H. *Near-Optimal Data-independent Point Locations for Radial Basis Function Interpolation*, Accepted by Adv. Comput. Math., 2003.

[8] Iske, A. *Optimal distribution of centers for radial basis function methods*, Tech. Rep. M0004, Technische Universität München, 2000.

[9] Iske, A. *On the Approximation Order and Numerical Stability of Local Lagrange Interpolation by Polyharmonic Splines*, To appear in the Proceedings of the *5th International Conference on Multivariate Approximation*, held at Bommerholz, Sept. 2002. Preprint, Technische Universität München, April 11, 2003.

[10] Leja, F. *Sur certaines suites liées aux ensembles plans er leur application à la représentation conforme*, Ann. Polon. Math. **4** (1957), 8–13.

[11] Mastroianni, G. and Occorsio, D. *Optimal systems of nodes for Lagrange interpolation on bounded intervals. A survey*, J. Comput. Appl. Math. **134**(1-2) (2001), 325–341.

[12] Micchelli, C. A. *Interpolation of scattered data: distance matrices and conditionally positive definite functions*, Constr. Approx., **2(1)** (1986), 11–22.

[13] Reichel, L., *Newton interpolation at Leja points*, BIT Numerical Analysis, **30** (1990), 332–346.

[14] Saff, E. B., Totik, V. *Logarithmic potential with external fields*, Springer Berlin, 1997.

[15] Schaback, R. *Reconstruction of Multivariate Functions from Scattered Data*, Manuscript available at *www.num.math.uni-goettingen.de/schaback/research/papers/rbfbook.ps.*

[16] Schaback, R. *Comparison of radial basis function interpolants*, In *Multivariate Approximations: from CAGD to Wavelets*, K. Jetter and F. Utreras (eds.), World Scientific Publishing Co., 1993.

[17] Schaback, R. *Native Hilbert spaces for radial basis functions I*, In *New developments in approximation theory*, 2nd Int. Dortmund meeting (IDoMAT), 1998, M. W. Müller and al., Eds., vol. 132 of Int. Ser. Numer. Math. Birkhäuser Verlag, Basel, (1999) 255–282.

[18] Wendland, H. *Error estimates for interpolation by compactly supported radial basis functions of minimal degree*, J. Approx. Theory **93** (1998), 258–272.

[19] Wu, Z., and Schaback, R. *Local error estimates for radial basis function interpolation of scattered data*, IMA Journal of Numerical Analysis **13** (1993), 13–27.

AMS Subject Classification: 41A05, 41A25, 41A63, 65D05, 65D15.

Stefano DE MARCHI
Department of Computer Science
University of Verona
S.da Le Grazie, 15
37134 Verona, ITALY
e-mail: `stefano.demarchi@univr.it`