THE ROLE OF INNER SUMMARIES IN THE FAST EVALUATION OF THIN-PLATE SPLINES

Michael J. Johnson

Department of Mathematics
Kuwait University
yohnson1963@hotmail.com

Nov. 13, 2011

Abstract. The driving force behind fast evaluation of thin-plate splines is the fact that a sum $\sum_{j=n_1}^{n_2} \lambda_j \phi(x-\xi_j)$, where $\phi(x) = \|x\|^2 \log \|x\|$, can be efficiently and accurately approximated by a truncated Laurent-like series (called an outer summary) when the data sites $\{\xi_j\}_{j=n_1}^{n_2}$ are clustered in a disk and when the evaluation point x lies well outside this disk. We present a means (called an inner summary) of approximating this sum when the evaluation point x lies inside the disk. The benefit of having an inner summary available (and of an improved error estimate for the outer summary), is that one can safely use uniform subdivision of clusters in the pre-processing phase without concern that an unfortunate distribution of data sites will lead to an unreasonably large number of clusters. A complete description and cost analysis of a hierarchical method for fast evaluation is then presented, where, thanks to uniform subdivision of clusters, the pre-processing is formulated in a way which is suitable for implementation in a high level computing language like Octave or Matlab.

1. Introduction

Let $\phi : \mathbb{R}^2 \to \mathbb{R}$ be given by $\phi(x) = \|x\|^2 \log \|x\|$. A *thin-plate spline* is a function of the form

$$s(x) = \sum_{j=1}^{n} \lambda_j \phi(x - \xi_j) + p(x), \qquad x \in \mathbb{R}^2, \tag{1.1}$$

where the *data sites* $\{\xi_j\}$ lie in $\mathbb{R}^2$, p is a linear polynomial and the coefficients $\{\lambda_j\}$ satisfy the auxiliary equations $\sum_{j=1}^{n} \lambda_j q(\xi_j) = 0$ for all linear polynomials q.

Such functions are employed in the problem known as "scattered data interpolation" (see [4], [10], and [7]), where one seeks a 'nice' function s satisfying $s(\xi_j) = y_j$, for some prescribed data $\{y_j\}$. In case the data sites $\{\xi_j\}$ are non-collinear, there exists a unique

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

thin-plate spline which satisfies the interpolation conditions $s(\xi_j) = y_j$. Although thin-plate splines have many endearing qualities, there are several drawbacks, one of which is the high evaluation cost. To appreciate this, one simply notes that a single evaluation of s, as given in (1.1), requires $O(n)$ floating point operations, and so if both n and the number of required evaluations of s are *large*, then the evaluation costs are $large^2$.

Beatson and Newsam [2] were the first to recognize that the fast algorithms being developed in computational physics by Greengard & Rokhlin [8] and van Dommelen & Rundensteiner [6] could be adapted to thin-plate splines, enabling their fast evaluation. The improved efficiency is obtained after an initial expenditure, called pre-processing, which then enables approximate evaluations of s to be performed at a per evaluation cost which is far less than $O(n)$ operations. The pre-processing involves arranging the data sites $\{\xi_j\}$ into a quad-tree of *clusters*, and computing, for each cluster, the coefficients in a truncated Laurent-like series. The utility of this can be described as follows. Suppose a cluster of data sites $\xi_{n_1}, \xi_{n_1+1}, \ldots \xi_{n_2}$ is contained in a disk and we wish to evaluate the cluster sum

$$\sum_{j=n_1}^{n_2} \lambda_j \phi(x - \xi_j) \tag{1.2}$$

at a point x which lies outside the disk. If the distance from x to the disk is sufficiently large, relative to the radius of the disk, it may be possible to obtain an accurate approximation to the cluster sum by simply evaluating the truncated Laurent-like series. In the sequel, the truncated Laurent-like series will be referred to as an *outer summary* of the cluster sum, where the term *outer* indicates that the evaluation point x lies outside the disk.

The quad-tree of clusters, produced during the pre-processing phase, is constructed according to the rule (applied recursively): *Any cluster containing at least Q data sites (Q is a given tuning parameter) will be subdivided into at most 4 child clusters.* In [8], [6] and [2] the subdivisioning is *uniform*, in that the bounding squares of the child clusters are obtained by partitioning the bounding square of the parent cluster into four congruent squares. Uniform subdivision is very desirable because it gives a certain stationarity to the pre-processing, reminiscent of the simplicity found in wavelet computations. There is, however, one very serious vulnerability associated with uniform subdivision, and that is that the number of clusters in the quad-tree can grow, without bound, as the distribution of the data sites becomes less and less uniform. This can be illustrated with the following example. Let the number of data sites be $Q + 1$ and let k be an arbitrarily large integer. Define $\xi_1 = (1, 1)$, $\xi_2 = (0, 0)$ and let the remaining $Q - 1$ data-sites lie in the square having opposite corners $(0, 0)$ and $(2^{-k}, 2^{-k})$. Then the first bounding square (containing all the data sites) will have $(0, 0)$ and $(1, 1)$ at its corners, it will have exactly one child (containing Q data sites), whose bounding square has corners $(0, 0)$ and $(2^{-1}, 2^{-1})$, and this will be repeated at least $k-1$ times until a cluster is produced whose bounding square has corners $(0, 0)$ and $(2^{-k}, 2^{-k})$. The resulting quad-tree contains more than k clusters, even though the number of data-sites is only $Q+1$. As a consequence of this vulnerability, it is not possible to obtain finite estimates on the pre-processing costs without imposing restrictions on the distribution of the the data sites. For example, assuming a uniform grid of data-sites, van Dommelen & Rundensteiner [6] show that their pre-processing requires $O(n \log n)$ operations.

Powell [9] proposes the following alternative to uniform subdivision. Any cluster containing more than $2Q$ data sites will be divided into exactly two child clusters, each of which contains at least Q data sites. The way this is performed is that the bounding rectangle of the parent is at first partitioned into two congruent rectangles. If this initial partitioning results in at least Q data sites in each child, then these are accepted. Otherwise, the partitioning line is adjusted until the smaller child contains exactly Q data-sites. Although Powell's modification ensures that the number of clusters is $O(n)$, the lack of stationarity greatly complicates the algorithm and this in turn complicates the task of obtaining meaningful cost estimates for both the pre-processing stage as well as the evaluation stage. Subsequently, Beatson & Light [1] extended Powell's method to polyharmonic splines in $\mathbb{R}^2$, and recently Beatson, Powell & Tan [3] proposed a method for polyharmonic splines in three dimensions.

In the present contribution, an alternate means of eliminating the vulnerability associated with uniform subdivision is proposed. Firstly, we suggest a slight modification to the way the Laurent-like series is truncated and obtain an estimate on the error in the outer summary which allows us to potentially employ the outer summary even when the evaluation point x is near, or on, the boundary of the disk containing the cluster of data sites. In contrast, the error estimate employed in [2], [9] becomes unbounded as x approaches the disk's boundary, and consequently, they do not use their outer summary when x is near, or on, the disk's boundary. Secondly, we introduce an *inner summary* of the cluster sum (1.2), which is defined when x lies inside the disk containing the cluster. The inner summary becomes more accurate as the radius of the disk becomes small, and this allows one to determine a maximum level ℓ_max such that the summarized value (regardless of whether x lies inside or outside the disk) of a cluster sum will be of sufficient accuracy whenever the level of the cluster (ie the number of its ancestors) equals ℓ_max. Consequently, the evaluation algorithm never visits a cluster of level more than ℓ_max, and therefore the pre-processing only needs to construct clusters of level less or equal to ℓ_max. The upshot is that we can safely employ uniform subdivision with the subdivision rule:

A cluster will be subdivided into at most 4 children if

1. the level of the cluster is less than ℓ_max, and

2. the cluster contains more than Q data sites.

The sequel is organized as follows. In section 2, the pre-processing is explained, for simplicity, as the task of constructing a catalog. The outer summaries and their associated error estimates are constructed in section 3, while section 4 is devoted to inner summaries. Section 5 explains the determination of certain *threshold radii* (two for each level) which are used in the evaluation algorithm (section 6) to determine whether a summarized value is sufficiently accurate. Estimates for the pre-processing costs and the per evaluation costs are obtained in sections 7 and 8, respectively. Finally, in section 9, we give some concluding remarks and describe several numerical experiments which shed some light on the effectiveness of inner summaries. We mention that the pre-processing has been formulated in such a way that it can be efficiently implemented in a high level language like Octave or Matlab. After the catalog has been constructed, it is expected that fast evaluations are implemented in a low level language like C or Fortran. Both the formulation and the error estimate for the inner summaries involve some numerically obtained values which are given as tables in the appendix. Since it might be helpful to start with a rough idea of how the evaluation algorithm unfolds, the reader is encouraged to first read section 6, casually, before continuing on to section 2.

For the reader's convenience, we mention those notations which carry a global meaning. The natural numbers are denoted $\mathbb{N} = \{1, 2, 3, \ldots\}$ and the non-negative integers are denoted $\mathbb{N}_0$. We identify $\mathbb{R}^2$ with the complex plane $\mathbb{C}$; in particular the function ϕ, defined at the beginning, becomes $\phi(z) = |z|^2 \log |z|$, $z \in \mathbb{C}$. The imaginary unit is denoted $\imath$ and the real and imaginary parts of z are denoted $\Re z$ and $\Im z$, respectively. The bounding squares which arise always have sides parallel to the coordinate axes, and the radius of such a square is defined to be the distance from its center to a vertex. The data sites are $\xi_1, \xi_2, \ldots, \xi_n$ (note that there are n of these) and our purpose is to approximately evaluate the function $s(z) = \sum_{j=1}^{n} \lambda_j \phi(z - \xi_j)$ with absolute error $\leq \delta$, where $\delta > 0$ is assumed to be a given tolerance. The integer m is related to the 'degree' of the truncated Laurent-like series used for outer summaries while m_0 plays a similar role for inner summaries. The univariate functions $\mathcal{E}_m$ and E_m, defined in section 3, are associated with the error estimates for outer summaries; while ε_{m_0} and $\widehat{\varepsilon}_{m_0}$, defined in section 4, go with inner summaries. The integer ℓ is used to denote a level, whereby r_ℓ denotes the radius of any cluster having level ℓ. The threshold radii, defined in section 5, are denoted t_ℓ (for inner summaries) and T_ℓ (for outer summaries).

2. Constructing the Catalog

The pre-processing required before fast evaluations can be performed may be visualized as the construction of a catalog whose pages are numbered $1, 2, \ldots, N$. The catalog is imbued with a tree structure such that every page, except page 1 (the root of the tree), has a parent and each page has at most four children. The entries on page p are the following:

$1-4.$	$p_I, \ldots, p_{IV}$	(page number of child q, $\quad q = I, II, III, IV$)
$5, 6.$	n_1, n_2	(associated cluster is $\{\xi_{n_1}, \xi_{n_1+1}, \ldots, \xi_{n_2}\}$)
7.	ℓ	(level of page)
8.	$c \in \mathbb{C}$	(center of bounding square Ω_p)
9.	$\alpha \in \mathbb{C}^{m+1}$	(standardized harmonic moments)
10.	$\beta \in \mathbb{C}^{m+1}$	(standardized biharmonic moments)

A value of 0 in any of the first four entries indicates that the corresponding child does not exist. The level ℓ equals the number of ancestors in the tree, so the first page has level 0 and the level of a child is one more than that of its parent. The square Ω_1 is a square, with radius $r_0 > 0$, which contains Ξ. The bounding square Ω_p (associated with page p) is the square with center c (entry 8) and radius $r_\ell = r_0 2^{-\ell}$, so Ω_p is determined by its center and the level of page p. The cluster associated with page p is $\Xi_p = \{\xi_{n_1}, \xi_{n_1+1}, \ldots, \xi_{n_2}\}$ which is bounded by Ω_p. Note that the mapping $\xi \mapsto \widetilde{\xi} := (\xi - c)/r_\ell$ maps the square Ω_p onto the standard square S, having center 0 and radius 1. The standardized moments (entries 9 and 10) are $\alpha = [\alpha_0, \alpha_1, \ldots, \alpha_m] \in \mathbb{C}^{m+1}$ and $\beta = [\beta_0, \beta_1, \ldots, \beta_m] \in \mathbb{C}^{m+1}$ given by

$$\alpha_k = \sum_{j=n_1}^{n_2} \lambda_j \widetilde{\xi}_j^{\,k} \quad \text{and} \quad \beta_k = \sum_{j=n_1}^{n_2} \lambda_j \left|\widetilde{\xi}_j\right|^2 \widetilde{\xi}_j^{\,k} \tag{2.1}$$

Remark 2.1. Our construction of the catalog ensures that if page p is a parent and if its bounding square Ω_p is divided into four congruent squares, then the bounding square associated with child I, if it exists, is the upper right square. If child II exists, then its bounding square is the upper left square, and so on, counter-clockwise.

Remark 2.2. Our construction also ensures that the induced tree of clusters $\{\Xi_1, \Xi_2, \ldots, \Xi_N\}$ forms a *partition-tree* of Ξ; that is,
(a) $\{\Xi_1, \Xi_2, \ldots, \Xi_N\}$ is a tree with root $\Xi_1 = \Xi$, and
(b) For each p, either Ξ_p is childless or the children of Ξ_p form a partition of Ξ_p.

Two immediate consequences of the latter remark are firstly that the childless clusters in $\{\Xi_1, \Xi_2, \ldots, \Xi_N\}$ form a partition of Ξ, and secondly that if distinct pages p and p' have the same level, then their associated clusters, Ξ_p and $\Xi_{p'}$, are disjoint.

The catalog can be efficiently stored in four matrices: a $7 \times N$ integer matrix CAT, a $1 \times N$ complex row vector $Center$ and two $(m+1) \times N$ complex matrices H_mom and bi_Hmom, where page p of the catalog is stored in column p of these matrices. We now give an algorithm for creating the catalog, where it is assumed that the integer $m \geq 2$, the centers $\Xi \in \mathbb{C}^{1\times n}$ and the coefficients $\lambda \in \mathbb{R}^{1\times n}$ are globally visible.

Step 1: Create globally visible variables and matrices $r_0 \in \mathbb{R}$, $N = 0 \in \mathbb{N}_0$, $CAT \in \mathbb{N}_0^{7\times 0}$, $Center \in \mathbb{C}^{1\times 0}$, and $H, biH \in \mathbb{C}^{(m+2)\times 0}$.

Step 2: Identify a smallest square Ω_1 which contains Ξ and let $c_1 \in \mathbb{C}$ be its center and $r_0 > 0$ its radius.

Step 3: Call the recursive function *create_page* (described below) with the command $p = create\_page(1, n, c_1, 0)$.

Step 4: Set $H\_mom = H(1 : m+1, :)$ and $biH\_mom = biH(2 : m+2, :)$.

Calls to the recursive function *create_page* have the format $p = create\_page(n_1, n_2, \ell, c)$. The immediate purpose of *create_page* is to create a new page, where entries 5-8 are given as inputs and the newly created page number p is returned as the output. After creating page p and registering entries 5-8, *create_page* must decide whether page p will have children. The decision rule is simply that page p will have children if its cluster contains at least Q data sites (Q is a fixed tuning parameter) and the level ℓ of page p is less than the maximum allowable level ℓ_max (an integer defined in section 6).

function $p = create\_page(n_1, n_2, \ell, c)$

1. Increment N and append a column of zeros to CAT, $Center$, H and biH.

2. Set $p = N$, $CAT(5:7, p) = \begin{bmatrix} n_1 \\ n_2 \\ \ell \end{bmatrix}$, $Center(p) = c$.

3a. If $n_2 - n_1 + 1 < Q$ or $\ell = \ell\_max$,
compute $H(:, p)$ and $biH(:, p)$ directly.

3b. Otherwise,
Reorder $\Xi(n_1 : n_2)$, and correspondingly $\lambda(n_1 : n_2)$, so that
$[\, n_1 \quad n_1+1 \quad \cdots \quad n_2 \,] = [\, J_I \quad J_{II} \quad J_{III} \quad J_{IV} \,]$ with
$\Xi(J_q) - c$ lying in quadrant q,

For $q \in \{I, II, III, IV\}$, with J_q nonempty,
Call *create_page* to create child q, with cluster $\Xi(J_q)$,
and register the output p_q in $CAT(q,p)$.
Add the contributions from child q to $H(:,p)$ and $biH(:,p)$.

end-function

The matrices H and biH, which each have a row more than H_mom and biH_mom, are defined by $H(:,p) = [\,\alpha_0 \quad \alpha_1 \quad \cdots \quad \alpha_{m+1}\,]^T$ and $biH(:,p) = [\,\overline{\alpha_1} \quad \beta_0 \quad \beta_1 \quad \cdots \quad \beta_m\,]^T$, and so it is clear that $H\_mom = H(1:m+1,:)$ and $biH\_mom = biH(2:m+2,:)$. In step 3a of *create_page*, the columns $H(:,p)$ and $biH(:,p)$ are computed directly using (2.1). In step 3b, however, these columns are to be obtained from those of the children. To see how this can be done, let α_k and β_k be as defined in (2.1) and put $a = [\,\alpha_0 \quad \alpha_1 \quad \cdots \quad \alpha_{m+1}\,]^T$ and $b = [\,\overline{\alpha_1} \quad \beta_0 \quad \beta_1 \quad \cdots \quad \beta_m\,]^T$. Let $\widetilde{\alpha}_k$, $\widetilde{\beta}_k$, $\widetilde{a}$ and $\widetilde{b}$ be defined the same, except use $\{\widetilde{\xi}_j\}$ in place of $\{\xi_j\}$ in (2.1), where $\widetilde{\xi}_j = \tau + \sigma\xi_j$ for some fixed $\sigma, \tau \in \mathbb{C}$. The problem of determining $\widetilde{a}$ and $\widetilde{b}$ from a and b has the following solution.

Theorem 2.3. *Let L be the $(m+2)\times(m+2)$ lower triangular matrix defined by*

$$L(i+1,j+1) = \binom{i}{j}\sigma^i\tau^{i-j}, \quad 0 \le i,j \le m+1.$$

Then $\widetilde{a} = La$ and $\widetilde{b} = L\,(\overline{\tau}a + \overline{\sigma}b)$.

Proof. We first observe that

$$\widetilde{\alpha}_k = \sum_{j=n_1}^{n_2} \lambda_j(\tau+\sigma\xi_j)^k = \sum_{j=n_1}^{n_2}\lambda_j\sum_{d=0}^{k}\binom{k}{d}\sigma^d\xi_j^d\tau^{k-d} = \sum_{d=0}^{k}\binom{k}{d}\sigma^d\tau^{k-d}\alpha_d = L(k+1,:)a,$$

and hence $\widetilde{a} = La$. With similar steps as above, we obtain

$$\begin{aligned}
\widetilde{\beta}_k &= \sum_{j=n_1}^{n_2}\lambda_j\left|\tau+\sigma\xi_j\right|^2(\tau+\sigma\xi_j)^k = \sum_{j=n_1}^{n_2}\lambda_j(\overline{\tau}+\overline{\sigma}\overline{\xi}_j)(\tau+\sigma\xi_j)^{k+1}\\
&= \sum_{d=0}^{k+1}\binom{k+1}{d}\sigma^d\tau^{k+1-d}\sum_{j=n_1}^{n_2}\lambda_j(\overline{\tau}\xi_j^d + \overline{\sigma}\overline{\xi}_j\xi_j^d)\\
&= \sum_{d=0}^{k+1}\binom{k+1}{d}\sigma^d\tau^{k+1-d}\left(\overline{\tau}\alpha_d + \overline{\sigma}\sum_{j=n_1}^{n_2}\lambda_j\overline{\xi}_j\xi_j^d\right).
\end{aligned}$$

Noting that $\sum_{j=n_1}^{n_2}\lambda_j\overline{\xi}_j\xi_j^d = \begin{cases}\overline{\alpha}_1 & \text{if } d=0\\ \beta_{d-1} & \text{if } d>0,\end{cases}$ we see that $\widetilde{\beta}_k = L(k+2,:)(\overline{\tau}a+\overline{\sigma}b)$ for $k = 0,1,\ldots,m$. Since L is lower triangular, $L(1,1) = 1$ and α_0 is real, we have $L(1,:)(\overline{\tau}a+\overline{\sigma}b) = \overline{\tau}\alpha_0 + \overline{\sigma}\overline{\alpha}_1 = \overline{\tau\alpha_0+\sigma\alpha_1}$. On the other hand, noting that $L(2,:) = [\,\tau \quad \sigma \quad 0 \quad \cdots \quad 0\,]$, we see that $\widetilde{\alpha}_1 = L(2,:)a = \tau\alpha_0 + \sigma\alpha_1$, and hence $L(1,:)(\overline{\tau}a+\overline{\sigma}b) = \overline{\widetilde{\alpha}_1}$. It now follows that $\widetilde{b} = L\,(\overline{\tau}a+\overline{\sigma}b)$. $\square$

Returning to our original concern of implementing the last item in step 3b, for $q = I, II, III, IV$, let L_q be the lower triangular matrix defined in Theorem 2.3 with $\sigma = \frac{1}{2}$ and $\tau_q = \dfrac{1}{2}e^{i(2q-1)\pi/4}$. It easily follows from Theorem 2.3 that the contribution from child q to $H(:,p)$ is $L_q * H(:,p_q)$, and the contribution to $biH(:,p)$ is $L_q * (\overline{\tau}_q H(:,p_q) + \frac{1}{2} biH(:,p_q))$. Noting that $\tau_q = L_q(2,1)$, the last item of step 3b can thus be written explicitly as

$$\begin{aligned} temp &= L_q * H(:,p_q) \\ H(:,p) &= H(:,p) + temp \\ biH(:,p) &= biH(:,p) + \overline{L_q(2,1)}\, temp + \frac{1}{2} L_q * biH(:,p_k) \end{aligned} \tag{2.2}$$

Remark. Formulae for efficiently obtaining the moments of a cluster from those of its children are given in [2, Lemma 3] and [9, eq. (3.15)]. The formulation given above, (2.2), has been specialized to the case of uniform subdivision and expressed as matrix multiplication to facilitate its implementation in a high level language like Octave or Matlab.

3. Outer Summaries

Let $\{\xi_{n_1}, \xi_{n_1+1}, \ldots, \xi_{n_2}\}$ be a cluster of data sites whose bounding square has center c and radius r_ℓ. An *outer summary* of the cluster sum $v(z) = \sum_{j=n_1}^{n_2} \lambda_j \phi(z - \xi_j)$ is a function F_v which approximates v outside the open disk $c + r_\ell U^o$, where U (resp. U^o) denotes the closed (resp. open) disk in $\mathbb{C}$ with center 0 and radius 1. Our choice of outer summary, which is very similar to that of [2], is built using an 'outer approximation' of the function $z \mapsto \phi(z - \xi)$. We first consider the *standard* case when $c = 0$ and $r_\ell = 1$.

For $s > 0$, we define $A_0(s) = s^2 \log s$, $A_1(s) = -s^2(1 + 2\log s)$, $A_k(s) = \dfrac{s^2}{k(k-1)}$, $k \geq 2$, $B_0(s) = 1 + \log s$, and $B_k = -\dfrac{1}{k(k+1)}$, $k \geq 1$. And for $k \in \mathbb{N}_0$, let $\phi_k : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ be defined by $\phi_k(0,0) = 0$ and

$$\phi_k(z,\xi) = \begin{cases} (\xi/z)^k (A_k(|z|) + |\xi|^2 B_k(|z|)) & \text{if } |\xi| \leq |z| \neq 0 \\ (\overline{z}/\overline{\xi})^k (A_k(|\xi|) + |z|^2 B_k(|\xi|)) & \text{if } |z| < |\xi| \end{cases}$$

Theorem 3.1. *The following hold:*
(i) $\phi_k(z,\xi) = \phi_k(\overline{\xi}, \overline{z}) = \overline{\phi_k(\xi, z)} = \phi_k(e^{i\theta} z, e^{i\theta}\xi)$ *for all* $z, \xi \in \mathbb{C}$, $\theta \in \mathbb{R}$, $k \in \mathbb{N}_0$.
(ii) ϕ_k *is continuous on* $\mathbb{C} \times \mathbb{C}$ *for all* $k \in \mathbb{N}_0$.
(iii) The series $\displaystyle\sum_{k=0}^{\infty} \phi_k$ *converges uniformly on compact subsets of* $\mathbb{C} \times \mathbb{C}$.
(iv) $\phi(z - \xi) = \displaystyle\sum_{k=0}^{\infty} \Re\phi_k(z,\xi)$ *for all* $z, \xi \in \mathbb{C}$.

Proof. (i) can be verified by inspection. For (ii), it is clear that ϕ_k is continuous at (z_0, ξ_0) in case $|z_0| \neq |\xi_0|$. The case $|z_0| = |\xi_0| \neq 0$ is also easy to verify after noting that

$(\xi_0/z_0)^k = (\overline{z}_0/\overline{\xi}_0)^k$ and $A_k(|z_0|)+|\xi_0|^2 B_k(|z_0|) = A_k(|\xi_0|)+|z_0|^2 B_k(|\xi_0|)$ hold in this case. The remaining case, $z_0 = \xi_0 = 0$, is not so simple. Let us write $z = se^{\imath\gamma}$ and $\xi = re^{\imath\theta}$ with $s, r > 0$. If $r, s \in (0, 1]$, then $|\phi_0(z,\xi)| \le \begin{cases} s^2 |\log s| + r^2(1 + |\log s|) & \text{if } r \le s, \\ r^2 |\log r| + s^2(1 + |\log r|) & \text{if } s < r \end{cases}$, and hence $|\phi_0(z,\xi)| \le r^2(1 + |\log r|) + s^2(1 + |\log s|) \to 0$ as $r, s \to 0$; hence ϕ_0 is continuous at $(0, 0)$. Similarly, $|\phi_1(z,\xi)| \le r^2(1+2 |\log r|)+s^2(1+2 |\log s|)$ and it follows that ϕ_1 is continuous at $(0, 0)$. If $k > 1$, then $|\phi_k(z,\xi)| \le \frac{1}{k(k-1)} \begin{cases} (r/s)^k(s^2 + r^2) & \text{if } r \le s \\ (s/r)^k(r^2 + s^2) & \text{if } s \le r \end{cases}$, and hence $|\phi_k(z,\xi)| \le \frac{1}{k(k-1)}(s^2 + r^2) \to 0$ as $r, s \to 0$. Thus ϕ_k is continuous at $(0, 0)$ and we have established (ii). We will prove (iii) by showing that $\sum_{k>1} \|\phi_k\|_{L_\infty(RU\times RU)} < \infty$ for all $R > 0$, where U denotes the closed unit disk in $\mathbb{C}$. If $k > 1$ and $|z|, |\xi| \le R$, then $|\phi_k(z,\xi)| \le \frac{1}{k(k-1)}(|z|^2 + |\xi|^2) \le \frac{2R^2}{k(k-1)}$, and hence $\sum_{k>1} \|\phi_k\|_{L_\infty(RU\times RU)} \le \sum_{k>1} \frac{2R^2}{k(k-1)} < \infty$. Beatson & Newsam [2, Lemma 1] have proved (iv), using slightly different notation, for the case $|z| > |\xi|$. In case $|z| < |\xi|$, then $\phi(z - \xi) = \phi(\xi - z) = \sum_{k=0}^{\infty} \Re\phi_k(\xi, z) = \sum_{k=0}^{\infty} \Re\phi_k(z,\xi)$, by (i). Now it follows from (ii) and (iii) that $\sum_{k=0}^{\infty} \Re\phi_k$ is continuous on $\mathbb{C} \times \mathbb{C}$. Since the function $(z,\xi) \mapsto \phi(z - \xi)$ is also continuous on $\mathbb{C} \times \mathbb{C}$, and equals the former whenever $|z| \ne |\xi|$ (ie on a dense subset of $\mathbb{C} \times \mathbb{C}$), it follows that the two are equal on all of $\mathbb{C} \times \mathbb{C}$. □

Given an integer $m \ge 2$, our outer approximation of $\phi(z - \xi)$, when $|\xi| \le 1 \le |z|$, is $\Re\Psi_m(z,\xi)$, where

$$\Psi_m(z,\xi) := \sum_{k=0}^{m} \phi_k(z,\xi) = \sum_{k=0}^{m} (\xi/z)^k(A_k(|z|) + |\xi|^2 B_k(|z|)).$$

Note the similarity to the outer approximation used in [2] and [9]: $\Re\left(\Psi_{m-2}(z,\xi) + (\xi/z)^{m-1}A_{m-1}(|z|) + (\xi/z)^m A_m(|z|)\right)$. We note that Ψ_m inherits property (i) of Theorem 3.1 from ϕ_k.

Our outer summary of $v(z) = \sum_{j=n_1}^{n_2} \lambda_j\phi(z-\xi_j)$ is obtained simply by replacing $\phi(z-\xi_j)$ with its outer approximation:

$$F_v(z) := \sum_{j=n_1}^{n_2} \lambda_j\Re\Psi_m(z,\xi_j) = \sum_{k=0}^{m} A_k(|z|)\Re(\alpha_k/z^k) + B_k(|z|)\Re(\beta_k/z^k), \tag{3.1}$$

where $\alpha_k = \sum_{j=n_1}^{n_2} \lambda_j\xi_j^k$ and $\beta_k = \sum_{j=n_1}^{n_2} \lambda_j |\xi_j|^2 \xi_j^k$. Our estimate on the error in the outer summary is expressed using the function

$$\mathcal{E}_m(t) = \frac{1}{m(m+1)}t^{1-m} + \frac{1}{(m+1)(m+2)}t^{-m}, \quad t \ge 1.$$

Proposition 3.2. *Let $m \ge 2$. If $|z| \ge 1$, then*

$$|v(z) - F_v(z)| \le \mathcal{E}_m(|z|) \sum_{j=n_1}^{n_2} |\lambda_j|.$$

The error estimate used in [2] and [9] is as above, but with $\mathcal{E}_m(|z|)$ replaced by $\mathcal{B}_m(|z|)$, where $\mathcal{B}_m(t) = \frac{1}{m(m+1)}\frac{t+1}{t-1}t^{1-m}$ (see [2, lemma 3]). We mention that $\mathcal{E}_m$ is an improvement in that $\mathcal{E}_m(t) < \mathcal{B}_m(t)$ for $t > 1$, and more importantly, that $\mathcal{E}_m(t)$ is well-behaved near $t = 1$, while $\mathcal{B}_m(t)$ is unbounded.

Our proof of Proposition 3.2 employs the error functional

$$\mathcal{F}_m(s) := \sup_{|z|\geq s} \max_{|\xi|\leq 1} |\phi(z-\xi) - \Re\Psi_m(z,\xi)|\,, \quad s \geq 1.$$

Lemma 3.3. *If* $|z| \geq 1$, *then*

$$|v(z) - F_v(z)| \leq \mathcal{F}_m(|z|) \sum_{j=n_1}^{n_2} |\lambda_j|\,.$$

Proof. The definition of $\mathcal{F}_m$ ensures that $|\phi(z-\xi_j) - \Re\Psi_m(z,\xi_j)| \leq \mathcal{F}_m(|z|)$, and hence

$$|v(z) - F_v(z)| \leq \sum_{j=n_1}^{n_2} |\lambda_j|\,|\phi(z-\xi_j) - \Re\Psi_m(z,\xi_j)| \leq \mathcal{F}_m(|z|) \sum_{j=n_1}^{n_2} |\lambda_j|\,.$$

□

Note that with Lemma 3.3 in hand, Proposition 3.2 can be proved simply by showing that $\mathcal{F}_m(s) \leq \mathcal{E}_m(s)$ for $s \geq 1$.

Lemma. *If* $k \geq 2$ *and* $|z| \geq 1$, *then* $\max_{|\xi|\leq 1} |\phi_k(z,\xi)| = \phi_k(z, z/|z|)$.

Proof. Assume $k \geq 2$ and $|\xi| \leq 1 \leq |z|$. Then

$$\phi_k(z,\xi) = \left(\frac{\xi}{z}\right)^k \left(\frac{|z|^2}{k(k-1)} - \frac{|\xi|^2}{k(k+1)}\right) = \frac{z^{-k}}{k(k-1)(k+1)}\left((k+1)\,|z|^2 - (k-1)\,|\xi|^2\right)\xi^k.$$

Writing $|\xi| = r$, we have $|\phi_k(z,\xi)| = \dfrac{|z|^{-k}}{k(k-1)(k+1)} f(r)$, where $f(r) = ((k+1)\,|z|^2 - (k-1)r^2)r^k$, $0 \leq r \leq 1$. Given that $k \geq 2$ and $|z| \geq 1$, it is a simple matter to verify that $f(0) = 0$, $f(1) > 0$ and $f'(r) > 0$ for $0 < r < 1$. Hence f is increasing on $[0,1]$ and therefore $\max_{|\xi|\leq 1} |\phi_k(z,\xi)| = |\phi_k(z,\zeta)|$ whenever $|\zeta| = 1$. The particular choice $\zeta = z/|z|$ renders $\phi_k(z,\zeta)$ real and positive, and hence the desired equality is obtained. □

Lemma 3.4. *For* $m \geq 2$, *let* $E_m(t) = \phi(t-1) - \Psi_m(t,1)$, $t \geq 1$. *The following hold:*

(*i*) $\mathcal{F}_m(s) = \max_{t\geq s} E_m(t)$, *for* $s \geq 1$.

(*ii*) $E_m(t) = \sum_{k>m} \phi_k(t,1) = \mathcal{E}_m(t) - \sum_{k=m+1}^{\infty} \dfrac{2t^{-k}}{k(k+1)(k+2)}$, *for* $t \geq 1$.

Proof. It follows from (iv) of Theorem 3.1 that

$$\phi(z-\xi) - \Re\Psi_m(z,\xi) = \sum_{k>m} \Re\phi_k(z,\xi), \tag{3.2}$$

and hence, by the lemma, that

$$\max_{|\xi|\le 1} |\phi(z-\xi) - \Re\Psi_m(z,\xi)| \le \sum_{k>m} \phi_k(z, z/|z|).$$

Substituting $\xi = z/|z|$ into (3.2) yields the reverse inequality, and therefore the above inequality holds with equality. But $\phi_k(z, z/|z|) = \phi_k(|z|, 1)$, by (i) of Theorem 3.1, and hence it follows that

$$\mathcal{F}_m(s) = \sup_{t\ge s} \sum_{k>m} \phi_k(t,1). \tag{3.3}$$

Before completing the proof of (i), let us first prove (ii), where we note that the first equality in (ii) is a special case of (3.2). For the second equality, we note that $\phi_k(t,1) = \dfrac{t^{2-k}}{k(k-1)} - \dfrac{t^{-k}}{k(k+1)}$, whence follows the expansion

$$\sum_{k>m} \phi_k(t,1) = \frac{t^{1-m}}{m(m+1)} + \frac{t^{-m}}{(m+1)(m+2)} + \sum_{k>m+2} \frac{t^{2-k}}{k(k-1)} - \sum_{k>m} \frac{t^{-k}}{k(k+1)}.$$

The proof of (ii) is then completed by combining the two series above into a single series. Now it follows from (ii) that $\sum_{k>m} \phi_k(t,1)$ converges to 0 as $t \to \infty$, and therefore the supremum in (3.3) is obtained, and we obtain (i) from (3.3) and the first equality in (ii). ☐

Proof of Proposition 3.2. As mentioned just after the proof of Lemma 3.3, it suffices to show that $\mathcal{F}_m(s) \le \mathcal{E}_m(s)$ for $s \ge 1$. It follows from Lemma 3.4 (ii) that $E_m(t) \le \mathcal{E}_m(t)$ for $t \ge 1$, and therefore, from (i), that $\mathcal{F}_m(s) \le \sup_{t\ge s} \mathcal{E}_m(t) = \mathcal{E}_m(s)$ for $s \ge 1$, where the equality arises since $\mathcal{E}_m$ is a decreasing function. ☐

Having settled the standard case, when $c = 0$ and $r_\ell = 1$, we consider now the general case, and to be more specific, let us assume that the cluster $\{\xi_{n_1}, \xi_{n_1+1}, \dots, \xi_{n_2}\}$ is associated with page p of our catalog whose bounding square Ω_p has center $c = Center(p)$ and radius $r_\ell = r_0 2^{-\ell}$, ℓ being the level of page p. For $w \in \mathbb{C}$, let $\widetilde{w} = (w-c)/r_\ell$ and note that the points $\{\widetilde{\xi}_{n_1}, \widetilde{\xi}_{n_1+1}, \dots, \widetilde{\xi}_{n_2}\}$ are bounded by the standard square (having center 0 and radius 1). It follows that the function $\widetilde{v}(w) = \sum_{j=n_1}^{n_2} \lambda_j \phi(w - \widetilde{\xi}_j)$ has the outer summary given in (3.1); specifically,

$$F_{\widetilde{v}}(w) = \sum_{k=0}^{m} A_k(|w|)\Re(\alpha_k/w^k) + B_k(|w|)\Re(\beta_k/w^k), \quad |w| \ge 1,$$

where α_k and β_k are as given in (2.1) and are stored in $H\_mom(:,p)$ and $biH\_mom(:,p)$, respectively. Now let v denote the cluster sum $v(z) = \sum_{j=n_1}^{n_2} \lambda_j \phi(z-\xi_j)$. Using the identity $\phi(rw) = (r^2 \log r)\,|w|^2 + r^2\phi(w)$, we see that

$$v(z) = \sum_{j=n_1}^{n_2} \lambda_j \phi(r_\ell(\widetilde{z}-\widetilde{\xi}_j)) = \sum_{j=n_1}^{n_2} \lambda_j (r_\ell^2 \log r_\ell) \left|\widetilde{z}-\widetilde{\xi}_j\right|^2 + r_\ell^2 \sum_{j=n_1}^{n_2} \lambda_j \phi(\widetilde{z}-\widetilde{\xi}_j),$$

and then substituting $\left|\widetilde{z}-\widetilde{\xi}_j\right|^2 = |\widetilde{z}|^2 - 2\Re(\widetilde{z}\overline{\widetilde{\xi}}_j) + \left|\widetilde{\xi}_j\right|^2$ yields

$$v(z) = (r_\ell^2 \log r_\ell)[\alpha_0 |\widetilde{z}|^2 - 2\Re(\widetilde{z}\overline{\alpha}_1) + \beta_0] + r_\ell^2 \widetilde{v}(\widetilde{z}), \quad z \in \mathbb{C}. \tag{3.4}$$

Our outer summary of v is now obtained simply by replacing $\widetilde{v}$ with its outer summary:

$$F_v(z) = (r_\ell^2 \log r_\ell)[\alpha_0 |\widetilde{z}|^2 - 2\Re(\widetilde{z}\overline{\alpha}_1) + \beta_0] + r_\ell^2 F_{\widetilde{v}}(\widetilde{z}), \quad |z-c| \geq r_\ell.$$

Our error estimate is a direct consequence of Proposition 3.2.

Corollary 3.5. *Let $m \geq 2$. If $|z-c| \geq r_\ell$, then*

$$|v(z) - F_v(z)| \leq r_\ell^2 \mathcal{E}_m \left(\frac{|z-c|}{r_\ell}\right) \sum_{j=n_1}^{n_2} |\lambda_j|.$$

4. Inner Summaries

An *inner summary* of the cluster sum $v(z) = \sum_{j=n_1}^{n_2} \lambda_j \phi(z-\xi_j)$ is a function f_v which approximates v on the ball $c + r_\ell U$. As in the previous section, we will first address the *standard case* when $c = 0$ and $r_\ell = 1$. Mimicking our construction of the outer summary, we will first 'define' an inner approximation $\Re\psi_{m_0}(z,\xi)$ of $\phi(z-\xi)$, for $z,\xi \in U$, and then stipulate that $f_v(z) = \sum_{j=n_1}^{n_2} \lambda_j \Re\psi_{m_0}(z,\xi_j)$, $z \in U$. We postulate that ψ_{m_0} has the form

$$\psi_{m_0}(z,\xi) = \sum_{k=0}^{m_0} \xi^k \left(\frac{|z|}{z}\right)^k (u_k(|z|) + |\xi|^2 v_k(|z|)), \quad z,\xi \in U, \tag{4.1}$$

where u_k and v_k are continuous and real-valued on $[0,1]$ and $u_k(0) = v_k(0) = 0$ for $k > 0$. In case $z = 0$, the value $\frac{|z|}{z}$ is understood to equal 1. The form of ψ_{m_0} is chosen to ensure that ψ_{m_0} is continuous and satisfies $\psi_{m_0}(e^{i\theta}z, e^{i\theta}\xi) = \psi_{m_0}(z,\xi)$, and so that $f_v(z)$ will equal a z-dependent linear combination of the moments $\alpha_k = \sum_{j=n_1}^{n_2} \lambda_j \xi_j^k$ and $\beta_k = \sum_{j=n_1}^{n_2} \lambda_j |\xi_j|^2 \xi_j^k$. Indeed,

$$f_v(z) = \sum_{k=0}^{m_0} u_k(|z|)\Re\left((|z|/z)^k \alpha_k\right) + v_k(|z|)\Re\left((|z|/z)^k \beta_k\right), \quad z \in U.$$

Our error estimate is obtained with the aid of the error functional $\varepsilon_{m_0} : [0,1] \to [0,\infty)$, defined by

$$\varepsilon_{m_0}(s) = \max_{|\xi|\le 1} \max_{s\le |z|\le 1} |\phi(z-\xi) - \Re\psi_{m_0}(z,\xi)|, \quad s \in [0,1].$$

Since $\phi(z-\xi)$ and $\Re\psi_{m_0}$ are both continuous on $U \times U$, it follows that ε_{m_0} is well-defined, continuous and, by definition, monotonically decreasing. The proof of the following error estimate is the same as that of Lemma 3.3, mutatis mutandis.

Proposition 4.1. *If $|z| \le 1$, then*

$$|v(z) - f_v(z)| \le \varepsilon_{m_0}(|z|) \sum_{j=n_1}^{n_2} |\lambda_j|.$$

Example. If u_k and v_k are chosen simply as $u_k = v_k = 0$, then $\varepsilon_{m_0}(s) = 4\log 2$ for all $s \in [0,1]$.

For other (non-trivial) choices of u_k and v_k, it is expected that ε_{m_0} will be computed numerically. As a means of reducing the difficulty of this numerical task, we offer the following.

Proposition 4.2. *Let $U_{up} := \{z \in U : \Im z \ge 0\}$ and define*

$$\gamma_{m_0}(t) := \max_{\xi\in U_{up}} |\phi(t-\xi) - \Re\psi_{m_0}(t,\xi)|, \quad t \in [0,1].$$

Then $\varepsilon_{m_0}(s) = \max\limits_{s\le t\le 1} \gamma_{m_0}(t)$, $s \in [0,1]$.

Proof. Fix $s \in [0,1]$ and put $\Gamma = \max_{s\le t\le 1}\gamma_{m_0}(t)$. It is clear, from the definition of ε_{m_0}, that $\varepsilon_{m_0}(s) \ge \Gamma$. Let $\xi, z \in U$, with $s \le |z| \le 1$, be such that $\varepsilon_{m_0}(s) = |\phi(z-\xi) - \Re\psi_m(z,\xi)|$, and let $\theta \in \mathbb{R}$ be such that $e^{i\theta}z = |z|$. Then

$$\begin{aligned}\varepsilon_{m_0}(s) &= \left|\phi(e^{i\theta}z - e^{i\theta}\xi) - \Re\psi_m(e^{i\theta}z, e^{i\theta}\xi)\right| \\ &= \left|\phi(|z| - e^{i\theta}\xi) - \Re\psi_m(|z|, e^{i\theta}\xi)\right| \le \gamma_{m_0}(|z|) \le \Gamma,\end{aligned}$$

where, in the first inequality, if $e^{i\theta}\xi$ does not belong to U_{up}, then $e^{-i\theta}\overline{\xi}$ does and the value $\phi(|z| - e^{i\theta}\xi) - \Re\psi_m(|z|, e^{i\theta}\xi)$ equals $\phi(|z| - e^{-i\theta}\overline{\xi}) - \Re\psi_m(|z|, e^{-i\theta}\overline{\xi})$. $\square$

With Proposition 4.2 in view, approximate values of ε_{m_0} can be obtained as follows:
Choose $M \in \mathbb{N}$ and put $s_j = j/M$, $j = 0,1,\ldots,M$.
Numerically compute $\gamma_{m_0}(s_j)$, $j = 0,1,\ldots,M$.
Then $\varepsilon_{m_0}(s_i) \approx \max\limits_{i\le j\le M} \gamma_{m_0}(s_j)$ for $i = 0,1,\ldots,M$.

We propose that u_k and v_k be chosen so that $\xi^k \left(\frac{|z|}{z}\right)^k (u_k(|z|) + |\xi|^2 v_k(|z|))$ approximates $\phi_k(z,\xi)$, for $z,\xi \in U$. Note that if $z = se^{i\omega}$ and $\xi = re^{i\theta}$, with $0 \le s, r \le 1$, then factoring out $e^{\imath k(\omega-\theta)}$ yields

$$\left|\phi_k(z,\xi) - \xi^k \left(\frac{|z|}{z}\right)^k (u_k(|z|) + |\xi|^2 v_k(|z|))\right| = \left|\phi_k(s,r) - r^k(u_k(s) + r^2 v_k(s))\right|,$$

and hence

$$\max_{\xi\in U}\left|\phi_k(z,\xi)-\xi^k\left(\frac{|z|}{z}\right)^k\left(u_k(|z|)+|\xi|^2\,v_k(|z|)\right)\right| = \max_{0\le r\le 1}\left|\phi_k(s,r)-r^k(u_k(s)+r^2v_k(s))\right|.$$

Our stated aim thus simplifies to that of choosing u_k and v_k so that $r^k(u_k(s)+r^2v_k(s))$ approximates $\phi_k(s,r)$, for $s,r\in[0,1]$. For $s\in[0,1]$ fixed, we let $u_k^*(s)$ and $v_k^*(s)$ be such that the function $[0,1]\ni r\mapsto r^k(u_k^*(s)+r^2v_k^*(s))$, which belongs to $W_k=span\{r^k,r^{k+2}\}$, is a best approximation of $r\mapsto\phi_k(s,r)$ from W_k. Specific values $u_k^*(s)$ and $v_k^*(s)$ are easy to compute numerically, and in Table 1 of the Appendix, we state polynomial approximations of u_k^* and v_k^* for $k=0,1,2,\dots,11$. With u_k and v_k chosen to equal these polynomials, we also compute the resulting function ε_{m_0}, using Proposition 4.2, and we state in Table 2 of the Appendix a range-limited polynomial $\widehat{\varepsilon}_{m_0}$ such that $\widehat{\varepsilon}_{m_0}$ is continuous, monotone decreasing and greater or equal to ε_{m_0} on $[0,1]$. For the remainder of the paper, we assume only that u_k and v_k have been chosen in a manner consistent with the requirements mentioned after (4.1), and we assume that a numerically accessible function $\widehat{\varepsilon}_{m_0}$ is at hand which is continuous, monotone decreasing and greater or equal to ε_{m_0} on $[0,1]$.

The transition from the standard case ($c=0$, $r_\ell=1$) to the general case ($c=Center(p)$, $r_\ell=r_0 2^{-\ell}$) is done the same as for the outer summary. In brief, we express $v(z)$ as in (3.4) and then replace $\widetilde{v}$ with its inner summary $f_{\widetilde{v}}$, culminating in

$$\begin{aligned} f_v(z) &= (r_\ell^2\log r_\ell)[\alpha_0\,|\widetilde{z}|^2-2\Re(\widetilde{z}\overline{\alpha}_1)+\beta_0] \\ &\qquad + r_\ell^2\sum_{k=0}^{m_0}u_k(|\widetilde{z}|)\Re\left((|\widetilde{z}|/\widetilde{z})^k\alpha_k\right)+v_k(|\widetilde{z}|)\Re\left((|\widetilde{z}|/\widetilde{z})^k\beta_k\right), \quad z\in c+r_\ell U, \end{aligned}$$

where $\widetilde{z}=(z-c)/r_\ell$ and α_k and β_k are as given in (2.1) and stored in $H\_mom(:,p)$ and $biH\_mom(:,p)$, respectively. As with the outer summary, our error estimate for the general case follows immediately from the error estimate for the standard case, namely Proposition 4.1.

Corollary 4.3. *If $|z-c|\le r_\ell$, then*

$$|v(z)-f_v(z)|\le r_\ell^2\varepsilon_m\left(\frac{|z-c|}{r_\ell}\right)\sum_{j=n_1}^{n_2}|\lambda_j|.$$

5. The Threshold Radii

During the evaluation phase of our algorithm, which is fully described in the next section, whenever a page of the catalog is visited, with an evaluation point $z\in\mathbb{C}$ and tolerance $\delta>0$ in hand, it must be decided whether or not the corresponding cluster sum $v(z)=\sum_{j=n_1}^{n_2}\lambda_j\phi(z-\xi_j)$ can be summarized. As usual, let p denote the page number of the catalog, ℓ the level of page p, c the center of the bounding square and $r_\ell=r_0 2^{-\ell}$ the radius of the bounding square. In this section we carefully examine this decision and determine threshold radii t_ℓ and T_ℓ whereby the decision is quickly taken simply by comparing the distance from z to c with the three radii t_ℓ, r_ℓ and T_ℓ. For the sake of clarity, we will consider, for the moment, the case $|z-c|\ge r_\ell$ where the outer summary is defined. Our starting point is the following rule.

Outer Summarizing Rule. The cluster sum $v(z)$ is summarized by $F_v(z)$ if the error estimate in Corollary 3.5 ensures that

$$|v(z) - F_v(z)| \le \frac{\delta}{\|\lambda\|_1} \sum_{j=n_1}^{n_2} |\lambda_j|\,,$$

where $\|\lambda\|_1 := \sum_{j=1}^{n} |\lambda_j|$.

With Corollary 3.5 in view, one easily sees that our Outer Summarizing Rule is equivalent to

$$\mathcal{E}_m\left(\frac{|z-c|}{r_\ell}\right) \le \rho_\ell, \quad \text{where } \rho_\ell := \frac{\delta}{r_\ell^2 \|\lambda\|_1}, \tag{5.1}$$

and since $\mathcal{E}_m$ is continuous and decreasing, it follows that this, in turn, is equivalent to

$$|z-c| \ge T_\ell, \quad \text{where } T_\ell := \begin{cases} r_\ell & \text{if } \mathcal{E}_m(1) \le \rho_\ell \\ r_\ell \mathcal{E}_m^{-1}(\rho_\ell) & \text{if } \mathcal{E}_m(1) > \rho_\ell \end{cases}$$

Remark 5.1. For the case $\mathcal{E}_m(1) \le \rho_\ell$, since $\mathcal{E}_m$ is C^∞, decreasing and convex, the equation $\mathcal{E}_m(s) = \rho_\ell$ can be fearlessly solved (numerically) using Newton's method initialized with $s_0 = 1$.

Turning now to the case $|z-c| < r_\ell$, where the inner summary is defined, we begin with the following rule.

Inner Summarizing Rule. The cluster sum $v(z)$ is summarized by $f_v(z)$ if the error estimate in Corollary 4.3, with $\widehat{\varepsilon}_{m_0}$ in place of ε_{m_0}, ensures that

$$|v(z) - f_v(z)| \le \frac{\delta}{\|\lambda\|_1} \sum_{j=n_1}^{n_2} |\lambda_j|\,.$$

Following the steps used for the outer summary above, and with ρ_ℓ as before, one directly arrives at the equivalent formulation:
$v(z)$ is summarized by $f_v(z)$ if $|z-c| \ge t_\ell$, where

$$t_\ell := \begin{cases} 0 & \text{if } \widehat{\varepsilon}_{m_0}(0) \le \rho_\ell \\ r_\ell & \text{if } \widehat{\varepsilon}_{m_0}(1) > \rho_\ell \\ r_\ell \min\{s \in [0,1] : \widehat{\varepsilon}_{m_0}(s) \le \rho_\ell\} & \text{otherwise} \end{cases}$$

If $\widehat{\varepsilon}_{m_0}$ is as given in Table 2 of the Appendix, then the above definition of t_ℓ simplifies to

$$t_\ell = \begin{cases} 0 & \text{if } e_{m_0}(0) \le \rho_\ell \\ r_\ell & \text{if } e_{m_0}(1) > \rho_\ell \\ r_\ell q_{m_0}^{-1}(\rho_\ell) & \text{otherwise} \end{cases},$$

and the author suggests, for the latter case, solving the equation $q_{m_0}(s) = \rho_\ell$ using the secant method with starting points 0 and 1.

The computation of the threshold radii T_ℓ and t_ℓ is part of the pre-processing phase and we include the values $t_\ell^2, r_\ell^2, T_\ell^2$, for $\ell = 0, 1, \ldots, \ell\_max$, as the appendix of the catalog.

6. Fast Evaluation

Given a tolerance $\delta > 0$, we consider the task of evaluating the function $s(z) = \sum_{j=1}^{n} \lambda_j \phi(z - \xi_j)$ to within tolerance δ; specifically, we wish to use the catalog, for a given evaluation point z, to quickly find a value V satisfying $|s(z) - V| \le \delta$. We adopt Powell's [9] method of evaluation which uses a stack, storing page numbers, along with an accumulation variable V. Starting with stack=[1] and $V = 0$, the algorithm proceeds iteratively, stopping when the stack is empty. At each iteration, a page number is popped off the stack and that page is *visited*. Each visit to a page results in one of the following actions:

(i) a value is added to V, or

(ii) the page numbers of the children are pushed onto the stack.

The choice is made by first comparing $|z - c|^2$ with the values $t_\ell^2, r_\ell^2, T_\ell^2$, which are stored in the catalog's appendix, to determine whether the cluster sum $v(z) = \sum_{j=n_1}^{n_2} \lambda_j \phi(z - \xi_j)$ can be summarized. If so, then the summarized value $F_v(z)$ (if $T_\ell^2 \le |z - c|^2$) or $f_v(z)$ (if $t_\ell^2 \le |z - c|^2 < r_\ell^2$) is added to V. If the cluster sum cannot be summarized, but our page has children, then action (ii) is performed. And if none of the above applies, we directly compute the cluster sum and add it to V.

Theorem 6.1. *The obtained value V satisfies $|s(z) - V| \le \delta$.*

Proof. Let $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ denote the set of pages which get visited, where $\mathcal{P}_1$ contains those pages resulting in action (i) and $\mathcal{P}_2$ contains those pages resulting in action (ii). Note that $\mathcal{P}$ is a sub-tree of our catalog, whose root is page 1, and $\mathcal{P}_1$ contains those pages of $\mathcal{P}$ which are childless (relative to $\mathcal{P}$) while $\mathcal{P}_2$ contains those pages of $\mathcal{P}$ which have children (in $\mathcal{P}$). Furthermore, the restriction of the partition tree $\{\Xi_p\}$ to $\mathcal{P}$ forms a partition tree of Ξ, and consequently (see Remark 2.2) the clusters $\{\Xi_p : p \in \mathcal{P}_1\}$ form a partition of Ξ. It follows that $s(z)$ can be written as $s(z) = \displaystyle\sum_{p\in\mathcal{P}_1} v_p(z)$, where $v_p(z) = \displaystyle\sum_{j=n_1(p)}^{n_2(p)} \lambda_j \phi(z - \xi_j)$ is the cluster sum associated with page p. On the other hand, the obtained value V arises as $V = \displaystyle\sum_{p\in\mathcal{P}_1} w_p$, where w_p denotes the value added to V when page p was visited. For each $p \in \mathcal{P}_1$, either w_p is a summary of $v_p(z)$, in which case $|v_p(z) - w_p| \le \frac{\delta}{\|\lambda\|_1} \sum_{j=n_1(p)}^{n_2(p)} |\lambda_j|$, or $w_p = v_p(z)$ (obtained by direct evaluation). Hence,

$$
\begin{aligned}
|s(z) - V| &= \left| \sum_{p\in\mathcal{P}_1} (v_p(z) - w_p) \right| \le \sum_{p\in\mathcal{P}_1} |v_p(z) - w_p| \\
&\le \sum_{p\in\mathcal{P}_1} \frac{\delta}{\|\lambda\|_1} \sum_{j=n_1(p)}^{n_2(p)} |\lambda_j| = \frac{\delta}{\|\lambda\|_1} \sum_{j=1}^{n} |\lambda_j| = \delta.
\end{aligned}
$$

□

An important observation, regarding the above evaluation algorithm, is that if a page, with level ℓ, is visited, and if $T_\ell = r_\ell$ and $t_\ell = 0$, then it is guaranteed that action (i) will be taken and consequently no children of this page will be visited. So if we define ℓ_max to be the first level ℓ for which both $t_\ell = 0$ and $T_\ell = r_\ell$, then the evaluation algorithm will never visit a page with a level greater than ℓ_max. This is why our catalog need only include pages of level less or equal to ℓ_max. With the definitions of T_ℓ and t_ℓ in view, we see that ℓ_max equals the smallest non-negative integer ℓ such that $\mathcal{E}_m(1) \le \rho_\ell$ and $\varepsilon_{m_0}(0) \le \rho_\ell$, where $\rho_\ell = \frac{\delta}{r_\ell^2\|\lambda\|_1}$. From this it is a simple matter to work out that

$$\ell\_max = \max\{0, \left\lceil \log_4 \frac{r_0^2 \varepsilon \|\lambda\|_1}{\delta} \right\rceil\}, \text{ where } \varepsilon := \max\{\mathcal{E}_m(1), \varepsilon_{m_0}(0)\}. \tag{6.1}$$

7. Pre-Processing Cost

In this section we estimate the cost of constructing the catalog, in terms of floating point operations (flops) and floating point comparisons (flo-comps). We adopt the viewpoint that m_0 and r_0 are fixed and any quantity which depends only on these values will be considered a constant. Consequently our estimates will involve the variable parameters m, n, δ and $\|\lambda\|_1$. For example, the integer ℓ_max, defined in (6.1), is $O(\log(2 + \|\lambda\|_1/\delta))$ since r_0 is constant and $\varepsilon = \max\{\mathcal{E}_m(1), \varepsilon_{m_0}(0)\}$ is bounded independently of m. We assume the following relations throughout: $2 \le m_0 \le m$, $r_0, \delta > 0$ and $Q, n \ge 2$. It will be seen that δ and $\|\lambda\|_1$ always appear in the ratio $\|\lambda\|_1/\delta$, so for convenience we set $\mu := \|\lambda\|_1/\delta$.

Let the pages of the catalog be partitioned as $C_1 \cup C_2$, where C_1 contains the childless pages and C_2 contains the child-bearing pages. We first estimate the number of flo-comps. These only arise in the construction of child-bearing pages, specifically in the first half of step 3a of the function *create_page*. The number of flo-comps which arise during the construction of a page $p \in C_2$ is proportional to the number of data sites in the cluster Ξ_p, and therefore, the total number of flo-comps needed to construct the catalog is $O(\sum_{p=1}^N \#\Xi_p)$. This sum can be stratified according to level, leading to the estimate

$$\sum_{p=1}^N \#\Xi_p = \sum_{\ell=0}^{\ell\_max} \sum \{\#\Xi_p : \text{page } p \text{ has level } \ell\} \le \sum_{\ell=0}^{\ell\_max} \#\Xi = (1 + \ell\_max)n.$$

We conclude therefore that the number of flo-comps needed to construct the catalog is $O(n\log(2+\mu))$.

The flops employed in constructing a page $p \in C_1$ arise entirely in step 3a of the function *create_page*, where the moments $H(:,p)$ and $biH(:,p)$ are computed directly using (2.1). The number of flops employed for this is bounded by a constant multiple of m times the number of data sites in the associated cluster Ξ_p, and it follows that the total number of flops needed to construct all the childless pages is $O\left(m \sum_{p\in C_1} \#\Xi_p\right)$. But the childless clusters $\{\Xi_p\}_{p\in C_1}$ form a partition of Ξ (see Remark 2.2), and hence the total number of flops employed in constructing the pages in C_1 is $O(mn)$.

The flops employed in constructing a page $p \in C_2$ arise entirely in step 3b of the function *create_page*, where the number employed is $O(m^2)$ since the moments are obtained from the children (using at most 4 executions of (2.2)). It follows that the total number of flops employed in constructing the pages in C_2 is $O(m^2\#C_2)$, and we therefore direct our attention to estimating $\#C_2$ (the number of child-bearing pages).

Proposition. *The number of child-bearing pages satisfies*

$$\#C_2 \leq \frac{n}{Q}\left(\frac{4}{3} + \log_4^+(4r_0^2\varepsilon\frac{Q}{n}\mu)\right), \tag{7.1}$$

where $\log_4^+ t$ *equals* $\log_4 t$ *if* $t \geq 1$ *and equals* 0 *if* $0 < t \leq 1$.

Proof. If $\#C_2 = 0$, then (7.1) holds trivially, so assume $\#C_2 > 0$. It follows then that $n \geq Q$ and $\ell\_max > 0$, and we then obtain from (6.1) that $4^{\ell\_max-1} < r_0^2\varepsilon\mu \leq 4^{\ell\_max}$. Let C_2 be partitioned as $C_2 = \bigcup_{0\leq\ell<\ell\_max} P_\ell$, where P_ℓ contains the child-bearing pages of level ℓ. Since each page has at most 4 children, it is clear that $\#P_\ell \leq 4^\ell$. On the other hand, since each child-bearing cluster contains at least Q data sites, and since clusters at the same level are pairwise disjoint, it follows that $\#P_\ell \leq n/Q$. Therefore,

$$\#C_2 = \sum_{0\leq\ell<\ell\_max} \#P_\ell \leq \sum_{0\leq\ell<\ell\_max} \min\{4^\ell, n/Q\}. \tag{7.2}$$

In case $4^{\ell\_max-1} \leq \dfrac{n}{Q}$, we obtain from (7.2) that $\#C_2 \leq \sum_{\ell=0}^{\ell\_max-1} 4^\ell < \frac{1}{3}4^{\ell\_max} \leq \frac{4}{3}\frac{n}{Q}$, and we see that (7.1) holds for this case. Considering the remaining case, $4^{\ell\_max-1} > \dfrac{n}{Q}$, let ℓ_1 be the smallest integer such that $\dfrac{n}{Q} \leq 4^{\ell_1}$, and note that ℓ_1 is non-negative since $n \geq Q$. We now obtain from (7.2) that

$$\#C_2 \leq \sum_{0\leq\ell<\ell_1} 4^\ell + \sum_{\ell_1\leq\ell<\ell\_max} \frac{n}{Q} < \frac{1}{3}4^{\ell_1} + \frac{n}{Q}(\ell\_max - \ell_1).$$

The desired estimate (7.1) is now obtained by employing the inequalities $4^{\ell_1} \leq 4\dfrac{n}{Q}$, $\ell\_max \leq \log_4(4r_0^2\varepsilon\mu)$ and $\ell_1 \geq \log_4 \frac{n}{Q}$. $\square$

Theorem 7.1. *The cost of constructing the catalog is at most* $O(n\log(2+\mu))$ *flo-comps and* $O(n)\left(m + \dfrac{m^2}{Q}\log(2+\dfrac{Q}{n}\mu)\right)$ *flops.*

Remark. If $\mu\log(2+\mu) \leq const\, n$ and if m and Q are chosen so that $Q \sim m \sim \log(2+\mu)$, then the number of flops mentioned in the theorem above simplifies to $O(n\log(2+\mu))$.

8. Per Evaluation Cost

In this section we estimate the cost, in terms of floating point operations (flops) and floating point comparisons (flo-comps), of finding a value $V \approx s(z)$ (with $|s(z) - V| \leq \delta$) using Powell's [9] evaluation method as described in section 6. As in the previous section, we set $\mu := \|\lambda\|_1/\delta$ and we adopt the viewpoint that $m_0 \geq 2$ and $r_0 > 0$ are fixed and any quantity which depends only on these values will be considered a constant.

Given $z \in \mathbb{C}$, let $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ be as in the proof of Theorem 6.1. As each page $p \in \mathcal{P}$ is visited, the evaluation algorithm first computes $|z - c_p|^2$ (using 5 flops) and compares this value with the values t_ℓ, r_ℓ, T_ℓ held in the catalog's appendix (using 2 flo-comps). If $p \in \mathcal{P}_2$, then no further flops or flo-comps are performed; while if $p \in \mathcal{P}_1$, then either a summarized value is computed and added to V, at a cost of $O(m)$ flops, or the cluster sum is computed directly and added to V at a cost of $O(Q)$ flops. The total cost, therefore, is at most $2(\#\mathcal{P}_2)$ flo-comps and $5(\#\mathcal{P}_2) + O(m+Q)(\#\mathcal{P}_1)$ flops. Since the parent of each page in $\mathcal{P}_1$ belongs to $\mathcal{P}_2$, it follows that $\#\mathcal{P}_1 \le 4(\#\mathcal{P}_2)$, and we thus arrive at the following.

Proposition. *The cost, per evaluation, is at most* $2(\#\mathcal{P}_2)$ *flo-comps and* $O(m+Q)(\#\mathcal{P}_2)$ *flops.*

In order to estimate $\#\mathcal{P}_2$, we partition $\mathcal{P}_2$ as $\mathcal{P}_2 = \bigcup_{0 \le \ell < \ell\_max} P_\ell$, where P_ℓ contains those pages in $\mathcal{P}_2$ of level ℓ. Note that if $p \in P_\ell$, then either $r_\ell \le |z - c_p| < T_\ell$ or $|z - c_p| < t_\ell$, but since $t_\ell \le r_\ell \le T_\ell$, we have $|z - c_p| < T_\ell$ in either case. Hence c_p lies in the disk with center z and radius T_ℓ for all $p \in P_\ell$.

Lemma. *The number of pages in* P_ℓ *satisfies* $\#P_\ell \le \dfrac{\pi}{2}\left(1 + \dfrac{T_\ell}{r_\ell}\right)^2$, $0 \le \ell < \ell\_max$.

Proof. For $p \in P_\ell$, let Ω_p^o be the open region bounded by the bounding square Ω_p associated with page p. Since Ω_p^o has center c_p and radius r_ℓ, it follows that Ω_p^o has area $2r_\ell^2$ and is contained in the disk having center z and radius $T_\ell + r_\ell$. Since the collection $\{\Omega_p^o\}_{p \in P_\ell}$ is pairwise disjoint, it follows that the sum of the areas, namely $2r_\ell^2(\#P_\ell)$, is less than the area of the mentioned disk. Hence $2r_\ell^2(\#P_\ell) \le \pi(T_\ell + r_\ell)^2$, whence follows the desired inequality. $\square$

Theorem. *The number of pages in* $\mathcal{P}_2$ *satisfies*

$$\#\mathcal{P}_2 \le \pi\left(4 + \left(\frac{2r_0^2\mu}{m^2}\right)^{2/(m-1)}\right)\ell\_max = O(1 + \mu^{2/(m-1)})\log(2+\mu).$$

Proof. Recall from section 5 that $\dfrac{T_\ell}{r_\ell} = \begin{cases} 1 & \text{if } \mathcal{E}_m(1) \le \rho_\ell \\ \mathcal{E}_m^{-1}(\rho_\ell) & \text{if } \mathcal{E}_m(1) > \rho_\ell \end{cases}$, where $\rho_\ell = \dfrac{1}{r_\ell^2\mu}$. It is easily verified that $\mathcal{E}_m(s) \le \frac{2}{m^2}s^{1-m}$, $s \ge 1$, and therefore

$$\frac{T_\ell}{r_\ell} \le \max\{1, \left(\frac{m^2}{2r_\ell^2\mu}\right)^{1/(1-m)}\} \le 1 + \left(\frac{2r_\ell^2\mu}{m^2}\right)^{1/(m-1)}.$$

It then follows from the lemma that $\#P_\ell \le \pi\left(4 + \left(\frac{2r_\ell^2\mu}{m^2}\right)^{2/(m-1)}\right)$, and then the desired estimate is a consequence of $\#\mathcal{P}_2 = \sum_{0 \le \ell < \ell\_max} \#P_\ell$, after noting that $r_\ell \le r_0$. $\square$

Remark 8.1. If m and Q are chosen so that $Q \sim m \sim \log(2+\mu)$, then $\mu^{2/(m-1)}$ is bounded independently of μ, and it follows from the above theorem and proposition that the number of flo-comps employed is $O(\log(2+\mu))$ and the number of flops employed is $O(\log(2+\mu))^2$.

9. Concluding Remarks

The fast evaluation algorithm presented above has been designed to be as simple as possible, while maintaining competitive performance when compared with the methods of Beatson & Newsam [2] and Powell [9]. We mention three noteworthy features of the algorithm.

1. For a given $z \in \mathbb{C}$, our algorithm produces a value V with the guaranteed accuracy

$$|s(z) - V| \le \delta \tag{9.1}$$

2. The clusters are obtained via uniform subdivision. Not only is this easy to implement, it also yields a certain stationarity in the relations between any parent cluster and its children. One important consequence of this is that the moments of a parent cluster can be obtained from those of its children, as detailed in (2.2), where, due to the stationarity, the four lower triangular matrices L_q ($q = I, II, III, IV$) are fixed, and so can be computed in advance.
3. The combination of uniform subdivision with Powell's method of evaluation allows the error estimates, for inner and outer summaries, to be designed so that the threshold radii depend only on the level of the cluster. Since the number of levels under consideration is quite small, we can calculate these threshold radii very precisely without any concern that these computations will add significantly to the total pre-processing costs.

It is interesting to compare the pre-processing and per-evaluation costs obtained in sections 7 and 8 with those of [2]. Although no cost estimates are specifically worked out in [2], it is expected that the cost estimates found in [5] would be applicable. The accuracy obtained in [5] takes the form $|s(z) - V| \le C_0\varepsilon\,|\log\varepsilon|\,\|\lambda\|_1$ for some constant C_0. Since it is not possible to bound the number of clusters (and hence the pre-processing costs) in the algorithm of [5] without placing some restriction on the distribution of the data sites, it is assumed in [5] that the minimum separation distance in the data sites is greater than ε. Under this restriction, they show that the total cost of pre-processing plus n evaluations (at the data sites) is bounded by $O(n\,|\log\varepsilon|^2)$ operations. In order to compare this with our algorithm, we consider the scenario where $\|\lambda\|_1 = n^\gamma/\log n$, for some constant $\gamma > 1/2$, with targeted accuracy (9.1), where $\delta > 0$ is constant. Choosing $\varepsilon_n = \frac{\delta}{C_0\gamma} n^{-\gamma}$, the above accuracy estimate (for [5]) simplifies, asymptotically, to (9.1) at a cost of $O(n(\log n)^2)$ operations. On the other hand, applying the current algorithm to the same scenario, with $q \sim m \sim \log(2+\mu)$, we obtain from Theorem 7.1 that the pre-processing costs are at most $O(n\log n)$ flo-comps and $O(n(\log n)^2)$ flops; while it follows from Remark 8.1 that the per-evaluation costs are at most $O(\log n)$ flo-comps and $O(\log n)^2$ flops. Thus the combined cost of pre-processing and n evaluations is $O(n\log n)$ flo-comps and $O(n(\log n)^2)$ flops. We note that this is exactly the same estimate as for [5].

Our fast evaluation algorithm has been implemented using a combination of the programming languages Octave (similar to Matlab) and C. The fast evaluation and the direct computation of moments (for childless pages) is written in C, while the construction of the catalog is in Octave. Experiments were run on an Intel-based notebook computer with Ξ containing 300,000 points. The tables below display, for various values of δ, the time T_0 (in cpu seconds) required to construct the catalog as well as the total execution time

$T = T_0 + T_1$, where T_1 is the time required to evaluate the function s at all points in Ξ. For each choice of tolerance δ, the displayed run corresponds to a choice of $m \in \{3, 4, 5, \ldots, 40\}$ and $Q \in \{32, 48, 72, \ldots, 243, 365, 547\}$ aimed at minimizing T (with $m_0 = \min(10, m)$). The other reported values are L (the maximum level in the catalog), N (the number of pages in the catalog), and max-error (the maximum absolute error in the computed values of s). We mention that direct evaluation of s at all points in Ξ (implemented in C) requires around 8000 cpu seconds.

Experiment A. The points in Ξ are chosen randomly in a square of radius $\sqrt{2}$.

δ	T	T_0	L	N	m	Q	max-error
0.1	0.70	0.27	1	5	4	≤ 243	0.0069
0.01	1.86	0.50	3	85	5	≤ 243	$6.7e-5$
$1e-4$	7.0	2.6	6	5461	11	≤ 243	$2.1e-7$
$1e-7$	15	2.9	6	5461	23	162	$4.0e-11$

Experiment B. The points in Ξ are chosen randomly on the curve $z = \sin 2t + \imath \cos t$.

δ	T	T_0	L	N	m	Q	max-error
0.1	0.58	0.26	1	5	4	≤ 243	0.0092
0.01	1.6	0.47	3	61	4	≤ 243	$1.9e-4$
$1e-4$	3.8	1.0	6	692	9	≤ 243	$6.1e-7$
$1e-7$	12	4.5	11	9041	18	108	$3.7e-10$

Experiment C. The points in Ξ are chosen randomly in the unit disk, but in a manner which clusters them near the origin. Specifically, $\xi_j = r_j^{200} \exp(\imath \theta_j)$, where r_j and θ_j are chosen randomly in $[0.5, 1]$ and $[0, 2\pi]$, respectively.

δ	T	T_0	L	N	m	Q	max-error
0.1	0.78	0.26	1	5	3	≤ 243	0.047
0.01	1.56	0.59	3	69	9	≤ 243	0.0060
0.0001	6.4	1.0	6	241	9	108	$8.4e-5$
$1e-7$	16	1.8	11	541	13	108	$7.6e-8$

In order to gage the contribution to performance made by the inner summaries, we run the same experiments as above, except that the inner summaries are removed from the algorithm. This is accomplished by applying uniform subdivision to any cluster having more than Q data sites and by setting the inner threshold radii to $t_\ell = r_\ell$.

Experiment A. (no inner summaries)

δ	T	T_0	L	N	m	Q	max-error
0.1	8.0	2.5	6	5461	3	108	0.0012
0.01	8.5	2.6	6	5457	5	243	$1.3e-5$
$1e-4$	10.1	2.7	6	5457	11	243	$2.1e-7$
$1e-7$	15	3.0	6	5461	24	162	$8.5e-12$

Experiment B. (no inner summaries)

δ	T	T_0	L	N	m	Q	max-error
0.1	9.3	4.2	11	9041	3	108	0.0012
0.01	9.6	4.2	11	9041	4	108	$5.2e-5$
$1e-4$	11	4.3	11	9041	9	108	$4.8e-7$
$1e-7$	13	3.5	11	6117	20	162	$1.1e-10$

Experiment C. (no inner summaries)

δ	T	T_0	L	N	m	Q	max-error
0.1	57	13	201	8373	3	108	0.039
0.01	58	13	201	8373	3	108	0.0087
$1e-4$	67	13	201	8373	5	108	$8.2e-5$

$1e-7$	85	13	201	8373	13	108	$7.6e-8$

Remark. The experiments above indicate that the use of inner summaries is advantageous when the tolerance δ is large (relative to machine epsilon) or when the data sites exhibit significant clustering (eg. along a curve or near a single point). We also observe that despite the use of Octave (rather than C) for the pre-processing, the pre-processing time T_0 has never exceeded more than half of the total execution time T.

10. Appendix

The functions u_k and v_k, mentioned after Proposition 4.2, have been rendered as polynomials of the form $u_k(s) = s^{\min(k,2)}(a_{k,5}s^5 + a_{k,4}s^4 + \cdots + a_{k,0})$ and $v_k(s) = s^{\min(k,2)}(b_{k,5}s^5 + b_{k,4}s^4 + \cdots + b_{k,0})$. The coefficients $a_{k,j}$ and $b_{k,j}$ are chosen so that the resultant polynomials u_k and v_k are best approximations of u_k^* and v_k^*, respectively, over $[0, 1]$.

Table 1. Coefficients $a_{k,j}$, $b_{k,j}$, $5 \geq j \geq 0$, $k = 0, 1, \ldots, 11$

j	$a_{0,j} \times 1000$	$a_{1,j} \times 1000$	$a_{2,j} \times 1000$	$a_{3,j} \times 1000$
5	554.367620292417	2893.55886412027	731.031003205713	2341.32661620336
4	−1676.09740708712	−9339.49407628645	−770.874327514406	−8821.49835122952
3	2470.09369659267	11793.6368491615	−3024.32276316878	13474.7641433104
2	−1203.52135249518	−6768.42560284372	7977.14618583141	−10013.6366810069
1	−52.8727006518332	−918.137296707126	−8446.33066898238	2676.59530316383
0	−91.9698566509581	1338.86126255552	4033.35057062844	509.11563622544

j	$a_{4,j} \times 1000$	$a_{5,j} \times 1000$	$a_{6,j} \times 1000$	$a_{7,j} \times 1000$
5	1.9805077189544	−232.610217855045	94.3420623565238	483.544548904333
4	−191.184716761647	1160.13164564459	281.605950358639	−1008.51853450323
3	1234.49631789906	−1574.04739676809	−977.255014605228	507.229394042683
2	−2136.22147856583	429.5647934734	599.720595671105	−46.6406891097388
1	973.142640846853	120.336243973952	−78.4709751669653	9.85058726571184
0	201.120062195939	146.624931531191	113.39071471926	78.3442172097633

j	$a_{8,j} \times 1000$	$a_{9,j} \times 1000$	$a_{10,j} \times 1000$	$a_{11,j} \times 1000$
5	820.188798357444	1030.62107386976	1074.17894164996	1012.30632121686
4	−2219.16234398542	−3022.02281999415	−3255.33606929124	−3141.36052810102
3	2081.52010711887	3184.47676269295	3568.21633474276	3536.67407614013
2	−915.819839589274	−1557.60789160516	−1796.32910237133	−1823.82132240484
1	209.005080795342	360.958684234922	416.178014730804	429.598811570711
0	42.125340160178	17.4630796905773	4.20299165015753	−4.30644933092202

j	$b_{0,j} \times 1000$	$b_{1,j} \times 1000$	$b_{2,j} \times 1000$	$b_{3,j} \times 1000$
5	−801.240385741798	−3601.73621825062	−704.256714216919	−2622.10856576555
4	2571.33601427813	11755.5745993905	416.475444860944	10079.2378604159
3	−3687.84001392965	−14722.6186348106	4384.74192822409	−15609.861547052
2	2839.00024485265	7444.9931927463	−10423.1407503236	11387.7296423684
1	78.7441405406701	1216.16141807741	10204.8847381105	−2730.9048968497
0	0	−2592.37435715301	−4045.37131332166	−587.425826450344

j	$b_{4,j} \times 1000$	$b_{5,j} \times 1000$	$b_{6,j} \times 1000$	$b_{7,j} \times 1000$
5	182.616381517056	475.909865650397	76.4883133075081	−413.337911284215
4	−255.873475183468	−1998.93810133627	−985.283989587266	599.060788149772
3	−1127.41887037728	2485.71814882485	1920.85559649614	165.377125423283
2	2432.00124725409	−762.836676183038	−1098.25882450744	−381.222265210761
1	−1053.25811694934	−66.7525754769067	195.299834704462	108.281714803425
0	−228.067166261059	−166.433994812363	−132.910454222927	−96.0165947386465

j	$b_{8,j} \times 1000$	$b_{9,j} \times 1000$	$b_{10,j} \times 1000$	$b_{11,j} \times 1000$
5	−864.442493894418	−1167.21500736846	−1243.56701410354	−1182.42948213144
4	2196.79262724745	3325.2112659316	3681.98001249953	3592.04612601419
3	−1880.25992188183	−3400.41253079081	−3945.19778814388	−3965.77513405712

2	739.334326334459	1617.03967945454	1942.5754218434	2008.2558307215
1	−153.664681727	−365.140480497584	−439.715909995292	−464.740057362562
0	−51.6487449675453	−20.5940378403927	−5.16563119112753	5.06695923967167

The function $\widehat{\varepsilon}_{m_0}$, mentioned prior to Corollary 4.3, has been rendered as the range-limited polynomial

$$\widehat{\varepsilon}_{m_0}(s) = \begin{cases} e_{m_0}(0) & \text{if } q_{m_0}(s) \geq e_{m_0}(0) \\ e_{m_0}(1) & \text{if } q_{m_0}(s) \leq e_{m_0}(1) \\ q_{m_0}(s) & \text{otherwise} \end{cases}$$

where q_{m_0} is a polynomial of degree 6, obtained as follows.
Recalling that e_{m_0} is continuous and monotonically decreasing on $[0,1]$, and assuming that $e_{m_0}(0) > e_{m_0}(1)$, put $a = \max\{s \in [0,1] : e_{m_0}(s) = e_{m_0}(0)\}$ and $b = \min\{s \in [0,1] : e_{m_0}(s) = e_{m_0}(1)\}$. Then q_{m_0} is chosen as a polynomial q, of degree ≤ 6, which minimizes $\int_a^b q(s)\, ds$ subject to $q' \leq 0$ on $[0,1]$ and $q \geq e_{m_0}$ on $[a,b]$. It follows from these constraints that $\widehat{e}_{m_0} \geq e_{m_0}$ on $[0,1]$, $\widehat{e}_{m_0}$ is monotonically decreasing on $[0,1]$ and $\widehat{e}_{m_0}(0) = e_{m_0}(0)$, $\widehat{e}_{m_0}(1) = e_{m_0}(1)$.

Table 2. Values of $e_{m_0}(0)$ and $e_{m_0}(1)$.

m_0	$e_{m_0}(0)\times 1000$	$e_{m_0}(1)\times 1000$	m_0	$e_{m_0}(0)\times 1000$	$e_{m_0}(1)\times 1000$
3	116.175303273073	83.333333333332	8	119.024609730425	13.8888888888867
4	117.423558079213	50.0000000000003	9	119.158482746193	11.1111111111101
5	118.130701067538	33.3333333333332	10	119.261469158511	9.09090909090689
6	118.558969878066	23.8095238095232	11	119.340234333432	7.57575757575392
7	118.845068773397	17.8571428571421	12	119.402563088249	6.4102564102522

Table 3. Values of coefficients in $q_{m_0}(s) = \sum_{j=0}^{6} c_{m_0,j} s^j$.

j	$c_{3,j}\times 1000$	$c_{4,j}\times 1000$	$c_{5,j}\times 1000$	$c_{6,j}\times 1000$	$c_{7,j}\times 1000$
6	12868.785375	2819.9143359	1262.8387275	623.44303794	322.43140678
5	−37750.034427	−10407.753697	−5536.1395235	−3333.8933502	−2195.8058886
4	42241.403598	14713.503567	9014.5164921	6181.1804295	4592.1883762
3	−22841.751141	−9572.007321	−6463.1429925	−4779.6249028	−3768.1691423
2	5904.3665471	2652.2069637	1827.8056824	1350.93794	1049.6788509
1	−711.6341796	−323.12376551	−220.58310372	−158.9143657	−119.16277733
0	148.93179931	132.84816696	129.17212618	127.03092397	125.76390294

j	$c_{8,j}\times 1000$	$c_{9,j}\times 1000$	$c_{10,j}\times 1000$	$c_{11,j}\times 1000$	$c_{12,j}\times 1000$
6	205.72993756	195.16526285	243.234007	318.46402077	387.18180856
5	−1678.5226738	−1535.6978285	−1601.647968	−1774.8882031	−1941.3057168
4	3788.1937377	3479.7426795	3447.5009659	3571.1852441	3703.2014337
3	−3219.6206525	−2975.4944643	−2898.3075705	−2924.2870039	−2962.4034168
2	880.56436968	800.54618761	767.47608001	766.10956411	769.05071215
1	−96.807988584	−86.082134776	−81.197513979	−80.442202133	−80.259176058
0	125.14999861	124.96074464	124.9669468	125.07685182	125.19238822

References

1. R.K. Beatson & W.A. Light, *Fast evaluation of radial basis functions; Methods for two-dimensional polyharmonic splines*, IMA J. Numer. Anal. **17** (1997), 343–372.
2. R.K. Beatson & G.N. Newsam, *Fast evaluation of radial basis function, I*, Computers Math Applic **24** (1992), 7–19.
3. R.K. Beatson, M.J.D. Powell & A.M. Tan, *Fast evaluation of polyharmonic splines in three dimensions*, IMA J. Numer. Anal. **27** (2007), 427–450.
4. M.D. Buhmann, *Radial basis functions: theory and implementations*, Cambridge Monographs on Applied and Computational Mathematics, **12**, Cambridge University Press, Cambridge, 2003.
5. J. Carrier, L. Greengard & V. Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Statist. Comput. **9** (1988), 669–686.

6. L. van Dommelen & E.A. Rundensteiner, *Fast, adaptive summation of point forces in the two-dimensional Poisson equation*, J. Comput. Phys. **83** (1989), 126–147.
7. G.E. Fasshauer, *Meshfree approximation methods with MATLAB*, Interdisciplinary Mathematical Sciences, **6**, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007.
8. L. Greengard & V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys. **73** (1987), 325–348.
9. M.J.D. Powell, *Truncated Laurent expansions for the fast evaluation of thin plate splines*, Numer. Algorithms **5** (1993), 99–120.
10. H. Wendland, *Scattered data approximation*, Cambridge Monographs on Applied and Computational Mathematics, **17**, Cambridge University Press, Cambridge, 2005.