CONSTRUCTIVE THEORY OF FUNCTIONS, Varna 2002
(B. Bojanov, Ed.), DARBA, Sofia, 2003, pp. 347-353.

Fast Evaluation of Discrete Integral Transforms by Chebyshev and Leja Polynomial Approximation

Stefano De Marchi and Marco Vianello

We present and compare two fast methods for the evaluation of a discrete integral transform $(T_n\mathbf{u})_i = \sum_{j=1}^{n} w_j K(x_i, t_j, u_j)$, $i = 1, ..., q$, where n and q are large, based on polynomial approximation of the action of the transform. The first resorts to *truncated Chebyshev series expansion* while the second reduces to *interpolation at Leja sequences*. Our approach has a different conception compared with other well-known fast methods, which usually work on the kernel and are restricted to linear transforms. Working instead on the action of T_n, we are able to exploit the "*smoothing* effect" of integration and to treat also *nonlinear* instances. Both Chebyshev and Leja polynomial approximation reduce the complexity from $\mathcal{O}(nq)$ to $\mathcal{O}(n + q)$ in several applications, but the Leja approach turns out to be more efficient and shows speed-up ratios, w.r.t. direct evaluation, up to two orders of magnitude.

1. Introduction

This work concerns with the efficient evaluation of discrete integral transforms like

$$(T_n\mathbf{u})_i = \sum_{j=1}^{n} w_j\, K(x_i, t_j, u_j)\,, \qquad 1 \le i \le q\,, \tag{1}$$

which approximate continuous transforms like $Tu(x_i) = \int_a^b K(x_i, t, u(t))\, dt$, where $\{w_j\}$ and $\{t_j\} \in \mathbb{R}^n$ are the quadrature weights and nodes, $\mathbf{u} \in \mathbb{R}^n$ the (approximated) values of $u(t)$ at the quadrature nodes, $u_j \approx u(t_j)$, and $\{x_i\} \in \mathbb{R}^q$ a set of "target" points in $[a, b]$. Since (1) has *dense* structure, its evaluation costs $\mathcal{O}(nq)$ flops, which becomes quite expensive when n and q are "large". For example, in the case of second-kind integral equations discretized by the Nyström method [2], typically $q = n$, so that a *quadratic complexity* arises. In the framework of *linear transforms*, that is, when $K(x, t, u) = H(x, t)u$, several *fast methods* have been proposed, which working *on the kernel* are able to reduce the computational cost to $\mathcal{O}(n \log^p n)$, $p = 1, 2$, or even $\mathcal{O}(n)$ flops. We quote, without any pretence of exhaustivity, the *fast multipole* method by Greengard and Rokhlin [7], *wavelet based* methods [1], and more recently $\mathcal{H}$*-matrix* and *mosaic-skeleton* methods [6, 10].

The *main idea* behind our approach is instead to *approximate* directly the *action* of the transform: computing the discrete transform $T_n\mathbf{u}$ in (1) can be seen in fact as evaluating the scalar function Φ_n,

$$\Phi_n(x) := \sum_{j=1}^{n} w_j \, K(x, t_j, u_j) \approx \Phi(x) := \int_a^b K(x, t, u(t)) \, dt \, , \quad x \in [a, b] \, , \tag{2}$$

at the target points $\{x_i\}$, $1 \leq i \leq q$. By *approximating globally* Φ_n in $[a, b]$ by a polynomial, say $P_{m,n}(x)$, whose construction and evaluation at the target points *costs* $\mathcal{O}(m(n+q))$ flops, with $m \ll n, q$, we obtain a *general purpose* fast method based on *univariate* techniques, which is able to exploit the *smoothing* effect of integration, and works also in *nonlinear* instances.

2. The Proposed Methods

We consider two different methods of *global polynomial approximation*: *truncated Chebyshev series expansion (C)*, and *polynomial interpolation at Leja sequences (L)*.

• Truncated Chebyshev series expansion (C)

This method has been already presented and discussed in [4], where we approximated $\Phi_n(x)$ in $[a, b]$ through its m-degree truncated Chebyshev series expansion,

$$P_{m,n}(x) = \frac{c_0}{2} + \sum_{s=1}^{m} c_s \, T_s \left(\frac{2x - a - b}{b - a} \right) , \tag{3}$$

where T_s are the first-kind Chebyshev polynomials, and the coefficients c_s are obtained by Gauss-Lobatto quadrature, i.e.,

$$c_s = \frac{2}{m \, \omega_s} \sum_{k=0}^{m} \frac{1}{\omega_k} \, \Phi_n(\xi_k^m) \, \cos \frac{\pi k s}{m} \, , \qquad s = 0, ..., m \, ,$$

with $\{\xi_k^m\}$ the *Chebyshev-Lobatto* nodes in $[a, b]$. We studied there the convergence of $P_{m,n}$ to $\Phi_n(x)$ in (3) as m increases and we observed that:

(i) if $K(x, t, u)$ is *smooth* in $x \in [a, b]$ for fixed $t \in [a, b]$ and $u \in D \subseteq \mathbb{R}$, then convergence occurs. In fact, in this case the function Φ_n is smooth itself, with (at least) the same degree of regularity as that of Φ;

(ii) if $K(x,t,u)$ is *nonsmooth* in x (e.g. K is linear weakly-singular, $H(x,t) = \log(|x-t|)$), then Φ_n is less smooth than Φ in (2), and the *error* $\|P_{m,n} - \Phi_n\|_\infty$ exhibits a *stalling* at increasing m, correspondingly to the size of the underlying *discretization error* $\|\Phi_n - \Phi\|_\infty$ which begins to dominate.

The algorithm and more details about *compression rates* and *speed-ups*, *a posteriori error estimation* and the *stalling phenomenon* that occurs in nonsmooth instances, can be found in [4].

• Polynomial interpolation at Leja sequences (L)

Leja [8], defined a sequence of "extremal" points in the following way. Let λ_0 be a point arbitrarily chosen in $[a,b]$; then $\lambda_s \in [a,b]$, $s = 1,2,\dots$, are such that

$$\prod_{k=0}^{s-1} |\lambda_s - \lambda_k| = \max_{x\in[a,b]} \prod_{k=0}^{s-1} |x - \lambda_k| \ .$$

For a Leja sequence $\Lambda = \{\lambda_s\}$ the following important properties hold, cf. [3, 9]: Λ has the same limit distribution as that of the Chebyshev points, and turns out to be "nearly optimal" for polynomial interpolation (slow increase of the Lebesgue constant); Λ is a *stable* sequence for the Newton form of the interpolant; Λ can be *extracted from* a (sufficiently dense) *discretization* of $[a,b]$, even in a "fast" way.

In what follows, with $P_{m,n}$ we denote the polynomial of degree m which interpolates $\Phi_n(x)$ at the *fast Leja points* $\lambda_s, s = 0,\dots,m$, (see [3]) extracted from the target interval, cf. (1). As known, the key feature of Leja-like interpolation is that the polynomial degree m can be incremented by computing the underlying function values *only* at the additional nodes [3]. Indeed, with Chebyshev-like methods this saving in terms of functional evaluations can not be achieved, unless a special degree increment (e.g. doubling) is chosen, which anyway leads to oversampling.

As in the case of truncated Chebyshev series, we studied *a posteriori error estimation* and we detected the *stalling phenomenon.* It is worth noting that the problem of reliably estimating the error of Leja-like interpolation does not seem to have been faced thoroughly in the numerical literature. Let $\Lambda_m = \{\lambda_0,...,\lambda_m\}$ be the set of the first $m+1$ fast Leja points among the target points $X = \{x_i\}$, for a given λ_0 (usually one of the extremal target points). The natural estimate of the approximation error given by $\|P_{m+1,n}(X) - P_{m,n}(X)\|_\infty / \|P_{m+1,n}(X)\|_\infty$ has proved to be misleading in several (even regular) examples. Neither the use of a degree incremental step greater than 1 solves clearly this problem, in our experience.

We tested two alternative strategies for estimating the approximation error. The first has been that of comparing the values of the last interpolant with those of the interpolated function Φ_n at the new c_s Leja nodes (where Φ_n will be in any case evaluated to update the approximation), c_s being a (small) *control* *step*. The relative error is then estimated by

$$E_m = \frac{\|P_{m,n}(\Lambda_{m+c_s} \setminus \Lambda_m) - \Phi_n(\Lambda_{m+c_s} \setminus \Lambda_m)\|_\infty}{\|\Phi_n(\Lambda_{m+c_s} \setminus \Lambda_m)\|_\infty} \tag{4}$$

where $\Lambda_{m+c_s} \setminus \Lambda_m$ is the difference of the two sets. The iteration is stopped either when E_m goes below a given tolerance, or when the quantity $|E_{m+1}/E_m - 1|$ is below a suitable $\theta \in (0,1)$ on two consecutive values of m, i.e., a stalling of convergence occurs. The above a posteriori error estimate E_m works quite well for smooth kernels (even with $c_s = 1$, see Figure 1 *up* and Table 1), but turned out to be often unreliable in nonsmooth instances. In these latter cases, we used the following alternative estimate

$$E_m = \|P_{m,n}(\Delta) - \Phi_n(\Delta)\|_\infty / \|\Phi_n(\Delta)\|_\infty \ , \tag{5}$$

where Δ is a fixed (small) set of *control* points in $[a,b]$.

3. Numerical Examples

In this section we present two illustrative examples, where $[a,b] = [0,1]$ and $\{t_j\} = \{x_i\}$ (that is, $q = n$), which correspond to quite different kernels appearing in the numerical literature on second-kind Fredholm equations [2]. The computational results concerning the truncated Chebyshev series expansion are taken from [4].

1. *Uryshon-type kernel (nonlinear and smooth)*: $K(x,t,u) = \frac{1}{x+t+0.3u}$, $u > 0$; the vectors $\{t_j\}$ and $\{w_j\}$ correspond to the Simpson quadrature rule at n (odd) equispaced nodes, and we chose $u(t) = 1/(1+t)$.

2. *Weakly-singular log kernel (linear and nonsmooth)*: here, $K(x,t,u) = \log(|x-t|)\,u(t)$, $\{t_j\}$ and $\{w_j\}$ correspond to the trapezoidal quadrature rule at n (even) equispaced nodes, and we chose $u(t) \equiv 1$; the stalling control parameter is set to $\theta = 0.25$.

In both cases we know the exact continuous transform $\Phi(x)$, cf. (2), and thus the underlying discretization error. In the Tables below we use the following notations ($X = \{x_i\}$ being the target points):

- *discretization error* : $DE = \|\Phi(X) - \Phi_n(X)\|_\infty / \|\Phi(X)\|_\infty$;
- *actual error* : $AE = \|\Phi_n(X) - P_{m,n}(X)\|_\infty / \|\Phi_n(X)\|_\infty$;
- *estimated error* at degree m: $EE = E_m$, cf. (4) for Table 1 and (5) for Table 2 as for Leja-like interpolation, while for the Chebyshev series approach we resorted to the remainder estimate in [4, formula (10)];

- *estimated speed-up* (w.r.t. direct evaluation): $SU = n/m$, m being the displayed exit degree, such that $E_m \leq tol$ or a stalling has been detected. When the error estimate (5) is used, then $SU = SU_\Delta = \frac{n^2}{nm+C_\Delta n+C_\Delta m} \approx \frac{n}{m+C_\Delta}$, C_Δ being the cardinality of the "control" set Δ (see Table 2 where Δ is chosen as the set of the first ten Chebyshev points in $[0,1]$).

n	129		513		2049	
DE	$1.8\,10^{-9}$		$6.9\,10^{-12}$		$3.0\,10^{-14}$	
	(C)	(L)	(C)	(L)	(C)	(L)
m	17	13	17	13	17	13
EE	$7.2\,10^{-7}$	$5.7\,10^{-7}$	$7.2\,10^{-7}$	$5.7\,10^{-7}$	$7.2\,10^{-7}$	$5.7\,10^{-7}$
AE	$1.1\,10^{-6}$	$5.1\,10^{-7}$	$1.1\,10^{-6}$	$5.1\,10^{-7}$	$1.1\,10^{-6}$	$5.1\,10^{-7}$
SU	3.3	9.9	13.2	39.5	52.5	157.6

Table 1: Compression of the discrete nonlinear Uryshon-type transform in Example 1, *tol* $= 10^{-6}$, with Chebyshev (C) and Leja (L).

n	128		512		2048	
DE	$3.2\,10^{-2}$		$9.5\,10^{-3}$		$2.8\,10^{-3}$	
	(C)	(L)	(C)	(L)	(C)	(L)
m	11	9	26	9	61	9
EE	$8.2\,10^{-3}$	$1.5\,10^{-2}$	$2.9\,10^{-3}$	$8.8\,10^{-3}$	$7.1\,10^{-4}$	$5.9\,10^{-3}$
AE	$4.0\,10^{-2}$	$3.7\,10^{-2}$	$1.0\,10^{-2}$	$1.1\,10^{-2}$	$3.1\,10^{-3}$	$4.4\,10^{-3}$
SU	5.8	6.7	7.9	26.9	12.3	107.8

Table 2: Compression of the discrete "weakly-singular log" linear transform in Example 2; here, the exit is forced by a stalling of the estimated error.

Some comments are now in order. First, we stress the effectiveness of polynomial compression for discrete integral transforms, especially Leja-like interpolation, which is more efficient than Chebyshev-like methods, and shows *speed-ups* up to *two orders of magnitude*. Even when we reach a degree which guarantees an error of the order of the underlying discretization error (see Figure 1), we get for the medium-size problem with $n \approx 512$ speed-ups ≥ 20. Notice also the reliability of the error estimates, and of the stalling detection in the weakly-singular instance. More details on the Leja-like method as well as a wider set of numerical examples can be found in the forthcoming paper [5].

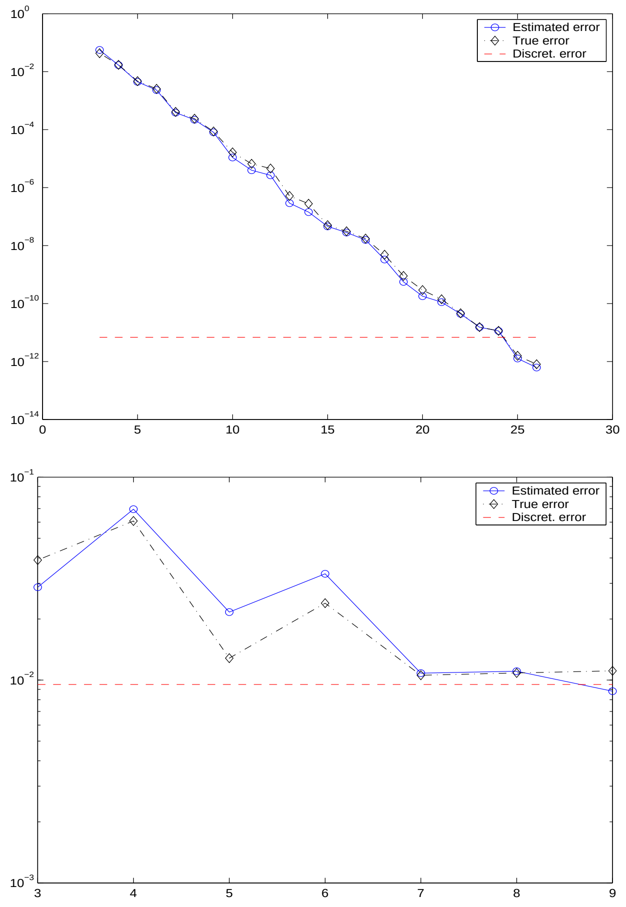

Figure 1: (Log-scale approx. errors vs. polynomial degree.) *Up*: Leja-like compression of the Uryshon-type transform, $n = 513$, error estimate (4); $SU = 21.4$. *Down*: Leja-like compression of the discrete "weakly-singular log" linear transform, $n = 512$, error estimate (5); $SU = 26.9$.

References

[1] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin, Wavelet-like bases for the fast solution of second-kind integral equations, *SIAM J. Sci. Comput.* **14** (1993), 159–184.

[2] K. E. Atkinson, A survey of numerical methods for solving nonlinear integral equations, *J. Integral Equations Appl.* **4** (1992), 15–46.

[3] J. Baglama, D. Calvetti, and L. Reichel, Fast Leja points, *Elect. Trans. Numer. Anal.* **7** (1998), 124–140.

[4] S. De Marchi and M. Vianello, Approximating the approximant: a numerical code for polynomial compression of discrete integral operators, *Numer. Algorithms* **28** (2001), 101–116.

[5] S. De Marchi and M. Vianello, Fast evaluation of discrete integral transforms by polynomial interpolation at Leja sequences, in preparation.

[6] W. Hackbusch and B. N. Khoromskij, Towards $\mathcal{H}$-matrix approximation of the linear complexity, *in* "Operator Theory: Advances and Applications", vol. 121, pp. 194–220, Birkhäuser, 2001.

[7] L. Greengard and V. Rohklin, A fast algorithm for particle simulations, *J. Comput. Phys.* **73** (1987), 325–348.

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

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

[10] E. E. Tyrtyshnikov, Incomplete cross approximation in the mosaic-skeleton method, *Computing* **64** (2000), 367–380.

Stefano De Marchi
Dept. of Computer Science
University of Verona
Strada Le Grazie, 15
37134 Verona - ITALY
E-mail: `demarchi@sci.univr.it`

Marco Vianello
Dept. of Pure and Applied Mathematics
University of Padova
Via Belzoni, 7
35131 Padova - ITALY
E-mail: `marcov@math.unipd.it`