Adaptive bivariate Chebyshev approximation*

Alvise Sommariva, Marco Vianello †
and Renato Zanovello
Department of Pure and Applied Mathematics
University of Padova
via Belzoni 7, 35131 - Padova (Italy)

Abstract

We propose an adaptive algorithm which extends Chebyshev series approximation to bivariate functions, on domains which are smooth transformations of a square. The method is tested on functions with different degrees of regularity and on domains with various geometries. We show also an application to the fast evaluation of linear and nonlinear bivariate integral transforms.

2000 AMS subject classification: 41A10, 65D10, 65R10.

1 Introduction.

Chebyshev series expansion

$$f(x) = \sum_{i=0}^{\infty} c_i T_i(x) \ , \quad x \in [-1,1] \ ,$$

$$c_i = \frac{2}{\pi} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} \, f(x) \, T_i(x) \, dx \ , \ i > 0 \ , \quad c_0 = \frac{1}{\pi} \int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}} \, dx \ , \qquad (1)$$

is a useful and popular tool for the approximation of sufficiently regular univariate functions (in (1), $T_i(x)$ denotes as usual the i-th degree first-kind Chebyshev polynomial). In fact, the partial sums of Chebyshev series provide asymptotically optimal polynomial approximations, which can be constructed in a very efficient way by resorting to the FFT algorithm; see, e.g., the classical reference [23], the comprehensive monograph [17], and the various algorithms in the

*Work supported by the research project CPDA028291 "Efficient approximation methods for nonlocal discrete transforms" of the University of Padova, and by the GNCS-INdAM.
†Corresponding author: e-mail: *marcov@math.unipd.it*

NETLIB and CERNLIB repositories for the practical implementations. It is also worth quoting here the "CHEBFUN" object-oriented Matlab system [4], which has been recently developed in order to extend basic Matlab functions to the continuous context, via Chebyshev approximation. On the other hand, concerning the extension of Chebyshev approximation to bivariate functions only some scattered results, restricted to the case of rectangular domains (see, e.g., [21]), seem to have appeared in the literature. In particular, an adaptive algorithm for bivariate Chebyshev approximation, taking into account the different behaviors of the underlying function in different parts of the domain, and working also on a wide class of domain geometries, does not seem to be yet available.

In the following subsections we describe the analytical construction of the approximation on the square $[-1,1]^2$, we discuss its practical implementation, and we present two numerical tests. In Section 2 we extend the Chebyshev approximation method to other bivariate domains, namely domains which are smooth transformations of a square, and we give several numerical examples concerning functions with different degree of regularity on domains with various geometries. In Section 3 we apply adaptive Chebyshev approximation to the fast evaluation of linear and nonlinear bivariate integral transforms.

2 Adaptive Chebyshev approximation on a square.

2.1 Analytical construction.

Let us begin with the simplest case, that is a function $f(x,y)$ defined on the square $[-1,1]^2$, the adaptation to rectangular domains being straightforward. The basic idea is simple: suppose that $f(x,y)$ can be expanded in Chebyshev series in y (uniformly in x), then we get $f(x,y)=\sum_{i=0}^{\infty} c_i(x)T_i(y)$. Moreover, suppose that we can expand each coefficient $c_i(x)$ in Chebyshev series (uniformly in i), i.e. $c_i(x)=\sum_{j=0}^{\infty} c_{ij}T_j(x)$. Then, it is natural to seek a polynomial approximation of the form

$$f(x,y) \approx \sum_{i=0}^{n} \sum_{j=0}^{m_i} c_{ij}\, T_j(x)\, T_i(y) \;, \tag{2}$$

obtained by suitable truncation of the univariate Chebyshev expansions described above. This is made rigorous by the following result, which rests on the natural bivariate extension of the well-known Dini-Lipschitz condition for uniform convergence of univariate Chebyshev series [17, 23].

Theorem 1 *Let $f:[-1,1]^2 \to \mathbb{R}$ be continuous and such that*

$$\log(n)\, osc(f;[-1,1]^2;1/n) \to 0 \;\; as \;\; n \to \infty \;, \tag{3}$$

where $osc(f;[-1,1]^2;1/n)$ denotes the $1/n$-oscillation of f in $[-1,1]^2$, i.e.

$$osc(f;[-1,1]^2;1/n) := \max_{|P-Q|\le 1/n} \{|f(P)-f(Q)|,\, P,Q \in [-1,1]^2\} \;. \tag{4}$$

For every $\varepsilon > 0$, there exist an index $n(\varepsilon)$, and a sequence $m_i = m(i;\varepsilon)$, such that the following bivariate polynomial approximation of $f(x,y)$ with degree $\max_i \{i+m_i\}$ *holds*

$$\max_{(x,y)\in[-1,1]^2} |f(x,y) - p(x,y)| \leq \varepsilon \;, \;\; p(x,y) = \sum_{i=0}^{n(\varepsilon)} \sum_{j=0}^{m_i} c_{ij}\, T_j(x)\, T_i(y) \;, \qquad (5)$$

where the coefficients $\{c_{ij}\}$ are given by

$$c_{ij} = \frac{2\,\hat{\delta}_j}{\pi} \int_{-1}^{1} \left(\frac{2\,\hat{\delta}_i}{\pi} \int_{-1}^{1} \frac{1}{\sqrt{1-y^2}}\, f(x,y)\, T_i(y)\, dy \right) \frac{1}{\sqrt{1-x^2}}\, T_j(x)\, dx \;, \qquad (6)$$

having defined $\hat{\delta}_k = 1 - \delta_{k0}/2$ by the usual Kronecker symbol.

Proof. First, we prove that for every fixed $\varepsilon > 0$, there exists an index $n(\varepsilon)$ such that

$$\left\| f(x,\cdot) - \sum_{i=0}^{n(\varepsilon)} c_i(x)\, T_i(\cdot) \right\|_\infty \leq \varepsilon/2 \;, \qquad (7)$$

uniformly in x. In view of Jackson's theorem [22], this is an immediate consequence of the chain of inequalities

$$\left\| f(x,\cdot) - \sum_{i=0}^{n} c_i(x)\, T_i(\cdot) \right\|_\infty \leq (1+\Lambda_n)\, \|f(x,\cdot) - p_n^*(\cdot)\|_\infty$$

$$\leq 6\,(1+\Lambda_n)\, \mathrm{osc}(f(x,\cdot);[-1,1];1/n) \leq 6\,(1+\Lambda_n)\, \mathrm{osc}(f;[-1,1]^2;1/n) \;, \qquad (8)$$

where p_n^* denotes the degree n polynomial of best uniform approximation for $f(x,\cdot)$ in $[-1,1]$, and the *Lebesgue constant* Λ_n is estimated as $\Lambda_n \leq 3 + (4/\pi^2)\log n$, cf. [17, 23].

Recalling that the Chebyshev coefficients $c_i(x)$ are defined by

$$c_i(x) = \frac{2\,\hat{\delta}_i}{\pi} \int_{-1}^{1} \frac{1}{\sqrt{1-y^2}}\, f(x,y)\, T_i(y)\, dy \;, \qquad (9)$$

with the same reasoning as above we get the chain of inequalities

$$\left\| c_i - \sum_{j=0}^{m} c_{ij}\, T_j \right\|_\infty \leq 6\,(1+\Lambda_m)\, \mathrm{osc}(c_i;[-1,1];1/m)$$

$$\leq \frac{12}{\pi}\,(1+\Lambda_m) \int_{-1}^{1} \frac{1}{\sqrt{1-y^2}}\, \mathrm{osc}(f(\cdot,y);[-1,1];1/m)\, dy$$

$$\leq 48\,(1+\frac{\log m}{\pi^2})\, \mathrm{osc}(f;[-1,1]^2;1/m) \;, \qquad (10)$$

where the Chebyshev coefficients c_{ij} of c_i are defined as in (6). By observing that, for every $\varepsilon > 0$, in view of (10) and (3) we can take $m_i = m(i, \varepsilon)$ such that

$$\left\| c_i - \sum_{j=0}^{m_i} c_{ij}\, T_j \right\|_\infty \leq \frac{\varepsilon}{2\,(1 + n(\varepsilon))} \,, \tag{11}$$

uniformly in i, we finally obtain

$$\max_{(x,y)\in[-1,1]^2} \left| f(x,y) - \sum_{i=0}^{n(\varepsilon)} \sum_{j=0}^{m_i} c_{ij}\, T_j(x)\, T_i(y) \right| \leq \max_{x\in[-1,1]} \left\| f(x,\cdot) - \sum_{i=0}^{n(\varepsilon)} c_i(x)\, T_i(\cdot) \right\|_\infty$$

$$+ \max_{(x,y)\in[-1,1]^2} \left| \sum_{i=0}^{n(\varepsilon)} \left(c_i(x) - \sum_{j=0}^{m_i} c_{ij}\, T_j(x) \right) T_i(y) \right|$$

$$\leq \frac{\varepsilon}{2} + \sum_{i=0}^{n(\varepsilon)} \left\| c_i - \sum_{j=0}^{m_i} c_{ij}\, T_j \right\|_\infty \leq \frac{\varepsilon}{2} + \sum_{i=0}^{n(\varepsilon)} \frac{\varepsilon}{2\,(1 + n(\varepsilon))} = \varepsilon \,. \tag{12}$$

□

Some comments on the approximation result above are now in order. First, observe that condition (3) is satisfied by any globally Hölder continuous function on the square. Moreover, it is clear that as in the univariate case, the smoother the function f, the faster the convergence of the method. This can be justified by estimating the decay rate of the Chebyshev coefficients. Indeed, it has been shown in [21] that when $f \in C^{m+\alpha}[-1,1]^2$, i.e. f has m-th partial derivatives which are Hölder continuous of order α, $m + \alpha > 0$, then $c_i(x) = O(i^{-m-\alpha})$ uniformly in x, and $c_{ij} = O(\min\{i^{-m-\alpha}, j^{-m-\alpha}\})$. We'll not deepen further this aspect, which could be important for example in order to derive *a priori* convergence estimates. On the other hand, the adaptive algorithm described in the next section is able to exploit automatically the regularity features of the function f, and is based on *a posteriori* error estimates.

Observe finally that all the information necessary to reconstruct the function f on the square, up to the given tolerance ε, is compressed in the coefficients matrix $\{c_{ij}\}$ $i = 0, \ldots, n(\varepsilon)$, $j = 0, \ldots, \max m_i$. This matrix in general is not full, but has a null "right-lower" portion. In fact, the coefficients $c_i(x)$ are infinitesimal (uniformly in x) as $i \to \infty$, so that the trend of the sequence m_i is nonincreasing. By using a suitable data structure, the memory allocation for the polynomial compression of $f(x,y)$ up to ε is given by $\sum_{i=0}^{n(\varepsilon)} (m_i + 1)$ floating-point numbers.

2.2 Implementation of the method.

The practical implementation of the bivariate Chebyshev approximation method follows strictly the construction in the proof of Theorem 1, and is based on efficient univariate Chebyshev approximation. For this purpose, we selected the robust Fortran code [13] from the CERNLIB, and we translated it in the C programming language with some minor adaptations in view of the bivariate embedding. The most important of these adaptations are: we have allowed discrete values of the function at the Chebyshev-Lobatto nodes as inputs (the original code requires an analytic definition), since the Chebyshev coefficients are evaluated as usual by Gauss-Lobatto quadrature, and only the discrete values at Chebyshev-Lobatto nodes are really required; moreover, we have introduced an additional stopping criterion based on a possible "stalling" of convergence. The "regular" exit is based on a classical and somehow empirical *a posteriori* estimate [6], namely looking for the first triple of consecutive coefficients whose sum of absolute values is below a given tolerance; on the other hand, the output error estimate uses for safety the complete tail of the last computed coefficients. Other important features of the univariate code are that it doubles iteratively the computed coefficients, minimizing the computational effort via a clever use of the FFT, and moreover that it minimizes a posteriori the final number of output Chebyshev coefficients, by eliminating backward those whose contribution to the tail (sum of absolute values) remains below a suitable threshold. Observe that, as it is well-known, constructing the discrete univariate Chebyhev approximation by Gauss-Lobatto quadrature entails that the corresponding polynomial is also interpolant at the Chebyshev-Lobatto nodes. Clearly, this feature will be inherited by the bivariate method, which becomes ultimately an interpolation method.

We can now give a brief description of the main features of the adaptive Chebyshev algorithm for constructing the bivariate approximating polynomial $p(x, y)$ in (5), by using a pseudo-code notation.

2.2.1 Adaptive algorithm for bivariate Chebyshev approximation:

(I) Set the initial number of Chebyshev subintervals $m := m_0$ (that is $m_0 + 1$ Chebyshev nodes, default $m_0 = 2$), the error tolerance ε, the stalling detection parameter σ (default $\sigma := 0.1$), the maximum number of doubling steps $itmax$; initialize the iteration counter $it := 0$.

(II) Take vertical "cuts" on the square, corresponding to the new Chebyshev-Lobatto nodes $\xi_j = \cos(j\pi/m)$, with $j = 0, 1, ..., m$ if $m = m_0$ ($it = 0$), and $j = 1, 3, 5, \ldots, m-1$ (only odd values) if $m > m_0$ ($it > 0$): approximate the function $f(x, y)$ by the univariate code as a truncated Chebyshev series $\tilde{f}(x, y) = \sum_{i=0}^{n(x;\theta\varepsilon)} c_i(x)T_i(y)$ at each cut $x = \xi_j$, up to a fraction θ of the global tolerance (say $\theta = 0.5$). This step provides the values $\{c_i(\xi_j)\}$ of the coefficient functions $c_i(x)$, $i = 0, 1, ..., n(\theta\varepsilon) =: \max_j n(\xi_j; \theta\varepsilon)$, at the Chebyshev x-nodes $\{\xi_j\}$, $j = 0, \ldots, m$ (the values corresponding to even indexes have already been computed in the previous iterations).

(III) For every $i = 0, 1, ..., n(\theta\varepsilon)$, passing the new values $\{c_i(\xi_j)\}$ to the univariate code, produce the Chebyshev approximation $\tilde{c}_i(x) = \sum_{j=0}^{m} c_{ij}T_j(x)$, and check whether

$$\|\tilde{c}_i - c_i\|_\infty \leq (1-\theta)\varepsilon/(1 + n(\theta\varepsilon)) \ . \quad (13)$$

Here we use the error estimate

$$\|f-p\|_\infty \le \|f-\tilde{f}\|_\infty + \|\tilde{f}-p\|_\infty \approx \max_j \|f(\xi_j,\cdot)-\tilde{f}(\xi_j,\cdot)\|_\infty + \sum_{i=0}^{n(\theta\varepsilon)} \|\tilde{c}_i - c_i\|_\infty$$

$$\le \theta\varepsilon + (1+n(\theta\varepsilon)) \max_i \|\tilde{c}_i - c_i\|_\infty \,. \qquad (14)$$

If the stopping criterion (13) is not satisfied (the approximation in the x-direction is not satisfactory for some index i), return to (II) with $m := 2m$, $it := it + 1$. □

It is worth noting the full adaptivity of this algorithm, which is further improved by some implementation tricks. For example, as usual the global tolerance is taken as a combination of a relative and an absolute one, $\varepsilon = \varepsilon_r \|f\|_\infty + \varepsilon_a$, where the estimate of the maximum norm of the function is dynamically updated when new cuts are added. Moreover, whenever a stalling of convergence occurs, for example during the Chebyshev approximation along a cut, the absolute tolerance ε_a is automatically modified and set to a value close to the estimated size of the error at stalling, in order to avoid overprecision (and thus saving computational work) on other cuts, or in the x-direction. The error stalling is detected by the univariate code, namely by comparing the triple of coefficients indexed by $n-2$, $n-1$, n, with those indexed by $2n-2$, $2n-1$, $2n$, where $2n$ is the last degree of the Chebyshev approximation. In practice, the code checks whether the ratio of two consecutive error estimates belongs to a given neighborhood of 1, say $[1-\sigma, 1+\sigma]$; the stalling phenomenon is typical of low regularity of the function, for example in the presence of noise, or in the application to discrete integral transforms with weakly-singular kernels (see [8, 25] for the univariate case, [16] and section 3 below for bivariate instances).

2.2.2 Two numerical examples.

In order to give a first illustration of the performance of our implementation of adaptive bivariate Chebyshev approximation, we apply the method to the reconstruction of Franke's function [11],

$$f(x,y) = \frac{3}{4}\, e^{-(9x-2)^2/4-(9y-2)^2/4} + \frac{3}{4}\, e^{-(9x-2)^2/49-(9y-2)^2/10}$$

$$+\frac{1}{2}\, e^{-(9x-7)^2/4-(9y-3)^2/4} - \frac{1}{5}\, e^{-(9x-4)^2-(9y-7)^2} \;,\;\; (x,y) \in [0,1]^2 \,, \qquad (15)$$

which is a popular test in the framework of bivariate interpolation. Recall that, as has been already stressed above, the fully discrete Chebyshev algorithm gives in practice also a polynomial interpolation technique.

In Table 1 we report the results corresponding to the application of the adaptive Chebyshev algorithm at various tolerances (extension of the method from $[-1,1]^2$ to general squares and rectangles is immediate by the usual interval transformation). The true relative errors have been computed on a suitable

Table 1: Adaptive Chebyshev approximation of Franke's function (15) at various relative tolerances ε_r.

	coeffs	nodes	est. error	rel. error
$\varepsilon_r = 10^{-3}$	336	625	$2 \cdot 10^{-3}$	$7 \cdot 10^{-4}$
$\varepsilon_r = 10^{-6}$	878	2145	$2 \cdot 10^{-6}$	$5 \cdot 10^{-7}$
$\varepsilon_r = 10^{-9}$	1441	2913	$1 \cdot 10^{-9}$	$3 \cdot 10^{-10}$

control grid. Observe that the overall number of function evaluations at the sampling Chebyshev nodes is higher than the number of output Chebyshev coefficients, since as already observed the code is able to discard insignificant coefficients.

For the sake of comparison with another widely used bivariate method, we show in Table 2 the relative errors given by interpolation of the same function with Radial Basis Functions (RBF) with a suitable scaling (see, e.g., [5]), at three uniform $N \times N$ grids in $[0,1]^2$, corresponding to the integer squares N^2 closest to the number of nodes (i.e. of function evaluations) required by the Chebyshev algorithm.

In order to illustrate the performance of the adaptive Chebyshev algorithm on a less regular function, we consider $f(x,y) = (x^2+y^2)^{5/2}$ on $[-1,1]^2$ and on $[0,2]^2$, which is of class C^4 with Lipschitz-continuous fourth partial derivatives, but has fifth partial derivatives that are discontinuous at the origin. The results are reported in Table 3, where one can observe that the algorithm performs better when the singularity is located at a corner of the square, since Chebyshev sampling nodes cluster at the sides and especially at the corners. This suggests that, whenever the location of a singularity (of the derivatives) is known, the square (rectangle) should be split into four subrectangles, with the singularity in the common vertex.

In Table 4 a comparison is made with classical bicubic interpolation on three uniform $N \times N$ grids in $[-1,1]^2$ (Matlab "INTERP2" function), again corresponding to the integer squares N^2 closest to the number of nodes (i.e. of function evaluations) required by the adaptive Chebyshev algorithm.

3 Other domains.

We can now face more general situations. Consider a function f defined on a bivariate compact domain Ω, that corresponds to the square $[-1,1]^2$ through the smooth surjective transformation

$$\tau : [-1,1]^2 \to \Omega \;, \quad (X,Y) \mapsto (x(X,Y), y(X,Y)) \;. \tag{16}$$

Table 2: Relative approximation errors of Franke's function (15) by RBF interpolation at three uniform $N \times N$ grids in $[0,1]^2$, with Thin-Plate Splines (TPS), Inverse Multi-Quadrics (IMQ), Wendland's C^2 compactly supported functions (W2).

	$625 = 25^2$ nodes	$2116 = 46^2$ nodes	$2916 = 54^2$ nodes
TPS	$1 \cdot 10^{-3}$	$5 \cdot 10^{-4}$	$3 \cdot 10^{-4}$
IMQ	$7 \cdot 10^{-6}$	$4 \cdot 10^{-7}$	$7 \cdot 10^{-7}$
W2	$7 \cdot 10^{-4}$	$2 \cdot 10^{-4}$	$1 \cdot 10^{-4}$

Then, we can construct by the method described above the Chebyshev-like polynomial approximation $p(X,Y) \approx F(X,Y) := f(x(X,Y), y(X,Y))$ on the square $[-1,1]^2 \ni (X,Y)$, and finally obtain an approximation of the form

$$f(x,y) \approx \phi(x,y) = p(X(x,y), Y(x,y)) = \sum_{i=0}^{n(\varepsilon)} \sum_{j=0}^{m_i} c_{ij}\, T_j(X(x,y))\, T_i(Y(x,y)) \;, \tag{17}$$

$(x,y) \in \Omega$, which is in general no longer polynomial. In (17), $(X(x,y), Y(x,y))$ denotes the "inverse" transformation $\tau^{-1} : \Omega \setminus \Omega_S \to [-1,1]^2$, which is allowed to be undefined only in a finite number of singular points Ω_S.

Observe that, from the theoretical point of view, when the function f is globally Hölder-continuous in Ω, a Hölder-continuous transformation suffices to ensure convergence of the Chebyshev-like approximation method. However, a key point in order to avoid loss of smoothness in this process and thus an artificial slowing down of convergence, is to choose a transformation as smooth as possible, and in any case with at least the same degree of regularity of the function f. This role of the transformation will be clarified by the example in Table 6 below. Now we are ready to describe four important classes of domain geometries, with corresponding transformations (the terminology being that usual in the field of numerical cubature).

3.1 Generalized rectangles (Cartesian coordinates).

The domain Ω is defined by

$$\Omega = \{(x,y) : \; a \leq x \leq b \,,\; g_1(x) \leq y \leq g_2(x)\} \;, \tag{18}$$

g_1 and g_2 being suitable functions (these are the typical domains where double integrals can be splitted). Here the transformation is

$$x(X,Y) = x(X) = a + (X+1)\frac{b-a}{2} \;,\;\; y(X,Y) = g_1(x) + (Y+1)\frac{g_2(x) - g_1(x)}{2} \;, \tag{19}$$

Table 3: Adaptive Chebyshev approximation of $f(x,y) = (x^2+y^2)^{5/2}$ on $[-1,1]^2$ and on $[0,2]^2$, at various relative tolerances.

		coeffs	nodes	est. error	rel. error
$\varepsilon_r = 10^{-3}$	dom=$[-1,1]^2$	51	289	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
	dom=$[0,2]^2$	48	81	$5 \cdot 10^{-6}$	$2 \cdot 10^{-6}$
$\varepsilon_r = 10^{-6}$	dom=$[-1,1]^2$	223	673	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$
	dom=$[0,2]^2$	99	249	$2 \cdot 10^{-7}$	$1 \cdot 10^{-7}$
$\varepsilon_r = 10^{-9}$	dom=$[-1,1]^2$	1399	4929	$3 \cdot 10^{-8}$	$3 \cdot 10^{-8}$
	dom=$[0,2]^2$	260	529	$1 \cdot 10^{-9}$	$6 \cdot 10^{-10}$

Table 4: Relative approximation errors of $f(x,y) = (x^2+y^2)^{5/2}$ by bicubic interpolation at three $N \times N$ uniform grids in $[-1,1]^2$.

	$289 = 17^2$ nodes	$676 = 26^2$ nodes	$5041 = 71^2$ nodes
bicubic interp.	$3 \cdot 10^{-3}$	$8 \cdot 10^{-4}$	$4 \cdot 10^{-5}$

Table 5: Adaptive Chebyshev approximation of $f(x,y) = e^x(\sin(y) + xy^2)$ on the generalized rectangle $-2 \leq x \leq 1$, $\sin(x) - 2 \leq y \leq \log(x+3)$ (Cartesian coordinates); the absolute and relative tolerances have been set to $\varepsilon_a = 10^{-8}$, $\varepsilon_r = 10^{-6}$.

	coeffs	nodes	est. error	rel. error
Cartesian	124	289	$6 \cdot 10^{-8}$	$2 \cdot 10^{-8}$

and its inverse is given by

$$X(x,y) = X(x) = -1 + 2\frac{x-a}{b-a} \ , \ \ Y(x,y) = -1 + 2\frac{y - g_1(x)}{g_2(x) - g_1(x)} \ . \qquad (20)$$

The latter is not defined at the possible points $(\overline{x}, \overline{y})$ where $g_1(\overline{x}) = g_2(\overline{x})$, but this is not a real problem, since in such cases the Chebyshev series on the corresponding X-cut is constant, and our algorithm manages the situation computing $p(X(\overline{x}), Y(\overline{x}, \overline{y})) = \sum_{j=0}^{m_0} c_{0j} \, T_j(X(\overline{x}))$, cf. (17), (20). The regularity of the transformation is clearly given by the regularity of the functions g_1 and g_2.

An example of adaptive Chebyshev approximation of an extremely regular function on a generalized rectangle can be found in Table 5.

3.2 Generalized sectors (polar coordinates).

These are defined by $\theta_1 \leq \theta \leq \theta_2$, $\rho_1(\theta) \leq \rho \leq \rho_2(\theta)$, and the transformation is the composition of one analogous to (19) with the change from Cartesian to polar coordinates, with inverse

$$X(x,y) = -1 + 2\,\frac{\theta - \theta_1}{\theta_2 - \theta_1} \ , Y(x,y) = -1 + 2\,\frac{\rho - \rho_1(\theta)}{\rho_2(\theta) - \rho_1(\theta)} \ ,$$

$$\text{where} \quad \rho = \rho(x,y) = \sqrt{x^2 + y^2} \ , \ \ \theta(x,y) = \arctan(y/x) \ . \qquad (21)$$

Observe that in practice we approximate the function $g(\rho, \theta) = f(\rho \cos\theta, \rho \sin\theta)$ on a generalized rectangle in polar coordinates, and that the "cuts" correspond to fixing the angles. The special case of the origin is managed by choosing $\theta(0,0) = 0$, while the angles where $\rho_1 = \rho_2$ are treated as above. Again, the regularity of the transformation is determined by the functions ρ_1 and ρ_2. The simplest case is that of a circle of radius R centered at the origin, i.e. $0 \leq \theta \leq 2\pi$, $0 \leq \rho \leq R$. Notice that the transformation is analytic in this case, while that corresponding to the circle represented directly in Cartesian coordinates is not even C^1, since we have $g_1(x) = -\sqrt{R^2 - x^2}$, $g_2(x) = \sqrt{R^2 - x^2}$, which have singular derivatives at $x = \pm R$.

We stress this fact by showing Table 6, which illustrates the importance of choosing of the right transformation. Notice that, while the function is extremely smooth, the choice of representing the unit circle in Cartesian coordinates leads to computational failure, since the singularity of the transformation

Table 6: Adaptive Chebyshev approximation of $f(x, y) = \cos(x + y)$ on the unit circle in Cartesian and polar coordinates; the absolute and relative tolerances have been set to $\varepsilon_a = 10^{-8}$, $\varepsilon_r = 10^{-6}$.

	coeffs	nodes	est. error	rel. error
Cartesian	4114	57081	$7 \cdot 10^{-6}$	$7 \cdot 10^{-6}$
polar	196	393	$3 \cdot 10^{-7}$	$1 \cdot 10^{-7}$

entails very slow convergence (the relative error in $\|\cdot\|_\infty$ is computed by comparison with the exact values on a suitable control grid).

3.2.1 Starlike domains in polar coordinates.

An important subclass of generalized sectors is given by starlike domains, i.e. $[\theta_1, \theta_2] = [0, 2\pi]$, $\rho_1(\theta) \equiv 0$ (up to a translation). Here a different transformation can be defined, which allows more symmetry in the cuts, which now cluster at both $\theta = 0$ and at $\theta = \pi$ instead of only at $\theta = 0$, and in addition avoids clustering of sampling nodes at the origin. This is obtained by varying the angle θ in $[0, \pi]$, and by allowing negative values of ρ in order to span the negative half-plane, in the following way

$$\theta = \theta(X, Y) = \frac{\pi}{2}(X+1) \ , \quad \rho = \rho(X, Y) = (Y+1)\,\frac{\rho_2(\theta) + \rho_2(\theta + \pi)}{2} - \rho_2(\theta+\pi) \ , \tag{22}$$

with $x(X, Y) = \rho(X, Y)\cos(\theta(X, Y))$, $y(X, Y) = \rho(X, Y)\sin(\theta(X, Y))$. The "inverse" transformation is given by

$$X(x, y) = -1 + \frac{2\,\theta}{\pi} \ , Y(x, y) = -1 + 2\,\frac{\rho - \rho_2(\theta + \pi)}{\rho_2(\theta) + \rho_2(\theta + \pi)} \ ,$$

$$\text{with} \quad \rho = \rho(x, y) = \text{sign}(y)\sqrt{x^2 + y^2} \ , \quad \theta = \theta(x, y) = \arctan(y/x) \ . \tag{23}$$

The advantage of using the transformation (22)-(23) for a starlike domain instead of standard polar coordinates, is illustrated by Table 7.

3.3 The triangle.

Besides their own interest, the importance of triangular domains stems from the fact that bivariate domains with complex geometries can be efficiently triangulated, by the computational geometry codes that are at the basis of finite element techniques. For simplicity, we refer here to a triangle with one vertex in the origin and another on the positive x-axis (the side length being denoted by b), and the third vertex $V = (v_1, v_2)$ in the first quadrant. It is clear that this configuration is always obtainable by a suitable rotation and translation of the

Table 7: Adaptive Chebyshev approximation of $f(x,y) = e^{-xy}(\cos(x) + \sin(y))$ on the cardioid $0 \leq \theta \leq 2\pi$, $0 \leq \rho \leq \cos((\theta - \pi)/2)$; the absolute and relative tolerances have been set to $\varepsilon_a = 10^{-8}$, $\varepsilon_r = 10^{-6}$.

	coeffs	nodes	est. error	rel. error
polar	302	681	$4 \cdot 10^{-7}$	$1 \cdot 10^{-6}$
starlike	191	473	$4 \cdot 10^{-7}$	$5 \cdot 10^{-7}$

original triangle. We adopt in this case a transformation which is essentially that proposed for the first time by Proriol [20], and later revived by various authors [10, 15, 19], that is

$$x(X,Y) = \frac{b}{2}(X+1)\left(1 - \frac{y(X,Y)}{v_2}\right) + \frac{v_1}{v_2}\,y(X,Y)\ ,\ \ y(X,Y) = \frac{v_2}{2}(Y+1)\ , \tag{24}$$

with inverse

$$X(x,y) = -1 + \frac{2}{b}\,\frac{x - y\,v_1/v_2}{1 - y/v_2}\ ,\ \ Y(x,y) = -1 + \frac{2y}{v_2}\ . \tag{25}$$

It is easily seen that here the cuts correspond to the segments connecting the vertex V with the Chebyshev-Lobatto nodes of the opposite side. The inverse transformation is not defined at the vertex V (where $y = v_2$), but in this case the algorithm chooses $X(V) = 1$ (observe that the vertex V corresponds to the whole upper side of the square $[-1,1]^2$), and computes $f(V) = p(1,1)$, since $(1,1)$ is one of the sampling nodes in the reference square.

Clearly the Proriol transformation, which is algebraic (quadratic) and gives finally through the Chebyshev method a *rational* approximation $\phi(x,y) = p(X(x,y), Y(y)) \approx f(x,y)$, cf. (17), is only one of the possibilities. For example, as it is well known, any triangle can be transformed into the unit reference triangle with vertices $(0,0)$, $(1,0)$ and $(0,1)$ by a nonsingular affine mapping. Then, the reference triangle can be seen as a generalized rectangle with $g_1 \equiv 0$, $g_2(x) = 1 - x$, obtaining again an algebraic transformation and a rational Chebyshev-like approximation. In any case, the Proriol transformation appears more "natural" for a triangle, and is advantageous, for example, when the function varies rapidly in a direction perpendicular to a single side and more slowly in a direction parallel to that side.

This is illustrated by Tables 8 and 9, where the triangle with vertices $(0,0)$, $(1,0)$ and $(1,1)$ is concerned. In Table 8 the function $f(x+y) = \cos(x+y)$ is extremely smooth, and the performances of the adaptive Chebyshev algorithm in Cartesian coordinates ($g_1 \equiv 0$, $g_2(x) = x$), and via the Proriol transformation, are quite similar. The situation changes completely in Table 9, where $f(x,y) = g(x-y)$ with $g(u)$ varying rapidly (strongly peaked) for $u \to 0$. In

Table 8: Adaptive Chebyshev approximation of $f(x,y) = \cos(x+y)$ on the triangle $(0,0)$, $(1,0)$, $(1,1)$, in Cartesian coordinates and with the Proriol transformation (24)-(25); the absolute and relative tolerances have been set to $\varepsilon_a = 10^{-8}$, $\varepsilon_r = 10^{-6}$.

	coeffs	nodes	est. error	rel. error
Cartesian	63	149	$4 \cdot 10^{-8}$	$2 \cdot 10^{-8}$
Proriol	70	149	$2 \cdot 10^{-8}$	$9 \cdot 10^{-9}$

Table 9: Adaptive Chebyshev approximation of $f(x,y) = \exp(-100(x-y)^2)$ on the triangle $(0,0)$, $(1,0)$, $(1,1)$, in Cartesian coordinates and with the Proriol transformation (24)-(25); the absolute and relative tolerances have been set to $\varepsilon_a = 10^{-8}$, $\varepsilon_r = 10^{-6}$.

	coeffs	nodes	est. error	rel. error
Cartesian	741	1661	$1 \cdot 10^{-6}$	$2 \cdot 10^{-7}$
Proriol	31	165	$2 \cdot 10^{-7}$	$1 \cdot 10^{-7}$

this case, taking as reference vertex the point $(1,0)$ (i.e., the vertex where the cuts originate), the function $F(\cdot,Y) = f(x(\cdot,Y),y(Y))$ and thus the Chebyshev coefficients $c_i(\cdot)$ are constant. This makes the adaptive Chebyshev algorithm much more efficient, as can be seen in Table 9, where a comparison is given with the direct use of Cartesian coordinates (it is worth reporting that the number of cuts required with the Proriol transformation is 5, versus 65 of the Cartesian case).

4 An application: fast evaluation of bivariate integral transforms.

At this point, we could summarize by observing that bivariate Chebyshev approximation is particularly useful when the domain is a smooth transformation of a square, and the function exhibits the following features: it is *sufficiently regular* to guarantee a satisfactory convergence rate; it can be evaluated *at any point* of the domain; the *evaluation is costly* (indeed, after the approximation process all the information, up to the tolerance, is compressed in the Chebyshev coefficients array). All these features are usually shown by functions coming from the action of bivariate integral operators, or of their discrete analogs,

$$f(P) = \iint_\Omega K(P,Q,u(Q))\,dQ \approx f_d(P) = \sum_{h=1}^{M}\sum_{k=1}^{N} w_{hk}\,K(P,Q_{hk},u_{hk})\ , \quad (26)$$

where $P = (x, y), Q = (t, s) \in \Omega$, $u : \Omega \to D \subseteq \mathbf{R}$, $K : \Omega \times \Omega \times D \to \mathbf{R}$ is a pentavariate kernel function, $\{w_{hk}\}$ and $\{Q_{hk}\}$ are suitable cubature weights and nodes, respectively, and $u_{hk} \approx u(Q_{hk})$.

Observe that computing the bivariate functions f or f_d in (26) at a large number of "target" points is a very costly process, since each evaluation corresponds to the computation of a double integral. For example, if the discrete integral transform f_d has to be evaluated at all the MN cubature points $\{Q_{hk}\}$, as is usual in the numerical solution of integral equations within iterative solvers of the corresponding discrete systems, a quadratic complexity like $O(M^2N^2)$ arises.

Starting from the basic work of Rokhlin and Greengard in the '80s on the fast multipole method [12], which represented a turning point in the simulation of large scale *linear* integral models, several *fast methods* have been proposed, all sharing the task of accelerating the evaluation of dense discrete linear operators: we may quote, e.g., wavelet-based methods [2], and more recently $\mathcal{H}$-matrix methods [18]. Roughly summarizing, these fast methods act on the kernel of the linear operator by approximation theory techniques, and are able to obtain impressive speed-ups in evaluating the target vector at a specified precision, reducing the complexity even from quadratic to linear. On the other hand, they are usually tailored on the specific structure of the kernel, and are conceived for linear operators.

In some recent papers, a different approach has been explored in the framework of univariate integral operators [8, 9, 25] and equations [24], that of accelerating the evaluation by approximating directly the action of the operator (i.e. the output function) via Chebyshev series or polynomial interpolation at Leja nodes. In the present paper we apply the same idea to bivariate integral operators, by means of our adaptive Chebyshev approximation algorithm described above. It is worth emphasizing some features of this approach (approximating the action instead of the kernel of the operator):

- we *compress* the functions f or f_d in (26) into the array of the μ Chebyshev coefficients $\{c_{ij}\}$, $\mu = \sum_{i=0}^{n(\varepsilon)} (1 + m_i)$, and *reduce* consequently the cost of evaluation of the discrete operator from $O(M^2N^2)$ to $O((\mu + \nu)MN)$, $\mu + \nu \ll MN$ (where $\nu = \nu(\varepsilon)$ denotes the overall number of sampling Chebyshev nodes used by the adaptive algorithm along the cuts);
- we exploit the *smoothing effect* of integration, a fact that has been often overlooked concerning the construction of fast methods;
- we are able to treat linear as well as *nonlinear* problems, because we operate *after integration*;
- even in linear instances, $K(P, Q, u) = H(P, Q)u$, we work in *lower dimension*, since we face a bivariate approximation problem while the kernel H is quadrivariate.

It is worth recalling here that the idea of approximating directly the action of the operator (the potential), has already appeared in the framework of computational methods of potential theory, cf. e.g. [1] where a suitable radial basis interpolation is adopted.

In order to illustrate the effectiveness of our approach, we present some examples, collected in Tables 10-12 below. The computations have been performed on a Pentium 4/2.6 Ghz processor; in all the examples the absolute and relative tolerances in the Chebyshev approximation code have been set to $\varepsilon_a = 10^{-8}$, $\varepsilon_r = 10^{-6}$. For the adaptive cubatures we have used the C++ package CubPack++ (see [7]), while the discrete operators have been obtained by a simple trapezoidal-like cubature formula on a $N \times N$ uniform grid in the square $[-1, 1]^2$, via a suitable change of variables by the transformations described above.

Table 10 concerns the adaptive Chebyshev approximation and consequent compression of a logarithmic potential with a constant density, $K(P, Q, u) = \log(|P - Q|)u$, $u(Q) \equiv 1$, and of a nonlinear Urysohn-like transform (cf., e.g., [3, 14]) with kernel $K(x, y, t, s, u) = \sin(yt + x|s| + u)/(1 + u)$ and argument $u(t, s) = |\sin(e^t + |s|(s + 1))|$, the domain Ω being the unit circle. As is well known, logarithmic potentials are solutions of the Poisson equation $\Delta f(x, y) = 2\pi u(x, y)$, cf. [26]. The relative error of Chebyshev approximation has been computed in $\|\cdot\|_\infty$ on a suitable control grid. Notice in particular that all the information necessary to reconstruct the logarithmic potential up to a relative error of the order of 10^{-7}, is completely contained in only 15 output Chebyshev coefficients (and has required the computation of 45 double integrals). Here we have adopted a high-precision adaptive cubature method, and this corresponds to approximating directly the function f in (26).

On the contrary, in Tables 11 and 12 we approximate the action of operators that have been discretized by a cubature formula on a fixed $N \times N$ grid $\{Q_{hk}\}$ (which is also the target grid), and this corresponds to working with the function f_d in (26). The transforms in Table 11 have respectively $K(x, y, t, s, u) = \exp(xt - ys)u$, $u(t, s) = \sin(t) + 1$ if $t \geq s$ and $u(t, s) = \sin(s) - 1$ if $t < s$, and $K(x, y, t, s, u) = \exp(u \sin(x + s) + y + t)/u$, $u(t, s) = e^t$ if $t^2 + s^2 \geq 1$ and $u(t, s) = \sin(s) + 2$ if $t^2 + s^2 < 1$. The domain, defined by $-2 \leq x \leq 2$, $-\sin(2x) - 2 \leq y \leq -\sin(3x) + 2$, is treated as a generalized rectangle in Cartesian coordinates. Observe that due to the discontinuity of the operator arguments $u(t, s)$ the cubature is not very precise, but the smoothness of the kernels entails that the bivariate Chebyshev approximation works satisfactorily, with errors very close to the required tolerance and good speed-ups relative to direct evaluation.

The *speed-up* SU is defined as (direct time)/(construction time + evaluation time), "direct" denoting the machine-time for computing $\{f_d(Q_{hk})\}$, "construction" that for computing the Chebyshev coefficients $\{c_{ij}\}$ of the approximating function $\phi_d(x, y) \approx f_d(x, y)$ (cf. (17)), and "evaluation" the machine-time for evaluating the output vector $\{\phi_d(Q_{hk})\}$. As for the relative errors, they are measured in the maximum norm, $\max_{hk} |f_d(Q_{hk}) - \phi_d(Q_{hk})| / \max_{hk} |f_d(Q_{hk})|$.

Notice that the evaluation time is a small fraction of the construction time, since the bulk of the algorithm is given by computation of f_d. Indeed, the observed speed-ups are not far from the rough speed-up estimate N^2/(number of nodes): this means that taking for example a 500×500 grid, we could expect

Table 10: Adaptive Chebyshev compression on the unit circle of a Urysohn-type nonlinear transform with smooth kernel and of a logarithmic potential, pointwise evaluated by adaptive cubature with tolerance 10^{-7}.

	number of coeffs	number of nodes	relative error
log potential	15	45	$2 \cdot 10^{-7}$
Urysohn	99	289	$4 \cdot 10^{-8}$

Table 11: Adaptive Chebyshev approximation of a linear discrete and of a Urysohn-type nonlinear discrete transform with smooth kernels and discontinuous arguments, on the generalized rectangle $-2 \leq x \leq 2$, $-\sin(2x) - 2 \leq y \leq -\sin(3x) + 2$ (100×100 grid).

	coeffs	nodes	rel. err.	cub. err.	constr.	eval.	dir.	SU
lin.	815	1329	$4 \cdot 10^{-7}$	$3 \cdot 10^{-3}$	7.1 s	0.2 s	53.2 s	7.2
Ury.	418	1065	$3 \cdot 10^{-6}$	$6 \cdot 10^{-4}$	8.8 s	0.1 s	97.3 s	10.9

an increase of the speed-up by a factor 25.

Finally, it is worth commenting on the examples in Table 12, where two discrete logarithmic potentials have been computed on a 500×500 grid in polar coordinates on the unit circle. The first row in the table corresponds to a constant density, while the second to a C^1 density which has partial second derivatives discontinuous on the Cartesian axes. Indeed, $u_2(t,s)$ is taken equal to $e^t + s^3$ in the first quadrant, $1 + t + s^3$ in the second, $1 + t - s^2$ in the third, and $e^t - s^2$ in the fourth. Here, we have a fixed cubature formula on a weakly-singular kernel, and the Chebyshev approximation error is not able to reach the required tolerance (a stalling phenomenon appears: the Chebyshev error stagnates at the size of the cubature error). This fact has already been observed in univariate instances, cf. [8, 9, 25] where it is analyzed and qualitatively explained; the key point is that in weakly-singular instances the discrete transform f_d is singular at the cubature points, while f can be regular. Observe that the stalling phenomenon does not represent a real disadvantage, since one usually is not interested in approximating beyond the underlying discretization error.

References

[1] G. Allasia, *Approximating potential integrals by cardinal basis interpolants on multivariate scattered data* in: "Radial basis functions and partial dif-

Table 12: Adaptive Chebyshev approximation of two discrete logarithmic potentials on the unit circle (500×500 polar grid): $K(P, Q, u) = \log(|P - Q|)u$, $u = u_1(Q) \equiv 1$, while $u = u_2 \in C^1$ has discontinuous second partial derivatives.

	coeffs	nodes	rel. err.	cub. err.	constr.	eval.	dir.	SU
u_1	273	337	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	39.5 s	2.1 s	8.5 hrs	734
u_2	352	825	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	123 s	1.3 s	10.2 hrs	294

ferential equations", Comput. Math. Appl. **43** (2002), 275-287.

[2] 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.

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

[4] Z. Battles and L.N. Trefethen, *An extension of Matlab to continuous functions and operators*, 2003, submitted to SIAM J. Sci. Comp. (preprint available online, Computing Laboratory, Oxford University).

[5] M.D. Buhmann, *Radial basis functions*, Acta Numer. **9** (2000), 1-38.

[6] C.W. Clenshaw and A.R. Curtis, *A method for numerical integration on an automatic computer*, Numer. Math. **2** (1960), 197-205.

[7] R. Cools, D. Laurie and L. Pluym, *A User Manual for Cubpack++*, Dept. of Computer Science, K.U. Leuven, 1997.

[8] 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.

[9] S. De Marchi and M. Vianello, *Fast evaluation of discrete integral transforms by Chebyshev and Leja polynomial approximation*, in: "Constructive Functions Theory" (Varna 2002), B. Bojanov Ed., DARBA, Sofia, 2003, pp. 347-353.

[10] M. Dubiner, *Spectral methods on triangles and other domains*, J. Sci. Comput. **6** (1991), 345-390.

[11] R. Franke, *Scattered data interpolation: tests of some methods*, Math. Comp. **38** (1982), 181-200.

[12] L. Greengard, *Fast algorithms for classical physics*, Science **265** (1994), 909-914.

[13] T. Håvie, *Chebyshev series coefficients of a function*, CERN Program Library, algorithm E406, 1986.

[14] M.C. Joshi and R.K. Bose, "Some topics in nonlinear functional analysis", Wiley Eastern Limited, New Delhi, 1985.

[15] T. Koornwinder, *Two-variable analogues of the classical orthogonal polynomials*, in "Theory and Applications of Special Functions", Academic Press, San Diego, 1975.

[16] A. Mardegan, A. Sommariva, M. Vianello and R. Zanovello, *Adaptive bivariate Chebyshev approximation and efficient evaluation of integral operators*, NACoM 2003 extended abstracts (Cambridge, UK, May 2003), WILEY-VCH, Weinheim, 2003.

[17] J.C. Mason and D.C. Handscomb, "Chebyshev polynomials", Chapman & Hall/CRC, Boca Raton, FL, 2003.

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

[19] R.G. Owens, *A posteriori error estimates for spectral element solutions to viscoelastic flow problems*, Comput. Methods Appl. Mech. Engrg. **164** (1998), 375-395.

[20] J. Proriol, *Sur une famille de polynomes á deux variables orthogonaux dans un triangle*, C.R. Acad. Sci. Paris **245** (1957), 2459-2461.

[21] L. Reichel, *Fast solution methods for Fredholm integral equations of the second kind*, Numer. Math. **57** (1990), 719-736.

[22] T. J. Rivlin, "An Introduction to the Approximation of Functions", Dover, New York, 1981.

[23] T. J. Rivlin, "Chebyshev polynomials. From approximation theory to algebra and number theory", second edition, Pure and Applied Mathematics, John Wiley & Sons, New York, 1990.

[24] A. Sommariva, *A fast Nyström-Broyden solver by Chebyshev compression*, Numer. Algorithms, in press.

[25] M. Vianello, *Chebyshev-like compression of linear and nonlinear discretized integral operators*, Neural, Parallel and Sci. Comput. **8** (2000), 327-353.

[26] V.S. Vladimirov, "Equations of mathematical physics", MIR, Moscow, 1984.