# Adaptive bivariate Chebyshev approximation and efficient evaluation of integral operators

**Alberto Mardegan, Alvise Sommariva** [1]**, Marco Vianello*** [1]**,** and **Renato Zanovello** [1]

[1] Dept. of Pure and Applied Mathematics, University of Padova, Via Belzoni 7- 35131, Padova (Italy)

Received ???, revised ???, accepted ???

We propose an adaptive algorithm which extends Chebyshev series approximation to bivariate functions on domains which are smooth transformations of a square. We apply the method for evaluating efficiently the action of linear and nonlinear bivariate integral operators.

Chebyshev series expansion

$$f(x) = \sum_{k=0}^{\infty} c_k T_k(x) \;, \quad x \in [-1,1] \;, \quad c_k = \frac{2}{\pi} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} \, f(x) \, T_k(x) \, dx \;, \; k > 0 \;, \tag{1}$$

$c_0 = \frac{1}{\pi} \int_{-1}^{1} f(x)/\sqrt{1-x^2} \, dx$, is a useful and popular tool for the approximation of sufficiently regular univariate functions. In fact, its partial sums provide an asymptotically optimal polynomial approximation, which can be constructed in a very efficient way by resorting to the FFT algorithm; see, e.g., the classical book [11], and e.g. the various algorithms in the Netlib and Cernlib repositories for the practical implementations. Concerning its extension to bivariate functions only some spread results, restricted to the case of rectangular domains (see, e.g., [10]), 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.

Let us begin by 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 $f(x,y) = \sum_{\alpha=0}^{\infty} c_\alpha(x) T_\alpha(y)$. Moreover, suppose that we can expand each coefficient $c_\alpha(x)$ in Chebyshev series (uniformly in $\alpha$), i.e. $c_\alpha(x) = \sum_{\beta=0}^{\infty} c_{\alpha\beta} T_\beta(x)$. It is not difficult to show that both expansions take place for example when $\log(n) \, \mathrm{osc}(f;1/n) \to 0$ as $n \to \infty$, $\mathrm{osc}(f;1/n) := \max_{|P-Q| \leq 1/n} \{|f(P) - f(Q)|, \, P, Q \in [-1,1]^2\}$. The latter is indeed the natural bivariate extension of the well-known Dini-Lipschitz condition for Chebyshev univariate approximation (see [11]), and is verified for example by any globally Hölder-continuous function. Under this assumption, we obtain indeed that, for every fixed tolerance $\varepsilon > 0$, there exist an index $\alpha(\varepsilon)$, and a sequence $\beta(\alpha) = \beta(\alpha;\varepsilon)$, such that the following bivariate polynomial approximation of $f(x,y)$ with degree $\max_\alpha \{\alpha + \beta(\alpha)\}$ holds (see the forthcoming paper [12] for a proof)

$$\|f - p\|_\infty = \max_{(x,y) \in [-1,1]^2} |f(x,y) - p(x,y)| \leq \varepsilon \;, \quad p(x,y) = \sum_{\alpha=0}^{\alpha(\varepsilon)} \sum_{\beta=0}^{\beta(\alpha)} c_{\alpha\beta} \, T_\beta(x) \, T_\alpha(y) \;. \tag{2}$$

As in the univariate case, the smoother the function $f$, the smaller the approximation degree at the same tolerance. Observe that all the information necessary to reconstruct the function $f$ on the square is now

* Corresponding author: e-mail: marcov@math.unipd.it. This work has been partially supported by the University of Padova, in particular by the project CPDA028291 “Efficient approximation methods for nonlocal discrete transforms”, and by the 2003 project “Innovative methods for structured problems in stationary and evolutive models” of the GNCS-INdAM.

compressed in the coefficients matrix $\{c_{\alpha\beta}\}$, which is in general not full, but has a "reversed nonincreasing histogram" shape.

The practical implementation of the method works as follows. First, one chooses a good univariate code for Chebyshev series approximation: we selected the robust Fortran code [7] in the Cernlib, and translated it in the C language with some minor adaptations in view of the bivariate embedding, the most remarkable being that of allowing discrete values of the function at the Chebyshev nodes as inputs (indeed, the original code requires an analytic definition), and of introducing an additional stopping criterion based on a possible "stalling" of convergence. The "regular" exit is based on a classical criterion [3], looking for the first triple of consecutive coefficients whose absolute values sum is below a given tolerance, while the output error estimate uses for safety the complete tail of the last computed coefficients. Other important features of the code [7] 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 those below a suitable threshold.

For the construction of the bivariate polynomial $p(x,y)$ in (2), we take $n$ "cuts" on the square, corresponding to the first $n$ Chebyshev-Lobatto nodes $\xi_j = \cos(j\pi/(n-1))$, $j = 0,1,...,n-1$. The function $f(x,y)$ is approximated by a truncated Chebyshev series $\tilde{f}(x,y) = \sum_{\alpha=0}^{\alpha(x;\theta\varepsilon)} c_\alpha(x)T_\alpha(y)$ at each cut $x = \xi_j$, by the univariate code up to a fraction $\theta$ of the global tolerance (say $\theta = 0.5$); this step provides the values $\{c_\alpha(\xi_j)\}$ of the coefficient functions $c_\alpha(x)$, $\alpha = 0,1,...,\alpha(\theta\varepsilon) =: \max_j \alpha(\xi_j;\theta\varepsilon)$, at the Chebyshev $x$-nodes $\{\xi_j\}$. Then, these values are passed to the univariate code, which produces the Chebyshev approximation $\tilde{c}_\alpha(x) = \sum_{\beta=0}^{n-1} c_{\alpha\beta}T_\beta(x)$, and checks whether $\|\tilde{c}_\alpha - c_\alpha\|_\infty \le (1-\theta)\varepsilon/(1+\alpha(\theta\varepsilon))$ for every $\alpha = 0,1,...,\alpha(\theta\varepsilon)$. In fact, we can consider the error estimate $\|f-p\|_\infty \le \|f-\tilde{f}\|_\infty + \|\tilde{f}-p\|_\infty \approx \max_j \|f(\xi_j,\cdot) - \tilde{f}(\xi_j,\cdot)\|_\infty + \sum_{\alpha=0}^{\alpha(\theta\varepsilon)} \|\tilde{c}_\alpha - c_\alpha\|_\infty \le \theta\varepsilon + (1+\alpha(\theta\varepsilon)) \max_\alpha \|\tilde{c}_\alpha - c_\alpha\|_\infty$. If the stopping criterion is not satisfied (the approximation in the $x$-direction is not satisfactory), then the number $n$ of Chebyshev cuts is doubled and the procedure repeated. It is worth noting the full adaptivity of this algorithm, which is even 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 dinamically 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 $\varepsilon_a$ 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, simply by comparing two consecutive error estimates, and is typical of low regularity of the function, for example in the presence of noise, or in the application to discrete integral operators with weakly-singular kernels, as we shall see below.

We can now face more general situations. Consider a function $f$ defined on a bivariate compact domain $\Omega$, that corresponds to the square $[-1,1]^2$ through the smooth surjective transformation $[-1,1]^2 \to \Omega$, $(X,Y) \mapsto (x(X,Y), y(X,Y))$. 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 like

$$f(x,y) \approx \phi(x,y) = p(X(x,y), Y(x,y)) = \sum_{\alpha=0}^{\alpha(\varepsilon)} \sum_{\beta=0}^{\beta(\alpha)} c_{\alpha\beta}\, T_\beta(X(x,y))\, T_\alpha(Y(x,y)) \ , \tag{3}$$

$(x,y) \in \Omega$, which is in general no more polynomial. In (3), $(X(x,y), Y(x,y))$ denotes the "inverse" transformation $\Omega \to [-1,1]^2$, which is allowed to be undefined only in a finite number of points. Observe that, from the theoretical point of view, when the function $f$ is globally Hölder-continuous in $\Omega$, then 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 1 below. Now we are ready to describe two important classes of domain geometries, with corresponding transformations.

- $x$-regular domains in Cartesian coordinates: the domain $\Omega$ is defined by $a \le x \le b$, $g_1(x) \le y \le g_2(x)$, $g_1$ and $g_2$ being suitable functions (these are the typical domains where double integrals can be splitted). Here the transformation is $x(X,Y) = x(X) = a + (X+1)(b-a)/2$, $y(X,Y) = g_1(x) + (Y+1)(g_2(x) - g_1(x))/2$, and its inverse $X(x,y) = X(x) = -1 + 2(x-a)/(b-a)$, $Y(x,y) = -1 + 2(y - g_1(x))/(g_2(x) - g_1(x))$; 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 $p(X(\overline{x}), Y(\overline{x}, \overline{y})) = \sum_{\beta=0}^{\beta(0)} c_{0\beta} T_\beta(X(\overline{x}))$, cf. (3). The regularity of the transformation is clearly given by the regularity of the functions $g_1$ and $g_2$.

- $\theta$-regular domains in polar coordinates: these are defined by $\theta_1 \le \theta \le \theta_2$, $\rho_1(\theta) \le \rho \le \rho_2(\theta)$, and the transformation is the composition of one analogous to that described above with the change from Cartesian to polar coordinates, with inverse $X(x,y) = -1 + 2(\theta - \theta_1)/(\theta_2 - \theta_1)$, $Y(x,y) = -1 + 2(\rho - \rho_1(\theta))/(\rho_2(\theta) - \rho_1(\theta))$, with $\rho = \rho(x,y) = \sqrt{x^2+y^2}$, $\theta(x,y) = \arctan(y/x)$. 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 $\rho_1$ and $\rho_2$. The simplest case is that of a circle centered at the origin, i.e. $0 \le \theta \le 2\pi$, $0 \le \rho \le R$ (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 $x = \pm R$).

A remarkable subclass of $\theta$-regular is given by star 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 to avoid unnecessary clustering of sampling nodes at the origin [9, 12]. For the sake of brevity, we omit a third type of domain, the triangle, whose importance stems from the possibility of triangulating efficiently domains with complex geometries: also in this case different transformations can be used, which lead to quite different performances of the approximation method; see again [9, 12], where domain splitting techniques are also discussed in order to manage possible singularities of the function $f$. The following table illustrates the importance of the choice of the transformation.

**Table 1** Adaptive Chebyshev approximation of $f(x,y) = e^x(\sin(y) + xy^2)$ on the unit circle in Cartesian and polar coordinates; the absolute and relative tolerances have been set to $\varepsilon_a = 10^{-8}$, $\varepsilon_r = 10^{-6}$.

| | number of coeffs | number of nodes | relative error |
|---|---|---|---|
| Cartesian | 6525 | 88409 | $8 \cdot 10^{-6}$ |
| polar | 197 | 457 | $2 \cdot 10^{-7}$ |

Notice that, while the function is extremely smooth, the choice of representing the unit circle as an $x$-regular domain leads to computational failure, since the singularity of the transformation entails very slow convergence (the relative error in $\|\cdot\|_\infty$ is computed by comparison with the exact values on a suitable control grid). 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 unsignificant coefficients.

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 versions,

$$f(P) = \iint_{\Omega} K(P, Q, u(Q))\, dQ \approx f_d(P) = \sum_{i=1}^{M} \sum_{j=1}^{N} w_{ij}\, K(P, Q_{ij}, u_{ij}) \;, \tag{4}$$

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, $\{w_{ij}\}$ and $\{Q_{ij}\}$ are suitable cubature weights and nodes, respectively, and $u_{ij} \approx u(Q_{ij})$. Observe that computing the bivariate functions $f$ or $f_d$ in (4) 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 $\{Q_{ij}\}$, as it 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 [6], which represented the 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 [8]. 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 taylored on the specific structure of the kernel, and are conceived for linear operators.

In some recent papers [4, 5, 13], a different approach has been explored in the framework of univariate integral operators, that is 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 (4) into the array of the $\mu$ Chebyshev coefficients $\{c_{\alpha\beta}\}$, $\mu = \sum_{\alpha=0}^{\alpha(\varepsilon)} (1 + \beta(\alpha))$, 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].

In order to illustrate the effectiveness of our approach, we present some examples, collected in Tables 2-4 below. The computations have been performed on a AMD K6-III 400 Mhz 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++, 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 2 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 transform 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 being the unit circle. As known, logarithmic potentials are solutions of the Poisson equation $\Delta f = 2\pi u$, cf. [14]. 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 approximate directly the function $f$ in (4).

On the contrary, in Tables 3 and 4 we approximate the action of operators that have been discretized by a cubature formula on a fixed $N \times N$ grid $\{Q_{ij}\}$ (which is also the target grid), and this corresponds to work with the function $f_d$ in (4). The transforms in Table 3 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 $x$-regular one. 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 w.r.t. direct evaluation. The speed-up is defined as (direct time)/(construction time + evaluation time), "direct" denoting the machine-time for computing $\{f_d(Q_{ij})\}$, "construction" that for computing the Chebyshev coefficients $\{c_{\alpha\beta}\}$ of the approximating function $\phi_d(x,y) \approx f_d(x,y)$ (cf. (3)), and "evaluation" the machine-time for evaluating the output vector $\{\phi_d(Q_{ij})\}$. As for the relative errors, they are measured in the maximum norm, $\max_{ij} |f_d(Q_{ij}) - \phi_d(Q_{ij})| / \max_{ij} |f_d(Q_{ij})|$. 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 $500 \times 500$ grid, we could expect an increase of the speed-up by a factor $25$.

Finally, it is worth to comment the examples in Table 4, where two discrete logarithmic potentials have been computed on a $500 \times 500$ 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, $u_2(t,s)$ is 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 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. [4, 5, 13] 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.

**Table 2** 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 3** Adaptive Chebyshev approximation of a linear discrete and of a Urysohn-type nonlinear discrete transform with smooth kernels and discontinuous arguments, on a $x$-regular domain ($100 \times 100$ grid).

| | coeffs | nodes | rel. err. | cub. err. | constr. | eval. | direct | speed-up |
|---|---|---|---|---|---|---|---|---|
| linear | 815 | 1329 | $4 \cdot 10^{-7}$ | $3 \cdot 10^{-3}$ | 43 sec | 1.25 sec | 319 sec | 7.2 |
| Urysohn | 418 | 1065 | $3 \cdot 10^{-6}$ | $6 \cdot 10^{-4}$ | 53 sec | 0.75 sec | 584 sec | 10.9 |

**Table 4** Adaptive Chebyshev approximation of two discrete logarithmic potentials on the unit circle ($500 \times 500$ polar grid): $K(P, Q, u) = \log (|P - Q|)u$, $u_1(Q) \equiv 1$, $u_2 \in C^1$ has discontinuous second partial derivatives.

| | coeffs | nodes | rel. err. | cub. err. | constr. | eval. | direct | speed-up |
|---|---|---|---|---|---|---|---|---|
| $u = u_1$ | 273 | 337 | $5 \cdot 10^{-4}$ | $5 \cdot 10^{-4}$ | 237 sec | 12.5 sec | 51 hours | 734 |
| $u = u_2$ | 352 | 825 | $1 \cdot 10^{-3}$ | $1 \cdot 10^{-3}$ | 738 sec | 7.5 sec | 61 hours | 294 |

## References

[1] G. Allasia, *Approximating potential integrals by cardinal basis interpolants on multivariate scattered data* in: "Radial basis functions and partial differential 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] C.W. Clenshaw and A.R. Curtis, *A method for numerical integration on an automatic computer*, Numer. Math. **2** (1960), 197-205.

[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 Chebyshev and Leja polynomial approximation*, in: "Constructive Functions Theory" (Varna 2002), B. Bojanov Ed., DARBA, Sofia, 2003, pp. 347-353.

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

[7] T. Håvie, *Chebyshev series coefficients of a function*, CERN Program Library, algorithm E406, 1986 (http://wwwinfo.cern.ch/asd/cernlib/mathlib.html).

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

[9] A. Mardegan, "Bivariate Chebyshev series and approximation of integral operators", Laurea Thesis in Mathematics (Italian), University of Padova, 2002 (advisors M. Vianello and A. Sommariva).

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

[11] T. J. Rivlin, "The Chebyshev polynomials". Pure and Applied Mathematics. Wiley-Interscience, New York-London-Sydney, 1974.

[12] A. Sommariva, M. Vianello and R. Zanovello, *Adaptive bivariate Chebyshev approximation*, in preparation.

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

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