Hyperinterpolation in the cube*

Marco Caliari, Stefano De Marchi
Dept. of Computer Science,
University of Verona (Italy)
e-mail: marco.caliari@univr.it, stefano.demarchi@univr.it

Marco Vianello
Dept. of Pure and Applied Mathematics,
University of Padua (Italy)
e-mail: marcov@math.unipd.it

Abstract

We construct an hyperinterpolation formula of degree n in the three-dimensional cube, by using the numerical cubature formula for the product Chebyshev measure given by the product of a (near) minimal formula in the square with Gauss-Chebyshev-Lobatto quadrature. The underlying function is sampled at $N \sim n^3/2$ points, whereas the hyperinterpolation polynomial is determined by its $(n+1)(n+2)(n+3)/6 \sim n^3/6$ coefficients in the trivariate Chebyshev orthogonal basis. The effectiveness of the method is shown by a numerical study of the Lebesgue constant, which turns out to increase like $\log^3(n)$, and by the application to several test functions.

Keywords: hyperinterpolation, Lebegsue constant, numerical cubature.

1 Hyperinterpolation over general regions

Polynomial hyperinterpolation of multivariate continuous functions over compact domains or manifolds, originally introduced by Sloan [24], is a discretized orthogonal projection on polynomial subspaces, which provides an approximation method more general than polynomial interpolation. Though the idea is very general and flexible, and the problem in some sense easier

*Work supported by the ex-60% funds of the Universities of Padova and Verona, and by the GNCS-INdAM.

than multivariate polynomial interpolation [17], till now it has been used effectively in few cases: the sphere [22, 25], the square [8, 10], and the disk [18].

Indeed, hyperinterpolation requires two basic ingredients, i.e. the explicit knowledge of a family of orthogonal polynomials w.r.t. any measure on the domain, and a "good" cubature formula for that measure (positive weights and high algebraic degree of exactness). It becomes an effective uniform approximation tool when its norm as a projection operator (the so-called Lebesgue constant) grows slowly.

The importance of these basic features can be understood by summarizing briefly the structure of hyperinterpolation. Let $\Omega \subset \mathbb{R}^d$ be a compact subset (or lower dimensional manifold), and μ a positive measure such that $\mu(\Omega) = 1$ (i.e., a normalized positive and finite measure on Ω). For every function $f \in C(\Omega)$ the μ-orthogonal projection of f on $\Pi_n^d(\Omega)$ (the subspace of d-variate polynomials of degree $\leq n$ restricted to Ω) can be written as

$$\mathcal{S}_n f(\boldsymbol{x}) = \int_\Omega K_n(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y})\, d\mu(\boldsymbol{y}) \quad \text{with } \mathcal{S}_n p = p \text{ for } p \in \Pi_n^d(\Omega)\ , \tag{1.1}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$, $\boldsymbol{y} = (y_1, y_2, \ldots, y_d)$, and the so-called reproducing kernel K_n is defined by

$$K_n(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=0}^{n} \sum_{|\alpha|=k} p_\alpha(\boldsymbol{x}) p_\alpha(\boldsymbol{y}), \quad \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d) \tag{1.2}$$

the set of polynomials $\{p_\alpha,\ |\alpha| = \alpha_1 + \ldots + \alpha_d = k,\ 0 \leq k \leq n\}$ being any μ-orthonormal basis of $\Pi_n^d(\Omega)$, with p_α of total degree $|\alpha|$ (concerning the theory of multivariate orthogonal polynomials, we refer the reader to the recent monograph by Dunkl and Xu [16]).

Now, given a cubature formula for μ with $N = N(n)$ nodes $\boldsymbol{\xi} \in \Xi \subset \Omega$, $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_d)$, and positive weights $\{w_{\boldsymbol{\xi}}\}$, which is exact for polynomials of degree $\leq 2n$,

$$\int_\Omega p(\boldsymbol{x})\, d\mu = \sum_{\boldsymbol{\xi} \in \Xi} w_{\boldsymbol{\xi}}\, p(\boldsymbol{\xi})\ , \quad \forall p \in \Pi_{2n}^d(\Omega)\ , \tag{1.3}$$

we obtain from (1.1) the polynomial approximation of degree n

$$f(\boldsymbol{x}) \approx L_n f(\boldsymbol{x}) = \sum_{\boldsymbol{\xi} \in \Xi} w_{\boldsymbol{\xi}} K_n(\boldsymbol{x}, \boldsymbol{\xi})\, f(\boldsymbol{\xi})\ , \tag{1.4}$$

where $L_n p = S_n p = p$ for every $p \in \Pi_n^d(\Omega)$. It is known that necessarily $N \geq \dim(\Pi_n^d(\Omega))$, and that (1.4) is a polynomial interpolation at Ξ whenever the equality holds [24].

The hyperinterpolation error in the uniform norm, due to exactness on $\Pi_{2n}^d(\Omega)$, can be easily estimated as

$$\|f - L_n f\|_\infty \leq (1 + \Lambda_n)\, E_n(f)\ , \quad E_n(f) = \inf_{p \in \Pi_n^d(\Omega)} \|f - p\|_{\infty,\Omega}\ ,$$

$$\Lambda_n = \|L_n\| = \max_{\boldsymbol{x} \in \Omega} \left\{ \sum_{\boldsymbol{\xi} \in \Xi} w_{\boldsymbol{\xi}}\ |K_n(\boldsymbol{x}, \boldsymbol{\xi})| \right\}\ , \tag{1.5}$$

where Λ_n is the operator norm of $L_n : (C(\Omega), \|\cdot\|_\infty) \to (\Pi_n^d(\Omega), \|\cdot\|_\infty)$, usually termed the "Lebesgue constant" in the interpolation framework.

The aim of this paper is to make a first step towards three-dimensional hyperinterpolation, in the special case of the cube. In the next section we describe the technique and we discuss its main features, namely growth of the Lebesgue constant, implementation and computational cost, theoretical and practical error estimates. In section 3 we show the effectiveness of hyperinterpolation on a set of trivariate test functions.

2 The case of the cube

In the case of the d-dimensional cube, orthogonal polynomials are explicitly known for product Jacobi measures [16]. The main difficulty dealing with hyperinterpolation of degree n is then to find a cubature formula (for any of such measures) with algebraic degree of exactness not smaller than $2n$, which uses a low number of nodes (as low as possible, in principle). In particular, *minimal* cubature formulas, i.e. formulas which have degree of exactness $2n + 1$ and match Möller's lower bound [19], seem the right choice. Unfortunately, minimal cubature in dimension $d > 1$ is very difficult to construct, and even computing numerically nodes and weights of formulas at least "close to minimal" is a challenging computational task [12, 13, 21].

Among the few minimal formulas with explicitly known nodes and weights, there is the formula for the product Chebyshev measure in dimension 2 (square), originally given by Morrow and Patterson [20] for even degree n; see also the construction of Cools and Schmid [14]. The nodes and weights for odd degrees were given by Xu [27], who also proved that such points are suitable for bivariate polynomial interpolation (though in a polynomial subspace $\mathcal{V}_n$, $\Pi_{n-1}^2 \subset \mathcal{V}_n \subset \Pi_n^2$). Theoretical and computational aspects of bivariate interpolation at Xu points has been addressed recently [5, 6, 15]. In particular, it has been proved rigorously that the corresponding Lebesgue constant is $\mathcal{O}(\log^2(n))$, which is the optimal rate for projection operators [26]. The same rate is obtained by the so-called "Padua points", which are the first known example of optimal unisolvent family of interpolation points in two variables [9, 4].

We have used the minimal cubature formula above for bivariate hyperinterpolation on the square [10] and more generally on rectangles [8]. The corresponding projection is not interpolant, but shares the good theoretical and computational features of Xu-like and Padua-like interpolation, in particular the optimal rate of the Lebesgue constant [8, 10]. All these polynomial approximations are completely determined by a number of sampling points and of coefficients growing asymptotically like $n^2/2$.

Extension of Xu-like or Padua-like interpolation to dimension 3 (cube) seems a very difficult task. Here, we try to extend hyperinterpolation, with the limitation that a minimal cubature formula for the product Chebyshev measure in the cube is not known.

As a first fundamental step towards three-dimensional hyperinterpolation of degree n, we begin by constructing a cubature formula with the required degree of exactness, trying to keep low the number of nodes. Consider the three-dimensional normalized product Chebyshev measure on $\Omega = [-1,1]^3$, factorized as

$$d\mu_3 = d\mu_2 \otimes d\mu_1 \ , \quad d\mu_2 = \frac{1}{\pi^2} \frac{dx_1 dx_2}{\sqrt{1-x_1^2}\sqrt{1-x_2^2}} \ , \quad d\mu_1 = \frac{1}{\pi} \frac{dx_3}{\sqrt{1-x_3^2}} \ . \tag{2.6}$$

The $N_1 = n+2$ Chebyshev-Lobatto points in $[-1,1]$

$$T_{n+1} = \{z_k\} \ , \quad z_k = z_{k,n+1} = \cos\frac{k\pi}{n+1} \ , \quad k = 0, \ldots, n+1 \ , \tag{2.7}$$

as it is well known are nodes of a minimal quadrature formula for $d\mu_1$ (degree of exactness $2n+1$), with corresponding weights $\{w_{z_k}\}$ are: $w_{-1} = w_1 = (n+1)^{-1}/2$, $w_{z_k} = (n+1)^{-1}$ for the interior points.

Moreover, there exists a (near) minimal cubature formula for $d\mu_2$ on $[-1,1]^2$ with degree of exactness $2n+1$, whose nodes (ξ_1, ξ_2) belong to the two-dimensional Chebyshev-like set

$$X_{n+1} = A \cup B, \qquad \mathrm{card}(X_{n+1}) = N_2, \tag{2.8}$$

where

- case n odd, $n = 2m - 1$

$$\begin{aligned} A_{\text{odd}} &= \{(z_{2i}, z_{2j+1}), \quad 0 \le i \le m, \qquad 0 \le j \le m-1\} \\ B_{\text{odd}} &= \{(z_{2i+1}, z_{2j}), \quad 0 \le i \le m-1, \quad 0 \le j \le m\} \end{aligned} \tag{2.9}$$

  with $N_2 = (n+1)(n+3)/2$, and weights $w_{(\xi_1,\xi_2)} = (n+1)^{-2}$ for the boundary nodes, and $w_{(\xi_1,\xi_2)} = 2(n+1)^{-2}$ for the interior nodes (minimal formula, N_2 equals Möller's lower bound [19]).

- case n even, $n = 2m$

$$\begin{aligned} A_{\text{even}} &= \{(z_{2i}, z_{2j}), \qquad\quad 0 \le i \le m, \quad 0 \le j \le m\} \\ B_{\text{even}} &= \{(z_{2i+1}, z_{2j+1}), \quad 0 \le i \le m, \quad 0 \le j \le m\} \end{aligned} \tag{2.10}$$

  with $N_2 = (n+2)^2/2$, and weights $w_{(\xi_1,\xi_2)} = (n+1)^{-2}/2$ for $(\xi_1, \xi_2) = (1, 1)$ and $(\xi_1, \xi_2) = (-1, -1)$, $w_{(\xi_1,\xi_2)} = (n+1)^{-2}$ for the other boundary nodes and $w_{(\xi_1,\xi_2)} = 2(n+1)^{-2}$ for the interior nodes (near minimal formula, the number of nodes is only one more than Möller's lower bound).

Observe that X_{n+1} is a symmetric set w.r.t. the diagonal, i.e. $\{(\xi_2, \xi_1) : (\xi_1, \xi_2) \in X_{n+1}\} = X_{n+1}$.

From the minimal formulas above, we can immediately obtain a product cubature formula for $d\mu_3$ (recall that $\boldsymbol{x} = (x_1, x_2, x_3)$ and $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$)

$$\int_{[-1,1]^3} f(\boldsymbol{x})\, d\mu_3 \approx \sum_{\boldsymbol{\xi} \in X_{n+1} \times T_{n+1}} w_{(\xi_1,\xi_2)} w_{\xi_3}\, f(\boldsymbol{\xi}) \;, \tag{2.11}$$

which is exact for polynomials of degree $\le 2n+1$ using

$$N_3 = \text{card}(X_{n+1} \times T_{n+1}) = N_1 N_2 \sim \frac{n^3}{2} \tag{2.12}$$

nodes. In fact, it is exact on $\Pi^2_{2n+1}([-1,1]^2) \otimes \Pi^1_{2n+1}([-1,1]) \supset \Pi^3_{2n+1}([-1,1]^3)$.

Then, we can construct the hyperinterpolation polynomial of degree n at $X_{n+1} \times T_{n+1}$ as in (1.4)

$$L_n f(\boldsymbol{x}) = \sum_{\boldsymbol{\xi} \in X_{n+1} \times T_{n+1}} w_{(\xi_1,\xi_2)} w_{\xi_3}\, K_n(\boldsymbol{x}, \boldsymbol{\xi})\, f(\boldsymbol{\xi}) \;, \tag{2.13}$$

where the reproducing kernel is defined as in (1.2) via the orthonormal basis

$$p_\alpha(\boldsymbol{x}) = \hat{T}_{\alpha_1}(x_1) \hat{T}_{\alpha_2}(x_2) \hat{T}_{\alpha_3}(x_3) \;, \quad 0 \le \alpha_1 + \alpha_2 + \alpha_3 \le n \;, \tag{2.14}$$

$\hat{T}_j$ being the normalized Chebyshev polynomial of degree j, i.e. $\hat{T}_0(\cdot) = 1$ and $\hat{T}_j(\cdot) = \sqrt{2}\cos(j \arccos(\cdot))$ for $j > 0$. The hyperinterpolation polynomial is not interpolant [24], since $N_3 > \dim(\Pi_n^3([-1,1]^3)) = \dim(\Pi_n^3) = (n+1)(n+2)(n+3)/6$.

Remark We have treated only the hyperinterpolation points $X_{n+1} \times T_{n+1}$ for simplicity, but there are clearly three possible families of such product points in the cube. Indeed, we can take also X_{n+1} in the second and third variables and T_{n+1} in the first, or X_{n+1} in the first and third variables and T_{n+1} in the second. Correspondingly, we have three hyperinterpolation polynomials of degree n.

2.1 Growth of the Lebesgue constant

The effectiveness of the projection operator (2.13) as an approximation tool in the uniform norm, depends on the growth of its norm (the so-called Lebesgue constant), cf. (1.5). Now, in view of (1.2), (2.14) and of the following bound for the weights, $w_{\boldsymbol{\xi}} = w_{(\xi_1,\xi_2)} w_{\xi_3} \leq 2(n+1)^{-3}$, we have the estimate

$$\Lambda_n \leq \max_{\boldsymbol{x}\in\Omega} \left\{ \sum_{k=0}^{n} \sum_{|\alpha|=k} |p_\alpha(\boldsymbol{x})| \sum_{\boldsymbol{\xi}\in X_{n+1}\times T_{n+1}} w_{(\xi_1,\xi_2)} w_{\xi_3} \, |p_\alpha(\boldsymbol{\xi})| \right\}$$

$$\leq \frac{16 N_3}{(n+1)^3} \dim(\Pi_n^3) = 16\, \frac{(n+2)^2(n+3)^2}{12(n+1)} \sim \frac{4}{3}\, n^3 \ , \tag{2.15}$$

which already shows that (2.13) is not a bad candidate for approximation in the uniform norm.

However, (2.15) turns out to be by far an overestimate of the actual Lebesgue constant. Indeed, a wide set of numerical experiments on the maximization of the Lebesgue function

$$\lambda_n(\boldsymbol{x}) = \sum_{\boldsymbol{\xi}\in X_{n+1}\times T_{n+1}} w_{(\xi_1,\xi_2)} w_{\xi_3} \, |K_n(\boldsymbol{x},\boldsymbol{\xi})| \ , \tag{2.16}$$

up to degree $n = 100$, has shown that the maximum seems to be attained at the vertices of the cube and to increase much more slowly, namely like $\mathcal{O}(\log^3(n))$. See Figure 1, where are displayed the computed Lebesgue constant and its least-square fitting by a cubic polynomial in $\log(n+1)$ on $n \in \{10,\ldots,70\}$, namely $(2/\pi)^3 \log^3(n+1) - 1.3\log^2(n+1) + 9.6\log(n+1) - 8.3$, up to degree $n = 100$.

These numerical results lead to state the following

Conjecture 2.1. *The Lebesgue function of the hyperinterpolation operator in the cube at the product points* $X_{n+1} \times T_{n+1}$ *can be bounded as*

$$\Lambda_n = \max_{\boldsymbol{x} \in [-1,1]^3} \lambda_n(\boldsymbol{x}) \lesssim \left(\frac{2}{\pi} \log(n+1) \right)^3 , \quad n \to \infty . \tag{2.17}$$

Moreover, the maximum is attained at the vertices of the cube.

Notice that this is just the asymptotic growth of the Lebesgue constant of trivariate tensor-product interpolation of degree n at $(n+1)^3$ product Chebyshev-Lobatto points [7].

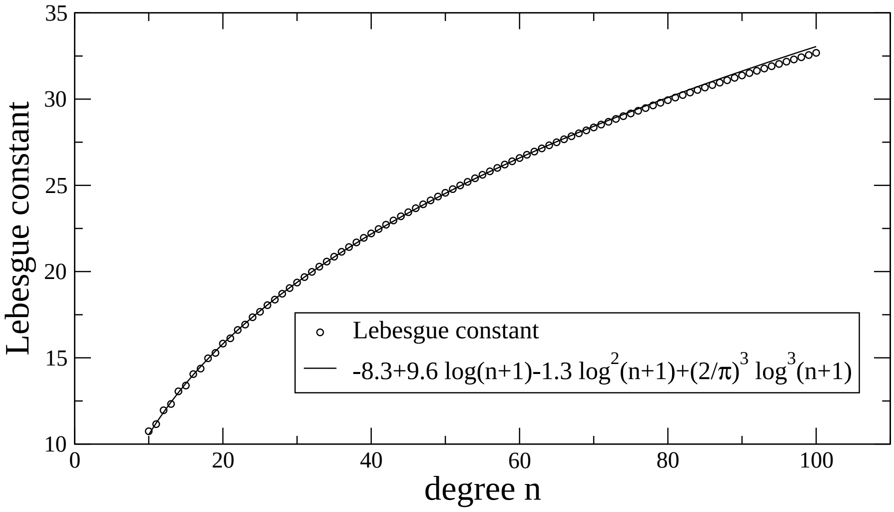

Figure 1: The Lebesgue constant of hyperinterpolation in the cube up to degree 100.

2.2 Implementation

From the computational point of view, the representation of the hyperinterpolation polynomial in the underlying orthonormal basis $\{p_\alpha\}$ is more convenient. This is immediately obtained by (2.13) and (1.2)

$$L_n f(\boldsymbol{x}) = \sum_{k=0}^{n} \sum_{|\alpha|=k} c_\alpha \, p_\alpha(\boldsymbol{x}) , \tag{2.18}$$

where

$$c_\alpha = c_\alpha(f) = \sum_{\boldsymbol{\xi} \in X_{n+1} \times T_{n+1}} w_{(\xi_1,\xi_2)} w_{\xi_3} \, p_\alpha(\boldsymbol{\xi}) \, f(\boldsymbol{\xi}) . \tag{2.19}$$

It is worth stressing that the coefficients $\{c_\alpha\}$ can be computed once and for all, as soon as the function has been sampled at $X_{n+1} \times T_{n+1}$. This means that, whereas the number of function evaluations is $N_3 \sim n^3/2$ (cf. (2.12)), the function is then "compressed", up to the hyperinterpolation error (see the next section), into $(n+1)(n+2)(n+3)/6 \sim n^3/6$ coefficients.

Given the sample, the computation of the coefficients $\{c_\alpha\}$ can be organized as follows. In view of (2.14), it is convenient to precompute $\{\hat{T}_s(\xi_i)\}$, $0 \le s \le n$, $i = 1, 2, 3$, for every hyperinterpolation point $\boldsymbol{\xi}$, by the three-term recurrence of the Chebyshev polynomials. This has a cost of the order of $2n^4$ flops. The bulk of the procedure is then given by the evaluation of the sum in (2.19) for every α, which has a total cost of about $3N_3 \dim(\Pi_n^3) \sim 5n^6/12$ flops.

A computational complexity growing like $\mathcal{O}(n^6)$ seems prohibitive when a high hyperinterpolation degree is needed. There is, however, a simple way to reduce the complexity to $\mathcal{O}(n^5)$ and the overall computing time. Indeed, by ordering the multiindexes $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ as $(i, j-i, k-j)$, we can rewrite the coefficients in the following way (cf. (2.14))

$$c_\alpha = c_{(i,j-i,k-j)} = \sum_{\xi_3 \in T_{n+1}} w_{\xi_3} \, \hat{T}_{k-j}(\xi_3) \, c_{i,j-i}(\xi_3) \;, \quad 0 \le i \le j \le k \le n \;,$$

$$c_{i,j-i}(\xi_3) = \sum_{(\xi_1,\xi_2) \in X_{n+1}} w_{(\xi_1,\xi_2)} \, \hat{T}_i(\xi_1) \, \hat{T}_{j-i}(\xi_2) \, f(\boldsymbol{\xi}) \;. \qquad (2.20)$$

Then, for every $\xi_3 \in T_{n+1}$ we can compute the intermediate coefficients $\{c_{i,j-i}(\xi_3)\}$ as elements of a triangular $(n+1) \times (n+1)$ matrix $C(\xi_3) = [c_{s,t}(\xi_3)]$, $0 \le s \le t \le n$, using *optimized linear algebra* routines instead of "`for`" loops. This is just the computational trick adopted for bivariate hyperinterpolation [8] (indeed the $\{c_{i,j-i}(\xi_3)\}$ are bivariate hyperinterpolation coefficients). In our Fortran implementation of hyperinterpolation in the cube [11], we have used optimized BLAS libraries [1], which allow to obtain considerable speed-ups at high degree. See Table 1, where we report the CPU time (seconds) for the computation of the hyperinterpolation coefficients $\{c_\alpha\}$ of a given sample at a sequence of degrees, by the basic algorithm (only "`for`" loops) and by the matrix algorithm with optimized matrix products (the tests have been done on a AMD Athlon 2800+ processor machine with 2Gb RAM).

Table 1: Computation of the hyperinterpolation coefficients $\{c_\alpha\}$ at a sequence of degrees.

	$n=10$	$n=20$	$n=30$	$n=40$	$n=50$	$n=60$
pts.	864	5324	16384	37044	70304	119164
coeffs.	286	1771	5456	12341	23426	39711
CPU (s.)						
basic	0.010	0.165	1.829	16.34	58.62	165.63
optBLAS	0.007	0.0018	0.089	0.269	0.672	1.481
speed-up	1.4	9.2	20.6	60.7	87.2	111.8

2.3 Error estimates

In view of the multivariate extension of Jackson's theorem (cf., e.g., Bagby et al. [2] and references therein), we have that for $f \in C^m([-1,1]^3)$, $m > 0$,

$$\|f - L_n f\|_\infty \leq (1 + \Lambda_n)\, E_n(f) \leq \mathcal{C}(f;m)\,(1+\Lambda_n)\; n^{-m}\;, \tag{2.21}$$

where $\mathcal{C}$ is a suitable constant (with n), dependent on f and m. In view of the conjecture above on the growth of the Lebesgue constant, we then expect convergence for such f, with the rate given in (2.21). However, this "a priori" estimate is essentially qualitative.

An important feature would be the availability of a reliable and if possible "a posteriori" quantitative estimate of the error. To this purpose, here we can use the fact that hyperinterpolation is a discretized truncated Fourier-Chebyshev expansion.

First, observe that for polynomials of degree not greater than n the $\{c_\alpha\}$ are exactly the Fourier-Chebyshev coefficients, i.e. for every α, $0 \leq |\alpha| \leq n$, we have $c_\alpha(f) = \varphi_\alpha(f)$, where

$$\varphi_\alpha(f) = \int_{[-1,1]^3} p_\alpha(\boldsymbol{x})\, f(\boldsymbol{x})\, d\mu_3\;,\quad \forall\, f \in \Pi_n^3\;. \tag{2.22}$$

Now, consider c_α and φ_α as linear functionals on $C([-1,1]^3, \|\cdot\|_\infty)$. In view of the fact that the cubature formula (2.11) is exact on constants, denoting by p_n^* the best uniform polynomial approximation of degree n to $f \in C([-1,1]^3)$,

$$|c_\alpha(f) - \varphi_\alpha(f)| \leq (\|c_\alpha\| + \|\varphi_\alpha\|)\,\|f - p_n^*\|_\infty$$

$$\leq 2\sqrt{2}\left(\sum_{\boldsymbol{\xi}\in X_{n+1}\times T_{n+1}} w_{(\xi_1,\xi_2)} w_{\xi_3} + \int_{[-1,1]^3} d\mu_3\right)\|f - p_n^*\|_\infty = 4\sqrt{2}E_n(f)\;, \tag{2.23}$$

for every α. Moreover, observing that L_n and the truncated Fourier-Chebyshev expansion $\mathcal{S}_n$ (cf. (1.1)) are both projection operators on Π_n^3, we have

$$\|L_n f - \mathcal{S}_n f\|_\infty \leq (\|L_n\| + \|\mathcal{S}_n\|) \|f - p_n^*\|_\infty = \left(\Lambda_n + \mathcal{O}\left(\log^3(n)\right) E_n(f)\right) . \tag{2.24}$$

Estimates of Λ_n have been discussed in section 2.1, whereas the fact that $\|\mathcal{S}_n\| = \mathcal{O}(\log^3(n))$ is within the much more general results recently proved by Vertesi [26]. We can now give the following *a posteriori* error estimate, by the chain of estimates

$$\|f - L_n f\|_\infty \approx \|f - \mathcal{S}_n f\|_\infty \leq 2\sqrt{2} \sum_{k=n+1}^{\infty} \sum_{|\alpha|=k} |\varphi_\alpha(f)|$$

$$\approx 2\sqrt{2} \sum_{k=n-1}^{n} \sum_{|\alpha|=k} |\varphi_\alpha(f)| \approx 2\sqrt{2} \sum_{k=n-1}^{n} \sum_{|\alpha|=k} |c_\alpha(f)| . \tag{2.25}$$

The passage from the first to the second row in (2.25), though empirical, is reminiscent of popular error estimates for one-dimensional Chebyshev expansions, based on the size of the last two coefficients [3]. Here, we resort indeed to the coefficients corresponding to the last two degrees k, namely $n-1$ and n. Notice that the first and the last approximation in (2.25) are justified by (2.21) and (2.24), and (2.23), respectively. The latter, in particular, seems to give an overestimate of the final error by an order of $\mathcal{O}(n^2 E_n(f))$. In practice, however, as in the one-dimensional case it happens that (2.25) tends to be an overestimate for smooth functions, and an underestimate for functions of low regularity (due to fast/slow decay of the Fourier-Chebyshev coefficients). The behavior of this error estimate has been satisfactory in almost all our numerical tests (see the next section).

3 Numerical tests

In Table 2 we show the hyperinterpolation errors on a suite of six trivariate test functions in $[0,1]^3$ which exhibit a variety of behavior at a sequence of degrees, $n = 10, 20, \ldots, 60$:

$$\begin{aligned}
F_1(x_1,x_2,x_3) &= .75\exp[-((9x_1-2)^2+(9x_2-2)^2+(9x_3-2)^2)/4] \\
&+ .75\exp[-(9x_1+1)^2/49-(9x_2+1)/10-(9x_3+1)/10] \\
&+ .5\exp[-((9x_1-7)^2+(9x_2-3)^2+(9x_3-5)^2)/4] \\
&- .2\exp[-(9x_1-4)^2-(9x_2-7)^2-(9x_3-5)^2]; \\
F_2(x_1,x_2,x_3) &= [\tanh(9x_3-9x_1-9x_2)+1]/9; \\
F_3(x_1,x_2,x_3) &= [1.25+\cos(5.4x_2)]\cos(6x_3)/[6+6(3x_1-1)^2]; \\
F_4(x_1,x_2,x_3) &= \exp[-(81/16)((x_1-.5)^2+(x_2-.5)^2+(x_3-.5)^2)]/3; \\
F_5(x_1,x_2,x_3) &= \exp[-(81/4)((x_1-.5)^2+(x_2-.5)^2+(x_3-.5)^2)]/3; \\
F_6(x_1,x_2,x_3) &= [64-81((x_1-.5)^2+(x_2-.5)^2+(x_3-.5)^2)]^{1/2}/9-.5
\end{aligned}$$

We notice that these functions were also considered by Renka in [23]. We observe that extending hyperinterpolation from the reference cube $[-1,1]^3$ to any *parallelepiped* is trivial via the usual affine componentwise change of variables. The errors are measured in the max-norm normalized to the max deviation of the function from its mean. The "true" errors have been computed on a $30 \times 30 \times 30$ uniform control grid. In the table we report also (in parenthesis) the a posteriori empirical error estimate for hyperinterpolation, given by the last term of (2.25) normalized as above. The tests have been done by the Fortran code `HyperCube` [11].

From the numerical tests we can see that the hyperinterpolation formula in the cube, computed as described above via its representation in the Chebyshev orthogonal basis, is a stable and efficient approximation tool for functions with some regularity that can be sampled without restrictions. It can be used at high degree without serious drawbacks, and is accompanied by a satisfactory error estimate.

References

[1] AMD Core Math Library (ACML), Version 3.1.0 (2006), available at http://developer.amd.com/acml.aspx.

[2] T. Bagby, L. Bos and N. Levenberg, Multivariate simultaneous approximation, Constr. Approx. 18 (2002) 569–577.

[3] Z. Battles and L.N. Trefethen, An extension of MATLAB to continuous functions and operators, SIAM J. Sci. Comput. 25 (2004) 1743–1770.

Table 2: Hyperinterpolation errors and their a posteriori estimates (in parenthesis) on Renka's test functions in the unit cube.

n	F_1	F_2	F_3	F_4	F_5	F_6
10	1.5E-1	2.1E-1	2.0E-2	4.9E-4	1.5E-1	1.5E-2
	(3.4E-1)	(8.7E-1)	(1.5E-1)	(4.1E-3)	(1.8E-1)	(1.5E-2)
20	3.4E-2	5.8E-2	2.8E-5	1.5E-9	9.7E-4	7.2E-4
	(3.8E-2)	(2.7E-1)	(2.3E-4)	(2.3E-8)	(2.5E-3)	(5.7E-4)
30	3.1E-3	1.6E-2	3.6E-8	7.2E-15	8.7E-7	4.5E-5
	(4.8E-3)	(8.0E-2)	(3.6E-7)	(1.4E-14)	(3.9E-6)	(3.2E-5)
40	1.3E-4	4.8E-3	6.2E-11	2.5E-14	2.0E-10	3.1E-6
	(2.7E-4)	(2.3E-2)	(5.5E-10)	(8.4E-15)	(1.4E-9)	(2.2E-6)
50	2.5E-6	1.4E-3	8.1E-14	3.5E-14	1.9E-14	2.4E-7
	(7.3E-6)	(6.6E-3)	(8.7E-13)	(1.1E-14)	(1.5E-13)	(1.6E-7)
60	2.4E-8	4.2E-4	3.3E-14	3.3E-14	5.0E-15	1.8E-8
	(1.1E-7)	(1.9E-3)	(1.1E-14)	(1.1E-14)	(1.6E-15)	(1.2E-8)

[4] L. Bos, M. Caliari, S. De Marchi, M. Vianello and Y. Xu, Bivariate Lagrange interpolation at the Padua points: The generating curve approach, J. Approx. Theory 141 (2006) 134–141.

[5] L. Bos, M. Caliari, S. De Marchi and M. Vianello, Bivariate interpolation at Xu points: results, extensions and applications, Electron. Trans. Numer. Anal. 25 (2006) 1–16.

[6] L. Bos, S. De Marchi and M. Vianello, On the Lebesgue constant for the Xu interpolation formula, J. Approx. Theory 141 (2006) 134–141.

[7] L. Brutman, Lebesgue functions for polynomial interpolation - a survey, Ann. Numer. Math. 4 (1997) 111–127.

[8] M. Caliari, S. De Marchi, R. Montagna and M. Vianello, `Hyper2d`: A numerical code for hyperinterpolation on rectangles, Appl. Math. Comput. 183 (2006) 1138–1147.

[9] M. Caliari, S. De Marchi and M. Vianello, Bivariate polynomial interpolation on the square at new nodal sets, Appl. Math. Comput. 165 (2005) 261–274.

[10] M. Caliari, S. De Marchi and M. Vianello, Hyperinterpolation on the square, J. Comput. Appl. Math. 210(1-2) (2007) 78–83.

[11] M. Caliari, S. De Marchi and M. Vianello, `HyperCube`: Fortran 77 code for polynomial hyperinterpolation in the cube, downloadable at: http://www.math.unipd.it/∼marcov/software.html.

[12] R. Cools, Constructing cubature formulae: the science behind the art, Acta numerica 6 (1997) 1–54.

[13] R. Cools, An encyclopaedia of cubature formulas, J. Complexity 19 (2003) 445–453.

[14] R. Cools and H.J. Schmid, Minimal cubature formulae of degree $2k-1$ for two classical functionals, Computing 43 (1989), 141–157.

[15] B. Della Vecchia, G. Mastroianni and P. Vertesi, Exact order of the Lebesgue constants for bivariate Lagrange interpolation at certain node systems, 2006, to appear.

[16] C.F. Dunkl and Y. Xu, Orthogonal Polynomials of Several Variables, Encyclopedia of Mathematics and its Applications, vol. 81, Cambridge University Press, Cambridge, 2001.

[17] M. Gasca and T. Sauer, Polynomial interpolation in several variables, Adv. Comput. Math. 12 (2000) 377–410.

[18] O. Hansen, K. Atkinson and D. Chien, On the norm of the hyperinterpolation operator on the unit disk, Reports on Computational Mathematics n.167 (2006), Dept of Mathematics, University of Iowa.

[19] H.M. Möller, Lower bounds for the number of nodes in cubature formulae, in: Numerische Integration (Tagung, Math. Forschungsinst., Oberwolfach, 1978), pp. 221–230, Internat. Ser. Numer. Math., 45, Birkhuser, Basel-Boston, Mass., 1979.

[20] C.R. Morrow and T.N.L. Patterson, Construction of algebraic cubature rules using polynomial ideal theory, SIAM J. Numer. Anal. 15 (1978) 953–976.

[21] I.P. Omelyan and V.B. Solovyan, Improved cubature formulae of high degrees of exactness for the square, J. Comput. Appl. Math. 188 (2006) 190–204.

[22] M. Reimer, Multivariate Polynomial approximation, International Series of Numerical Mathematics, vol. 144, Birkhäuser, 2003.

[23] R.J. Renka, Multivariate Interpolation of Large Sets of Scattered Data, ACM Trans. Math. Software 14 (1988) 139–148.

[24] I. Sloan, Interpolation and Hyperinterpolation over General regions, J. Approx. Theory 83 (1995) 238–254.

[25] I. Sloan and R. Womersley, Constructive polynomial approximation on the sphere, J. Approx. Theory 103 (2000) 91–118.

[26] P. Vertesi, On multivariate projection operators, 2006, to appear.

[27] Y. Xu, Lagrange interpolation on Chebyshev points of two variables, J. Approx. Theory 87 (1996) 220–238.