

Article

dCATCH—A Numerical Package for d -Variate near G-Optimal Tchakaloff Regression via Fast NNLS

Monica Dessole, Fabio Marcuzzi  and Marco Vianello *

Department of Mathematics “Tullio Levi Civita”, University of Padova, Via Trieste 63, 35131 Padova, Italy; mdessole@math.unipd.it (M.D.); marcuzzi@math.unipd.it (F.M.)

* Correspondence: marcov@math.unipd.it

Received: 11 June 2020; Accepted: 7 July 2020; Published: 9 July 2020



Abstract: We provide a numerical package for the computation of a d -variate near G-optimal polynomial regression design of degree m on a finite design space $X \subset \mathbb{R}^d$, by few iterations of a basic multiplicative algorithm followed by Tchakaloff-like compression of the discrete measure keeping the reached G-efficiency, via an accelerated version of the Lawson-Hanson algorithm for Non-Negative Least Squares (NNLS) problems. This package can solve on a personal computer large-scale problems where $\text{card}(X) \times \dim(P_{2m}^d)$ is up to 10^8 – 10^9 , being $\dim(P_{2m}^d) = \binom{2m+d}{d} = \binom{2m+d}{2m}$. Several numerical tests are presented on complex shapes in $d = 3$ and on hypercubes in $d > 3$.

Keywords: multivariate polynomial regression designs; G-optimality; D-optimality; multiplicative algorithms; G-efficiency; Caratheodory-Tchakaloff discrete measure compression; Non-Negative Least Squares; accelerated Lawson-Hanson solver

1. Introduction

In this paper we present the numerical software package *dCATCH* [1] for the computation of a d -variate near G-optimal polynomial regression design of degree m on a finite design space $X \subset \mathbb{R}^d$. In particular, it is the first software package for general-purpose Tchakaloff-like compression of d -variate designs via Non-Negative Least Squares (NNLS), freely available on the Internet. The code is an evolution of the codes in Reference [2] (limited to $d = 2, 3$), with a number of features tailored to higher dimension and large-scale computations. The key ingredients are:

- use of d -variate Vandermonde-like matrices at X in a discrete orthogonal polynomial basis (obtained by discrete orthonormalization of the total-degree product Chebyshev basis of the minimal box containing X), with automatic adaptation to the actual dimension of $\mathbb{P}_m^d(X)$;
- few tens of iterations of the basic Titterton multiplicative algorithm until near G-optimality of the design is reached, with a checked G-efficiency of say 95% (but with a design support still far from sparsity);
- Tchakaloff-like compression of the resulting near G-optimal design via NNLS solution of the underdetermined moment system, with concentration of the discrete probability measure by sparse re-weighting to a support $\subset X$, of cardinality at most $\mathbb{P}_{2m}^d(X)$, keeping the same G-efficiency;
- iterative solution of the large-scale NNLS problem by a new accelerated version of the classical Lawson-Hanson active set algorithm, that we recently introduced in Reference [3] for $2d$ and $3d$ instances and here we validate on higher dimensions.

Before giving a more detailed description of the algorithm, it is worth recalling in brief some basic notions of optimal design theory. Such a theory has its roots and main applications within statistics, but also strong connections with approximation theory. In statistics, a design is a probability

measure μ supported on a (discrete or continuous) compact set $\Omega \subset \mathbb{R}^d$. The search for designs that optimize some properties of statistical estimators (optimal designs) dates back to at least one century ago, and the relevant literature is so wide and still actively growing and monographs and survey papers are abundant in the literature. For readers interested in the evolution and state of the art of this research field, we may quote, for example, two classical treatises such as in References [4,5], the recent monograph [6] and the algorithmic survey [7], as well as References [8–10] and references therein. On the approximation theory side we may quote, for example, References [11,12].

The present paper is organized as follows—in Section 2 we briefly recall some basic concepts from the theory of Optimal Designs, for the reader’s convenience, with special attention to the deterministic and approximation theoretic aspects. In Section 3 we present in detail our computational approach to near G-optimal d -variate designs via Caratheodory-Tchakaloff compression. All the routines of the *dCATCH* software package here presented, are described. In Section 4 we show several numerical results with dimensions in the range 3–10 and a Conclusions section follows.

For the reader’s convenience we also display Tables 1 and 2, describing the acronyms used in this paper and the content (subroutine names) of the *dCATCH* software package.

Table 1. List of acronyms.

LS	Least Squares
NNLS	Non-Negative Least Squares
LH	Lawson-Hawson algorithm for NNLS
LHI	Lawson-Hawson algorithm with unconstrained LS Initialization
LHDM	Lawson-Hawson algorithm with Deviation Maximization acceleration

Table 2. *dCATCH* package content.

dCATCH	d -variate CARatheodory-TCHakaloff discrete measure compression
dCHEBVAND	d -variate Chebyshev-Vandermonde matrix
dORTHVAND	d -variate Vandermonde-like matrix in a weighted orthogonal polynomial basis
dNORD	d -variate Near G-Optimal Regression Designs
LHDM	Lawson-Hawson algorithm with Deviation Maximization acceleration

2. G-Optimal Designs

Let $\mathbb{P}_m^d(\Omega)$ denote the space of d -variate real polynomials of total degree not greater than n , restricted to a (discrete or continuous) compact set $\Omega \subset \mathbb{R}^d$, and let μ be a design, that is, a probability measure, with $supp(\mu) \subseteq \Omega$. In what follows we assume that $supp(\mu)$ is *determining* for $\mathbb{P}_m^d(\Omega)$ [13], that is, polynomials in \mathbb{P}_m^d vanishing on $supp(\mu)$ vanish everywhere on Ω .

In the theory of optimal designs, a key role is played by the diagonal of the reproducing kernel for μ in $\mathbb{P}_m^d(\Omega)$ (also called the Christoffel polynomial of degree m for μ)

$$K_m^\mu(x, x) = \sum_{j=1}^{N_m} p_j^2(x), \quad N_m = \dim(\mathbb{P}_m^d(\Omega)), \tag{1}$$

where $\{p_j\}$ is any μ -orthonormal basis of $\mathbb{P}_m^d(\Omega)$. Recall that $K_m^\mu(x, x)$ can be proved to be independent of the choice of the orthonormal basis. Indeed, a relevant property is the following estimate of the L^∞ -norm in terms of the L_μ^2 -norm of polynomials

$$\|p\|_{L^\infty(\Omega)} \leq \sqrt{\max_{x \in \Omega} K_m^\mu(x, x)} \|p\|_{L_\mu^2(\Omega)}, \quad \forall p \in \mathbb{P}_m^d(\Omega). \tag{2}$$

Now, by (1) and μ -orthonormality of the basis we get

$$\int_{\Omega} K_m^\mu(x, x) d\mu = \sum_{j=1}^{N_m} \int_{\Omega} p_j^2(x) d\mu = N_m, \tag{3}$$

which entails that $\max_{x \in \Omega} K_m^\mu(x, x) \geq N_m$.

Then, a probability measure $\mu_* = \mu_*(\Omega)$ is then called a G-optimal design for polynomial regression of degree m on Ω if

$$\min_{\mu} \max_{x \in \Omega} K_m^\mu(x, x) = \max_{x \in \Omega} K_m^{\mu_*}(x, x) = N_m. \tag{4}$$

Observe that, since $\int_{\Omega} K_m^\mu(x, x) d\mu = N_m$ for every μ , an optimal design has also the following property $K_m^{\mu_*}(x, x) = N_m, \mu_*$ -a.e. in Ω .

Now, the well-known Kiefer-Wolfowitz General Equivalence Theorem [14] (a cornerstone of optimal design theory), asserts that the difficult min-max problem (4) is equivalent to the much simpler maximization problem

$$\max_{\mu} \det(G_m^\mu), \quad G_m^\mu = \left(\int_{\Omega} \phi_i(x)\phi_j(x) d\mu \right)_{1 \leq i, j \leq N_m},$$

where G_m^μ is the Gram matrix (or information matrix in statistics) of μ in a fixed polynomial basis $\{\phi_i\}$ of $\mathbb{P}_m^d(\Omega)$. Such an optimality is called D-optimality, and ensures that an optimal measure always exists, since the set of Gram matrices of probability measures is compact and convex; see for example, References [5,12] for a general proof of these results, valid for continuous as well as for discrete compact sets.

Notice that an optimal measure is neither unique nor necessarily discrete (unless Ω is discrete itself). Nevertheless, the celebrated Tchakaloff Theorem ensures the existence of a positive quadrature formula for integration in $d\mu_*$ on Ω , with cardinality not exceeding $N_{2m} = \dim(\mathbb{P}_{2m}^d(\Omega))$ and which is exact for all polynomials in $\mathbb{P}_{2m}^d(\Omega)$. Such a formula is then a design itself, and it generates the same orthogonal polynomials and hence the same Christoffel polynomial of μ_* , preserving G-optimality (see Reference [15] for a proof of Tchakaloff Theorem with general measures).

We recall that G-optimality has two important interpretations in terms of statistical and deterministic polynomial regression.

From a statistical viewpoint, it is the probability measure on Ω that minimizes the maximum prediction variance by polynomial regression of degree m , cf. for example, Reference [5].

On the other hand, from an approximation theory viewpoint, if we call $\mathcal{L}_m^{\mu_*}$ the corresponding weighted least squares projection operator $L^\infty(\Omega) \rightarrow \mathbb{P}_m^d(\Omega)$, namely

$$\|f - \mathcal{L}_m^{\mu_*} f\|_{L_{\mu_*}^2(\Omega)} = \min_{p \in \mathbb{P}_m^d(\Omega)} \|f - p\|_{L_{\mu_*}^2(\Omega)}, \tag{5}$$

by (2) we can write for every $f \in L^\infty(\Omega)$

$$\begin{aligned} \|\mathcal{L}_m^{\mu_*} f\|_{L^\infty(\Omega)} &\leq \sqrt{\max_{x \in \Omega} K_m^{\mu_*}(x, x)} \|\mathcal{L}_m^{\mu_*} f\|_{L_{\mu_*}^2(\Omega)} = \sqrt{N_m} \|\mathcal{L}_m^{\mu_*} f\|_{L_{\mu_*}^2(\Omega)} \\ &\leq \sqrt{N_m} \|f\|_{L_{\mu_*}^2(\Omega)} \leq \sqrt{N_m} \|f\|_{L^\infty(\Omega)}, \end{aligned}$$

(where the second inequality comes from μ_* -orthogonality of the projection), which gives

$$\|\mathcal{L}_m^{\mu_*}\| = \sup_{f \neq 0} \frac{\|\mathcal{L}_m^{\mu_*} f\|_{L^\infty(\Omega)}}{\|f\|_{L^\infty(\Omega)}} \leq \sqrt{N_m}, \tag{6}$$

that is a G-optimal measure minimizes (the estimate of) the weighted least squares uniform operator norm.

We stress that in this paper we are interested in the fully discrete case of a finite design space $\Omega = X$, so that any design μ is identified by a set of positive weights (masses) summing up to 1 and integrals are weighted sums.

3. Computing near G-Optimal Compressed Designs

Since in the present context we have a finite design space $\Omega = X = \{x_1, \dots, x_M\} \subset \mathbb{R}^d$, we may think a design μ as a vector of non-negative weights $u = (u_1, \dots, u_M)$ attached to the points, such that $\|u\|_1 = 1$ (the support of μ being identified by the positive weights). Then, a G-optimal (or D-optimal) design μ_* is represented by the corresponding non-negative vector u_* . We write $K_m^u(x, x) = K_m^u(x, x)$ for the Christoffel polynomial and similarly for other objects (spaces, operators, matrices) corresponding to a discrete design. At the same time, $L^\infty(\Omega) = \ell^\infty(X)$, and $L_\mu^2(\Omega) = \ell_u^2(X)$ (a weighted ℓ^2 functional space on X) with $\|f\|_{\ell_u^2(X)} = \left(\sum_{i=1}^M u_i f^2(x_i)\right)^{1/2}$.

In order to compute an approximation of the desired u_* , we resort to the basic multiplicative algorithm proposed by Titterton in the '70s (cf. Reference [16]), namely

$$u_i(k+1) = u_i(k) \frac{K_m^{u(k)}(x_i, x_i)}{N_m}, \quad 1 \leq i \leq M, \quad k = 0, 1, 2, \dots, \tag{7}$$

with initialization $u(0) = (1/M, \dots, 1/M)^T$. Such an algorithm is known to be convergent sublinearly to a D-optimal (or G-optimal by the Kiefer-Wolfowitz Equivalence Theorem) design, with an increasing sequence of Gram determinants

$$\det(G_m^{u(k)}) = \det(V^T \text{diag}(u(k)) V),$$

where V is a Vandermonde-like matrix in any fixed polynomial basis of $\mathbb{P}_m^d(X)$; cf., for example, References [7,10]. Observe that $u(k+1)$ is indeed a vector of positive probability weights if such is $u(k)$. In fact, the Christoffel polynomial $K_m^{u(k)}$ is positive on X , and calling μ_k the probability measure on X associated with the weights $u(k)$ we get immediately $\sum_i u_i(k+1) = \frac{1}{N_m} \sum_i u_i(k) K_m^{u(k)}(x_i, x_i) = \frac{1}{N_m} \int_X K_m^{u(k)}(x, x) d\mu_k = 1$ by (3) in the discrete case $\Omega = X$.

Our implementation of (7) is based on the functions

- $C = \text{dCHEBVAND}(n, X)$
- $[U, jvec] = \text{dORTHVAND}(n, X, u, jvec)$
- $[pts, w] = \text{dNORD}(m, X, gtol)$

The function `dCHEBVAND` computes the d -variate Chebyshev-Vandermonde matrix $C = (\phi_j(x_i)) \in \mathbb{R}^{M \times N_n}$, where $\{\phi_j(x)\} = \{T_{v_1}(\alpha_1 x_1 + \beta_1) \dots T_{v_d}(\alpha_d x_d + \beta_d)\}, 0 \leq v_i \leq n, v_1 + \dots + v_d \leq n$, is a suitably ordered total-degree product Chebyshev basis of the minimal box $[a_1, b_1] \times \dots \times [a_d, b_d]$ containing X , with $\alpha_i = 2/(b_i - a_i), \beta_i = -(b_i + a_i)/(b_i - a_i)$. Here we have resorted to the codes in Reference [17] for the construction and enumeration of the required "monomial" degrees. Though the initial basis is then orthogonalized, the choice of the Chebyshev basis is dictated by the necessity of controlling the conditioning of the matrix. This would be on the contrary extremely large with the standard monomial basis, already at moderate regression degrees, preventing a successful orthogonalization.

Indeed, the second function `dORTHVAND` computes a Vandermonde-like matrix in a u -orthogonal polynomial basis on X , where u is the probability weight array. This is accomplished essentially by numerical rank evaluation for $C = \text{dCHEBVAND}(n, X)$ and QR factorization

$$\text{diag}(\sqrt{u}) C_0 = QR, \quad U = C_0 R^{-1}, \tag{8}$$

(with Q orthogonal rectangular and R square invertible), where $\sqrt{u} = (\sqrt{u_1}, \dots, \sqrt{u_M})$. The matrix C_0 has full rank and corresponds to a selection of the columns of C (i.e., of the original basis polynomials) via QR with column pivoting, in such a way that these form a basis of $\mathbb{P}_n^d(X)$, since $\text{rank}(C) = \dim(\mathbb{P}_n^d(X))$. A possible alternative, not yet implemented, is the direct use of a rank-revealing QR factorization. The in-out parameter "jvec" allows to pass directly the column index

vector corresponding to a polynomial basis after a previous call to dORTHVAND with the same degree n , avoiding numerical rank computation and allowing a simple “economy size” QR factorization of $diag(\sqrt{u}) C_0 = diag(\sqrt{u}) C(:, jvec)$.

Summarizing, U is a Vandermonde-like matrix for degree n on X in the required u -orthogonal basis of $\mathbb{P}_n^d(X)$, that is

$$[p_1(x), \dots, p_{N_n}(x)] = [\phi_{j_1}(x), \dots, \phi_{j_{N_n}}(x)] R^{-1}, \tag{9}$$

where $jvec = (j_1, \dots, j_{N_n})$ is the multi-index resulting from pivoting. Indeed by (8) we can write the scalar product $(p_h, p_k)_{\ell_u^2(X)}$ as

$$(p_h, p_k)_{\ell_u^2(X)} = \sum_{i=1}^M u_i p_h(x_i) p_k(x_i) = (U^T diag(u) U)_{hk} = (Q^T Q)_{hk} = \delta_{hk},$$

for $1 \leq h, k \leq N_n$, which shows orthonormality of the polynomial basis in (9).

We stress that $rank(C) = \dim(\mathbb{P}_n^d(X))$ could be strictly smaller than $\dim(\mathbb{P}_n^d) = \binom{n+d}{d}$, when there are polynomials in \mathbb{P}_n^d vanishing on X that do not vanish everywhere. In other words, X lies on a lower-dimensional algebraic variety (technically one says that X is not \mathbb{P}_n^d -determining [13]). This certainly happens when $card(X)$ is too small, namely $card(X) < \dim(\mathbb{P}_n^d)$, but think for example also to the case when $d = 3$ and X lies on the 2-sphere S^2 (independently of its cardinality), then we have $\dim(\mathbb{P}_n^d(X)) \leq \dim(\mathbb{P}_n^d(S^2)) = (n + 1)^2 < \dim(\mathbb{P}_n^3) = (n + 1)(n + 2)(n + 3)/6$.

Iteration (7) is implemented within the third function dNORD whose name stands for d -dimensional Near G-Optimal Regression Designs, which calls dORTHVAND with $n = m$. Near optimality is here twofold, namely it concerns both the concept of G-efficiency of the design and the sparsity of the design support.

We recall that G-efficiency is the percentage of G-optimality reached by a (discrete) design, measured by the ratio

$$G_m(u) = \frac{N_m}{\max_{x \in X} K_m^u(x, x)},$$

knowing that $G_m(u) \leq 1$ by (3) in the discrete case $\Omega = X$. Notice that $G_m(u)$ can be easily computed after the construction of the u -orthogonal Vandermonde-like matrix U by dORTHVAND, as $G_m(u) = N_m / (\max_i \|row_i(U)\|_2^2)$.

In the multiplicative algorithm (7), we then stop iterating when a given threshold of G-efficiency (the input parameter “gtol” in the call to dNORD) is reached by $u(k)$, since $G_m(u(k)) \rightarrow 1$ as $k \rightarrow \infty$, say for example $G_m(u(k)) \geq 95\%$ or $G_m(u(k)) \geq 99\%$. Since convergence is sublinear and in practice we see that $1 - G_m(u(k)) = \mathcal{O}(1/k)$, for a 90% G-efficiency the number of iterations is typically in the tens, whereas it is in the hundreds for 99% one and in the thousands for 99,9%. When a G-efficiency very close to 1 is needed, one could resort to more sophisticated multiplicative algorithms, see for example, References [9,10].

In many applications however a G-efficiency of 90–95% could be sufficient (then we may speak of near G-optimality of the design), but though in principle the multiplicative algorithm converges to an optimal design μ_* on X with weights u_* and cardinality $N_m \leq card(supp(\mu_*)) \leq N_{2m}$, such a sparsity is far from being reached after the iterations that guarantee near G-optimality, in the sense that there is a still large percentage of non-negligible weights in the near optimal design weight vector, say

$$u(\bar{k}) \text{ such that } G_m(u(\bar{k})) \geq gtol. \tag{10}$$

Following References [18,19], we can however effectively compute a design which has the same G-efficiency of $u(\bar{k})$ but a support with a cardinality not exceeding $N_{2m} = \dim(\mathbb{P}_{2m}^d(X))$, where in many applications $N_{2m} \ll card(X)$, obtaining a remarkable compression of the near optimal design.

The theoretical foundation is a generalized version [15] of Tchakaloff Theorem [20] on positive quadratures, which asserts that for every measure on a compact set $\Omega \subset \mathbb{R}^d$ there exists an algebraic

quadrature formula exact on $\mathbb{P}_n^d(\Omega)$, with positive weights, nodes in Ω and cardinality not exceeding $N_n = \dim(\mathbb{P}_n^d(\Omega))$.

In the present discrete case, that is, where the designs are defined on $\Omega = X$, this theorem implies that for every design μ on X there exists a design ν , whose support is a subset of X , which is exact for integration in $d\mu$ on $\mathbb{P}_n^d(X)$. In other words, the design ν has the same basis moments (indeed, for any basis of $\mathbb{P}_n^d(\Omega)$)

$$\int_X p_j(x) d\mu = \sum_{i=1}^M u_i p_j(x_i) = \int_X p_j(x) d\nu = \sum_{\ell=1}^L w_\ell p_j(\xi_\ell), \quad 1 \leq j \leq N_n,$$

where $L \leq N_n \leq M$, $\{u_i\}$ are the weights of μ , $\text{supp}(\nu) = \{\xi_\ell\} \subseteq X$ and $\{w_\ell\}$ are the positive weights of ν . For $L < M$, which certainly holds if $N_n < M$, this represents a compression of the design μ into the design ν , which is particularly useful when $N_n \ll M$.

In matrix terms this can be seen as the fact that the underdetermined $\{p_j\}$ -moment system

$$U_n^T v = U_n^T u \tag{11}$$

has a non-negative solution $v = (v_1, \dots, v_M)^T$ whose positive components, say $w_\ell = v_{i_\ell}$, $1 \leq \ell \leq L \leq N_n$, determine the support points $\{\xi_\ell\} \subseteq X$ (for clarity we indicate here by U_n the matrix U computed by dORTHVAND at degree n). This fact is indeed a consequence of the celebrated Caratheodory Theorem on conic combinations [21], asserting that a linear combination with non-negative coefficients of M vectors in \mathbb{R}^N with $M > N$ can be re-written as linear positive combination of at most N of them. So, we get the discrete version of Tchakaloff Theorem by applying Caratheodory Theorem to the columns of U_n^T in the system (11), ensuring then existence of a non-negative solution v with at most N_n nonzero components.

In order to compute such a solution to (11) we choose the strategy based on Quadratic Programming introduced in Reference [22], namely on sparse solution of the Non-Negative Least Squares (NNLS) problem

$$v = \operatorname{argmin}_{z \in \mathbb{R}^M, z \geq 0} \|U_n^T z - U_n^T u\|_2^2$$

by a new accelerated version of the classical Lawson-Hanson active-set method, proposed in Reference [3] in the framework of design optimization in $d = 2, 3$ and implemented by the function LHDM (Lawson-Hanson with Deviation Maximization), that we tune in the present package for very large-scale d -variate problems (see the next subsection for a brief description and discussion). We observe that working with an orthogonal polynomial basis of $\mathbb{P}_n^d(X)$ allows to deal with the well-conditioned matrix U_n in the Lawson-Hanson algorithm.

The overall computational procedure is implemented by the function

- $[pts, w, momerr] = \text{dCATCH}(n, X, u)$,

where dCATCH stands for d -variate Caratheodory-TCHakaloff discrete measure compression. It works for any discrete measure on a discrete set X . Indeed, it could be used, other than for design compression, also in the compression of d -variate quadrature formulas, to give an example. The output parameter $pts = \{\xi_\ell\} \subset X$ is the array of support points of the compressed measure, while $w = \{w_\ell\} = \{v_{i_\ell} > 0\}$ is the corresponding positive weight array (that we may call a d -variate near G-optimal Tchakaloff design) and $momerr = \|U_n^T v - U_n^T u\|_2$ is the moment residual. This function is called LHDM.

In the present framework we call dCATCH with $n = 2m$ and $u = u(\bar{k})$, cf. (10), that is, we solve

$$v = \operatorname{argmin}_{z \in \mathbb{R}^M, z \geq 0} \|U_{2m}^T z - U_{2m}^T u(\bar{k})\|_2^2. \tag{12}$$

In such a way the compressed design generates the same scalar product of $u(\bar{k})$ in $\mathbb{P}_m^d(X)$, and hence the same orthogonal polynomials and the same Christoffel function on X keeping thus invariant the G-efficiency

$$\mathbb{P}_{2m}^d(X) \ni K_m^v(x, x) = K_m^{u(\bar{k})}(x, x) \quad \forall x \in X \implies G_m(v) = G_m(u(\bar{k})) \geq gtol \tag{13}$$

with a (much) smaller support.

From a deterministic regression viewpoint (approximation theory), let us denote by p_m^{opt} the polynomial in $\mathbb{P}_m^d(X)$ of best uniform approximation for f on X , where we assume $f \in C(D)$ with $X \subset D \subset \mathbb{R}^d$, D being a compact domain (or even lower-dimensional manifold), and by $E_m(f; X) = \inf_{p \in \mathbb{P}_m^d(X)} \|f - p\|_{\ell^\infty(X)} = \|f - p_m^{opt}\|_{\ell^\infty(X)}$ and $E_m(f; D) = \inf_{p \in \mathbb{P}_m^d(D)} \|f - p\|_{L^\infty(D)}$ the best uniform polynomial approximation errors on X and D .

Then, denoting by $\mathcal{L}_m^{u(\bar{k})}$ and $\mathcal{L}_m^w f = \mathcal{L}_m^v f$ the weighted least squares polynomial approximation of f (cf. (5)) by the near G-optimal weights $u(\bar{k})$ and w , respectively, with the same reasoning used to obtain (6) and by (13) we can write the operator norm estimates

$$\|\mathcal{L}_m^{u(\bar{k})}\|, \|\mathcal{L}_m^w\| \leq \sqrt{\tilde{N}_m} \leq \sqrt{\frac{N_m}{gtol}}, \quad \tilde{N}_m = \frac{N_m}{G_m(u(\bar{k}))} = \frac{N_m}{G_m(v)}.$$

Moreover, since $\mathcal{L}_m^w p = p$ for any $p \in \mathbb{P}_m^d(X)$, we can write the near optimal estimate

$$\begin{aligned} \|f - \mathcal{L}_m^w f\|_{\ell^\infty(X)} &\leq \|f - p_m^{opt}\|_{\ell^\infty(X)} + \|p_m^{opt} - \mathcal{L}_m^w p_m^{opt}\|_{\ell^\infty(X)} + \|\mathcal{L}_m^w p_m^{opt} - \mathcal{L}_m^w f\|_{\ell^\infty(X)} \\ &= \|f - p_m^{opt}\|_{\ell^\infty(X)} + \|\mathcal{L}_m^w p_m^{opt} - \mathcal{L}_m^w f\|_{\ell^\infty(X)} \leq (1 + \|\mathcal{L}_m^w\|) E_m(f; X) \\ &\leq \left(1 + \sqrt{\frac{N_m}{gtol}}\right) E_m(f; X) \leq \left(1 + \sqrt{\frac{N_m}{gtol}}\right) E_m(f; D) \approx \left(1 + \sqrt{N_m}\right) E_m(f; D). \end{aligned}$$

Notice that $\mathcal{L}_m^w f$ is constructed by sampling f only at the compressed support $\{\xi_\ell\} \subset X$. The error depends on the regularity of f on $D \supset X$, with a rate that can be estimated whenever D admits a multivariate Jackson-like inequality, cf. Reference [23].

Accelerating the Lawson-Hanson Algorithm by Deviation Maximization (LHDM)

Let $A \in \mathbb{R}^{N \times M}$ and $b \in \mathbb{R}^N$. The NNLS problem consists of seeking $x \in \mathbb{R}^M$ that solves

$$x = \operatorname{argmin}_{z \geq 0} \|Az - b\|_2^2. \tag{14}$$

This is a convex optimization problem with linear inequality constraints that define the *feasible region*, that is the positive orthant $\{x \in \mathbb{R}^M : x_i \geq 0\}$. The very first algorithm dedicated to problem (14) is due to Lawson and Hanson [24] and it is still one of the most often used. It was originally derived for solving overdetermined linear systems, with $N \gg M$. However, in the case of underdetermined linear systems, with $N \ll M$, this method succeeds in sparse recovery.

Recall that for a given point x in the feasible region, the index set $\{1, \dots, M\}$ can be partitioned into two sets: the active set Z , containing the indices of active constraints $x_i = 0$, and the passive set P , containing the remaining indices of inactive constraints $x_i > 0$. Observe that an optimal solution x^* of (14) satisfies $Ax^* = b$ and, if we denote by P^* and Z^* the corresponding passive and active sets respectively, x^* also solves in a least square sense the following unconstrained least squares subproblem

$$x_{P^*}^* = \operatorname{argmin}_y \|A_{P^*} y - b\|_2^2, \tag{15}$$

where A_{P^*} is the submatrix containing the columns of A with index in P^* , and similarly $x_{P^*}^*$ is the subvector made of the entries of x^* whose index is in P^* . The remaining entries of x^* , namely those whose index is in Z^* , are null.

The Lawson-Hanson algorithm, starting from a null initial guess $x = 0$ (which is feasible), incrementally builds an optimal solution by moving indices from the active set Z to the passive set P and vice versa, while keeping the iterates within the feasible region. More precisely, at each iteration first order information is used to detect a column of the matrix A such that the corresponding entry in the new solution vector will be strictly positive; the index of such a column is moved from the active set Z to the passive set P . Since there's no guarantee that the other entries corresponding to indices in the former passive set will stay positive, an inner loop ensures the new solution vector falls into the feasible region, by moving from the passive set P to the active set Z all those indices corresponding to violated constraints. At each iteration a new iterate is computed by solving a least squares problem of type (15): this can be done, for example, by computing a QR decomposition, which is substantially expensive. The algorithm terminates in a finite number of steps, since the possible combinations of passive/active set are finite and the sequence of objective function values is strictly decreasing, cf. Reference [24].

The *deviation maximization* (DM) technique is based on the idea of adding a whole set of indices T to the passive set at each outer iteration of the Lawson-Hanson algorithm. This corresponds to select a block of new columns to insert in the matrix A_P , while keeping the current solution vector within the feasible region in such a way that sparse recovery is possible when dealing with non-strictly convex problems. In this way, the number of total iterations and the resulting computational cost decrease. The set T is initialized to the index chosen by the standard Lawson-Hanson (LH) algorithm, and it is then extended, within the same iteration, using a set of candidate indices C chosen in such a way that the corresponding entries are likely positive in the new iterate. The elements of T are then chosen carefully within C : note that if the columns corresponding to the chosen indices are linearly dependent, the submatrix of the least squares problem (15) will be rank deficient, leading to numerical difficulties. We add k new indices, where k is an integer parameter to tune on the problem size, in such a way that, at the end, for every pair of indices in the set T , the corresponding column vectors form an angle whose cosine in absolute value is below a given threshold *thres*. The whole procedure is implemented in the function

- $[x, resnorm, exitflag] = \text{LHDM}(A, b, options)$.

The input variable *options* is a structure containing the user parameters for the LHDM algorithm; for example, the aforementioned k and *thres*. The output parameter x is the least squares solution, *resnorm* is the squared 2-norm of the residual and *exitflag* is set to 0 if the LHDM algorithm has reached the maximum number of iterations without converging and 1 otherwise.

In the literature, an accelerating technique was introduced by Van Benthem and Keenan [25], who presented a different NNLS solution algorithm, namely "fast combinatorial NNLS", designed for the specific case of a large number of right-hand sides. The authors exploited a clever reorganization of computations in order to take advantage of the combinatorial nature of the problems treated (multivariate curve resolution) and introduced a nontrivial initialization of the algorithm by means of unconstrained least squares solution. In the following section we are going to compare such an approach, briefly named LHI, and the standard LH algorithm with the LHDM procedure just summarized.

4. Numerical Examples

In this section, we perform several tests on the computation of d -variate near G-optimal Tchakaloff designs, from low to moderate dimension d . In practice, we are able to treat, on a personal computer, large-scale problems where $\text{card}(X) \times \dim(P_{2m}^d)$ is up to 10^8 – 10^9 , with $\dim(P_{2m}^d) = \binom{2m+d}{d} = \binom{2m+d}{2m}$.

Recall that the main memory requirement is given by the $N_{2m} \times M$ matrix U^T in the compression process solved by the LHDM algorithm, where $M = \text{card}(X)$ and $N_{2m} = \dim(P_{2m}^d(X)) \leq \dim(P_{2m}^d)$.

Given the dimension $d > 1$ and the polynomial degree m , the routine LHDM empirically sets the parameter k as follows $k = \lceil (2^{m+d}) / (m(d-1)) \rceil$, while the threshold is $\text{thres} = \cos(\frac{\pi}{2} - \theta)$, $\theta \approx 0.22$. All the tests are performed on a workstation with a 32 GB RAM and an Intel Core i7-8700 CPU @ 3.20 GHz.

4.1. Complex 3d Shapes

To show the flexibility of the package *dCATCH*, we compute near G-optimal designs on a “multibubble” $D \subset \mathbb{R}^3$ (i.e., the union of a finite number of non-disjoint balls), which can have a very complex shape with a boundary surface very difficult to describe analytically. Indeed, we are able to implement near optimal regression on quite complex solids, arising from finite union, intersection and set difference of simpler pieces, possibly multiply-connected, where for each piece we have available the indicator function via inequalities. Grid-points or low-discrepancy points, for example, Halton points, of a surrounding box, could be conveniently used to discretize the solid. Similarly, thanks to the adaptation of the method to the actual dimension of the polynomial spaces, we can treat near optimal regression on the surfaces of such complex solids, as soon as we are able to discretize the surface of each piece by point sets with good covering properties (for example, we could work on the surface of a multibubble by discretizing each sphere via one of the popular spherical point configurations, cf. Reference [26]).

We perform a test at regression degree $m = 10$ on the 5-bubble shown in Figure 1b. The initial support X consists in the $M = 18,915$ points within 64,000 low discrepancy Halton points, falling in the closure of the multibubble. Results are shown in Figure 1 and Table 3.

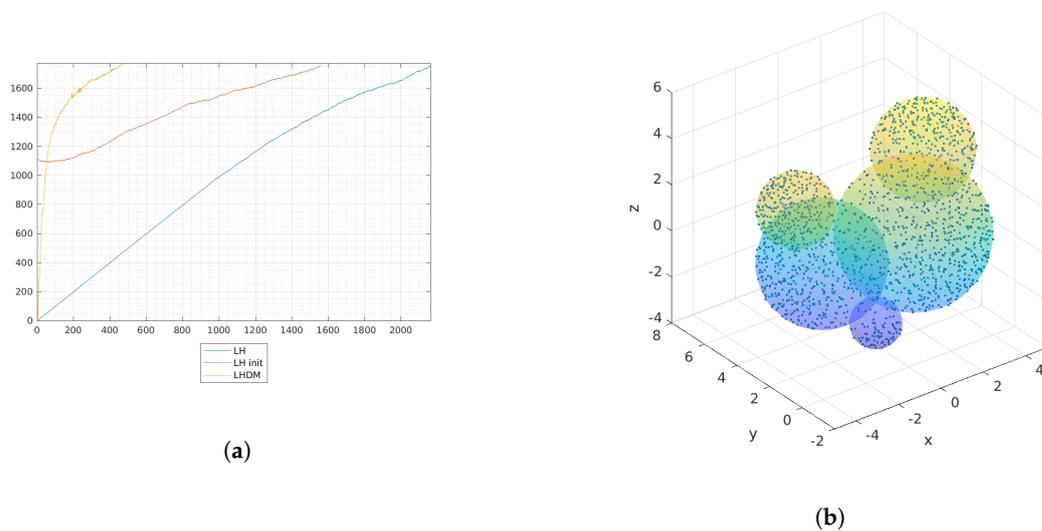


Figure 1. Multibubble test case, regression degree $m = 10$. (a) The evolution of the cardinality of the passive set P along the iterations of the three LH algorithms. (b) Multibubble with 1763 compressed Tchakaloff points, extracted from 18,915 original points.

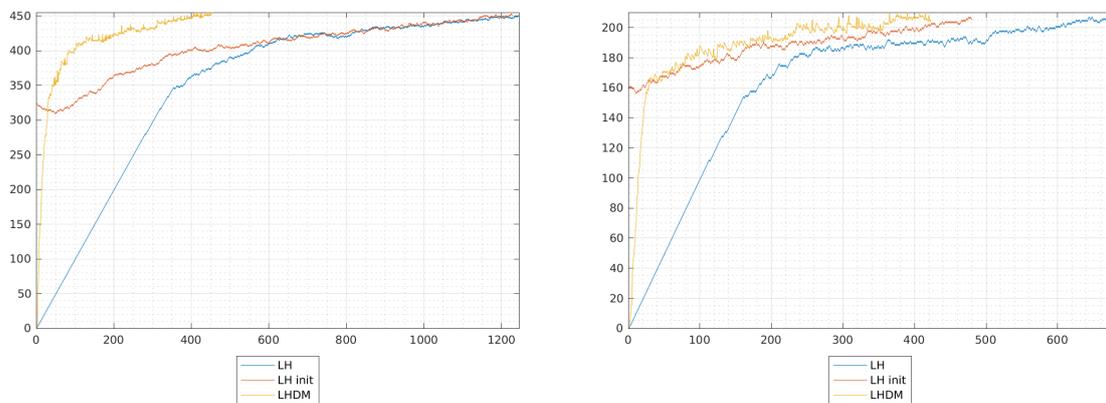
Table 3. Results for the multibubble numerical test: $\text{compr} = M / \text{mean}(\text{cpts})$ is the mean compression ratio obtained by the three methods listed; t_{LH} / t_{Tit} is the ratio between the execution time of LH and that of the Titterington algorithm; t_{LH} / t_{LHDM} (t_{LHI} / t_{LHDM}) is the ratio between the execution time of LH (LHI) and that of LHDM; cpts is the number of compressed Tchakaloff points and momerr is the final moment residual.

Test		LH					LHI		LHDM		
m	M	compr	t_{LH} / t_{Tit}	t_{LH} / t_{LHDM}	cpts	momerr	t_{LHI} / t_{LHDM}	cpts	momerr	cpts	momerr
10	18,915	11/1	40.0/1	2.7/1	1755	3.4×10^{-8}	3.2/1	1758	3.2×10^{-8}	1755	1.5×10^{-8}

4.2. Hypercubes: Chebyshev Grids

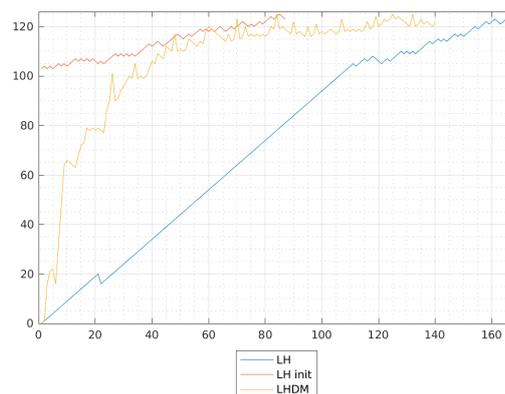
In a recent paper [19], a connection has been studied between the statistical notion of G-optimal design and the approximation theoretic notion of admissible mesh for multivariate polynomial approximation, deeply studied in the last decade after Reference [13] (see, e.g., References [27,28] with the references therein). In particular, it has been shown that near G-optimal designs on admissible meshes of suitable cardinality have a G-efficiency on the whole d -cube that can be made convergent to 1. For example, it has been proved by the notion of Dubiner distance and suitable multivariate polynomial inequalities, that a design with G-efficiency γ on a grid X of $(2km)^d$ Chebyshev points (the zeros of $T_{2km}(t) = \cos(2km \arccos(t))$, $t \in [-1, 1]$), is a design for $[-1, 1]^d$ with G-efficiency $\gamma(1 - \pi^2 / (8k^2))$. For example, taking $k = 3$ a near G-optimal Tchakaloff design with $\gamma = 0.99$ on a Chebyshev grid of $(6m)^d$ points is near G-optimal on $[-1, 1]^d$ with G-efficiency approximately $0.99 \cdot 0.86 \approx 0.85$, and taking $k = 4$ (i.e., a Chebyshev grid of $(8m)^d$ points) the corresponding G-optimal Tchakaloff design has G-efficiency approximately $0.99 \cdot 0.92 \approx 0.91$ on $[-1, 1]^d$ (in any dimension d).

We perform three tests in different dimension spaces and at different regression degrees. Results are shown in Figure 2 and Table 4, using the same notation above.



(a) $d = 3, n = 6, M = 110,592$.

(b) $d = 4, n = 3, M = 331,776$.



(c) $d = 5, n = 2, M = 1,048,576$.

Figure 2. The evolution of the cardinality of the passive set P along the iterations of the three LH algorithms for Chebyshev nodes' tests.

Table 4. Results of numerical tests on $M = (2km)^d$ Chebyshev’s nodes, with $k = 4$, with different dimensions and degrees: $compr = M/mean(cpts)$ is the mean compression ratio obtained by the three methods listed; t_{LH}/t_{Titt} is the ratio between the execution time of LH and that of Titterington algorithm; t_{LH}/t_{LHDM} (t_{LHI}/t_{LHDM}) is the ratio between the execution time of LH (LHI) and that of LHDM; $cpts$ is the number of compressed Tchakaloff points and $momerr$ is the final moment residual.

d	Test		compr	LH			LHI		LHDM			
	m	M		t_{LH}/t_{Titt}	t_{LH}/t_{LHDM}	cpts	momerr	t_{LHI}/t_{LHDM}	cpts	momerr		
3	6	110,592	250/1	0.4/1	3.1/1	450	5.0×10^{-7}	3.5/1	450	3.4×10^{-7}	450	1.4×10^{-7}
4	3	331,776	1607/1	0.2/1	2.0/1	207	8.9×10^{-7}	3.4/1	205	9.8×10^{-7}	207	7.9×10^{-7}
5	2	1,048,576	8571/1	0.1/1	1.4/1	122	6.3×10^{-7}	1.5/1	123	3.6×10^{-7}	122	3.3×10^{-7}

4.3. Hypercubes: Low-Discrepancy Points

The direct connection of Chebyshev grids with near G-optimal designs discussed in the previous subsection suffers rapidly of the curse of dimensionality, so only regression at low degree in relatively low dimension can be treated. On the other hand, in sampling theory a number of discretization nets with good space-filling properties on hypercubes has been proposed and they allow to increase the dimension d . We refer in particular to Latin hypercube sampling or low-discrepancy points (Sobol, Halton and other popular sequences); see for example, Reference [29]. These families of points give a discrete model of hypercubes that can be used in many different deterministic and statistical applications.

Here we consider a discretization made via Halton points. We present in particular two examples, where we take as finite design space X a set of $M = 10^5$ Halton points, in $d = 4$ with regression degree $m = 5$, and in $d = 10$ with $m = 2$. In both examples, $\dim(P_{2m}^d) = \binom{2m+d}{d} = \binom{2m+d}{2m} = \binom{14}{4} = 1001$, so that the largest matrix involved in the construction is the $1001 \times 100,000$ Chebyshev-Vandermonde matrix C for degree $2m$ on X constructed at the beginning of the compression process (by dORTHVAND within dCATCH to compute U_{2m} in (12)).

Results are shown in Figure 3 and Table 5, using the same notation as above.

Remark 1. The computational complexity of dCATCH mainly depends on the QR decompositions, which clearly limit the maximum size of the problem and mainly determine the execution time. Indeed, the computational complexity of a QR factorization of a matrix of size $n_r \times n_c$, with $n_c \leq n_r$, is high, namely $2(n_c^2 n_r - n_c^3/3) \approx 2n_c^2 n_r$ (see, e.g., Reference [30]).

Titterington algorithm performs a QR factorization of a $M \times N_m$ matrix at each iteration, with the following overall computational complexity

$$C_{Titt} \approx 2\bar{k} M N_m^2,$$

where \bar{k} is the number of iterations necessary for convergence, that depends on the desired G-efficiency.

On the other hand, the computational cost of one iteration of the Lawson-Hanson algorithm, fixed the passive set P , is given by the solution of an LS problem of type (15), which approximately is $2N_{2m}|P|^2$ that is the cost of a QR decomposition of a matrix of size $N_{2m} \times |P|$. However, as experimental results confirm, the evolution of the set P along the execution of the algorithm may vary significantly depending on the experiment settings, so that the exact overall complexity is hard to estimate. Lower and upper bounds are available, but may lead to heavy under- and over-estimations, respectively; cf. Reference [31] for a discussion on complexity issues.

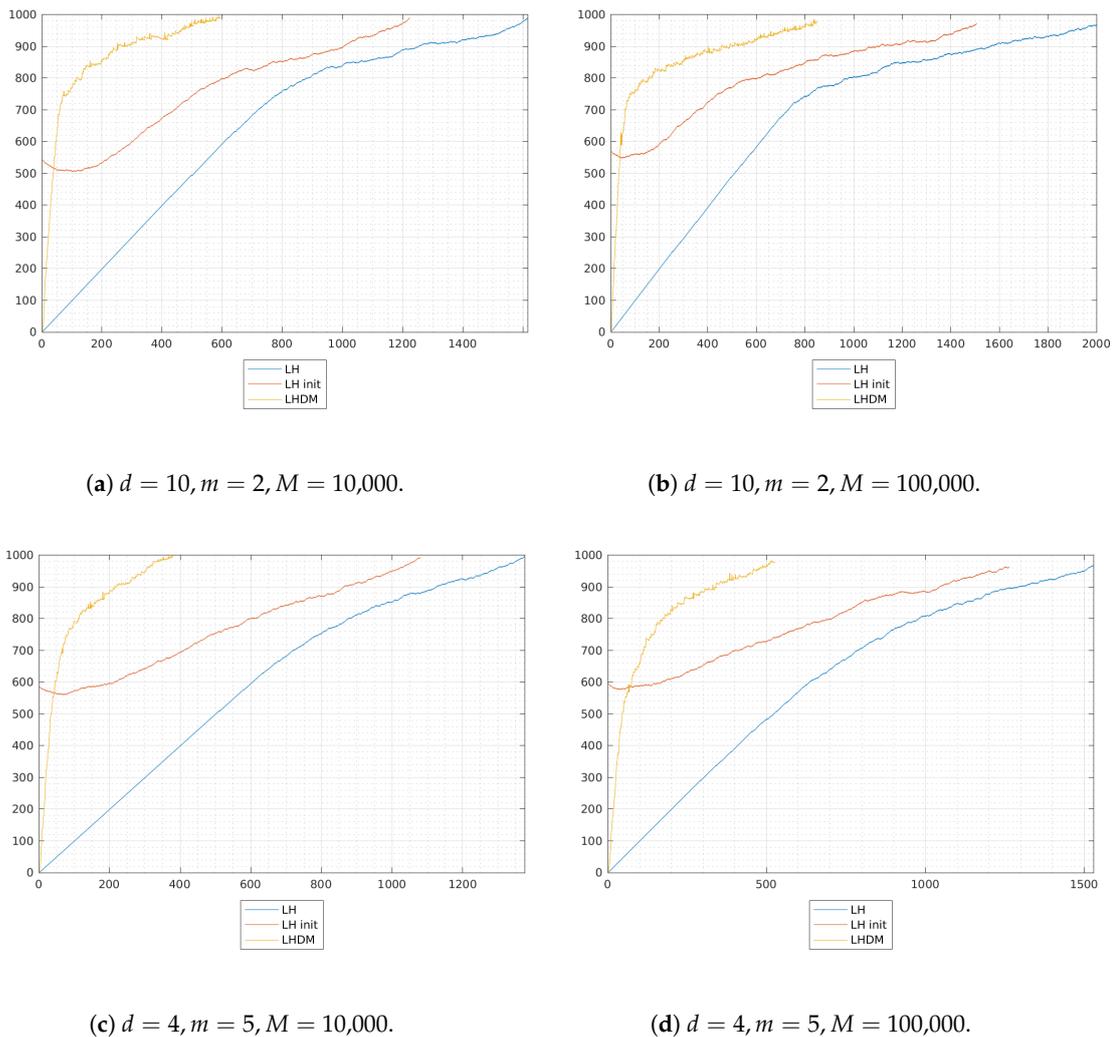


Figure 3. The evolution of the cardinality of the passive set P along the iterations of the three LH algorithms for Halton points’ tests.

Table 5. Results of numerical tests on Halton points: $compr = M/mean(cpts)$ is the mean compression ratio obtained by the three methods listed; t_{LH}/t_{TitH} is the ratio between the execution time of LH and that of Titterington algorithm; t_{LH}/t_{LHDM} (t_{LHI}/t_{LHDM}) is the ratio between the execution time of LH (LHI) and that of LHDM; $cpts$ is the number of compressed Tchakaloff points and $momerr$ is the final moment residual.

d	Test			LH				LHI		LHDM		
	m	M	$compr$	t_{LH}/t_{TitH}	t_{LH}/t_{LHDM}	$cpts$	$momerr$	t_{LHI}/t_{LHDM}	$cpts$	$momerr$	$cpts$	$momerr$
10	2	10,000	10/1	41.0/1	1.9/1	990	1.1×10^{-8}	1.9/1	988	9.8×10^{-9}	990	9.4×10^{-9}
10	2	100,000	103/1	6.0/1	3.1/1	968	3.6×10^{-7}	2.8/1	973	2.7×10^{-7}	968	4.2×10^{-7}
4	5	10,000	10/1	20.2/1	2.3/1	997	9.7×10^{-9}	2.4/1	993	1.3×10^{-8}	997	2.1×10^{-9}
4	5	100,000	103/1	2.0/1	3.8/1	969	6.6×10^{-7}	3.8/1	964	6.3×10^{-7}	969	5.3×10^{-7}

5. Conclusions

In this paper, we have presented *dCATCH* [1], a numerical software package for the computation of a d -variate near G-optimal polynomial regression design of degree m on a finite design space $X \subset \mathbb{R}^d$. The mathematical foundation is discussed connecting statistical design theoretic and approximation theoretic aspects, with a special emphasis on deterministic regression (Weighted Least Squares). The package takes advantage of an accelerated version of the classical NNLS Lawson-Hanson solver developed by the authors and applied to design compression.

As a few examples of use cases of this package we have shown the results on a complex shape (multibubble) in three dimensions, and on hypercubes discretized with Chebyshev grids and with Halton points, testing different combinations of dimensions and degrees which generate large-scale problems for a personal computer.

The present package, *dCATCH* works for any discrete measure on a discrete set X . Indeed, it could be used, other than for design compression, also in the compression of d -variate quadrature formulas, even on lower-dimensional manifolds, to give an example.

We may observe that with this approach we can compute a d -variate compressed design starting from a high-cardinality sampling set X , that discretizes a continuous compact set (see Sections 4.2 and 4.3). This design allows an m -th degree near optimal polynomial regression of a function on the whole X , by sampling on a small design support. We stress that the compressed design is function-independent and thus can be constructed “once and for all” in a pre-processing stage. This approach is potentially useful, for example, for the solution of d -variate parameter estimation problems, where we may think to model a nonlinear cost function by near optimal polynomial regression on a discrete d -variate parameter space X ; cf., for example, References [32,33] for instances of parameter estimation problems from mechatronics applications (*Digital Twins* of controlled systems) and references on the subject. Minimization of the polynomial model could then be accomplished by popular methods developed in the growing research field of Polynomial Optimization, such as Lasserre’s SOS (Sum of Squares) and measure-based hierarchies, and other recent methods; cf., for example, References [34–36] with the references therein.

From a computational viewpoint, the results shown in Tables 3–5 show relevant speed-ups in the compression stage, with respect to the standard Lawson-Hanson algorithm, in terms of the number of iterations required and of computing time within the Matlab scripting language. In order to further decrease the execution times and to allow us to tackle larger design problems, we would like in the near future to enrich the package *dCATCH* with an efficient C implementation of its algorithms and, possibly, a CUDA acceleration on GPUs.

Author Contributions: Investigation, M.D., F.M. and M.V. All authors have read and agreed to the published version of the manuscript.

Funding: Work partially supported by the DOR funds and the biennial project Project BIRD192932 of the University of Padova, and by the GNCS-INdAM. This research has been accomplished within the RITA “Research Italian network on Approximation”.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dessole, M.; Marcuzzi, F.; Vianello, M. *dCATCH: A Numerical Package for Compressed d-Variate Near G-Optimal Regression*. Available online: <https://www.math.unipd.it/~marcov/MVsoft.html> (accessed on 1 June 2020).
2. Bos, L.; Vianello, M. CaTchDes: MATLAB codes for Caratheodory—Tchakaloff Near-Optimal Regression Designs. *SoftwareX* **2019**, *10*, 100349. [CrossRef]
3. Dessole, M.; Marcuzzi, F.; Vianello, M. Accelerating the Lawson-Hanson NNLS solver for large-scale Tchakaloff regression designs. *Dolomit. Res. Notes Approx. DRNA* **2020**, *13*, 20–29.
4. Atkinson, A.; Donev, A.; Tobias, R. *Optimum Experimental Designs, with SAS*; Oxford University Press: Oxford, UK, 2007.
5. Pukelsheim, F. *Optimal Design of Experiments*; SIAM: Philadelphia, PA, USA, 2006.
6. Celant, G.; Broniatowski, M. *Interpolation and Extrapolation Optimal Designs 2-Finite Dimensional General Models*; Wiley: Hoboken, NJ, USA, 2017.
7. Mandal, A.; Wong, W.K.; Yu, Y. Algorithmic searches for optimal designs. In *Handbook of Design and Analysis of Experiments*; CRC Press: Boca Raton, FL, USA, 2015; pp. 755–783.

8. De Castro, Y.; Gamboa, F.; Henrion, D.; Hess, R.; Lasserre, J.B. Approximate optimal designs for multivariate polynomial regression. *Ann. Stat.* **2019**, *47*, 127–155. [[CrossRef](#)]
9. Dette, H.; Pepelyshev, A.; Zhigljavsky, A. Improving updating rules in multiplicative algorithms for computing D-optimal designs. *Comput. Stat. Data Anal.* **2008**, *53*, 312–320. [[CrossRef](#)]
10. Torsney, B.; Martin-Martin, R. Multiplicative algorithms for computing optimum designs. *J. Stat. Plan. Infer.* **2009**, *139*, 3947–3961. [[CrossRef](#)]
11. Bloom, T.; Bos, L.; Levenberg, N.; Waldron, S. On the Convergence of Optimal Measures. *Constr. Approx.* **2008**, *32*, 159–169. [[CrossRef](#)]
12. Bos, L. Some remarks on the Fejér problem for lagrange interpolation in several variables. *J. Approx. Theory* **1990**, *60*, 133–140. [[CrossRef](#)]
13. Calvi, J.P.; Levenberg, N. Uniform approximation by discrete least squares polynomials. *J. Approx. Theory* **2008**, *152*, 82–100. [[CrossRef](#)]
14. Kiefer, J.; Wolfowitz, J. The equivalence of two extremum problems. *Can. J. Math.* **1960**, *12*, 363–366. [[CrossRef](#)]
15. Putinar, M. A note on Tchakaloff’s theorem. *Proc. Am. Math. Soc.* **1997**, *125*, 2409–2414. [[CrossRef](#)]
16. Titterington, D. Algorithms for computing D-optimal designs on a finite design space. In *Proceedings of the 1976 Conference on Information Science and Systems*; John Hopkins University: Baltimore, MD, USA, 1976; Volume 3, pp. 213–216.
17. Burkardt, J. MONOMIAL: A Matlab Library for Multivariate Monomials. Available online: https://people.sc.fsu.edu/~jburkardt/m_src/monomial/monomial.html (accessed on 1 June 2020).
18. Bos, L.; Piazzon, F.; Vianello, M. Near optimal polynomial regression on norming meshes. In *Sampling Theory and Applications 2019*; IEEE Xplore Digital Library: New York, NY, USA, 2019.
19. Bos, L.; Piazzon, F.; Vianello, M. Near G-optimal Tchakaloff designs. *Comput. Stat.* **2020**, *35*, 803–819. [[CrossRef](#)]
20. Tchakaloff, V. Formules de cubatures mécaniques à coefficients non négatifs. *Bull. Sci. Math.* **1957**, *81*, 123–134.
21. Carathéodory, C. Über den Variabilitätsbereich der Fourier’schen Konstanten von positiven harmonischen Funktionen. *Rendiconti Del Circolo Matematico di Palermo (1884–1940)* **1911**, *32*, 193–217. [[CrossRef](#)]
22. Sommariva, A.; Vianello, M. Compression of Multivariate Discrete Measures and Applications. *Numer. Funct. Anal. Optim.* **2015**, *36*, 1198–1223. [[CrossRef](#)]
23. Pleśniak, W. Multivariate Jackson Inequality. *J. Comput. Appl. Math.* **2009**, *233*, 815–820. [[CrossRef](#)]
24. Lawson, C.L.; Hanson, R.J. *Solving Least Squares Problems*; SIAM: Philadelphia, PA, USA, 1995; Volume 15.
25. Van Benthem, M.H.; Keenan, M.R. Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. *J. Chemom.* **2004**, *18*, 441–450. [[CrossRef](#)]
26. Hardin, D.; Michaels, T.; Saff, E. A Comparison of Popular Point Configurations on S^2 . *Dolomit. Res. Notes Approx. DRNA* **2016**, *9*, 16–49.
27. Bloom, T.; Bos, L.; Calvi, J.; Levenberg, N. Polynomial Interpolation and Approximation in \mathbb{C}^d . *Ann. Polon. Math.* **2012**, *106*, 53–81. [[CrossRef](#)]
28. De Marchi, S.; Piazzon, F.; Sommariva, A.; Vianello, M. Polynomial Meshes: Computation and Approximation. In *Proceedings of the CMMSE 2015, Rota Cadiz, Spain, 6–10 July 2015*; pp. 414–425.
29. Dick, J.; Pillichshammer, F. *Digital Nets and Sequences-Discrepancy Theory and Quasi—Monte Carlo Integration*; Cambridge University Press: Cambridge, UK, 2010.
30. Golub, G.H.; Van Loan, C.F. *Matrix Computations*, 3rd ed.; Johns Hopkins University Press: Baltimore, MD, USA, 1996.
31. Slawski, M. Nonnegative Least Squares: Comparison of Algorithms. Available online: <https://sites.google.com/site/slawskimartin/code> (accessed on 1 June 2020).
32. Beghi, A.; Marcuzzi, F.; Martin, P.; Tinazzi, F.; Zigliotto, M. Virtual prototyping of embedded control software in mechatronic systems: A case study. *Mechatronics* **2017**, *43*, 99–111. [[CrossRef](#)]
33. Beghi, A.; Marcuzzi, F.; Rampazzo, M. A Virtual Laboratory for the Prototyping of Cyber-Physical Systems. *IFAC-PapersOnLine* **2016**, *49*, 63–68. [[CrossRef](#)]
34. Lasserre, J.B. The moment-SOS hierarchy. *Proc. Int. Cong. Math.* **2018**, *4*, 3791–3814.

35. De Klerk, E.; Laurent, M. A survey of semidefinite programming approaches to the generalized problem of moments and their error analysis. In *World Women in Mathematics 2018-Association for Women in Mathematics Series*; Springer: Cham, Switzerland, 2019; Volume 20, pp. 17–56.
36. Martinez, A.; Piazzon, F.; Sommariva, A.; Vianello, M. Quadrature-based polynomial optimization. *Optim. Lett.* **2020**, *35*, 803–819. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).