Semidefinite Programming for Gradient and Hessian Computation in Maximum Entropy Estimation

Jean B. Lasserre

***Abstract*— We consider the classical problem of estimating a density on $[0,1]$ via some maximum entropy criterion. For solving this convex optimization problem with algorithms using first-order or second-order methods, at each iteration one has to compute (or at least approximate) moments of some measure with a density on $[0,1]$, to obtain gradient and Hessian data. We propose a numerical scheme based on semidefinite programming that avoids computing quadrature formula for this gradient and Hessian computation.**

I. INTRODUCTION

We consider the maximum entropy estimation problem. Namely, let $f \in L_1([0,1])$ be a nonnegative function only known via the first $2n+1$ moments of its associated measure $d\mu = f dx$ on $[0,1]$. From that partial knowledge one wishes (a) to provide an estimate f_n of f such that the first $2n+1$ moments of $f_n dx$ match those of $f dx$, and (b) to analyze the asymptotic behavior of f_n as $n \to \infty$. This problem has important applications in various areas of physics, engineering, and signal processing in particular; see e.g. the discussions in [2], [3], [4], [7], [11], [12] and [13].

An elegant and appealing methodology is to search for f_n in a parametrized family $\{f_n(\lambda,x)\}_\lambda$ of functions: $[0,1] \to \mathbb{R}_+$, and optimize over the unknown parameters λ of a finite dimensional space, via a suitable criterion. For instance, one may wish to select an estimate f_n that maximizes some appropriate *entropy*. If one chooses the Boltzmann-Shannon entropy, then the optimal estimate f_n^* is shown to be the exponential of some polynomial of degree $2n$, whose coefficient vector $\lambda \in \mathbb{R}^{2n+1}$ is then an optimal solution of a finite-dimensional *convex* problem; other choices of entropy functional are possible as long as one obtains a convex problem in finitely many coefficients λ_i's. For more details on this approach the interested reader is referred to e.g. Byrnes and Lindquist [5], Borwein and Lewis [2], [3], [4], Georgeou [7], Mead and Papanicolaou [11], and Tagliani [12], [13]. As early as in [11], it was recognized that such entropy methods may outperform classical Padé-like approximations.

Except for the homotopy approach developed in Georgiou [7], optimization algorithms using first or second-order methods to search for λ, need evaluate the gradient or the gradient and Hessian of the function $g_n : \mathbb{R}^{2n+1} \to \mathbb{R}$ defined by:

$$\lambda \mapsto g_n(\lambda) := \int_0^1 f_n(\lambda,x)\,dx \tag{1}$$

at the current iterate λ. In the present context it reduces to evaluating the first $2n$ moments of the measure $f_n(\lambda,x)dx$ on $[0,1]$. This can be done via several integration techniques and in particular, via quadrature formula.

A typical example of this approach is the Newton method described in Mead and Papanicolaou [11] which uses quadrature formula (or alternatively, Newton-Cotes integration) for gradient and Hessian evaluation. In principle, the quadrature formula should be with respect to the weight function $f_n(\lambda,\bullet)$ on $[0,1]$ and so, one has to repeatedly compute such a quadrature formula at *each* current iterate λ of the algorithm. In particular, it requires to repeatedly compute orthogonal polynomials with respect to the measure $f_n(\lambda,x)dx$ on $[0,1]$, not an easy task in general. This is why in [11] the authors rather use some Gaussian quadrature formula with respect to the *constant* weight function 1 on $[0,1]$ and incorporate $f_n(\lambda,\bullet)$ in the integrand to evaluate. Hence, in doing so, they estimate both gradient and Hessian with some unknown error. However, in [11] they obtain very good numerical results because Gaussian quadrature formula seem to perform well for integrating exponentials of polynomials.

Contribution. In this paper we provide a numerical scheme to evaluate (or rather *approximate*) both gradient and Hessian of the function g_n in (1), at each current iterate λ of a maximizing entropy algorithm. We do not use quadrature formula and thus avoid computing orthogonal polynomials associated with the measure $d\mu = f_n(\lambda;x)dx$ on $[0,1]$. We instead use the approach defined in [1] for approximating integrals of exponentials of multivariate polynomials. It allows us to compute an estimate of the moments of the measure $f_n(\lambda,\cdot)dx$ by solving *two* single semidefinite programs (SDP) $\mathbf{U}_r$ and $\mathbf{L}_r$, each with $2n$ variables and two Linear Matrix Inequalities involving Hankel matrices of size r. This produces a monotone sequence of upper and lower bounds ($\mathbf{u}_r$ and $\mathbf{l}_r$ respectively) both converging to the same value as $r \to \infty$, and in addition, both optimal solutions of $\mathbf{U}_r$ and $\mathbf{L}_r$ also converge to the vector of moments of the measure $f_n(\lambda,\cdot)dx$ on $[0,1]$, at the current iterate λ. The desired accuracy is controlled by the size r of the Hankel matrices.

In addition, still following ideas of [1], the methodology easily extends to the multivariate case and also on compact domains more complicated than just boxes of $\mathbb{R}^p$, but of course at a higher computational cost. In particular, and in contrast to the univariate case, the number of variables is not fixed anymore (i.e. now depends on r) and moreover,

This work was supported the (French) ANR

Jean B. Lasserre is with LAAS-CNRS and the Institute of Mathematics, University of Toulouse, France. lasserre@laas.fr

the size of the moment and localizing matrices in $\mathbf{U}_r$ and $\mathbf{L}_r$ increases fast with the dimension p, typically like $O(r^p)$.

Therefore, especially in view of the present status of SDP solvers still in their infancy, applicability of this methodology is still limited to multivariate problems with dimension $p \leq 3$ and with a reasonable number of parameters λ_j.

II. Problem statement

A. Background

Consider the problem of estimating an unknown density $f : [0,1] \to \mathbb{R}_+$, based only on the knowledge of its first $2n+1$ *moments*, $m = (m_0, \dots, m_{2n})$ only. That is,

$$m_j = \int_0^1 x^j f(x)\,dx, \qquad j = 0, \dots, 2n, \tag{2}$$

where in general $m_0 = 1$ (as f is the density of some probability measure μ on $[0,1]$). The approach we follow is to compute an estimate f_n that maximizes some appropriate *entropy*. Namely, we consider the usual Boltzmann-Shannon entropy $\mathbf{H} : L_1([0,1]) \to [-\infty, +\infty)$:

$$g \mapsto \mathbf{H}[g] := -\int_0^1 g(x) \ln g(x)\,dx, \tag{3}$$

a strictly concave functional. Therefore the problem reduces to solving the concave maximization problem:

$$\max_g \left\{ \mathbf{H}[g] : \int_0^1 x^j g(x)\,dx = m_j, \quad j = 0, \dots, 2n \right\}. \tag{4}$$

The structure of this infinite-dimensional convex optimization problem permits to search for an optimal solution f_n^* of the form

$$f_n^*(x) = \exp \sum_{j=0}^{2n} \lambda_j^* x^j, \tag{5}$$

where $\lambda^* \in \mathbb{R}^{2n+1}$ is unique when g_n^* has to satisfy the constraints of (4). This is because the Legendre-Fenchel conjugate of

$$x \mapsto \psi(x) := \begin{cases} x \ln x & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ +\infty & \text{if } x < 0 \end{cases}$$

is $y \mapsto \psi^*(y) = \exp(y-1)$. And so, the conjugate functional $\mathbf{H}^*$ of $-\mathbf{H}$ is

$$g \mapsto \mathbf{H}^*(g) = \int_0^1 \exp(g(x) - 1)dx.$$

For a detailed discussion, see e.g. Borwein and Lewis [2, p. 257-258]. In particular $g_n^* \to f$ weakly in $L_1([0,1])$ as $n \to \infty$.

B. A finite-dimensional convex problem

So, in view of what precedes, with n fixed and given $\lambda \in \mathbb{R}^{2n+1}$, define $f_n(\lambda, \cdot) : \mathbb{R}_+ \to \mathbb{R}$ to be the function:

$$x \mapsto f_n(\lambda, x) := \exp \sum_{j=0}^{2n} \lambda_j x^j, \qquad x \in \mathbb{R}. \tag{6}$$

Then Problem (4) reduces to solving the *finite-dimensional* optimization problem:

$$\mathbf{P} : \quad \max_{\lambda \in \mathbb{R}^{2n+1}} \left\{ \langle m, \lambda \rangle - \int_0^1 f_n(\lambda, x)\,dx \right\}. \tag{7}$$

Notice that solving $\mathbf{P}$ is just evaluating $g_n^*(m)$, where g_n^* is the Legendre-Fenchel transform of the function $g_n : \mathbb{R}^{2n+1} \to \mathbb{R}$,

$$\lambda \mapsto g_n(\lambda) := \int_0^1 f_n(\lambda, x)\,dx, \tag{8}$$

i.e., the mass of the measure $d\mu_n := f_n(\lambda, x)dx$ on $[0,1]$.

Lemma 1: Let f_n, g_n be defined as in (6) and (8) respectively, and let

$$\gamma_k^{(n)} := \int_0^1 x^k f_n(\lambda, x)\,dx, \quad k = 0, 1, \dots \tag{9}$$

Then g_n is convex with

$$\frac{\partial g_n(\lambda)}{\partial \lambda_j} = \int_0^1 x^j f_n(\lambda, x)\,dx = \gamma_j^{(n)} \tag{10}$$

for all $j = 0, \dots, 2n$, and

$$\frac{\partial^2 g_n(\lambda)}{\partial \lambda_i \partial \lambda_j} = \int_0^1 x^{i+j} f_n(\lambda, x)\,dx = \gamma_{i+j}^{(n)} \tag{11}$$

for all $i, j = 0, \dots, 2n$.

Notice that computing the first $2n+1$ moments of the measure $d\mu = f_n dx$ yields the gradient ∇g_n, which in turn permits to implement a first-order minimization algorithm. Computing an additional $2n$ moments provides us with the Hessian $\{\partial^2 g_n / \partial \lambda_i \partial \lambda_j\}$ as well, which in turn permits to implement second-order methods (e.g. Newton's method) for minimization.

In fact, we only need compute the first $2n$ moments as every other moment can be expressed as a linear combination of $\gamma_j^{(n)}$, $j = 0, \dots, 2n-1$, at no additional computational cost.

Lemma 2: Let g_n and $\gamma^{(n)} = (\gamma_k^{(n)})$ be as in (8) and (9), and assume that $\lambda_{2n} \neq 0$. Then, for all $k \geq 0$

$$\gamma_{2n+k}^{(n)} = s_k + \langle \psi_k, (\gamma_0^{(n)}, \gamma_1^{(n)}, \dots, \gamma_{2n-1}^{(n)}) \rangle \tag{12}$$

for some $(s_k, \psi_k) \in \mathbb{R} \times \mathbb{R}^{2n}$, easily obtained by an iterative process.

Proof: The proof is by induction, which also gives a simple recipe to obtain (s_k, ψ_k). We start with the case $\gamma_{2n}^{(n)}$. From the definition of g_n, integration by parts yields

$$\begin{aligned} \gamma_0^{(n)} &= f_n(\lambda, 1) - \sum_{j=1}^{2n} j\lambda_j \int_0^1 x^j f_n(\lambda, x)\,dx \\ &= s_0 - \sum_{j=1}^{2n} j\lambda_j \gamma_j^{(n)}, \end{aligned} \tag{13}$$

and so, with $s_0 = f_n(\lambda, 1)$ (and as $\lambda_{2n} \neq 0$),

$$\gamma_{2n}^{(n)} = s_0 + \langle \psi_0, (\gamma_0^{(n)}, \dots, \gamma_{2n-1}^{(n)}) \rangle,$$

with $2n\lambda_n \psi_0(0) := -1$ and $2n\lambda_n \psi_0(j) := -j\lambda_j$, for all $j = 1, \dots, 2n-1$. Next, assume that (12) holds until $k > 1$. Then, again:

$$(k+2)\gamma_{k+1}^{(n)} = \int_0^1 (k+2) x^{k+1} f_n(\lambda, x)\,dx$$

$$\begin{aligned} &= f_n(\lambda,1) - \sum_{j=1}^{2n} j\lambda_j \int_0^1 x^{j+k+1} f_n(\lambda,x)\,dx \\ &= f_n(\lambda,1) - \sum_{j=1}^{2n} j\lambda_j \gamma^{(n)}_{j+k+1}, \end{aligned}$$

and so, $2n\lambda_{2n}\gamma^{(n)}_{2n+k+1} = f_n(\lambda_1) - (k+2)\gamma^{(n)}_{k+1} - \sum_{j=1}^{2n-1} j\lambda_j\gamma^{(n)}_{j+k+1}$. Next, use the induction hypothesis for $\gamma^{(n)}_{2n+j}$, $j = 0,\ldots,k$ to get the desired result. ■

III. Computing the maximum entropy estimator

To solve the convex optimization problem $\mathbf{P}$ defined in (7), we can use a first or second-order method. For instance one may wish to implement Newton's method which in view of (10)-(11), yields the iterates

$$\lambda^{(k+1)} = \lambda^{(k)} + H_k^{-1}\widehat{\gamma}_k \qquad (14)$$

where $\widehat{\gamma}_k(j) = m_j - \gamma^{(n)}_j$, and $H_k(i,j) = \gamma^{(n)}_{i+j-1}$, for all $i,j = 0,\ldots 2n$.

In any case, one has to compute $2n+1$ or $4n+1$ moments. In fact, from Lemma 2, all moments can be obtained from the first $2n-1$ moments only, via (12). Hence only a good approximation of the first $2n-1$ moments is required.

A first possibility for evaluating (9) is to use some quadrature formula with respect to the weight function $x \mapsto f_n(\lambda,x)$ on $[0,1]$. But this requires computing *orthogonal* polynomials w.r.t. to $f_n(\lambda,\bullet)$ on $[0,1]$, a highly nontrivial task. Recall that $\{p_k\}_{k=0,1,\ldots} \subset \mathbb{R}[x]$ is a family of orthogonal polynomials w.r.t. a positive weight function on $[0,1]$ if each p_k is of degree k, and

$$\langle p_j, p_k\rangle := \int_0^1 p_j(x)p_k(x)\,w(x)dx \left\{ \begin{array}{ll} = 0 & \text{if } j \neq k \\ > 0 & \text{otherwise} \end{array}\right. ,$$

for all $j,k = 0,1,\ldots$; see e.g. [6].

Another possibility is to approximate (9) by integrating *directly* the function $x \mapsto x^k f_n(\lambda,x)$ via some quadrature formula with respect to the constant weight function 1 on $[0,1]$. In this case, well-known quadrature formula are available but then one produces some error with the exact value. This is the successful approach taken in [11] with a double precision 24-point Gaussian quadrature formula. Indeed numerical experiments suggest that Gaussian quadratures seem to perform rather well for integrating exponentials of polynomials. Newton-Cotes integration is also a viable alternative proposed in [11] with good performances.

In the sequel, we provide a numerical approximation scheme for computing $\gamma^{(n)}_k$ in (9), $k = 0,\ldots,2n-1$, by using two sequences of semidefinite programs $\mathbf{U}_r$ and $\mathbf{L}_r$ whose size is indexed by $r \in \mathbb{N}$. Both $\mathbf{U}_r$ and $\mathbf{L}_r$ have the same fixed number $2n$ of variables and as $r \to \infty$, their respective optimal values $\mathbf{u}_r$ and $\mathbf{l}_r$ both converge monotonically to the same value. Moreover, their optimal solutions also converge to the desired vector of the first $2n$ moments of the measure $d\mu = f_n(\lambda,x)dx$ on $[0,1]$.

A. *Computing moments of $d\mu = f_n(\lambda,x)\,dx$*

Let $r \geq n$, $\lambda \in \mathbb{R}^{2n+1}$ be fixed and with no loss of generality assume that $\lambda_{2n} \neq 0$. Let $y = (y_0,\ldots,y_{2n-1}) \in \mathbb{R}^{2n}$, and define $\tilde{y} \in \mathbb{R}^{2r+1}$ by

$$\tilde{y}_j := \left\{ \begin{array}{lll} y_j & 0 \leq j \leq 2n-1 & \\ s_k + \langle \psi_k, y\rangle & j = 2n+k, & k = 0,\ldots,2(r-n), \end{array}\right. \qquad (15)$$

where (s_k,ψ_k) is defined in (12). Next, consider the $(r+1)\times(r+1)$ Hankel matrix $H_r(y) \in \mathscr{S}_{r+1}$ defined by:

$$H_r(y)(i,j) := \tilde{y}_{i+j-2}, \qquad \forall 1 \leq i,j \leq r+1, \qquad (16)$$

and the $r\times r$ Hankel matrix $\Delta_r(y) \in \mathscr{S}_r$ defined by:

$$\Delta_r(y)(i,j) := \tilde{y}_{i+j-1} - \tilde{y}_{i+j}, \qquad \forall 1 \leq i,j \leq r. \qquad (17)$$

$H_r(y)$ is the *moment* matrix associated with y, whereas $\Delta_r(y)$ is the *localizing* matrix associated with y and the polynomial $t(1-t)$. The conditions $H_r(y), \Delta_r(y) \succeq 0$ are necessary for y to be the $(2r+1)$-moment vector of some measure ν on $[0,1]$; see e.g. Lasserre [9].

Define $\mathbf{L}_r$ and $\mathbf{U}_r$ to be the semidefinite programs

$$\mathbf{L}_r: \quad \inf_y \{\, \langle m, \tilde{y}\rangle \,:\, H_r(y) \succeq 0; \quad \Delta_r(y) \succeq 0 \,\} \qquad (18)$$

$$\mathbf{U}_r: \quad \sup_y \{\, \langle m, \tilde{y}\rangle \,:\, H_r(y) \succeq 0; \quad \Delta_r(y) \succeq 0 \,\} \qquad (19)$$

with $m \in \mathbb{R}^{2n+1}$ as in (2), and denote by $\mathbf{l}_r$ and $\mathbf{u}_r$ their respective optimal value.

Theorem 3: Let $\mathbf{L}_r$ and $\mathbf{U}_r$ be as in (18)-(19). Then:

(a) Both $\mathbf{L}_r$ and $\mathbf{U}_r$ are solvable, and $\mathbf{u}_r - \mathbf{l}_r \downarrow 0$ as $r \to \infty$.

(b) In both cases, denote by y^r an optimal solution of $\mathbf{L}_r$ or $\mathbf{U}_r$. For every fixed j, $\tilde{y}^r_j \to \gamma^{(n)}_j$ as $r \to \infty$.

Theorem 3 is a particular case of a more general result in [1] where integrals of exponentials of multivariate polynomials on special bounded domains are considered. In fact the criterion $\langle m, \tilde{y}\rangle$ of $\mathbf{U}_r$ and $\mathbf{L}_r$ can be chosen arbitrarily and other choices are possible.

Therefore, one may obtain a good approximation of the first $2n$ moments, as close as desired, the gap $\mathbf{u}_r - \mathbf{l}_r$ controlling the error. Hence, we are now ready to implement Newton's method (14) to solve the convex optimization $\mathbf{P}$. Below we display results for three simple cases, with 2 and 4 moments.

Example 1: Consider the case $f \equiv 1$ and $n = 1$, i.e., approximating the uniform distribution from knowledge of only its first three moments $m = (1, 1/2, 1/3)$. For practical purposes we only solve the SDP $\mathbf{L}_r$ with $r = 5$, i.e., with at most 10 moments. Results are displayed in Table I where the last row displays the exact moments of $f dx$. Notice that in spite of an initial state far from the optimal solution, one obtains very good results after 10 iterations only. The gradient is like $O(10^{-4})$ and the first seven moments of $f_n dx$ read $(1.0, 0.5, 0.3333, 0.25, 0.2, 0.1667, 0.1429)$, an excellent approximation of the exact ones. The optimal solution is just $f_n^* \approx \exp(0) = 1 = f$.

TABLE I

EXAMPLE 1 : $f \equiv 1$ AND $n = 1$

k	moments			coefficients $\lambda^{(k)}$		
0	111.48	98.243	88.691	1.0000	1.0000	5.0000
1	41.401	36.330	32.751	0.3805	0.1521	5.4722
2	15.653	13.561	12.176	0.1772	-1.5642	6.3995
3	6.2276	5.1957	4.6113	0.3704	-3.8238	7.4644
4	2.7965	2.1360	1.8397	0.5192	-4.9419	7.3895
5	1.5828	1.0417	0.8397	0.4329	-3.9514	5.3942
6	1.1775	0.6633	0.4859	0.2191	-1.8705	2.5216
7	1.0379	0.5339	0.3645	0.0545	-0.4478	0.6132
8	1.0027	0.5024	0.3355	0.0038	-0.0308	0.0429
9	1.0000	0.5000	0.3333	0.0000	-0.0002	0.0003
$\star$	1.0000	0.5000	0.3333	0.0000	0.0000	0.0000

Example 2: Consider now the case $f \equiv x$ and $n = 1$. With $r = 5$, i.e., with at most 10 moments, results are displayed in Table II. Again, in spite of an initial state far from the optimal solution, one obtains very good results after 10 iterations only. The gradient is like $O(10^{-4})$ and the first seven moments of $f_n dx$ read $(0.5, 0.3333, 0.25, 0.1994, 0.1654, 0.1411)$. Figure 1 plots the curves of $f \equiv x$ and f_n on $[0,1]$. One can see that they agree fairly well despite we only use 3 moments.

TABLE II

EXAMPLE 2 : $f \equiv x$ AND $n = 1$

k	moments			coefficients $\lambda^{(k)}$		
0	111.5	98.24	88.69	1.0000	1.000	5.000
1	41.19	36.26	32.72	0.059	0.915	5.024
2	15.34	13.46	12.13	-0.7856	0.7049	5.0781
3	5.838	5.081	4.557	-1.4209	0.2524	5.1603
4	2.357	2.004	1.776	-1.7230	-0.3590	5.0577
5	1.103	.8873	.7630	-1.8393	-0.1437	3.9236
6	.6769	.4984	.4050	-2.1274	1.9119	1.1588
7	.5411	.3710	.2852	-2.4973	4.2597	-1.5490
8	.5040	.3369	.2533	-2.6750	5.2501	-2.6443
9	.5001	.3334	.2501	-2.6972	5.3664	-2.7705
10	.5000	.3333	.2500	-2.6975	5.3677	-2.7719
$\star$	.5000	.3333	.2500			

Example 3: Next, consider the case $f \equiv (1+x)^{-1}$. With $n = 1$ and $r = 5$, i.e., with at most 10 moments, results are displayed in Table III. Again, one obtains very good results after 10 iterations only. Gradient components are about $O(10^{-5})$ or $O(10^{-6})$, and the first seven moments of $f_n dx$ read $(0.6931, 0.3068, 0.1931, 0.1402, 0.1099, 0.0903, 0.0766)$, a fairly good approximation. Figure 2 plots $f_n - f$ on $[0,1]$ because both curves of f_n and f are almost indistinguishable. Indeed, the scale in Figure 2 is 10^{-3}. Finally, results with $n = 2$, i.e. with 5 parameters, are displayed in Table IV, again with very good results. However, we had to set $r = 7$ (i.e. 14 moments) for the SDP-relaxations $\mathbf{L}_r$. In case the Hessian would be ill-conditioned when close to the optimal solution, more sophisticated second-order methods (or even first-order methods) might be preferable for larger r. At iteration 8 the gradient is $O(10^{-10})$.

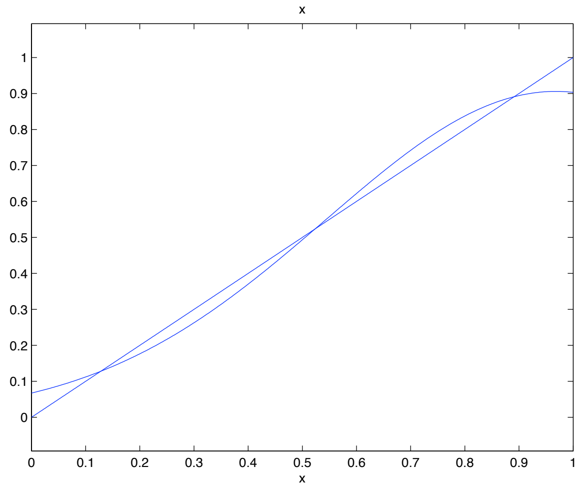

Fig. 1. Example 2: $n = 1$; $f \equiv x$ and f_n on $[0,1]$

TABLE III

EXAMPLE 3 : $f \equiv (1+x)^{-1}$ AND $n = 1$

k	moments			coefficients $\lambda^{(k)}$		
0	111.5	98.24	88.69	1.0000	1.0000	5.0000
1	41.28	36.25	32.69	0.3323	0.2306	5.4424
2	15.48	13.45	12.10	0.0569	-1.4160	6.3745
6	.8925	.4975	.3726	0.2484	-3.6629	4.2036
7	.7528	.3608	.2428	0.0986	-1.9125	1.6898
8	.7023	.3149	.2005	0.0125	-1.0867	0.4837
9	.6935	.3071	.1934	-0.0049	-0.9257	0.2461
10	.6931	.3068	.1931	-0.0055	-0.9205	0.2383
$\star$	.6931	.3069	.1931			

Figure 3 displays $f_n - f$ on $[0,1]$ because again both curves of f_n and f are almost indistinguishable. Indeed, the scale in Figure 3 is 10^{-4}.

IV. CONCLUSION

We have provided a relatively simple numerical scheme to obtain (in principle arbitrary close) approximations of gradient and Hessian data needed in first and second-order optimization methods for maximum entropy estimation.

- It is based on the semidefinite programming approach defined in [1] for evaluating integrals of exponentials of multivariate polynomials on simple compact domains.
- It avoids computing orthogonal polynomials needed at each current iterate λ, in standard approaches that use

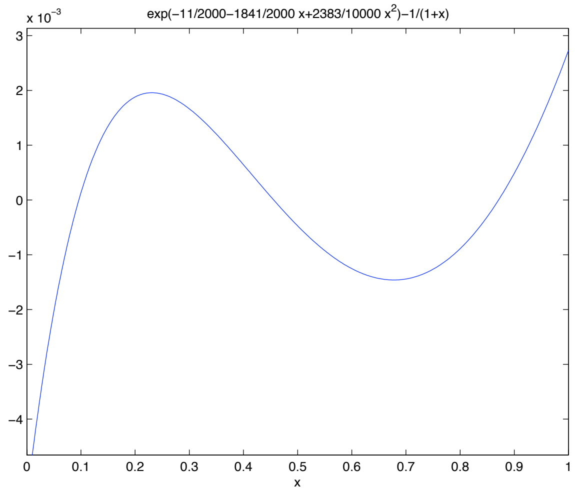

Fig. 2. Example 3, $n=1$: $f_n - f$ on $[0,1]$

TABLE IV

EXAMPLE 3 : $f \equiv (1+x)^{-1}$ AND $n=2$

k	moments				
0	8.2861	6.1806	5.1359	4.4543	3.9579
1	3.3332	2.3940	1.9635	1.6918	1.4974
2	1.5477	1.0112	0.8010	0.6781	0.5939
4	0.7504	0.3588	0.2408	0.1842	0.1507
8	0.6931	0.3069	0.1931	0.1402	0.1098
$\star$	0.6931	0.3069	0.1931	0.1402	0.1098

k	coefficients $\lambda^{(k)}$				
0	0.7500	0.7500	0.7500	0.7500	0.7500
1	0.2198	-0.0129	0.9905	0.7967	0.7684
2	0.0171	-0.6777	0.3247	1.2359	0.8959
4	0.0195	-1.5589	4.0336	-8.0020	5.5820
8	-0.0001	-0.9958	0.4658	-0.2191	0.0562

quadrature formula with respect to the weight function $f_n(\lambda,\bullet)$ on$[0,1]$.

- As it uses results from [1], the methodology easily extends to the multivariate case and on compact domains more complicated than just boxes of $\mathbb{R}^p$.

However,

- Its numerical analysis as well as its sensitivity to the number n of parameters λ is still to be investigated.
- In view of the present status of SDP solvers, its applicability is still limited to problems in dimension not larger than 3, and with a reasonable number of parameters λ.

ACKNOWLEDGEMENTS

For solving the SDP (18) we used the new version of GloptiPoly3, a public domain software dedicated to solving the generalized problem of moments as defined in [10]. See `www.laas.fr/~henrion/software/gloptipoly`. The authors wishes to thank D. Henrion for helpful discussions.

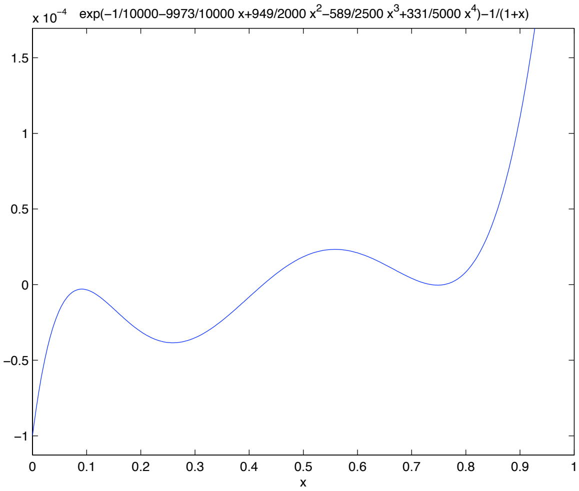

Fig. 3. Example 3, $n=2$: $f_n - f$ on $[0,1]$

REFERENCES

[1] D. Bertsimas, Xuan Vinh Doan, J.B. Lasserre, Approximating integrals of multivariate exponentials: a moment approach, *Oper. Res. Letters*, to appear. doi: 10.1016/j.orl.2007.07.002. Also available at `www.optimization-online.org`

[2] J. Borwein and A.S. Lewis, On the convergence of moment problems, *Trans. Am. Math. Soc.*, vol 325, 1991, pp 249–271.

[3] J. Borwein and A.S. Lewis, Convergence of best entropy estimates, *SIAM J. Optim.*, vol 1, 1991, pp 191–205.

[4] J. Borwein and A.S. Lewis, Partially finite convex programming, Part I: Quasi relative interiors and duality, *Math. Prog.*, vol 57, 1992, pp 15–48.

[5] C.I. Byrnes and A. Lindquist, The generalized moment problem with complexity constraint, *Int. Equat. Oper. Theory* vol 56, 2006, pp 163–180.

[6] C.F. Dunkl and Yuan Xu, *Orthogonal Polynomials of Several Variables*, Cambridge University Press, Cambridge, 2001.

[7] T. Georgiou, Relative entropy and the multi-variable multi-dimensional moment problem, *IEEE Trans. Inform. Theory* vol 52, 2006, pp 1052–1066.

[8] H.K. Kesavan and J.N. Kapur, *Entropy Optimization Principles with Applications*, Academic Press, 1992.

[9] J.B. Lasserre, Global optimization and the problem of moments, *SIAM J. Optim.* vol 11, 2001, pp 796–817.

[10] J.B. Lasserre, A semidefinite programming approach to the generalized problem of moments, *Math. Program.* vol 112, 2008, pp 65–92.

[11] L.R. Mead and N. Papanicolaou, Maximum entropy in the problem of moments, *J. Math. Phys.* vol 25, 1984, pp 2404–2417.

[12] A. Tagliani, Entropy estimate of probability densities having assigned moments: Hausdorff case, *Appl. Math. Lett.* vol 15, 2002, pp 309–314.

[13] A. Tagliani, Entropy estimate of probability densities having assigned moments: Stieltjes case, *Appl. Math. Comput.* vol 130, 2002, pp 201–211.