COMMUN. MATH. SCI.
Vol. 8, No. 2, pp. 377–392

THE MULTIDIMENSIONAL MAXIMUM ENTROPY MOMENT PROBLEM: A REVIEW ON NUMERICAL METHODS*

RAFAIL V. ABRAMOV†

Dedicated to the sixtieth birthday of Professor Andrew Majda

Abstract. Recently the author developed a numerical method for the multidimensional moment-constrained maximum entropy problem, which is practically capable of solving maximum entropy problems in the two-dimensional domain with moment constraints of order up to 8, in the three-dimensional domain with moment constraints of order up to 6, and in the four-dimensional domain with moment constraints of order up to 4, corresponding to the total number of moment constraints of 44, 83 and 69, respectively. In this work, the author brings together key algorithms and observations from his previous works as well as other literature in an attempt to present a comprehensive exposition of the current methods and results for the multidimensional maximum entropy moment problem.

Key words. maximum entropy problem, moment constraints, numerical methods.

AMS subject classifications. 49M, 65K, 90C.

1. Introduction

The moment-constrained maximum entropy problem yields an estimate of a probability density with highest uncertainty among all densities satisfying supplied moment constraints. The moment constrained maximum entropy problem arises in a variety of settings in solid state physics [7, 22, 23, 39], econometrics [34, 40], statistical description of gas flows [27, 33], geophysical applications such as weather and climate prediction [4, 5, 21, 25, 28, 29, 35, 36], and many other areas. The approximation of the probability density itself is obtained by maximizing the Shannon entropy under the constraints established by measured moments (phase space-averaged monomials of problem variables) [32]. A standard formalism [41] transforms the constrained maximum entropy problem into the unconstrained minimization problem of the dual objective function. More details on theoretical aspects of the maximum entropy moment problem can be found in [8, 15, 14, 38].

Recently, the author developed new algorithms for the multidimensional moment-constrained maximum entropy problem [1, 2, 3]. While the method in [1] is somewhat primitive and is only capable of solving two-dimensional maximum entropy problems with moments of order up to 4, the improved algorithm in [2] uses a suitable orthonormal polynomial basis in the space of Lagrange multipliers to improve convergence of its iterative optimization process. It is capable of practically solving two-dimensional problems with moments of order up to 8, three-dimensional problems with moments of order up to 6, and four-dimensional problems of order up to 4, totalling 44, 83 and 69 moment constraints, respectively, not counting the normalization constraint for a probability density. In [3], further improvements were introduced to the algorithm for the multidimensional moment-constrained maximum entropy problem, such as multiple Broyden-Fletcher-Goldfarb-Shanno (BFGS) iterations [9, 10, 12, 19, 37] to adaptively progress between points of polynomial reorthonormalization, as opposed to single Newton steps introduced in [2], and suitable constraint rescaling to reduce difference in magnitude between high and low order moment constraints. These two

*Received: November 30, 2009; accepted (in revised version): March 7, 2009.

†Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago (abramov@math.uic.edu).

improvements were observed in [3] to significantly reduce computational wall clock time (typically by a factor of 5-6) for the same maximum entropy problem.

In this work we bring together key algorithms and observations from [1, 2, 3] as well as other literature in an attempt to give the reader a comprehensive exposition of the current numerical methods and results for the multidimensional moment-constrained maximum entropy problem.

The manuscript is organized as follows. In section 2 we formulate two maximum entropy problems for the Shannon entropy and the relative entropy, or the Kullback-Leibler divergence [26], respectively, with general constraints. Section 3 shows the equivalence of the two maximum entropy problems in the framework of moment constraints. In section 4 we sketch the preconditioning method for the input moment constraints which maps an arbitrary maximum entropy problem into one with zero mean state and a diagonal covariance matrix with comparable magnitudes of high and low order moment constraints. Section 5 outlines the numerical algorithm with an orthonormal polynomial basis for the maximum entropy moment problem to improve stability and convergence. In section 6 we give an exposition of the modified Gram-Schmidt process for maintaining orthogonality of the polynomial basis. In section 7 we introduce the BFGS optimization algorithm, which improves computation speed of the basic algorithm with the polynomial basis. In section 8 we explain how to use the Gauss-Hermite quadratures to compute the integrals arising in section 5, given the problem scaling from section 4. Section 9 presents a schematic outline of the whole algorithm. Section 10 presents a few vivid examples of multidimensional maximum entropy problems with highly non-Gaussian states. Finally, section 11 outlines future directions of this work.

2. Maximum entropy problems for general constraints

Before providing a detailed exposition of the moment-constrained entropy problem, a more general optimization framework for an arbitrary set of constraints is formulated in this section for both the Shannon entropy and relative entropy. Let the partial knowledge of a probability density $\rho(\vec{x})$, where $\vec{x}$ denotes a vector of phase space coordinates, be available in the form of linear constraints

$$\begin{aligned} \mathcal{F}_0(\rho) &= \int_{\mathbb{R}^N} \rho(\vec{x})\,\mathrm{d}\vec{x} = f_0 = 1, \\ \mathcal{F}_i(\rho) &= \int_{\mathbb{R}^N} C_i(\vec{x})\rho(\vec{x})\,\mathrm{d}\vec{x} = f_i, \quad 1 \le i \le L, \end{aligned} \tag{2.1}$$

where $C_i(\vec{x})$ are such that the integrals (2.1) are finite, linearly independent, and arbitrary otherwise. Then the two maximum entropy problems are formulated as:

1. *Maximum entropy problem of the first kind*: find the optimal probability density ρ_S^* which maximizes the Shannon entropy

$$S(p) = -\int_{\mathbb{R}^N} \rho(\vec{x}) \ln \rho(\vec{x})\,\mathrm{d}\vec{x} \tag{2.2}$$

(i.e. such that $S(\rho_S^*) = \max_\rho S(\rho)$) and simultaneously satisfies the constraints in (2.1);

2. *Maximum entropy problem of the second kind*: find the optimal probability density ρ_P^* which minimizes the relative entropy

$$P(\rho,\Pi) = \int_{\mathbb{R}^N} \rho(\vec{x}) \ln\left[\frac{\rho(\vec{x})}{\Pi(\vec{x})}\right] \mathrm{d}\vec{x} \tag{2.3}$$

(i.e. such that $P(\rho_P^*,\Pi)=\min\limits_{\rho} P(\rho,\Pi)$) and simultaneously satisfies the constraints in (2.1), where $\Pi(\vec{x})$ is some known probability density.

A standard optimization approach allows the reformulation of the above two problems for the dual space of the Lagrange multipliers in the unconstrained form [32, 41]. In particular, the constrained optimization problems of the first and second kind in (2.2) and (2.3), respectively, are reduced to the unconstrained minimization of the dual objective functions in the form

$$\mathcal{L}_S(\vec{\lambda})=\int_{\mathbb{R}^N}\exp\left(\sum_{i=0}^{L}\lambda_i C_i(\vec{x})\right)\mathrm{d}\vec{x}-\sum_{i=0}^{L}\lambda_i f_i, \tag{2.4a}$$

$$\mathcal{L}_P(\vec{\theta})=\int_{\mathbb{R}^N}\Pi(\vec{x})\exp\left(\sum_{i=0}^{L}\theta_i C_i(\vec{x})\right)\mathrm{d}\vec{x}-\sum_{i=0}^{L}\theta_i f_i, \tag{2.4b}$$

over their sets of Lagrange multipliers $\vec{\lambda}$ and $\vec{\theta}$, respectively. The Hessians (matrices of second derivatives) of (2.4a) and (2.4b) are positive definite [41], and therefore the optimization problems of both the first and second kind are convex with unique optima (if the optima exist). The optimal probability densities ρ_S^* and ρ_P^* are then

$$\rho_S^*(\vec{x})=\exp\left(\sum_{i=0}^{L}\lambda_i C_i(\vec{x})\right), \tag{2.5a}$$

$$\rho_P^*(\vec{x})=\Pi(\vec{x})\exp\left(\sum_{i=0}^{L}\theta_i C_i(\vec{x})\right). \tag{2.5b}$$

It is easy to see that if the optima for (2.4a) and (2.4b) are reached (i.e. the gradients of (2.4a) and (2.4b) are zero), then the constraints in (2.1) are automatically satisfied by (2.5a) and (2.5b).

Note that since $C_0(x)=1$ in (2.5), λ_0 and θ_0 can be found as explicit functions of all other Lagrange multipliers:

$$\lambda_0=-\ln\int_{\mathbb{R}^N}\exp\left(\sum_{i=1}^{L}\lambda_i C_i(\vec{x})\right)\mathrm{d}\vec{x}, \tag{2.6a}$$

$$\theta_0=-\ln\int_{\mathbb{R}^N}\Pi(\vec{x})\exp\left(\sum_{i=1}^{L}\theta_i C_i(\vec{x})\right)\mathrm{d}\vec{x}. \tag{2.6b}$$

However, for the purpose of simplicity and better adaptation to numerical methods, developed below, here we prefer to treat λ_0 and θ_0 as generic Lagrange multipliers.

3. The unified maximum entropy problem for moment constraints

For practical applications it is common for the constraints in (2.1) to be various products of powers of phase space coordinates (moments). Since optimization problems with moment constraints are less general than those in (2.4a) and (2.4b), additional simplifications can be made in the theoretical formulation of the optimization problems before proceeding with their numerical implementation. First, however, we need to introduce a concise written form of an arbitrary moment. Here we follow

notations introduced earlier in [1, 2, 3], where for $\vec{x}\in\mathbb{R}^N$, with N being the dimension of the domain, an arbitrary monomial of $\vec{x}$ (the product of arbitrary powers of components of $\vec{x}$) is concisely written as

$$\vec{x}^{\vec{i}}=\prod_{k=1}^{N}x_k^{i_k},\qquad \vec{i}\in\mathbb{I}^N, \tag{3.1}$$

such that the monomial order $|\vec{i}|$ is the total power of all vector components, i.e.

$$|\vec{i}|=\sum_{k=1}^{N}i_k. \tag{3.2}$$

Using the above notation, for a probability density p we write a set of arbitrary moment constraints up to the total power M as

$$\mu_{\vec{i}}(\rho)=\int_{\mathbb{R}^N}\vec{x}^{\vec{i}}\rho(\vec{x})\,\mathrm{d}\vec{x}=m_{\vec{i}},\quad |\vec{i}|=0\ldots M. \tag{3.3}$$

With the set of moment constraints in (3.3), the unconstrained optimization problems in (2.4a) and (2.4b) become

$$\mathcal{L}_S(\vec{\lambda})=\int_{\mathbb{R}^N}\exp\left(\sum_{|\vec{i}|=0}^{M}\lambda_{\vec{i}}\vec{x}^{\vec{i}}\right)\mathrm{d}\vec{x}-\sum_{|\vec{i}|=0}^{M}\lambda_{\vec{i}}m_{\vec{i}}, \tag{3.4a}$$

$$\mathcal{L}_P(\vec{\theta})=\int_{\mathbb{R}^N}\Pi(\vec{x})\exp\left(\sum_{|\vec{i}|=0}^{M}\theta_{\vec{i}}\vec{x}^{\vec{i}}\right)\mathrm{d}\vec{x}-\sum_{|\vec{i}|=0}^{M}\theta_{\vec{i}}m_{\vec{i}}, \tag{3.4b}$$

with corresponding optimal probability densities

$$\rho_S^*(\vec{x})=\exp\left(\sum_{i=0}^{M}\lambda_i\vec{x}^{\vec{i}}\right), \tag{3.5a}$$

$$\rho_P^*(\vec{x})=\Pi(\vec{x})\exp\left(\sum_{i=0}^{M}\theta_i\vec{x}^{\vec{i}}\right). \tag{3.5b}$$

Note that the probability densities in (3.5a) and (3.5b) differ from each other solely due to the $\Pi(\vec{x})$ factor in the latter. In particular, if $\Pi(\vec{x})$ itself is of the form (3.5a), then the optimization problem of the second kind in (3.4b) can be reduced to the optimization problem of the first kind in (3.4a), which we demonstrate below. Let Π be of the form

$$\Pi(\vec{x})=\exp\left(\sum_{|\vec{i}|=0}^{M}\beta_{\vec{i}}\vec{x}^{\vec{i}}\right). \tag{3.6}$$

This choice of Π is natural if Π is itself the optimal probability density for the optimization problem of the first kind in (3.4a) with different constraints. Then the optimization problem of the second kind in (3.4b) can be written as

$$\mathcal{L}_P(\vec{\theta})=\int\exp\left(\sum_{|\vec{i}|=0}^{M}(\theta_{\vec{i}}+\beta_{\vec{i}})\vec{x}^{\vec{i}}\right)\mathrm{d}x-\sum_{|\vec{i}|=0}^{M}\theta_{\vec{i}}m_{\vec{i}}. \tag{3.7}$$

Changing variables $\xi_{\vec{i}}=\theta_{\vec{i}}+\beta_{\vec{i}}$ we obtain for (3.4b)

$$\mathcal{L}_P(\vec{\xi})=\int \exp\left(\sum_{|\vec{i}|=0}^{M}\xi_{\vec{i}}\vec{x}^{\vec{i}}\right)\mathrm{d}x-\sum_{|\vec{i}|=0}^{M}\xi_{\vec{i}}m_{\vec{i}}+\sum_{|\vec{i}|=0}^{M}\beta_{\vec{i}}m_{\vec{i}}, \tag{3.8}$$

where the last sum is constant and does not affect the location of the minimum of (3.8). Now it can be seen that $\mathcal{L}_P(\vec{\xi})$ in (3.8) and $\mathcal{L}_S(\vec{\lambda})$ in the optimization problem of the first kind (3.4a) differ by a constant, which means that, for the same set of initial constraints, the optimal set of Lagrange multipliers $\vec{\xi}$ for (3.8) coincides with the set $\vec{\lambda}$ for (3.4a).

Once the optimal sets of Lagrange multipliers for $\rho_S(\vec{x})$ and $\rho_P(\vec{x})$ are found, the corresponding Shannon and relative entropies are computed as follows:

$$S(\rho_S^*)=-\sum_{|\vec{i}|=0}^{M}\lambda_{\vec{i}}m_{\vec{i}}, \tag{3.9a}$$

$$P(\rho_P^*,\Pi)=\sum_{|\vec{i}|=0}^{M}(\xi_{\vec{i}}-\beta_{\vec{i}})m_{\vec{i}}. \tag{3.9b}$$

4. Preconditioning of input constraints

Before proceeding with the numerical algorithm for optimization, it is desirable to simplify and standardize the initial problem as much as possible via linear transformations of coordinates in the phase space $\mathbb{R}^N$. This preconditioning removes many implementation problems and improves the dynamic range of applicability and convergence of optimization algorithms.

The first step we suggest is to perform a coordinate change so that the mean state of the target probability density ρ^* is shifted to zero. Let $\vec{m}$ be the mean state of ρ^* and $\tilde{\rho}^*$ be the shifted probability density

$$\tilde{\rho}^*(\vec{x})=\rho^*(\vec{x}+\vec{m}) \tag{4.1}$$

so that the mean state of $\tilde{\rho}^*$ is automatically zero. Our purpose then is to obtain the set of moment constraints up to the same power for $\tilde{\rho}^*$ from the given moments of ρ^*. One can directly verify that the new moments are obtained as linear combinations of old moments by

$$\mu_{\vec{i}}(\tilde{\rho}^*)=\sum_{\vec{j}\le\vec{i}}(-1)^{|\vec{j}|}\prod_{k=1}^{N}\binom{i_k}{j_k}\mu_{\vec{i}-\vec{j}}(\rho^*)\vec{m}^{\vec{j}}. \tag{4.2}$$

At the end of computations the following direct formula can be used for the backward transformation of the computed Lagrange multipliers $\tilde{\lambda}_{\vec{i}}$ to the Lagrange multipliers $\lambda_{\vec{i}}$ of the original coordinates:

$$\lambda_{\vec{i}}=\sum_{\vec{j}\ge\vec{i}}(-1)^{|\vec{j}-\vec{i}|}\prod_{k=1}^{N}\binom{j_k}{i_k}\tilde{\lambda}_{\vec{j}}\vec{m}^{\vec{j}-\vec{i}}. \tag{4.3}$$

After the mean shift has been performed as described above, the next step of preconditioning is to rotate and stretch the coordinate system in such a way that the

constraint covariance matrix (the matrix of second moments) becomes the identity matrix in the new coordinates. This is achieved by setting

$$\vec{x}_{old} = E\Lambda^{1/2}\vec{x}_{new}, \tag{4.4}$$

where E is the matrix of eigenvectors, and Λ is the diagonal matrix of eigenvalues for the constraint covariance matrix. Let $\hat{\rho}^*$ be the mapping of the probability density $\tilde{\rho}^*$ in the rotated coordinates with rotation matrix $A = (E\Lambda^{1/2})^{-1}$, that is,

$$\hat{\rho}^*(\vec{x}) = \frac{1}{\det A}\tilde{\rho}^*(A^{-1}\vec{x}). \tag{4.5}$$

Then, one can verify that the formula for the constraint transformation is

$$\mu_{\vec{i}}(\hat{\rho}^*) = \sum_{|\vec{j}_1|=i_1} \cdots \sum_{|\vec{j}_N|=i_N} \prod_{k=1}^{N} \left[\binom{i_k}{\vec{j}_k} A_k^{\vec{j}_k} \right] \mu_{\sum_{k=1}^N \vec{j}_k}(\tilde{\rho}^*), \tag{4.6}$$

where $\vec{A}_k$ is the k-th row of the rotation matrix A and the multinomial coefficients are defined as

$$\binom{n}{\vec{j}} = \frac{n!}{\prod_{k=1}^N (j_k!)}. \tag{4.7}$$

At the end of computations the following direct formula can be used for the backward transformation of the computed Lagrange multipliers $\hat{\lambda}_{\vec{i}}$ to the Lagrange multipliers $\tilde{\lambda}_{\vec{i}}$ of the mean-shifted coordinates:

$$\tilde{\lambda}_{\vec{i}} = \sum_{\substack{\vec{j}_1 \ldots \vec{j}_N \\ \sum_l \vec{j}_l = \vec{i}}} \prod_{k=1}^{N} \left[\binom{|\vec{j}_k|}{\vec{j}_k} A_k^{\vec{j}_k} \right] \tilde{\lambda}_{|\vec{j}_1|,\ldots,|\vec{j}_N|}. \tag{4.8}$$

Note that for linear coordinate transformations, recomputation of the moment constraints of order M in the new coordinates requires only the moment constraints up to order M in the old coordinates. Since no extra information beyond the set of moment constraints has to be provided for this preconditioning to work, its user-transparent encapsulation into the main optimization routine makes it very convenient in practical use.

However, shifting the mean state of a maximum entropy moment problem to zero and setting its covariance matrix to identity is not sufficient to achieve computational stability, as one can observe even with zero mean state the moment constraints of high power to differ from the moment constraints of low power by several orders of magnitude, which in general negatively impacts convergence of the algorithm. As an example, the moments of a simple one-dimensional Gaussian distribution with zero mean state and unit variance obey the relation

$$\mu_M(\rho_G) = (M-1)!!, \qquad M \text{even}, \tag{4.9}$$

where $(M-1)!!$ denotes the factorial over odd numbers up to $M-1$. As we can see, the Gaussian moments in (4.9) grow extremely fast with increasing M. This situation is further complicated by the fact that for a maximum entropy problem of moment constraints of order up to M the polynomial reorthonormalization procedure

requires computation of moments of order up to $2M$. For instance, when the 8-order maximum entropy problem is started with the initial Gaussian distribution of zero mean and unit variance, the polynomial reorthonormalization algorithm computes $\mu_{16} = 15!! = 2{,}027{,}025$, which exceeds μ_0 and μ_2 (computed by the same algorithm) by 6 orders of magnitude.

To partially remedy the difference in magnitude between moments of different order, we suggest the following strategy. Instead of constraining the problem with the original set of moments, we will look for the optimal probability distribution

$$\rho^*_\alpha(\vec{x}) = \alpha^N \hat{\rho}^*(\alpha\vec{x}), \tag{4.10}$$

with α being a scalar parameter, with moment constraints

$$\mu_{\vec{i}}(\rho^*_\alpha) = \alpha^{-|\vec{i}|}\mu_{\vec{i}}(\hat{\rho}^*), \qquad 0 \le |\vec{i}| \le M. \tag{4.11}$$

α can be chosen to minimize the difference in magnitude between different moments. Note that for polynomial reorthonormalization the moments of total power up to $2M$ have to be computed, and therefore are needed to be taken into account for determining α.

Note that the more general way to minimize difference in magnitude between moments would be to set α to an $N \times N$ matrix and then choose its entries at each polynomial reorthonormalization so that the difference between magnitudes of all the moments of order up to $2M$ is minimized, with subsequent rescaling of moment constraints. While such an elaborate strategy for α will probably be addressed by the author in the future, it is somewhat sophisticated and requires a separate optimization subproblem to be solved for α at each polynomial reorthonormalization, which could drive the computational cost of the problem further up.

Here the scaling coefficient α is chosen once in the beginning of the optimization process, where the Gaussian distribution with zero mean state and identity covariance matrix is chosen as the starting guess. As the moments of the target probability distribution of order greater than M are unknown at the start, instead we choose α to set the common value of the highest-order corner moments μ^n_{2M} to the value of the normalization constant μ_0 of the starting guess of the iterations, where the corner moments μ^n_{2M} are defined as follows:

$$\mu^n_{2M}(\rho) = \int_{\mathbb{R}^N} x_n^{2M}\rho(\vec{x})\,\mathrm{d}x. \tag{4.12}$$

With the starting Gaussian distribution $\hat{\rho}^*$ of zero mean and identity covariance matrix, all the corner moments of order $2M$ are equal to $(2M-1)!!$ while the normalization constant $\mu_0 = 1$. With this, α becomes

$$\alpha = \sqrt[2M]{(2M-1)!!}. \tag{4.13}$$

One can verify that setting α to the value in (4.13) automatically equates the new normalization constant and all corner moments of order $2M$ for the starting Gaussian distribution.

With the rescaling in (4.13), one can check that for the 8-th order moment problem with Gaussian distribution, the 6-order difference in magnitudes of the moments is reduced to a single order. As we can see, the scaling in (4.13) is quite efficient, despite its apparent simplicity. After $\rho^*_\alpha(\vec{x})$ is found, it is easy to convert it back into $\rho^*(\vec{x})$ via (4.3), (4.8), and (4.10).

5. Polynomial basis for the moment-constrained maximum entropy problem

Although the minimization problem in (3.4a) is convex, the straightforward optimization of (3.4a) via an iterative technique like the Newton method [13] or the BFGS algorithm [10] directly over its Lagrange multipliers encounters numerical difficulty due to sensitivities of the value of the dual objective function $\mathcal{L}(\vec{\lambda})$ to changes in different Lagrange multipliers $\lambda_{\vec{i}}$. Indeed, it is easy to see that the value of the dual objective function is not likely to respond as much to a change in a first-level Lagrange multiplier $\lambda_{\vec{i}}$ with $|i|=1$, as it is likely to respond to a change in, say, a fifth-level Lagrange multiplier with $|i|=5$, due to the fact that the exponential function in the integrand of the phase-space integral in (3.4a) is more sensitive to changes in higher powers of $\vec{x}$. As a result, optimizing over the Lagrange multipliers directly often fails when the maximum moment order M exceeds the value of 6 or even 4.

Thus, in order to successfully iterate the optimization problem in (3.4a) one has to replace the set of monomials $\vec{x}^{\vec{i}}$ with a different basis in which the dual objective function in (3.4a) has roughly same sensitivity to changes in any of the coordinates of such a basis. A simple solution is to replace basis monomials $\vec{x}^{\vec{i}}$ with a set of M-th order polynomials $p_k(\vec{x})$,

$$\{\vec{x}^{\vec{i}},\lambda_{\vec{i}}\},\ \vec{i}\in\mathbb{I}^N,\ 0\le|i|\le M\ \rightarrow\ \{p_k(\vec{x}),\gamma_k\},\ 1\le k\le K,\qquad K=\frac{(M+N)!}{M!\,N!},\tag{5.1}$$

where γ_k are the Lagrange multipliers of the new basis. Since each basis polynomial $p_k(\vec{x})$ is of M-th order, the dual objective function should have comparable sensitivity to changes in different γ_k. The dual objective function (3.4a) is written in the new polynomial coordinates as

$$\mathcal{L}(\vec{\gamma})=\int_{\mathbb{R}^N}\exp\left(\sum_{k=1}^{K}\gamma_k p_k(\vec{x})\right)\mathrm{d}\vec{x}-\sum_{k=1}^{K}\gamma_k p_k(\vec{\mu}),\tag{5.2}$$

where $p_k(\vec{\mu})$ above in (5.2) denotes the polynomial $p_k(\vec{x})$ where all the powers $\vec{x}^{\vec{i}}$ are replaced with corresponding constraints $\mu_{\vec{i}}$ from (3.3). The corresponding optimal probability density function is now written as

$$\rho^*(\vec{x},\vec{\gamma})=\exp\left(\sum_{k=1}^{K}\gamma_k p_k(\vec{x})\right).\tag{5.3}$$

For the one-dimensional setting (i.e., when $\vec{x}$ is a scalar), it is common to use the shifted Chebyshev polynomials [7] or the Lagrange interpolation polynomials with suitably spaced roots [39] so that comparable sensitivity of the dual objective function to changes in new coordinates is ensured. Although it might be possible to generalize the Chebyshev polynomials or the Lagrange interpolants to the multidimensional setting, here we instead use an adaptive system of K general orthogonal polynomials, tailored for the optimization problem.

Below we adapt the key properties of a suitable polynomial system to the Newton method. First, we write the formula of the Newton iterations for (5.2):

$$\vec{\gamma}^{n+1}=\vec{\gamma}^{n}-\zeta_n\left(H^{-1}\nabla\mathcal{L}\right)|_{\vec{\gamma}^n},\tag{5.4}$$

where $\vec{\gamma}^n$ denotes the vector of Lagrange multipliers in the new basis at the n-th Newton iteration, and ζ_n is a stepping distance parameter at the n-th optimization

step. Here the entries of the gradient $\nabla\mathcal{L}$ and Hessian (matrix of second derivatives) H are given by

$$(\nabla\mathcal{L})_k = \frac{\partial\mathcal{L}}{\partial\gamma_k} = Q_{\vec{\gamma}}(p_k) - p_k(\vec{\mu}), \tag{5.5a}$$

$$(H)_{kl} = \frac{\partial^2\mathcal{L}}{\partial\gamma_k\partial\gamma_l} = Q_{\vec{\gamma}}(p_k p_l), \tag{5.5b}$$

where the quadrature $Q_{\vec{\gamma}}(g)$ for an arbitrary function $g(\vec{x})$ is computed as

$$Q_{\vec{\gamma}}(g) = \int_{\mathbb{R}^N} g(\vec{x})\rho^*(\vec{x},\vec{\gamma})\,\mathrm{d}\vec{x}. \tag{5.6}$$

It is easy to see that the optimization problem in new coordinates remains convex; the inner product $\vec{v}^T H\vec{v}$ for an arbitrary vector $\vec{v}$ is

$$\vec{v}^T H\vec{v} = v_k Q(p_k p_l)v_l = Q(v_k p_k p_l v_l) = Q((v_k p_k)^2) \geq 0,$$

where, by convention, a summation is performed over each pair of identical indices. In order to improve numerical stability of the Newton iterations in (5.4), we require the following orthogonality condition for the polynomials p_k at each step of the Newton iterations:

$$Q(p_k p_l) = \delta_{pl}, \tag{5.7}$$

where δ_{pl} is the usual Kronecker delta-symbol. The orthogonality requirement in (5.7) also provides roughly same sensitivity of the dual objective function in (5.2) to changes in different γ_k due to identical curvatures of the second-order surface approximation tangent to $\mathcal{L}$ in all directions at the current iteration point, and turns the formula in (5.5b) for the Hessian matrix into

$$(H)_{kl} = (H^{-1})_{kl} = \delta_{kl}. \tag{5.8}$$

With (5.8), the Newton method in (5.4) coincides with the steepest descent method:

$$\vec{\gamma}^{n+1} = \vec{\gamma}^n - \zeta_n\,(\nabla\mathcal{L})\,|_{\vec{\gamma}^n}. \tag{5.9}$$

6. Reorthonormalization of the polynomial basis

In section 5 we have formulated the orthogonality condition in (5.7) for the basis polynomials in (5.1) in order to improve the numerical stability and convergence of the minimization of the dual objective function in (5.2). However, the quadratures Q in (5.6) are weighted by the probability density function of the form (5.3) which changes between different points of the optimization path, and thus the orthogonal relation in (5.7) is not necessarily preserved as the optimization process evolves. Therefore, a Gram-Schmidt type of polynomial reorthonormalization of p_k is required to keep polynomials p_k orthogonal in the sense of (5.7). Poor numerical stability of the classical Gram-Schmidt reorthonormalization is widely known, and common numerically stable tools to reorthonormalize a set of Euclidean vectors usually involve the Householder reflections or Givens rotations [20]. However, it is not clear whether an analog of Householder or Givens decomposition exists for polynomials of an arbitrary dimension and order, and a suitable stabilized modification of the Gram-Schmidt algorithm will be used instead.

Recent work [11, 16, 17, 18] demonstrates that the modified Gram-Schmidt algorithm yields a good precision for vectors which are not "too ill-conditioned". In particular, it is demonstrated in [17] that the modified Gram-Schmidt algorithm with reorthogonalization yields errors which are small multiples of the machine round-off error.

According to the classical Gram-Schmidt method, K arbitrary linearly independent polynomials a_k are converted into the orthonormal (in the sense of (5.7)) set of polynomials p_k as

$$p_k = \frac{a_k - \sum_{l=1}^{k-1} Q(a_k p_l) p_l}{Q\left(\left[a_k - \sum_{l=1}^{k-1} Q(a_k p_l) p_l\right]^2\right)^{1/2}}, \qquad 1 \le k \le K. \tag{6.1}$$

As mentioned before, the classical Gram-Schmidt method is numerically unstable, and here we use a suitably modified version of the Gram-Schmidt method with reorthogonalization from [18], tailored for a polynomial basis. It is not convenient to illustrate the modified Gram-Schmidt method as a formula in (6.1) due to recursive nature of its computational implementation, and instead we give a step-by-step program of the algorithm as in [18] in a fashion which resembles modern computer languages:

Modified Gram-Schmidt algorithm

```
for k = 1,...,K {
    for l = 1,r
        for m = 1,...,k-1
            a_k = a_k - Q(a_k p_m) p_m
    p_k = Q(a_k^2)^{-1/2} a_k
}
```

The parameter r above is 1 if the reorthonormalization is not performed, and 2 otherwise. We observed that the modified Gram-Schmidt algorithm either with or without reorthogonalization produces sufficiently orthogonal polynomials for the maximum entropy problems considered in [2, 3]. Note that the modified Gram-Schmidt algorithm without reorthogonalization is more easily parallelizable, and therefore is a better candidate for parallel implementations of the maximum entropy problem.

7. The BFGS optimization algorithm

The optimization algorithm with the polynomial basis described above in section 5 performs the Gram-Schmidt reorthonormalization of the basis polynomials at each step of the steepest descent iterations in (5.9), which, on one hand, yields second order convergence in the vicinity of the optimal point, but, on the other hand, increases computational expense of the method, since the computational cost of polynomial reorthonormalization roughly matches the cost of Hessian computation in the monomial basis. Thus, an obvious way to reduce the computational expense of the basic algorithm is to perform several descent iterations between the Gram-Schmidt reorthonormalizations. However, the steepest descent method in (5.9) in this case loses precision down to the first order and becomes vulnerable to curvature anisotropy due to its lack of affine invariance. On the other hand, we need to avoid computing the Hessian during the iterations between polynomial basis reorthonormalizations, which suggests using one of the quasi-Newton methods such as BFGS.

The Broyden-Fletcher-Goldfarb-Shanno formula [9, 10, 12, 19, 37] is a quasi-Newton method, widely used in optimization algorithms. It needs the Hessian (or an approximation to it) at the starting point of iterations and only requires computation of the gradient of the dual objective function in (5.2) at each successive iteration, which significantly reduces computational cost in our case. The structure of the BFGS algorithm is the following:

- In the first iteration, provide the starting gradient of the dual objective function $(\nabla\mathcal{L})_0$ and starting Hessian H_0, which is an identity matrix after polynomial reorthonormalization (see (5.5b));
- At iteration m, perform the following steps:
  a) Find the direction of descent by solving

$$H_m\vec{d}_m = -(\nabla\mathcal{L})_m; \tag{7.1}$$

  b) Perform a line search for the step distance ζ_m and find the next iterate $\vec{\gamma}_{m+1}$ as

$$\vec{\gamma}_{m+1} = \vec{\gamma}_m + \zeta_m\vec{d}_m; \tag{7.2}$$

  c) At the new iterate $\vec{\gamma}_{m+1}$, compute the gradient $(\nabla\mathcal{L})_{m+1}$;
  d) Compute the new iterate of the pseudo-Hessian as

$$H_{m+1} = H_m + \frac{\vec{y}_m \otimes \vec{y}_m}{\vec{s}_m\vec{y}_m} - \frac{(H_m\vec{s}_m)\otimes(H_m\vec{s}_m)}{\vec{s}_m H_m\vec{s}_m}, \tag{7.3}$$

  where $\vec{s}_m = \vec{\gamma}_{m+1} - \vec{\gamma}_m$ and $\vec{y}_m = (\nabla\mathcal{L})_{m+1} - (\nabla\mathcal{L})_m$.

In practice, to compute the descent direction in (7.1), we apply the Sherman-Morrison formula to the pseudo-Hessian in (7.3) and obtain

$$\begin{aligned} H_{m+1}^{-1} = H_m^{-1} &+ \frac{\vec{s}_m\vec{y}_m + \vec{y}_m H_m^{-1}\vec{y}_m}{(\vec{s}_m\vec{y}_m)^2}(\vec{s}_m\otimes\vec{s}_m) - \\ &- \frac{1}{\vec{s}_m\vec{y}_m}\left[(H_m^{-1}\vec{y}_m)\otimes\vec{s}_m + \vec{s}_m\otimes(H_m^{-1}\vec{y}_m)\right]. \end{aligned} \tag{7.4}$$

Then, the descent direction in (7.1) is computed as

$$\vec{d}_m = -H_m^{-1}(\nabla\mathcal{L})_m. \tag{7.5}$$

For details on the Sherman-Morrison formula see, for example, [20].

As successive BFGS steps change the current iterate of the probability distribution ρ^* in (5.3), the current set of polynomials p_k in (5.1), which stays the same, gradually loses orthogonality with respect to changing ρ^*, which negatively impacts convergence of the BFGS iterations due to increased numerical errors in computation of the descent direction. Thus, when the error in the descent direction becomes too large, the basis polynomials in (5.1) have to be reorthonormalized with respect to the current iterate of (5.3) and the BFGS process has to be restarted. This adaptive reorthonormalization approach requires a computationally cheap estimate of the numerical error in the descent direction to be available at each step of the BFGS iterations. One can observe from (7.5) that errors in the search direction $\vec{d}_m$ originate from the errors in the

computed gradient of the dual objective function $\nabla\mathcal{L}$, amplified by the inverse pseudo-Hessian H_m^{-1}. The main source of errors in the gradient of the dual objective function are the Gauss-Hermite quadrature errors in computing moments of ρ^*, which usually remain bounded since the size of the quadrature and locations of abscissas are fixed during the course of computation. Thus, errors in the computed descent direction increase mainly when amplified by the inverse pseudo-Hessian H^{-1} in (7.5), and may grow significantly when H^{-1} becomes too ill-conditioned.

In the present algorithm, we monitor the condition number κ of the inverse pseudo-Hessian H^{-1} during the BFGS iterations and reorthonormalize the polynomial basis when κ exceeds the value of 20 (this threshold value of κ is empirically found to be small enough to preserve numerical stability of iterations, and at the same time large enough to allow multiple successive BFGS iterations between polynomial reorthonormalizations). Following [20], we compute the condition number in the L_∞ norm as

$$\kappa = \|H\|_\infty \|H^{-1}\|_\infty, \qquad \|H\|_\infty = \max_i \sum_j |H_{ij}|. \tag{7.6}$$

Thanks to the Sherman-Morrison formula, both H and H^{-1} are readily available through (7.3) and (7.4), respectively, and the computation of the condition number κ is inexpensive.

8. Quadrature computations

The integrals in (5.6) are computed using the Gauss-Hermite quadrature (for a standard reference, see, for example, [6]). The standard formula for the T-point Gauss-Hermite integration is

$$\int_{-\infty}^{+\infty} f(x)\exp\left(-\frac{1}{2}x^2\right)\mathrm{d}x \approx \sum_{j=1}^{T} f(x_j)w_j. \tag{8.1}$$

The Gauss-Hermite abscissae x_j are found as roots of the T-th order Hermite polynomial, which by itself is defined through the recurrence

$$H_0 = 1, \quad H_{j+1} = xH_j - jH_{j-1}, \tag{8.2}$$

and weights w_j are computed through the formula

$$w_j = \frac{(H_{N-1}, H_{N-1})}{H_{N-1}(x_j)H'_N(x_j)}, \tag{8.3}$$

where the inner product is defined in the sense of (8.1). In order to adapt the Gauss-Hermite quadrature to the α-rescaling from (4.13), the abscissas and weights are trivially scaled by α. The choice of the Gauss-Hermite quadrature is based on the following observation: recall that the optimization problem is preconditioned in section 4 to have a zero mean state and diagonal covariance matrix; thus, if the optimal probability density is Gaussian, then the Gauss-Hermite quadrature for such a density becomes exact for $T > M$ up to the machine round-off error. For non-Gaussian constraints the Gauss-Hermite quadrature loses exactness. However, the location of the abscissas and magnitude of the weights are important. In the vicinity of the optimal point, the integrands in (5.6) decay rapidly away from the origin due to preconditioning. The Gauss-Hermite abscissas are concentrated near the origin and become

sparse away from it, and the Gauss-Hermite weights approach zero away from the origin (where the integrands in (5.6) approach zero as well). Thus, the Gauss-Hermite quadrature offers a consistent sampling of an integrand of the form (5.6), which makes it more suitable for our problem than other standard high-order quadratures.

It is worth to mention that, while the moment-constrained maximum entropy problem is formulated on $\mathbb{R}^N$ (i.e. it is the upper entropy bound of the multidimensional Hamburger moment problem), the numerical integration through the Gauss-Hermite quadratures occures via the summation over the finite number of Gaussian abscissas. Potentially, the use of a finite-point numerical integration may lead to a situation where the computed estimate of the maximum entropy state is not normalizable on $\mathbb{R}^N$, although the finite-point quadratures remain bounded. This situation for the one-dimensional 4-moment maximum entropy problem has been studied in [24], and, to some extent, in [2]. While in the one-dimensional case a negative value of the Lagrange multiplier for the highest moment power is sufficient (but not necessary) to guarantee integrability, it is not as trivial in the multidimensional setting, where imposing the same constraint on the Lagrange multipliers of highest-order corner moments does not guarantee integrability, since there may exist noncompact manifolds in $\mathbb{R}^N$ on which the maximum entropy estimate grows exponentially away from the origin while being integrable along any of the basis directions. Currently, verification of integrability of the computed maximum entropy estimate is not implemented due to this difficulty, although it may be addressed by the author in the future.

9. Schematic outline of the algorithm

This section illustrates a schematic step-by-step outline of the algorithm, suitable as a set of general guidelines for its practical numerical implementation.

1. Precondition input constraints by setting zero mean and diagonal covariance as in section 4 above;
2. Generate a set of K random linearly independent polynomials p_k of M-th order;
3. Choose the starting set of the Lagrange multipliers γ_k corresponding to the Gaussian distribution matching the mean and covariance constraints of the preconditioned problem in the new polynomial basis p_k;
4. Reorthogonalize the set of polynomials p_k according to (5.7) with respect to the current iterate of ρ, using the Gauss-Hermite quadratures to compute integrals Q in (5.6), and recompute the set of Lagrange multipliers γ_k with respect to the reorthogonalized basis;
5. Compute the gradient of the new iterate of the dual objective function $\mathcal{L}$;
6. Perform BFGS steps until the minimum is reached or the condition number of the BFGS pseudo-Hessian becomes too ill-conditioned;
7. If the minimum of the dual objective function is not reached, return to step 4; otherwise, recompute the optimal γ_k into the set of standard Lagrange multipliers $\lambda_{\vec{i}}$ for the monomial basis $\vec{x}^{\vec{i}}$, to match the format of the input constraints.

The algorithm is implemented for an arbitrary phase space dimension N and an arbitrary order of input constraints M, using the object-oriented style of the C++ programming language. However, practical limitations on N and M are imposed by computational speed of machine-specific hardware.

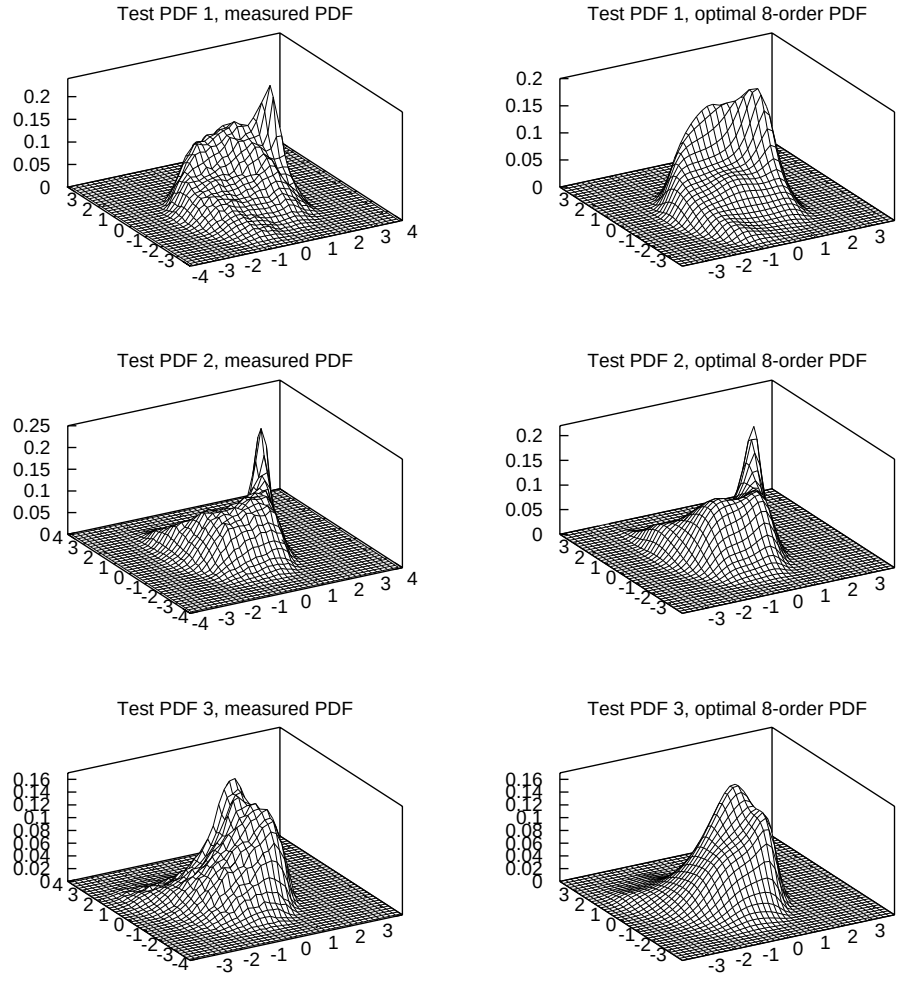

FIG. 10.1. *The two-dimensional test PDF and their maximum entropy estimates. Left: measured directly from the long-term model simulation through standard bin-counting. Right: computed by the 8-order maximum entropy problem.*

10. Computational ability

The algorithm for the maximum entropy problem schematically outlined above in section 9 has been used in [3] to produce moment-constrained maximum entropy estimates for the model variables of the model of large-scale oceanic currents [30, 31]. The details of the model physics, implementation and variables are beyond the scope of this review; for more details, see [3, 30, 31]. Here we pick three two-dimensional highly non-Gaussian test PDFs from [3] to demonstrate the computational ability of the maximum entropy alogirthm. In figure 10.1 in the left column we show the directly measured two-dimensional joint PDFs, computed by the standard bin-counting. The algorithm was found in [3] to be capable of computing 2-dimensional maximum entropy problems with moments of order up to 8 (44 moment constraints in total,

not counting the normalization requirement for a PDF), 3-dimensional problems with moments of order up to 6 (83 constraints), and 4-dimensional problems with moments of order up to 4 (69 constraints). The computed optimal 2-dimensional PDFs of 8th order are shown in figure 10.1 in the right column. Here we do not show the pictures of computed 3- and 4-dimensional PDFs; for detailed information about timing performance, accuracy and more pictures the reader is referred to [3].

11. Future work

Future work in this direction will be aimed at the ability of the developed suite of algorithms to converge for problems with higher dimension and moment constraint order. With higher dimension and moment order, the number of iterations needed to converge for a higher-moment maximum entropy problem increases, as well as the time to perform a single iteration in a higher-dimensional setting. We would especially like to stress that the "envelope of convergence" of the algorithm depending on the problem dimension and order (that is, 2D – 8th order, 3D – 6th order, 4D – 4th order) presented in section 10 does not imply that the algorithm fails to converge outside of the envelope. Rather, the computational cost of a single polynomial reorthonormalization or a BFGS iteration becomes so high that it is not feasible to perform computations with the existing implementation, which is not currently adapted to parallel computation on multiple processors.

On the other hand, it is trivial to parallelize the computations of the Gauss-Hermite quadratures onto multiple processors, and somewhat less trivial, but possible, to parallelize the modified Gram-Schmidt algorithm for polynomial reorthonormalization. In the future, various options for speeding up the iteration process will be studied, mainly focusing on parallel implementation of the Gauss-Hermite quadratures and the modified Gram-Schmidt process in a parallel computational environment. In particular, the use of contemporary graphic processing units (GPU), will be considered for accelerating the maximum entropy algorithm.

Acknowledgment. The author is supported by the National Science Foundation Grant DMS-0608984 and the Office of Naval Research Grant N00014-06-1-0286.

REFERENCES

[1] R. Abramov, *A practical computational framework for the multidimensional moment-constrained maximum entropy principle*, J. Comp. Phys., 211, 198–209, 2006.
[2] R. Abramov, *An improved algorithm for the multidimensional moment-constrained maximum entropy problem*, J. Comp. Phys., 226, 621–644, 2007.
[3] R. Abramov, *The multidimensional moment-constrained maximum entropy problem: a BFGS algorithm with constraint scaling*, J. Comp. Phys., 228, 96–108, 2009.
[4] R. Abramov and A. Majda, *Quantifying uncertainty for non-Gaussian ensembles in complex systems*, SIAM J. Sci. Comp., 26, 411–447, 2003.
[5] R. Abramov, A. Majda and R. Kleeman, *Information theory and predictability for low frequency variability*, J. Atmos. Sci., 62, 65–87, 2005.
[6] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*, Dover Publications, 1965.
[7] K. Bandyopadhyay, A. Bhattacharya, P. Biswas and D. Drabold, *Maximum entropy and the problem of moments: a stable algorithm*, Phys. Rev. E, 71, 2005.
[8] J. Borwein and A. Lewis, *On the convergence of moment problems*, Trans. Amer. Math. Soc., 325, 249–271, 1991.
[9] C. Broyden, *The convergence of a class of double-rank minimization algorithms*, J. Inst. Math. Appl., 6, 76–90, 1970.
[10] R. Byrd, P. Lu and J. Nocedal, *A limited memory algorithm for bound-constrained optimization*, SIAM J. Sci. Comp., 16, 1190–1208, 1995.

[11] A. Dax, *A modified Gram-Schmidt algorithm with iterative orthogonalization and column pivoting*, Lin. Alg. Appl., 310, 25–42, 2000.
[12] R. Fletcher, *A new approach to variable metric algorithms*, Comp. J., 13, 317–322, 1970.
[13] R. Fletcher, *Practical Methods of Optimization*, Wiley, New York, 1987.
[14] M. Frontini and A. Tagliani, *Maximum entropy in the finite Stieltjes and Hamburger moment problem*, J. Math. Phys., 35, 6748–6756, 1994.
[15] B. Fuglede, *The multidimensional moment problem*, Expo. Math., 1, 47–65, 1983.
[16] L. Giraud and J. Langou, *When modified Gram-Schmidt generates a well-conditioned set of vectors*, IMA J. Num. Anal., 22, 521–528, 2002.
[17] L. Giraud, J. Langou and M. Rozložník, *On the round-off error analysis of the Gram-Schmidt algorithm with reorthogonalization*, Tech. Report TR/PA/02/33, CERFACS, 2002.
[18] L. Giraud, J. Langou and M. Rozložník, *On the loss of orthogonality in the Gram-Schmidt orthogonalization process*, Tech. Report TR/PA/03/25, CERFACS, 2003.
[19] D. Goldfarb, *A family of variable metric updates derived by variational means*, Math. Comp., 24, 23–26, 1970.
[20] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 3rd ed., 1996.
[21] K. Haven, A. Majda and R. Abramov, *Quantifying predictability through information theory: small sample estimation in a non-Gaussian framework*, J. Comp. Phys., 206, 334–362, 2005.
[22] R. Haydock and C. Nex, *Comparison of quadrature and termination for estimating the density of states within the recursion method*, J. Phys. C: Solid State Phys., 17, 4783–4789, 1984.
[23] R. Haydock and C. Nex, *A general terminator for the recursion method*, J. Phys. C: Solid State Phys., 18, 2235–2248, 1985.
[24] M. Junk, *Domain of definition of Levermore's five-moment problem*, J. Stat. Phys., 93, 1143–1167, 1998.
[25] R. Kleeman, *Measuring dynamical prediction utility using relative entropy*, J. Atmos. Sci., 59, 2057–2072, 2002.
[26] S. Kullback and R. Leibler, *On information and sufficiency*, Ann. Math. Stat., 22, 79–86, 1951.
[27] C. Levermore, *Moment closure hierarchies for kinetic theories*, J. Stat. Phys., 83, 1021–1065, 1996.
[28] A. Majda, R. Abramov and M. Grote, *Information Theory and Stochastics for Multiscale Nonlinear Systems*, CRM Monograph Series of Centre de Recherches Mathématiques, Université de Montréal, American Mathematical Society, ISBN 0-8218-3843-1, 25, 2005.
[29] A. Majda, R. Kleeman and D. Cai, *A framework for predictability through relative entropy*, Meth. Appl. Anal., 9, 425–444, 2002.
[30] J. McCalpin, *The statistics and sensitivity of a double-gyre model: the reduced gravity, quasi-geostrophic case*, J. Phys. Oceanogr., 25, 806–824, 1995.
[31] J. McCalpin and D. Haidvogel, *Phenomenology of the low-frequency variability in a reduced-gravity, quasigeostrophic double-gyre model*, J. Phys. Oceanogr., 26, 739–752, 1996.
[32] L. Mead and N. Papanicolaou, *Maximum entropy in the problem of moments*, J. Math. Phys., 25, 2404–2417, 1984.
[33] I. Müller and T. Ruggeri, *Extended Thermodynamics*, Springer Tracts in Natural Philosophy, Springer, Heidelberg, 1st ed., 1993.
[34] D. Ormoneit and H. White, *An efficient algorithm to compute maximum entropy densities*, Econ. Rev., 18, 127–140, 1999.
[35] M. Roulston and L. Smith, *Evaluating probabilistic forecasts using information theory*, Mon. Wea. Rev., 130, 1653–1660, 2002.
[36] T. Schneider and S. Griffies, *A conceptual framework for predictability studies*, J. Clim., 12, 3133–3155, 1999.
[37] D. Shanno, *Conditioning of quasi-Newton methods for function minimization*, Math. Comp., 24, 647–656, 1970.
[38] A. Tagliani, *Hausdorff moment problem and maximum entropy: a unified approach*, Appl. Math. and Comp., 105, 291–305, 1999.
[39] I. Turek, *A maximum-entropy approach to the density of states with the recursion method*, J. Phys. C, 21, 3251–3260, 1988.
[40] X. Wu, *Calculation of maximum entropy densities with application to income distribution*, J. Econ., 115, 347–354, 2003.
[41] Z. Wu, G. Phillips, R. Tapia and Y. Zhang, *A fast Newton algorithm for entropy maximization in phase determination*, SIAM Rev., 43, 623–642, 2001.