Journal of Computational Physics 228 (2009) 9079–9091

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Maximum entropy algorithm with inexact upper entropy bound based on Fup basis functions with compact support

Hrvoje Gotovac [a,b,*], Blaž Gotovac [b]

[a] Department of Land and Water Resources Engineering, KTH, Brinellvagen 32,10044 Stockholm, Sweden
[b] Department of Civil and Architectural Engineering, University of Split, Matice hrvatske 15, 21000 Split, Croatia

ARTICLE INFO

Article history:
Received 16 May 2009
Received in revised form 7 September 2009
Accepted 8 September 2009
Available online 15 September 2009

Keywords:
Maximum entropy algorithm
Classic Moment Problem
Fup basis functions
Compact support

ABSTRACT

The maximum entropy (MaxEnt) principle is a versatile tool for statistical inference of the probability density function (pdf) from its moments as a least-biased estimation among all other possible pdf's. It maximizes Shannon entropy, satisfying the moment constraints. Thus, the MaxEnt algorithm transforms the original constrained optimization problem to the unconstrained dual optimization problem using Lagrangian multipliers. The Classic Moment Problem (CMP) uses algebraic power moments, causing typical conventional numerical methods to fail for higher-order moments ($m > 5$—10) due to different sensitivities of Lagrangian multipliers and unbalanced nonlinearities. Classic MaxEnt algorithms overcome these difficulties by using orthogonal polynomials, which enable roughly the same sensitivity for all Lagrangian multipliers. In this paper, we employ an idea based on different principles, using Fup_n basis functions with compact support, which can exactly describe algebraic polynomials, but only if the *Fup* order-n is greater than or equal to the polynomial's order. Our algorithm solves the CMP with respect to the moments of only low order Fup_2 basis functions, finding a Fup_2 optimal pdf with better balanced Lagrangian multipliers. The algorithm is numerically very efficient due to localized properties of Fup_2 basis functions implying a weaker dependence between Lagrangian multipliers and faster convergence. Only consequences are an iterative scheme of the algorithm where power moments are a sum of Fup_2 and residual moments and an inexact entropy upper bound. However, due to small residual moments, the algorithm converges very quickly as demonstrated on two continuous pdf examples – the beta distribution and a bi-modal pdf, and two discontinuous pdf examples – the step and double Dirac pdf. Finally, these pdf examples present that *Fup* MaxEnt algorithm yields smaller entropy value than classic MaxEnt algorithm, but differences are very small for all practical engineering purposes.

1. Introduction

Many physical processes cannot be characterized deterministically due to the presence of intrinsic or parametric uncertainty due to their physical nature, interpretation or measurements. Therefore, results are usually given in the form of a certain number of the first few statistical power moments or rarely as a probability density function (pdf). Jaynes [15] defined the Maximum Entropy (MaxEnt) principle as a versatile tool for statistical inference of the probability density function (pdf) from its moments (Classic Moment Problem – CMP) by maximizing the Shannon entropy [20]. This provides a least-biased

* Corresponding author. Address: Department of Civil and Architectural Engineering, University of Split, Matice hrvatske 15, 21000 Split, Croatia. Tel.: +385 21 303 354; fax: +385 21 465 117, tel.: +46 8 790 86 90; fax: +46 8 790 86 89.
E-mail addresses: hrvoje.gotovac@gradst.hr, gotovac@kth.se (H. Gotovac), blaz.gotovac@gradst.hr (B. Gotovac).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.09.011

estimation among all other possible distributions that satisfy constrained moments but ignore all unknown information. In other words, the MaxEnt pdf presents the pdf with the highest uncertainty and appears to be a robust tool for pdf prediction in terms of statistical moments [5].

In the last few decades, a number of MaxEnt algorithms have been developed and applied in many branches of science, including solid-state physics [6,23], geophysical applications [1–3], econometrics [17,24] and transport planning [21]. The first robust algorithm was developed in [16] for up to 10–12 moments. However, the MaxEnt algorithms for a higher number of moments are subjected to high unbalanced nonlinearities, ill-conditioned Jacobian and Hessian matrices in Newton algorithms, and many other numerical problems such as insufficient arithmetic precision. In order to overcome these difficulties, the MaxEnt algorithm uses orthogonal polynomials instead of classic monomials. Among others, this "state of the art" approach is presented in [23,6], who employed Lagrangian and Chebyshev polynomials, respectively. Recently, Abramov [1–3], in his series of papers, extended these improvements to multi-dimensional problems using the generalized orthogonal polynomials.

In this paper, we employ a different and original idea using the finite and localized basis functions with compact support closely related to the algebraic polynomials, which makes possible an efficient MaxEnt algorithm with more balanced nonlinearities and the ability to solve a higher number of moments. We are focused here on localized basis functions with compact support, such as wavelets and splines. Apart from wavelets and splines, there is a relatively lesser known class of atomic or R_{bf} basis functions (Rvachev's basis functions) [18,19]. Atomic functions are classified between classic polynomials and spline functions. However, in practice, their application as basis functions is closer to splines or wavelets. In this study, we shall use *Fup* basis functions, which are one type of atomic basis functions; recent review in [12]. Gotovac and Kozulić [8] systemized the existing knowledge on atomic functions and presented the transformation of basis functions into a numerically applicable form. The application of Fup basis functions has been demonstrated in signal processing [14], for solving the integral Fredholm equations [13], in initial value problems [9], and in the collocation methods for boundary value problems [10]. Recently, Fup basis functions were applied to the Monte-Carlo methodology and stochastic processes of flow and transport in heterogeneous porous media [11].

This paper is organized as follows. In the next section, the novel Fup Maximum Entropy Algorithm (FMEA) will be presented. In Section 3, two continuous pdf examples – the beta distribution and a bi-modal pdf and two discontinuous pdf examples – step and double Dirac pdf show ability of the method to solve ill-posed maximum moment entropy problem only with low order Fup_2 basis functions enabling a stable and efficient algorithm. We end the paper with conclusions and an Appendix A, which presents all basic properties of the *Fup* basis functions needed for development of the FMEA.

2. MaxEnt algorithm

In this section, we will present the maximum entropy principle and discuss numerical schemes for the classic MaxEnt algorithm with orthogonal polynomials. Last, we will present a novel MaxEnt algorithm that uses Fup_2 basis functions with compact support.

2.1. Maximum entropy principle

The maximum entropy principle (MaxEnt) is widely recognized as an efficient stochastic tool, especially in information theory. Furthermore, MaxEnt is particularly useful for pdf characterization, since it shows how many conventional statistical moments are needed in order to accurately describe all pdf properties, such as its shape, tailings, peakedness, number of peaks, skewness and/or kurtosis. Despite direct and sometimes contradictory relations with classical physical entropy [15], which describes chaos in a physical system and presents the second law of thermodynamics, Shannon information entropy [20] is defined in a broader sense as

$$H(f) = -\int_{x_{\min}}^{x_{\max}} \ln(f(x)) f(x) dx \tag{1}$$

where $I = \ln(f(x))$ is a quantity of information and f is the pdf. Shannon information entropy is the expected information quantity. The logarithm is chosen arbitrarily according to some desired properties of entropy: (a) decreasing probability causes an increase of information, (b) higher uncertainty causes higher entropy and (c) the total entropy of two independent events is equal to the sum of the individual entropies [22].

The MaxEnt is defined by *Jaynes* [15] such that the pdf with highest entropy is selected to give the most information among all other possible pdf's that satisfy known constraints. In other words, the MaxEnt principle states that among the probability distributions that satisfy our incomplete information about the system, the pdf that maximizes entropy is the least-biased estimate that can be made. It agrees with everything that is known but carefully avoids anything that is unknown [5,22]. If these constraints are known statistical moments of arbitrary basis functions $(h_j(x); j = 0, \ldots, m)$, MaxEnt can be defined as the following optimization problem

$$\max H(f) \tag{2a}$$

$$\int_{x_{\min}}^{x_{\max}} h_i(x) f(x) dx = \mu_i; \quad i = 0, \ldots, m \tag{2b}$$

Optimization problem (2) can be solved by introducing the Lagrangian function and corresponding multipliers λ_j

$$L(f,\lambda) = H(f) - \sum_{j=0}^{m} \lambda_j \left(\int_{x_{\min}}^{x_{\max}} h_j(x) f(x) dx - \mu_j \right) \tag{3}$$

The problem is reduced to solving the maximum of the Lagrangian function with respect to all possible functions f that satisfy constraints (2b)

$$\frac{\partial L(f,\lambda)}{\partial f} = \int_{x_{\min}}^{x_{\max}} (-1 - \ln(f(x)) - \sum_{j=0}^{m} \lambda_j h_j(x)) dx = 0 \tag{4}$$

The analytical form of the MaxEnt pdf is

$$f(x) = \exp\left(-1 - \sum_{j=0}^{m} \lambda_j h_j(x) \right) \tag{5}$$

Finally, for a given constraints, the MaxEnt problem requires solving the Lagrangian multipliers from the nonlinear system with $(m+1)$ equations, which is formed by introducing the MaxEnt pdf (5) into constraints (2b)

$$\int_{x_{\min}}^{x_{\max}} h_i(x) \exp\left(-1 - \sum_{j=0}^{m} \lambda_j h_j(x) \right) dx = \mu_i; \quad i = 0, \ldots, m \tag{6}$$

For a larger number of moments $(m > 2)$ and/or arbitrary basis functions $(h_j(x);\ j = 0, \ldots, m)$, the integrals in (6) are analytically intractable, and we use here adaptive Romberg integration

$$\sum_{j=1}^{N} a_{ij} w_j \exp\left(-1 - \sum_{k=0}^{m} a_{kj} \lambda_k \right) - \mu_i = 0; \quad i = 0, ..., m \tag{7}$$

where N is a number of integration points, x_j and w_j are the x-coordinates and weighting coefficients of integration points, respectively, and a_{ij} are the values of $h_i(x_j)$. System (7) shows that the original constrained optimization problem (2) can be transformed into the unconstrained convex optimization problem, which finds a minimum of the dual variable:

$$D(\lambda) = \sum_{j=1}^{N} w_j \exp\left(-1 - \sum_{i=0}^{m} a_{ij} \lambda_i \right) - \sum_{i=0}^{m} \mu_i \lambda_i \tag{8}$$

It can be shown that a minimum of dual variable $D(\lambda)$ yields optimal solution λ_j^*, which is also the solution of the original problem (2), according to the Kuhn–Tucker theorem and solution (5) for a wide range of validity [5]. Note that a minimization of $D(\lambda)$, i.e. their derivatives over all Lagrangian multipliers are zero, yields system (7). If basis functions $h_i(x)$ are monomials x^i, then $a_{ij} = x_j^i$, along with the original (2) and dual problem (8), are transformed into the Classic Moment Problem (CMP) where the zeroth normalized condition is given as $\int_{x_{\min}}^{x_{\max}} f(x) dx = 1$. The MaxEnt pdf now has the following form

$$f(x) = \exp\left(-1 - \lambda_0 - \sum_{j=1}^{m} \lambda_j x^j \right) \tag{9}$$

2.2. Classic MaxEnt algorithm

CMP and the corresponding nonlinear system (7) can be solved by a number of classic optimization techniques, such as generalized or improved iterative scaling, gradient descent, the standard or modified Newton method and the BFGS procedure [2,3,5]. In this paper, we use improved iterative scaling, an iterative procedure that solves only one moment equation (in system 7) in each nonlinear step and finds a correction of the corresponding Lagrangian multiplier [6]. The procedure consists of four basic steps, as follows.

Algorithm 1

Step 1: Choose the initial vector $\boldsymbol{\lambda}^0$ (usually the null vector) and sufficiently small threshold $\eta_1 > 0$, set the iteration counter $k = 0$ and calculate the initial value of pdf (9) for $\boldsymbol{\lambda}^0$.

Step 2: Set the index of the current equation as $i = (k \bmod m)$ and find a correction of λ^k using the classic Newton method

$$\phi_i^k\left(\lambda^k\right) = \sum_{j=1}^{N} a_{ij} f^k(x_j) \exp\left(a_{ij}\lambda^k\right) - \mu_i = 0 \tag{10}$$

Step 3: Add the calculated correction to the current solution vector

$$\lambda_l^{k+1} = \lambda_l^k + \delta_{li}\lambda^k; \quad l = 0, \ldots, m \tag{11}$$

Step 4: Calculate the MaxEnt pdf (9) for vector λ^{k+1}. If all moment equations in (7) for a new vector λ^{k+1} have values less than η_1, then stop the procedure and set the current solution as a final solution (9). Otherwise, set $k = k + 1$ and go to step 2.

This iterative procedure is very attractive and computationally efficient because there is no matrix inversion. The procedure is very stable and not sensitive to the initial vector as in the Newton method, but it usually requires more iteration steps and exhibits slower convergence. However, all the aforementioned iterative procedures suffer from these numerical difficulties and lack of convergence for CMP problems with $m > 5$—10 due to unbalanced nonlinearities of the Lagrangian multipliers. These difficulties are closely related to the different behavior of monomials x^i in the exponent of pdf (9) [2]. Therefore, for a larger number of moments, the Jacobian and/or Hessian matrices become ill-conditioned with significant influence from arithmetic precision, even in the case where extended precision is used [16,23]. These numerical difficulties can be significantly reduced by using the orthogonal polynomials instead of monomials x^i. In this way, Lagrangian multipliers provide similar sensitivities to the solution changes in the iterative nonlinear procedure, implying that the stable algorithm is well suited for a larger number of moments. It is conventional to use Chebyshev [6] or Lagrange polynomials [23] for 1-D moment problems. Abramov [2,3] has used generalized orthogonal polynomials for the multi-dimensional problems, maintaining the orthogonality by the modified Gramm–Schmidt procedure in each iteration step.

Without loss of generality, we can use shifted Chebyshev polynomials where interval $[-1,1]$ is transformed to $[0,1]$ for solving the CMP and corresponding Lagrangian multipliers [6]. In that case,$a_{ij} = T_i(x_j)$, and there is a direct relation between Chebyshev polynomials and monomials

$$T_i(x) = \sum_{j=0}^{m} \alpha_{ij} x^j; \quad i = 0, \ldots, m \tag{12}$$

where α_{ij} is the connection matrix. The reverse relation is given as

$$x^i = \sum_{j=0}^{m} \beta_{ij} T_j(x); \quad i = 0, \ldots, m \tag{13}$$

where $\beta_{ij} = \alpha_{ij}^{-1}$. Due to the linear character of relation (13), it is possible to relate moments in the same way

$$\mu_i = \sum_{j=0}^{m} \beta_{ij} t_j; \quad i = 0, \ldots, m \tag{14}$$

Relations (13) and (14) imply that satisfaction of Chebyshev moments also satisfies classic power moments. CMP is now reduced to the maximum entropy problem over the Chebyshev polynomials

$$\int_0^1 T_j(x) \exp\left(-1 - \sum_{l=0}^{m} \gamma_l T_l(x)\right) dx = t_j; \quad j = 0, \ldots, m \tag{15}$$

Finally, the optimal MaxEnt pdf that satisfies both power and Chebyshev moments (15) has the following form

$$f(x) = \exp\left(-1 - \sum_{j=0}^{m} \gamma_j T_j(x)\right) \tag{16}$$

Note that these relations are valid for all types of orthogonal polynomials, as well as for all other basis functions that satisfy relations (12) and (13).

2.3. Fup MaxEnt algorithm

Although the orthogonal polynomials significantly increase the efficiency of all aforementioned iterative procedures for the CMP (7), many numerical difficulties are still present. Moreover, the drawback remains of a strong connection between polynomials and their influence on the moment changes. For example, in light of Algorithm 1, correction λ^k belongs to the current moment equation in system (7), but in each iteration step, it may significantly changes all the other moment equations, which can substantially decrease computational efficiency, especially for a larger number of moments and multi-dimensional problems [2].

The original idea behind the proposed algorithm is to use finite and localized basis functions $h_i(x)$ with compact support so that correction λ^k in each iteration step changes only a few moment equations in system (7), namely, the ones that belong to the neighboring basis functions. This concept can significantly decrease the number of iteration steps in Algorithm 1. Classic candidates are B-splines and wavelets, which exactly describe polynomials. In this paper, we use *Fup* basis functions, which belong to the atomic class of basis functions [8]. *Fup* basis functions also exactly describe polynomials if the *Fup* order is equal to or higher than the polynomial order, as shown in Appendix A [8]. However, this means that *Fup* basis functions, as well as other basis functions with compact support, require a higher order for a larger number of moments and have loose

localized properties in this particular case (the compact support is greater than the considered interval from 0 to 1). Moreover, the iterative procedure shows even poorer performance than the algorithms with classic orthogonal polynomials. In order to use the full capabilities of *Fup* basis functions, it is better to use low *Fup* order (we use here $n = 2$), but the exact relation (13) still does not hold for the description of arbitrary polynomials of the mth order $m > 2$ or monomials with $m > 2$

$$x^i = \sum_{j=0}^{m} d_{ij} Fup_{2j}(x) + \varepsilon_i(x); \quad i = 0, \ldots, m \tag{17}$$

where $\varepsilon_i(x)$ are residual functions that describe the difference between monomials of the mth order and their Fup_2 approximation. Appendix A shows that a linear combination of displaced Fup_2 basis functions describes exactly the monomials up to $m = 2$, but all higher-order monomials are described approximately. Eq. (17) presents the connection matrix d_{ij}, which depends on the Fup_2 approximation and on the location and number of basis functions and, consequently, moments. For instance, the collocation procedure [10] calculates exactly monomials in the collocation points, and residual functions fluctuate around the zero, with significantly smaller values than *Fup* basis functions (Appendix A and Theorem 1). For increasing numbers of moments and basis functions, the residual functions converge to zero. In the limit, for $m \to \infty$, Eq. (17) converges to the exact relation (13). Using Eq. (17), we can relate classic power moments and Fup_2 moments

$$\mu_i - \Delta\mu_i^{(l-t1)} = \sum_{j=0}^{m} d_{ij} \mu_j^{Fup_2(l)}; \quad i = 0, \ldots, m \tag{18}$$

where $\mu_i, \Delta\mu_i^{(l-1)}$ and $\mu_j^{Fup_2(l)}$ are moments of monomials, residual and Fup_2 basis functions, respectively. Since moments of residual and Fup_2 basis functions are unknown, the algorithm must be defined in an iterative way, where l is a counter of iteration steps. An algorithm starts with an initial pdf guess ($l = 0$). In each iteration step, the residual moments are first calculated from the previous iteration or initial conditions, then Fup_2 moments are obtained from the system (18), and finally the MaxEnt nonlinear system is solved with respect to only Fup_2 moments

$$\int_0^1 Fup_{2i}(x) \exp\left(-1 - \sum_{j=0}^{m} \gamma_j Fup_{2j}(x)\right) dx = \mu_i^{Fup_2}; \quad i = 0, \ldots, m \tag{19}$$

The procedure is repeated until convergence is achieved. The *Fup* MaxEnt algorithm is reduced to CMP over the moments of Fup_2 basis functions. Finally, the optimal pdf has the form

$$f^*(x) = \exp\left(-1 - \sum_{j=0}^{m} \gamma_j Fup_{2j}(x)\right) \tag{20}$$

where classic power moments are satisfied exactly according to the relations (17) and (18)

$$\mu_i = \int_0^1 x^i f^*(x) dx = \sum_{j=0}^{m} d_{ij} \int_0^1 Fup_{2j}(x) f^*(x) dx + \int_0^1 \varepsilon_i(x) f^*(x) dx; \quad i = 0, \ldots, m \tag{21}$$

This algorithm solves higher-order MaxEnt moment problem with only low order Fup_2 basis functions which is opposite to all existing MaxEnt algorithms. Only consequence is an iterative algorithm defined by Eq. (18), while classic MaxEnt algorithms directly solve CMP due to exact relation (13) between monomials and orthogonal polynomials. However, iterative scheme (18) converges very quickly due to small residual moments (see Appendix A) and their weak influence on changes of Fup_2 moments. Therefore, an initial pdf guess as Gaussian normal distribution leads to fast convergence and relatively small number of outer iterations as will be shown in the sequel.

Algorithm 1 now has the better convergence properties because the correction λ^k will change only six moments of the closest Fup_2 basis functions. This implies that for a larger number of moments, the compact support is smaller, while the connections between moment equations in system (7) are weaker. This results in a considerably smaller number of iterations and more efficient the nonlinear Algorithm 1. *Fup* moments have a direct mathematical interpretation because they present the average value of the pdf over the compact support. In the limit, the *Fup* basis function converges to the Dirac function, while the compact support reduces to the point and corresponding moment describes the exact value of the pdf at that point. This algorithm is called the Fup MaxEnt Algorithm (FMEA), and it is described by the following iterative procedure:

Algorithm 2

Step 1: For a given set of $m + 1$ classic power moments on general interval $[x_{\min}, x_{\max}]$, calculate their values on unit interval $[0,1]$ by applying the simple linear transformation.
Step 2: Set $m + 1$ uniformly displaced Fup_2 basis functions (Appendix A) on unit interval $[0,1]$ and calculate connection matrix d_{ij} and residual functions (17). Choose the initial pdf ($l = 0$).
Step 3: Calculate residual moments in the lth iteration step.
Step 4: $l = l + 1$; Solve the system (18) and find Fup_2 moments.

Step 5: Solve the MaxEnt problem with Fup_2 moments (19) using Algorithm 1 and $a_{ij} = Fup_{2j}(x - x_i^0)$, where x_i^0 are vertices of the corresponding basis functions. Also, obtain the optimal pdf (20).

Step 6: Calculate the difference between power moments (Eq. 21) in the l and $l-1$ iteration step. If this difference is below some prescribed threshold η_2, stop the procedure. Otherwise, go to step 3.

Finally, note that *Fup* optimal pdf in (20) is inexact although exactly satisfies prescribed set of moments because only optimal pdf in (9) yields upper entropy bound according to Eqs. (1)–(8) and Section 2.1. In the sequel, we will discuss what consequences imply inexact entropy bound in practical pdf examples.

3. Pdf examples

In this section, we will present an application of the Fup MaxEnt Algorithm (FMEA) on two illustrative continuous pdf examples: the unimodal beta distribution and a bi-modal pdf, and two illustrative discontinuous pdf examples: the step distribution and a double Dirac pdf. We concentrate mainly on the accuracy of the algorithm in terms of reproduction of the exact pdf and their original power moments. FMEA uses Fup_2 basis functions, which means that at least 4 basis functions are needed for Algorithm 2 ($m = 3$). It is well known that the Gaussian distribution is indeed also the MaxEnt pdf for $m = 2$. Therefore, the choice of the Fup_2 basis functions seems reasonable, as this allows the maximum use of localized basis functions, while retaining good approximation properties. Threshold η_1 in Algorithm 1 is chosen up to the limit of the machine's double precision (around 10^{-15}). Threshold η_2 in Algorithm 2 then depends on η_1 and iterative algorithm (18). We will show accuracy of the algorithm if $\eta_1 = 10^{-15}$. The initial pdf is the Gaussian pdf. The most efficient strategy is to first solve the MaxEnt pdf with $m = 3$. In that case, the Gaussian pdf is a very good starting vector, implying fewer inner iterations in Algorithm 1 and outer iterations in Algorithm 2. Then, we gradually increase the number of moments, so that the previous pdf solution is the initial pdf for the next simulation. In this paper, we use a dyadic grid ($m = 2^j + 2$ where $j = 0, 1, \ldots$ is a resolution level), which implies that the number of moments increases from $m = 3$ to $m = 4, 6, 10, 18$, and so on, until the *Fup* optimal pdf converges to the exact pdf.

In the first example, we consider the beta distribution

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{22}$$

which is defined on interval $[0,1]$ and depends on two parameters α and β, while Γ is the gamma function. In this example, $\alpha = 3$ and $\beta = 5$, defining a slightly skewed pdf that requires more than two moments for its accurate description. Fig. 1 shows that three moments overestimate the pdf peak and produce a larger error at the left boundary. Four moments reduce the differences between the exact and optimal pdf, especially for the right tail. Six moments quite accurately describe all pdf features, while 10 moments completely reproduce the exact pdf.

Table 1 presents an accuracy and convergence analysis for the Fup MaxEnt approximation of the beta distribution for six moments. We calculate the absolute moment error between the given and calculated power moments over all moments for

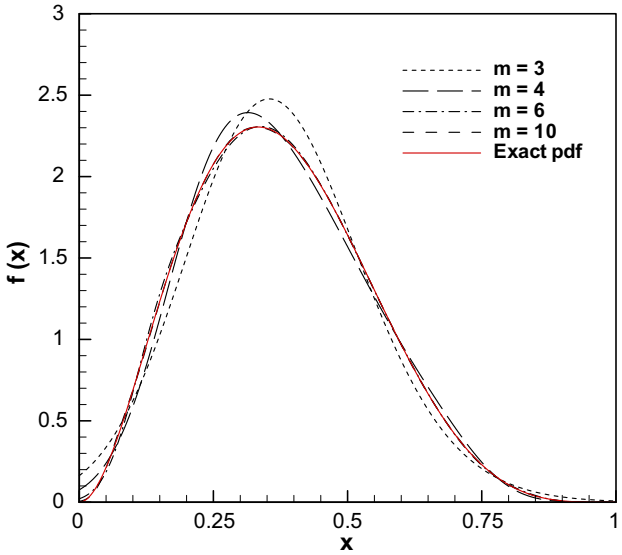

Fig. 1. MaxEnt approximation of the beta pdf using Fup_2 moments up to $m = 10$.

$\eta_1 = 10^{-15}$. Table 1 shows how the actual moment error decreases with the number of outer iterations in Algorithm 2. For $l = 5$—10, the moment error is already small and acceptable for many practical and engineering purposes. After 50 outer iterations, the moment actual error, 2.5×10^{-12}, reaches the asymptotic limit that is influenced by Fup threshold $\eta_1 = 10^{-15}$. In each outer iteration, there are $(20—50) \cdot m$ inner iterations in Algorithm 1 in order to produce the double precision accuracy of Fup_2 calculated moments in (18). Algorithm 1 is characterized by a stable nonlinear solver due to localized properties of Fup_2 basis functions and converges regardless the initial vector which is the most important property for an ill-posed MaxEnt problem. After only $(2—5) \cdot m$ inner iterations, Algorithm 1 obtained the single precision accuracy due to weak connections between basis functions which produces a nonstiff and well-balanced nonlinear system. This relatively large number of total iterations can be significantly reduced if thresholds η_1 and η_2 are chosen to be large enough that the obtained moment errors satisfy our current requirements.

In the second example, we define a more complex bi-modal pdf as a sum of two Gaussian pdfs, normalizing its total area to unity in a similar way as in [2]. For instance, a bi-modal pdf can be a real example of the concentration pdf in turbulent diffusion or in porous media as a consequence of the pore-scale dispersion [4]. Fig. 2 shows the approximation of the bi-modal pdf by $m = 3, 4, 6, 10$ and 18. The MaxEnt pdf with three moments remains unimodal, while bi-modality is obtained for all pdfs with more than three moments. This means that four moments are usually needed to describe the bi-modal nature of the pdf. Higher-order moments $(m > 4)$ have significant influence on the accuracy, where the MaxEnt pdf with 10 moments describes very accurately all features of the pdf, while the MaxEnt pdf with $m = 18$ completely reproduces the exact pdf.

Table 2 presents the accuracy of the FMEA in terms of maximum absolute error defined for the bi-modal pdf, $m = 10$ and $\eta_1 = 10^{-15}$. As in the first example, 5—10 outer iterations ensure that the moment error is quite small for many practical and engineering purposes. In this case, after only 20 outer iterations, the actual moment error is comparable with the usual threshold in Newton algorithms.In each outer iteration, there are $(20—50) \cdot m$ inner iterations in Algorithm 1 for the double precision accuracy. Furthermore, increasing the number of moments yields a pdf that converges to the exact one. The number of iterations is large compared to the Newton method, but this algorithm uses a stable iterative procedure with low order basis functions, which can solve a larger number of moments without sensitivity to the initial vector or classic numerical difficulties such as lack of convergence, arithmetic precision or ill-conditioned related matrices.

Fig. 3 presents the approximation of the step discontinuous pdf by $m = 6, 10$ and 18. This example is quite difficult due to discontinuity at $x = 0.5$. Increasing number of moments clearly presents a better description of the exact step pdf, particularly its boundaries and region around the front. Localized properties of Fup_2 basis functions enable the efficient resolving of all discontinuities and nondifferentiable regions relating only few particular Lagrangian multipliers to these places. Contrary, the classic MaxEnt algorithms use orthogonal polynomials and related Lagrangian multipliers defined on the whole domain

Table 1
Absolute moment error for beta pdf and $m = 6$ with respect to number of outer iteration steps and $\eta_1 = 10^{-15}$.

Outer iteration step (l)	5	10	20	50	100	200
Absolute moment error	1.4×10^{-4}	9.4×10^{-7}	1.5×10^{-11}	2.7×10^{-12}	2.5×10^{-12}	2.5×10^{-12}

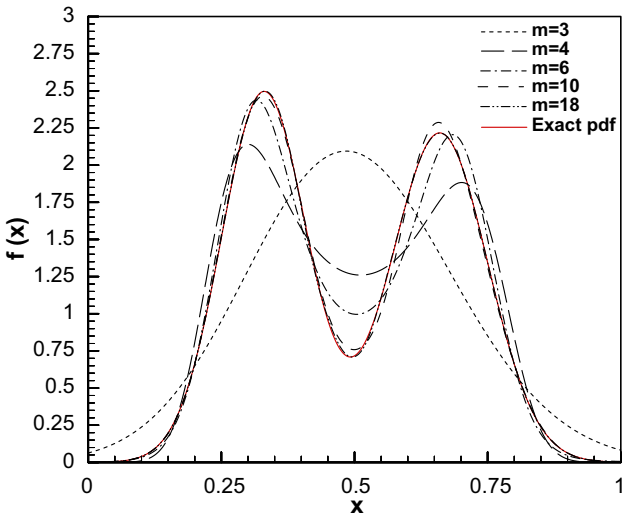

Fig. 2. MaxEnt approximation of the bi-modal pdf using Fup_2 moments up to $m = 18$.

Table 2
Absolute moment error for bi-modal pdf and $m = 10$ with respect to the number of outer iteration steps and $\eta_1 = 10^{-15}$.

Outer iteration step (l)	5	10	20	50	100	200
Absolute moment error	4.8×10^{-6}	7.7×10^{-9}	2.1×10^{-12}	4.3×10^{-13}	4.2×10^{-13}	4.2×10^{-13}

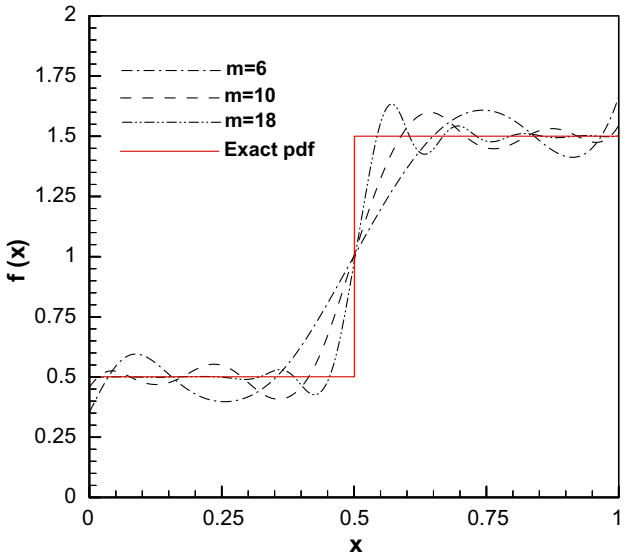

Fig. 3. MaxEnt approximation of the step pdf using Fup_2 moments up to $m = 18$.

Table 3
Absolute moment error for step pdf and $m = 10$ with respect to the number of outer iteration steps and $\eta_1 = 10^{-15}$.

Outer iteration step (l)	5	10	20	50	100	200
Absolute moment error	3.5×10^{-6}	4.7×10^{-8}	1.7×10^{-10}	1.5×10^{-11}	1.3×10^{-11}	1.3×10^{-11}

which cannot always efficiently resolve pdf's with spatially different features. Despite pdf discontinuities, Table 3 presents a quite accurate calculation of power moments and consequently small corresponding absolute moment error.

Discussion for MaxEnt approximation of more demanding discontinuous pdf examples has been given by Bandyopadhyay et al. [6]. Therefore, we choose one their example (Fig. 1 in [6]), the double Dirac pdf with pulses at $x = 0.25$ and 0.75 in order to show robustness and accuracy of our algorithm. Fig. 4 presents that Fup MaxEnt approximation for $m = 18$ correctly describes Dirac pulses without any numerical oscillations at boundaries or between them. Moreover, pulses are symmetric, while peak value is around 175. By comparison in [6], pulses are slightly nonsymmetrical and peak value is around 47 using the 25 shifted Chebyshev moments. Higher peak value implies more accurate MaxEnt approximation of double Dirac pdf. Furthermore, absolute moment error is around 10^{-10} which means that algorithm keeps accuracy, even for very demanding and discontinuous pdf examples. All four examples present ability of FMEA to accurately describe different pdf's and all its features keeping the stability of algorithm for a larger number of moments.

Finally, we need to check consequences of an inexact nature of our algorithm with respect to the upper entropy bound comparing entropy values for classic MaxEnt Algorithm and pdf in (9) or (16) and for FMEA and pdf in (20). In the first example (beta pdf), we firstly check the entropy values for basic algorithm with monomials and pdf (9) and algorithm with Chebyshev polynomials and pdf (16). For instance, both algorithms with $m = 4$ yield $H = -0.428134$. Therefore, for all other pdf examples and higher number of moments we will use classic MaxEnt algorithm based on Chebyshev polynomials [6] for comparative purposes. On the other side, our algorithm yields somewhat smaller entropy value $H = -0.428421$. Accordingly, for the bi-modal pdf and $m = 10$, classic algorithm yields $H = -0.423428$, since our algorithm yields $H = -0.423583$. Also, for the step pdf and $m = 10$, classic algorithm yields $H = -0.123142$, since our algorithm yields $H = -0.123424$. The double Dirac pdf is not suitable for comparison due to larger differences between our pdf solution and one in [6]. First three pdf examples demonstrate that our algorithm is inexact because it yields smaller entropy value than classic MaxEnt algorithms, as in [6]. However, in all these particular pdf examples, entropy differences are around 10^{-4} which yield small entropy value

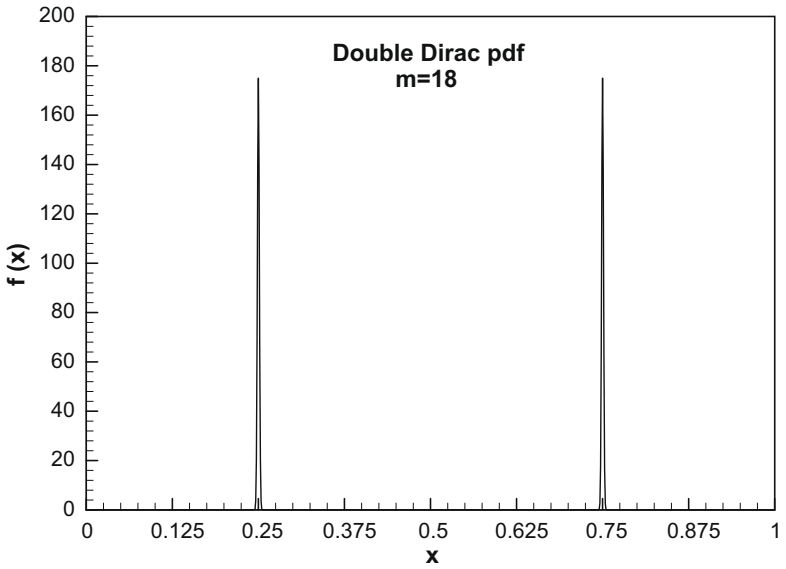

Fig. 4. MaxEnt approximation of the double Dirac pdf (pulses at $x = 0.25$ and 0.75) using Fup_2 moments up to $m = 18$.

error implying "almost" obtained upper entropy bound and similar pdf's in (9), (16) and (20), with differences irrelevant for all practical purposes.

4. Conclusions

The Classic Moment Problem (CMP) causes typical conventional numerical methods to fail for higher-order moments ($m > 5$—10) due to different sensitivities of Lagrangian multipliers and unbalanced nonlinearities. Classic MaxEnt algorithms overcome these difficulties by using orthogonal polynomials, which enable roughly the same sensitivity for all Lagrangian multipliers. In this paper, we present the Fup MaxEnt Algorithm (FMEA). This algorithm is based on low order Fup_2 basis functions with compact support, which exactly describe polynomials up to the second order but approximately expressed other higher-order polynomials. Therefore, the MaxEnt nonlinear algorithm possesses more balanced nonlinearities due to localized Fup_2 basis functions with compact support and enables a robust and efficient numerical procedure well suited for a larger number of moments. As a consequence, the algorithm exactly satisfies power moments over the Fup optimal MaxEnt pdf using the iterative scheme (Eq. (18)) where algebraic power moments are calculated as a sum of Fup_2 and residual moments. However, due to the excellent approximation properties of Fup_2 basis functions for description of polynomials, the residual moments are quite small implying a fast convergence of presented iterative stable algorithm. Robustness and accuracy of our algorithm is demonstrated on two continuous pdf examples: beta and bi-modal pdf, and two discontinuous pdf examples: step and double Dirac pdf. Particularly, presented algorithm does not satisfy upper entropy bound in comparison with classic MaxEnt algorithms [6]. However, presented pdf examples clearly demonstrated that Fup_2 optimal pdf in (20) "almost" yields upper entropy bound with value differences irrelevant for practical engineering purposes.

This new type of MaxEnt algorithm can be further developed for multi-dimensional problems [2,3] and an extreme number of moments ($m > 20$), as needed in solid-state physics [6,23]. Finally, the potential usefulness of *Fup* basis functions and other functions with compact support should be explored for new algorithms in stochastic modeling, such as Bayesian Maximum Entropy [7].

Appendix A. Fup basis functions

A.1. Calculation of the Fup_2 basis functions

Atomic basis functions are compactly supported and infinitely differentiable functions [18,19]. Atomic functions $y(.)$ are defined as solutions of differential-functional equations of the following type:

$$L_D y(x) = \lambda_D \sum_{k=1}^{M} C_k y(ax - b_k) \tag{A.1}$$

where L_D is a linear differential operator with constant coefficients, λ_D is a nonzero scalar, C_k are coefficients of the linear combination, $a > 1$ is a parameter defining the length of the compact support and b_k are coefficients that determine the

displacements of the basis functions. Rvachev and Rvachev [18], in their pioneering work, called these basis functions "atomic" because they span vector spaces of all three fundamental functions in mathematics: algebraic, exponential and trigonometric polynomials. Also, atomic functions can be divided into an infinite number of small pieces that maintain all their properties, implying a so-called "atomic structure."

The simplest function, which is the most studied among atomic basis functions, is the *up(x)* function. Function *up(x)* is a smooth function with compact support $[-1,1]$, which is obtained as a solution of a differential-functional equation:

$$up'(x) = 2up(2x+1) - 2up(2x-1) \tag{A.2}$$

with the normalized condition $\int_{-\infty}^{\infty} up(x)dx = \int_{-1}^{1} up(x)dx = 1$. Function *up(x)* can be expressed as an inverse Fourier transform

$$up(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{itx}\prod_{j=1}^{\infty}\left(\frac{\sin(t2^{-j})}{t2^{-j}}\right)dt \tag{A.3}$$

Since Eq. (A.3) represents an exact but mathematically intractable expression, Rvachev [19] and Gotovac and Kozulic [8] provided tractable means for calculating the function *up(x)*

$$up(x) = 1 - \sum_{k=1}^{\infty}(-1)^{1+p_1+\cdots+p_k}p_k\sum_{j=0}^{k}C_{jk}(x-0,p_1\ldots p_k)^j \tag{A.4}$$

where coefficients C_{jk} are rational numbers determined according to the following expression

$$C_{jk} = \frac{1}{j!}2^{j(j+1)/2}up(-1+2^{-(k-j)}); \quad j = 0,1,\ldots,k, \;\; k = 1,2,\ldots,\infty \tag{A.5}$$

Calculation of the $up(-1+2^{-r});\ r\in[0,\infty]$ in binary-rational points (Eq. (A.5)), as well as all details regarding the calculation of the function *up(x)* values, is provided in [8]. The argument $(x-0,p_1\ldots p_k)$ in Eq. (A.4) is the difference between the real value of coordinate *x* and its binary form with *k* bits, where $p_1\ldots p_k$ are digits, 0 or 1, of the binary representation of the *x* coordinate's value. Therefore, the accuracy of the *x* coordinate computation, and, thus, the accuracy of the up(x) function at an arbitrary point, depends on the machine's accuracy.

From Eq. (A.2), it can be seen that the derivatives of the *up(x)* function can be calculated simply from the values of the function itself. The general expression for the derivative of the *m*th degree is

$$up^{(m)}(x) = 2^{C_{m+1}^2}\sum_{k=1}^{2^m}\delta_k up(2^m x + 2^m + 1 - 2k), \quad m\in N \tag{A.6}$$

where $C_{m+1}^2 = m(m+1)/2$ is the binomial coefficient and δ_k are the coefficients with ±1 value according to the recursive formulas $\delta_{2k-1} = \delta_k,\ \delta_{2k} = -\delta_k,\ k\in N,\ \delta_1 = 1$. It can be observed that the derivatives consist of the *up(x)* function compressed to an interval of 2^{-m+1} length with ordinates extended with the $2^{C_{m+1}^2}$ factor.

The $Fup_n(x)$ function satisfies the following differential-functional equation

$$Fup_{n'}(x) = 2\sum_{k=0}^{n+2}\left(C_{n+1}^k - C_{n+1}^{k-1}\right)Fup_n(x)\left(2x - 2^{-n-1}k + 2^{-n-2}(n+2)\right) \tag{A.7}$$

where *n* is the *Fup* order. For $n=0,\ Fup_0(x) = up(x)$, since $Fup_n(x)$ and its derivatives can be calculated using a linear combination of displaced *up(x)* functions instead of using their Fourier transforms

$$Fup_n(x) = \sum_{k=0}^{\infty}C_k^*(n)up\left(x - 1 - \frac{k}{2^n} + \frac{n+2}{2^{n+1}}\right) \tag{A.8}$$

where $C_0^*(n) = 2^{C_{n+1}^2} = 2^{n(n+1)/2}$ and $C_k^*(n) = C_0^*(n)\cdot C_k'(n)$, where a recursive formula is used for calculating auxiliary coefficients $C_k'(n)$

$$C_0'(n) = 1, \text{ when } k = 0; \text{ i.e.when } k > 0$$
$$C_k'(n) = (-1)^k C_{n+1}^k - \sum_{j=1}^{\min\{k;2^{n+1}-1\}}C_{k-j}'(n)\cdot\delta_{j+1} \tag{A.9}$$

The $Fup_n(x)$ is defined on the compact support $\left[-(n+2)2^{-n-1};(n+2)2^{-n-1}\right]$. Fig. A.1 shows the $Fup_2(x)$ function and its first two derivatives, which are used in this paper. In order to define $Fup_2(x)$ basis functions on interval $[0,1]$ in Algorithms 1 and 2, the location of each basis function is actually determined by the location of the vertex and defined by $b_k = (k-1)\Delta x$ where $k = 0,\ldots,m$ is the corresponding moment index, while $\Delta x = 1/(m-2)$ is the characteristic interval or distance between vertices of basis functions (Fig. A.2). The calculation of basis function values at a general characteristic interval Δx should be done in the following form with respect to a basic characteristic interval 2^{-n}

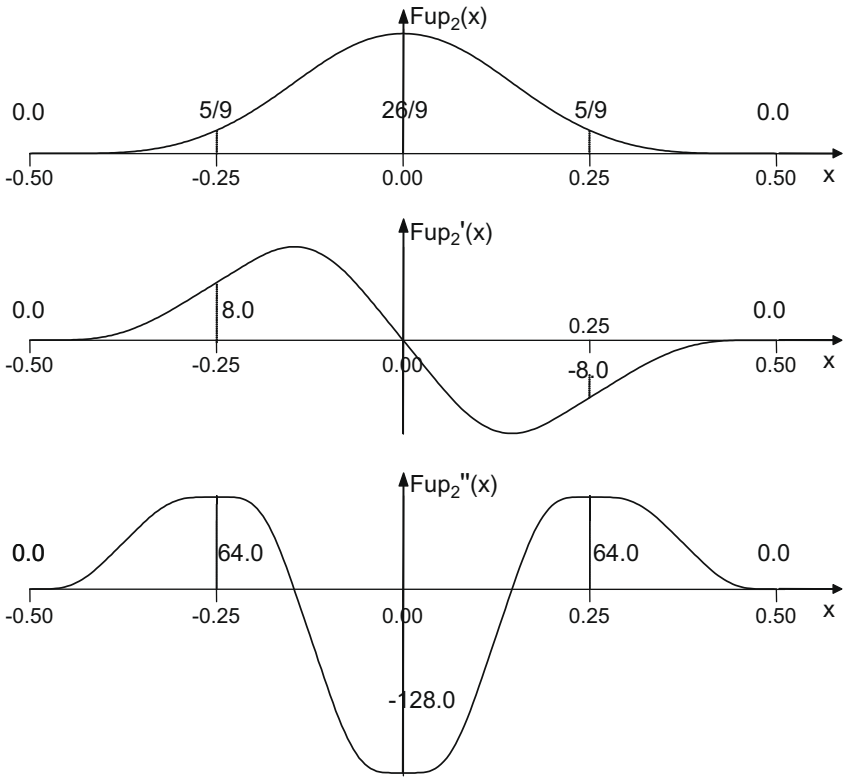

Fig. A.1. Function $Fup_2(x)$ and its first two derivatives.

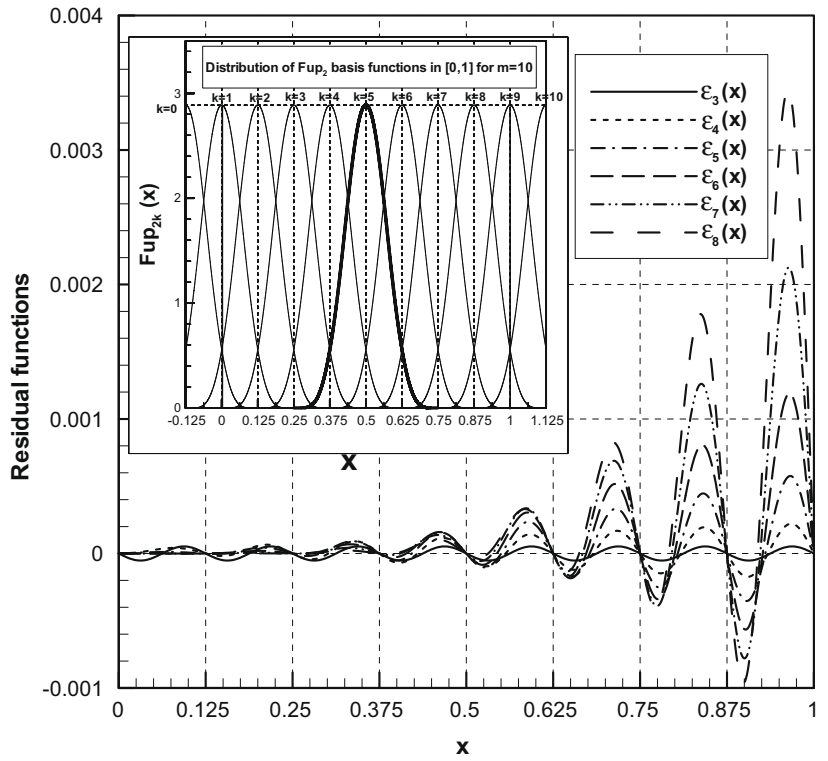

Fig. A.2. Residual functions defined for approximation of monomials on interval $[0,1]$ and $m = 10$. Also, locations and distribution of $Fup_2(x)$ basis functions are presented.

$$Fup_{2k}(x) = \frac{1}{(4\Delta x)} Fup_2\left(\frac{x-b_k}{4\Delta x}\right) \tag{A.10}$$

A.2. Approximation properties of the Fup_2 basis functions

Index n also denotes the highest degree of the polynomial that can be expressed exactly in the form of a linear combination of $n+2$ $Fup_n(x)$ basis functions, uniformly displaced by a characteristic interval Δx. In this paper, we use low order $Fup_2(x)$ basis functions, which exactly express polynomials (or monomials in this specific case) up to the second order on interval $[0,1]$ by a collocation procedure presented in [8]

$$x^0 = \frac{1}{4}\sum_{k=0}^{m} Fup_2\left(\frac{x}{4\Delta x} - \frac{k^*}{4}\right) \tag{A.11}$$

$$x = \frac{\Delta x}{4}\sum_{k=0}^{m} k^* Fup_2\left(\frac{x}{4\Delta x} - \frac{k^*}{4}\right) \tag{A.12}$$

$$x^2 = \frac{\Delta x^2}{4}\sum_{k=0}^{m}\left(k^{*^2} - \frac{5}{18}\right) Fup_2\left(\frac{x}{4\Delta x} - \frac{k^*}{4}\right) \tag{A.13}$$

where $k^* = k-1$. Other higher-order monomials are expressed approximately. Differences between monomials and their $Fup_2(x)$ approximation are defined by residual functions. Fig. A.2 shows a distribution of uniformly displaced $Fup_2(x)$ basis functions for the approximation of monomials in $[0,1]$ and $m=10$. Note that basis functions defined for $k=0$ and $k=m$ are external basis functions, while other functions are internal basis functions. Their influence is only considered within the domain $[0,1]$. Higher-order monomials up to the 8-order can be presented by a collocation procedure [8] as follows

$$x^3 = \frac{\Delta x^3}{4}\sum_{k=0}^{m}\left(k^{*3} - \frac{5}{6}k^*\right) Fup_2\left(\frac{x}{4\Delta x} - \frac{k^*}{4}\right) + \varepsilon_3(x) \tag{A.14}$$

$$x^4 = \frac{\Delta x^4}{4}\sum_{k=0}^{m}\left(k^{*^4} - \frac{5}{3}k^{*^2} + \frac{5}{27}\right) Fup_2\left(\frac{x}{4\Delta x} - \frac{k^*}{4}\right) + \varepsilon_4(x) \tag{A.15}$$

$$x^5 = \frac{\Delta x^5}{4}\sum_{k=0}^{m}\left(k^{*^5} - \frac{25}{9}k^{*^3} + \frac{25}{27}k^*\right) Fup_2\left(\frac{x}{4\Delta x} - \frac{k^*}{4}\right) + \varepsilon_5(x) \tag{A.16}$$

$$x^6 = \frac{\Delta x^6}{4}\sum_{k=0}^{m}\left(k^{*^6} - \frac{25}{6}k^{*^4} + \frac{25}{9}k^{*^2} + \frac{35}{324}\right) Fup_2\left(\frac{x}{4\Delta x} - \frac{k^*}{4}\right) + \varepsilon_6(x) \tag{A.17}$$

$$x^7 = \frac{\Delta x^7}{4}\sum_{k=0}^{m}\left(k^{*^7} - \frac{35}{6}k^{*^5} + \frac{175}{27}k^{*^3} + \frac{245}{324}k^*\right) Fup_2\left(\frac{x}{4\Delta x} - \frac{k^*}{4}\right) + \varepsilon_7(x) \tag{A.18}$$

$$x^8 = \frac{\Delta x^8}{4}\sum_{k=0}^{m}\left(k^{*^8} - \frac{70}{9}k^{*^6} + \frac{350}{27}k^{*^4} + \frac{245}{81}k^{*^2} - \frac{1865}{729}\right) Fup_2\left(\frac{x}{4\Delta x} - \frac{k^*}{4}\right) + \varepsilon_8(x) \tag{A.19}$$

Fig. 2 presents the residual functions $\varepsilon_3(x) - \varepsilon_8(x)$ for $m=10$. Their values are zero in the collocation points which are located in the vertices of basis functions [8,10]. Between collocation points, the nonzero residual functions are a considerably smaller than the monomials. Convergence of the iterative scheme (18) in presented Algorithm 2 mainly depends on the approximation properties of the basis functions. Smaller residual functions imply the faster convergence in Algorithm 2. Choice of the $Fup_2(x)$ basis functions seems reasonable due to their low order implying a stable algorithm, but still retains good approximation properties because they span the vector space which is closely related to the vector space of monomials. Accuracy of the collocation algorithm on the uniform grid has been shown in [13] by the following theorem:

Theorem 1. *If $f(x) \in C^n$, then exist Fup coefficients (d_{ij}) in* Eq. (17) *such that*

$$\|\varepsilon_i(x)\| \leqslant K_n \Delta x^{n-1} \omega_n(f;\Delta x); \quad i = 0,\ldots,m \tag{A.20}$$

where K_n is a real number independent of Δx, but depends on n-Fup order, while $\omega_n(f;\Delta x)$ is the modulus of continuity

$$\omega_n(f;\Delta x) = \max_{|x_0 - x_0^*| \leqslant \Delta x}\left|\frac{\partial^n}{\partial x^n}\left(f(x_0) - f(x_0^*)\right)\right| \tag{A.21}$$

It means that generally *Fup* approximation depends on three factors: n – *Fup order*, Δx and derivative differences of the *n-order* inside the Δx. In the limit when $m \to \infty$, then $\Delta x \to 0$ and consequently the derivative differences converge to zero, i.e. $\omega_n(f;\Delta x) \to 0$. It implies that for $m \to \infty$, the residual functions are zero and Eq. (17) becomes the exact relation (13) between *$Fup_2(x)$* basis functions and monomials.

References

[1] R. Abramov, A practical computational framework for the multidimensional moment-constrained maximum entropy principle, J. Comp. Phys. 211 (1) (2006) 198–209.
[2] R. Abramov, An improved algorithm for the multidimensional moment-constrained maximum entropy problem, J. Comp. Phys. 226 (2007) 621–644.
[3] R. Abramov, The multidimensional moment-constrained maximum entropy problem: a BFGS algorithm with constraint scaling, J. Comp. Phys. 228 (2009) 96–108.
[4] R. Andricevic, Exposure concentration statistics in the subsurface transport, Adv. Water Res. 31 (8) (2008) 714–725.
[5] A.L. Berger, S.A. Della Pietra, V.J. Della Pietra, A maximum entropy approach to natural language processing, Comp. Linguistics 22 (1) (1996) 39–71.
[6] K. Bandyopadhyay, A. Bhattacharya, P. Biswas, D. Drabold, Maximum entropy and the problem of moments. A stable algorithm, Phys. Rev. E 71 (5) (2005).
[7] G. Christakos, Modern Spatiotemporal Statistics, Oxford University Press, New York, 2000.
[8] B. Gotovac, V. Kozulić, On a selection of basis functions in numerical analyses of engineering problems, Int. J. Eng. Model. 12 (1–4) (1999) 25–41.
[9] B. Gotovac, V. Kozulić, Numerical solving the initial value problems by R_{bf} basis functions, Struct. Eng. Mech. 14 (3) (2002) 263–285.
[10] H. Gotovac, R. Andricevic, B. Gotovac, Multi-resolution adaptive modeling of groundwater flow and transport problems, Adv. Water Res. 30 (2007) 1105–1126.
[11] H. Gotovac, V. Cvetkovic, R. Andricevic, Adaptive Fup multi-resolution approach to flow and advective transport in highly heterogeneous porous media: methodology accuracy and convergence, Adv. Water Res. 32 (2009) 885–905.
[12] V.M. Kolodiazhny, V.A. Rvachev, Atomic functions: generalization to the multivariable case and promising applications, Cybern. Syst. Anal. 43 (6) (2007) 893–911.
[13] V.F. Kravchenko, M.A. Basarab, Approximations by atomic functions and numerical methods for Fredholm integral equations of the second kind, Differen. Equat. 37 (10) (2001) 1480–1490.
[14] V.F. Kravchenko, M.A. Basarab, H. Perez-Meana, Spectral properties of atomic functions used in digital signal processing, J. Commun. Technol. Electron. 46 (2001) 494–511.
[15] E.T. Jaynes, Information theory and statistical mechanics, Phy. Rev. 106 (1957) 620–630.
[16] L. Mead, N. Papanicolaou, Maximum entropy in the problem of moments, J. Math. Phys. 25 (1984) 2404–2417.
[17] D. Ormoneit, H. White, An efficient algorithm to compute maximum entropy densities, Econ. Rev. 18 (2) (1999) 127–140.
[18] V.L. Rvachev, V.A. Rvachev, Pro odnu finitnu funkciju, DAN URSR. Ser. A 6 (1971) 705–707 (in Russian).
[19] V.L. Rvachev, Teorija R-funkcij i nekotorija jeje priloženija (in Russian), Kiev: Naukova dumka (1982) 551.
[20] C.E. Shannon, Bell Syst. Tech. J. 27 (1948) 379–623. reprinted in C.E. Shannon and W. Weawer, The Mathematical Theory of Communication (University of Illinois Press, Urbana, 1949).
[21] W.-H. Steeb, F. Solms, R. Stoop, Chaotic systems and maximum entropy formalism, J. Phys. A: Math. Gen. 27 (1994) 399–402.
[22] Y.-K. Tung, B.-K. Yen, C.S. Melching, Hydrosystems Engineering Reliability Assessment and Risk Analysis, McGraw-Hill, New York, 2006. p. 495.
[23] I. Turek, A maximum-entropy approach to the density of states with the recursion method, J. Phys. C 21 (1988) 3251–3260.
[24] X. Wu, Calculation of maximum entropy densities with application to income distribution, J. Econ. 115 (2003) 347–354.