Chemical Engineering Science 75 (2012) 327–333

Contents lists available at SciVerse ScienceDirect

Chemical Engineering Science

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

On the efficiency and robustness of the core routine of the quadrature method of moments (QMOM)

Volker John [a,b,*], Ferdinand Thein [c]

[a] Weierstrass Institute for Applied Analysis and Stochastics, Leibniz Institute in Forschungsverbund Berlin e. V. (WIAS), Mohrenstr. 39, 10117 Berlin, Germany
[b] Free University of Berlin, Department of Mathematics and Computer Science, Arnimallee 6, 14195 Berlin, Germany
[c] Otto-von-Guericke University Magdeburg, Faculty of Mathematics, Institute for Analysis and Numerics, Postfach 4120, 39016 Magdeburg, Germany

ARTICLE INFO

Article history:
Received 4 January 2012
Received in revised form
19 March 2012
Accepted 20 March 2012
Available online 30 March 2012

Keywords:
Quadrature method of moments
Optimal quadrature rules
Product-difference algorithm
Long quotient-modified difference algorithm
Wheeler algorithm
Golub–Welsch algorithm

ABSTRACT

Three methods are reviewed for computing optimal weights and abscissas which can be used in the quadrature method of moments (QMOM): the product-difference algorithm (PDA), the long quotient-modified difference algorithm (LQMDA, variants are also called Wheeler algorithm or Chebyshev algorithm), and the Golub–Welsch algorithm (GWA). The PDA is traditionally used in applications. It is discussed that the PDA fails in certain situations whereas the LQMDA and the GWA are successful. Numerical studies reveal that the LQMDA is also more efficient than the PDA.

1. Introduction—the motivation for using the quadrature method of moments

Many processes in nature and industry involve particles. There are several ways for the numerical simulation of such processes. An individual modeling of the particles results generally in very complicated algorithms whose applicability is often restricted to a moderate number of particles. However, the behavior of individual particles is often not of interest in applications, but instead the average behavior of the particles. An approach with this objective consists in modeling the particles by a function called particle size distribution (PSD). This approach leads to so-called population balance systems that are often used as model, e.g., in chemical engineering (Ramkrishna, 2000) or in cloud physics (Rogers and Yau, 1996).

However, the numerical simulation of population balance systems is still challenging since the PSD f depends not only on time and space but also on properties of the particles, the so-called internal coordinates. Let us consider a model for a particulate process that takes into account the flow field (Navier–Stokes equations), balance laws for scalar quantities like energy or concentrations (system of scalar convection-diffusion equations), and an equation for the PSD, e.g., like in Hackbusch et al. (in press). Then, the flow field, energy, and concentrations depend on time and on the three-dimensional spatial coordinate, whereas the PSD depends additionally on the internal coordinate. After having discretized these equations in time, one has to solve in each discrete time equations for the flow field, energy, and concentrations that depend only on the spatial coordinate, while the equation for the PSD depends additionally on the internal coordinate. Hence, this equation is defined in a domain which is at least four-dimensional. In applications, currently the case of one internal coordinate (uni-variate PSD) is considered most often. This case will be studied in this note. A typical equation for f has the form, e.g., see Nayak et al. (2011):

$$\frac{\partial f(t,\mathbf{x},e)}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{u}(t,\mathbf{x}) f(t,\mathbf{x},e)) = s(t,\mathbf{x},e) - \frac{\partial (G(t,\mathbf{x},e) f(t,\mathbf{x},e))}{\partial e} \quad \text{in } (0,T) \times \Omega \times \Omega_e. \tag{1}$$

In (1), T is a final time, $\Omega \subset \mathbb{R}^3$ is a domain, $\Omega_e = (a,b)$, with $a < b$, often with $a \geq 0$, is the domain of the internal coordinate, $\mathbf{u}(t,\mathbf{x})$ is a velocity field, $s(t,\mathbf{x},e)$ is a source term, and $G(t,\mathbf{x},e)$ is the growth rate of the particles. In applications, (1) is defined in each discrete time in a four-dimensional domain. Hence, the solution of (1)

* Corresponding author at: Weierstrass Institute for Applied Analysis and Stochastics, Leibniz Institute in Forschungsverbund Berlin e. V. (WIAS), Mohrenstr. 39, 10117 Berlin, Germany. Tel.: +49 30 20372 561.
E-mail addresses: john@wias-berlin.de (V. John), ferdinand.thein@st.ovgu.de (F. Thein).

0009-2509/$ - see front matter
http://dx.doi.org/10.1016/j.ces.2012.03.024

might be time-consuming, although meanwhile some methods for solving (1) with direct discretizations can be found in the literature (Kulikov et al., 2005; John and Roland, 2010; Hackbusch et al., in press). Moreover, available software generally does not support equations in more than three dimensions.

For reasons like these, it was proposed in Hulburt and Katz (1964) to replace (1) by a system for the first moments of f, where the k-th moment is defined by $\int_a^b e^k f(t,\mathbf{x},e)\,de$. This approach is called method of moments (MOM). Multiplying (1) by e^k and integrating over Ω_e leads to

$$\frac{\partial m_k(t,\mathbf{x})}{\partial t}+\nabla_{\mathbf{x}}\cdot(\mathbf{u}(t,\mathbf{x})m_k(t,\mathbf{x}))$$
$$=\int_a^b e^k s(t,\mathbf{x},e)\,de+\int_a^b g(t,\mathbf{x},e)f(t,\mathbf{x},e)\,de \quad \text{in } (0,T)\times\Omega, \tag{2}$$

$k=0,1,2,\ldots$. The derivation of (2) applied integration by parts for the growth term where $\lim_{e\to a+0}f(t,\mathbf{x},e)=\lim_{e\to b-0}f(t,\mathbf{x},e)=0$ was assumed. The modified growth function is given by $g(t,\mathbf{x},e)=ke^{k-1}G(t,\mathbf{x},e)$. In this way, a system for the moments is obtained. On the one hand, the first moments are often of importance in practice since they correspond to physical quantities, like to the number of particles (0-th moment) or to their volume (3-rd moment). But on the other hand, the reconstruction of f from its first moments is generally an ill-posed problem and only few numerical schemes are available for this purpose (Alopaeus et al., 2008; John et al., 2007; de Souza et al., 2010). Since the unknown PSD still appears in the second term on the right hand side of (2), system (2) is not yet closed. A direct closure can be obtained only for some special growth functions, see Hulburt and Katz (1964).

The rise of moment-based methods started with the proposal of the quadrature method of moments (QMOM) in McGraw (1997). The idea of QMOM consists in replacing the second term on the right hand side of (2) by a quadrature formula:

$$\int_a^b g(t,\mathbf{x},e)f(t,\mathbf{x},e)\,de\approx\sum_{i=1}^n g(t,\mathbf{x},e_i)w_i(t,\mathbf{x}), \tag{3}$$

where e_i denote the quadrature points (abscissas) and w_i are the weights. In order to keep the quadrature error as small as possible, the abscissas and the weights should be chosen such that the optimal order $(2n-1)$ of the numerical quadrature is obtained.

This note will start by shortly reviewing the derivation of optimal-order quadrature rules. It turns out that in essence an eigenvalue problem with a symmetric tridiagonal matrix has to be solved whose coefficients have to be computed efficiently. In McGraw (1997), the product-difference algorithm (PDA) from Gordon (1968) was proposed for the computation of the coefficients. To our best knowledge, this algorithm has been used most often since then in combination with the QMOM. In this note, two alternatives to the PDA will be studied: the long quotient-modified difference algorithm (LQMDA) from Sack and Donovan (1972) and the Golub–Welsch (1969) algorithm (GWA). Variants of implementing the LQMDA are also called Wheeler algorithm (1974) and Chebyshev algorithm (Upadhyay, 2012). The advantages and drawbacks of the algorithms will be discussed. Numerical examples will be presented which compare primarily the efficiency of the considered algorithms. Also the robustness of the algorithms with respect to the number of moments is addressed and observations recently reported in Upadhyay (2012) will be supported. The note concludes with a summary.

2. Optimal-order quadrature rules

2.1. General approach

For simplicity of notation, the dependency of the functions on the spatial variable $\mathbf{x}$ and on the time t will be suppressed henceforth. The goal consists in defining the weights w_i and the abscissas e_i of the quadrature rule

$$\int_a^b g(e)f(e)\,de\approx\sum_{i=1}^n g(e_i)w_i \tag{4}$$

in such a way that if $g(e)$ is a polynomial of degree less or equal than $(2n-1)$, then the quadrature is exact. In (4), the function $g(e)$ is known. It will be assumed that

- f is measurable and non-negative in (a,b),
- the moments m_k, $k=0,1,\ldots$, of f exist and are finite,
- for all polynomials $p(e)\geq 0$ in $[a,b]$ with $\int_a^b p(e)f(e)\,de=0$ it follows that $p(e)\equiv 0$.

Then, f is called weight function. If the PSD is represented by a continuous function with non-negative values, these conditions are met.

The derivation of optimal-order quadrature rules will reveal that the complete knowledge of f is not necessary. It will be sufficient to know the first $2n$ moments of f. In practice, the moments computed in the previous discrete time can be used for this purpose.

Starting point of deriving optimal-order quadrature rules for (4) is the definition of an inner product which is induced by the weight function:

$$\langle p,q\rangle := \int_a^b p(e)q(e)f(e)\,de. \tag{5}$$

In the next step, one considers orthogonal polynomials $\{p_k\}_{k=0}^n$ with respect to the inner product $\langle\cdot,\cdot\rangle$ with degree$(p_k)=k$. The polynomials $\{p_k\}_{k=0}^n$ are normalized such that the coefficient in front of the term with the highest power is 1. It can be shown that a necessary condition for (4) being of order $(2n-1)$ is that the abscissas are the roots of the n-th order orthogonal polynomial $p_n(e)$.

For this reason, an efficient way for computing the roots of $p_n(e)$ is necessary. To this end, the recursion property of orthogonal polynomials is used:

$$p_{-1}(e)=0,\quad p_0(e)=1,$$

$$p_{k+1}(e)=(e-\beta_k)p_k(e)-\alpha_k^2 p_{k-1}(e),\quad k=0,1,\ldots \tag{6}$$

with the coefficients

$$\beta_k=\frac{\langle ep_k,p_k\rangle}{\langle p_k,p_k\rangle},\quad k\geq 0,\quad \alpha_k^2=\begin{cases}1, & k=0,\\ \dfrac{\langle p_k,p_k\rangle}{\langle p_{k-1},p_{k-1}\rangle}, & k=1,\ldots.\end{cases}$$

Note that, given $p_0,\ldots,p_{n-1}$, the coefficients β_{n-1} and α_{n-1}^2 can be computed by knowing the first $2n$ moments of f. Now, p_n can be computed by (6).

A simple rewriting of the three-term recursion (6) up to $k=n-1$ leads to the representation of (6) by a linear system of equations:

$$(\tilde{A}_n-eI)\begin{pmatrix}p_0(e)\\ \vdots\\ p_{n-1}(e)\end{pmatrix}=\begin{pmatrix}0\\ \vdots\\ 0\\ -p_n(e)\end{pmatrix}\quad \text{with}$$

$$\tilde{A}_n = \begin{pmatrix} \beta_0 & 1 & 0 & \dots & \dots & 0 \\ \alpha_1^2 & \beta_1 & 1 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & \alpha_{n-2}^2 & \beta_{n-2} & 1 \\ 0 & \dots & \dots & 0 & \alpha_{n-1}^2 & \beta_{n-1} \end{pmatrix}. \tag{7}$$

From (7) it can be seen that for the roots e_i of $p_n(e)$ the right hand side is homogeneous. That means, e_i is a root of $p_n(e)$ if and only if e_i is an eigenvalue of $\tilde{A}_n$.

The eigenvalue problem can be even converted to an eigenvalue problem for a symmetric matrix, which is preferable from the numerical point of view. Defining the diagonal matrix $D=(d_i)_{i=0}^{n-1}$ with $d_0=1$, $d_i=(\alpha_1 \cdots \alpha_i)^{-1}$, $i=1,\dots,n-1$, then (7) can be transformed for $e=e_i$ into

$$(A_n - eI)\begin{pmatrix} \hat{p}_0(e) \\ \vdots \\ \hat{p}_{n-1}(e) \end{pmatrix} = 0 \quad \text{with}$$

$$A_n = D\tilde{A}_n D^{-1} = \begin{pmatrix} \beta_0 & \alpha_1 & 0 & \dots & \dots & 0 \\ \alpha_1 & \beta_1 & \alpha_2 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & \alpha_{n-2} & \beta_{n-2} & \alpha_{n-1} \\ 0 & \dots & \dots & 0 & \alpha_{n-1} & \beta_{n-1} \end{pmatrix} \tag{8}$$

and, using the definition of d_k,

$$\hat{p}_k(e_i) = d_k p_k(e_i) = \frac{\langle p_0, p_0 \rangle^{1/2}}{\langle p_k, p_k \rangle^{1/2}} p_k(e_i), \quad k=0,\dots,n-1. \tag{9}$$

Let $\{\overline{p}_k\}_{k=0}^{n-1}$ be an orthogonal set of polynomials which are normalized such that $\langle \overline{p}_k, \overline{p}_k \rangle = 1$, $k=0,\dots,n-1$. Then it follows from the Christoffel–Darboux formula that the weights are given by

$$w_i = \left(\sum_{k=0}^{n-1} \overline{p}_k^2(e_i) \right)^{-1}, \quad i=1,\dots,n. \tag{10}$$

Normalizing $\{p_k\}_{k=0}^{n-1}$ and using (9) gives

$$\overline{p}_k(e_i) = \frac{p_k(e_i)}{\langle p_k, p_k \rangle^{1/2}} = \frac{\hat{p}_k(e_i)}{\langle p_0, p_0 \rangle^{1/2}}. \tag{11}$$

Let $\mathbf{q}=(q_{i0},\dots,q_{i,n-1})^T$ be an eigenvector to the eigenvalue e_i which is, e.g., computed by a numerical method. Since $\hat{p}_0(e_i)=1$, it follows that the eigenvector $(\hat{p}_0(e_i),\dots,\hat{p}_{n-1}(e_i))^T$ is q_{i0}^{-1} times $\mathbf{q}$. Using (10) and (11), one obtains

$$w_i = \left(\sum_{k=0}^{n-1} \frac{\hat{p}_k^2(e_i)}{\langle p_0, p_0 \rangle} \right)^{-1} = \langle p_0, p_0 \rangle q_{i0}^2 \left(\sum_{k=0}^{n-1} q_{ik}^2 \right)^{-1}$$
$$= m_0 q_{i0}^2 \left(\sum_{k=0}^{n-1} q_{ik}^2 \right)^{-1}, i=1,\dots,n$$

In summary, the complete information concerning the abscissas and the weights of (4) are obtained from the solution of the eigenvalue problem (8).

It remains to find an efficient and stable algorithm for computing the entries of A_n. By the definition of the inner product $\langle \cdot, \cdot \rangle$ it follows that the entries depend on the first $2n$ moments of f. In the application of the QMOM, these moments are generally different in each spatial point and they generally change in each time step. Thus, the algorithm for computing the coefficients of A_n has to be applied over and over again.

For the remainder of this note, it will be assumed that so-called valid or realizable sets of moments $\{m_i\}_{i=0}^{2n}$ or $\{m_i\}_{i=0}^{2n+1}$ with $m_0 \neq 0$ are given. A set of moments is called valid or realizable if there exists a function f such that $\{m_i\}$ are the moments of f. Results concerning the existence and uniqueness of a solution of this so-called truncated Hausdorff moment problem can be found in Curto and Fialkow (1991). It is well known that invalid sets of moments can be obtained, e.g., in numerical simulations of transport-dominated equations for the moments (Wright, 2007). From the first property of the weight function it follows that $m_0 \geq 0$. However, the case $m_0=0$ contradicts the last property for $p(e) \equiv 1$. Hence, it can be assumed even that $m_0 > 0$.

2.2. The product-difference algorithm

The PDA was introduced by Gordon (1968). In the first step of this algorithm, a matrix $B=(b_{ij}) \in \mathbb{R}^{2n \times (2n+1)}$ is initialized. The elements of the first and second column are set as follows

$$b_{i1} = \delta_{i1}, \quad b_{i2} = (-1)^{i-1} m_{i-1}, \quad i=1,\dots,2n,$$

where δ_{ij} is the Kronecker delta. The other components are obtained by applying the following product-difference recursion formula:

$$b_{ij} = \begin{cases} b_{1,j-1} b_{i+1,j-2} - b_{1,j-2} b_{i+1,j-1}, & j=3,\dots,2n+1, \\ & i=1,\dots,2n+2-j, \\ 0, \quad \text{else.} & \end{cases} \tag{12}$$

The resulting matrix has the form

$$B = \begin{pmatrix} 1 & m_0 & b_{13} & \dots & \dots & b_{1,2n+1} \\ 0 & -m_1 & b_{23} & \dots & b_{2,2n} & 0 \\ \vdots & \vdots & \vdots & \iddots & \iddots & \vdots \\ \vdots & m_{2n-2} & b_{2n-1,3} & 0 & & \vdots \\ 0 & -m_{2n-1} & 0 & 0 & \dots & 0 \end{pmatrix}.$$

In the next step, coefficients c_i are determined by

$$c_i = \begin{cases} m_0, & i=1, \\ \dfrac{b_{1,i+1}}{b_{1i} b_{1,i-1}}, & i=2,\dots,2n. \end{cases}$$

With these coefficients, one can compute the entries of A_n in the following way:

$$\beta_{i-1} = \begin{cases} c_2, & i=1, \\ c_{2i}+c_{2i-1}, & i=2,\dots,n, \end{cases} \quad \alpha_i = \sqrt{c_{2i+1} c_{2i}}, \quad i=1,\dots,n-1.$$

The derivation of the PDA is based on the study of the integral (Stieltjes transform)

$$I(z) = \int_0^\infty \frac{f(e)}{z+e} \, de,$$

where $f(z)$ is a weight function such that $I(z)$ is finite, see Gordon (1968). From the properties of a weight function given above, it follows that z does not belong to the domain of integration, because otherwise the function in the integral is singular for $e=-z>0$ and the integral itself is not well defined.

In the first step, the term $(z+e)^{-1}$ is expanded into a formal series with respect to z^{-1}, which is in the next step replaced by a continued fraction. The coefficients which appear in the PDA are determined by comparing the continued fraction and the formal series. Then, it can be shown that from the continued fraction the coefficients of the eigenvalue problem (8) can be derived (Gordon, 1968; Wall, 1948). In this way, one obtains for all z a quadrature formula for $I(z)$ with weights and abscissas independent of z:

$$I(z) \approx \sum_{i=1}^{n} \frac{w_i}{z+e_i}.$$

It can be concluded, using connections between Stieltjes transforms and optimal quadrature rules that one can use the same weights and abscissas if instead $(z+e_i)^{-1}$ any other function

$g(e)$ in the integral is given, provided $g(e)$ is analytic on the positive real axis.

It should be noted that the PDA might fail if the condition $a \geq 0$ is not fulfilled. A simple example is the consideration of Gauss–Legendre quadrature, i.e. $f(e)=1$, in $(-1,1)$ with four moments $m_0=2,\ m_1=0,\ m_2=2/3,\ m_3=0$. The well-known data of the quadrature formula are $e_i=\pm 1/\sqrt{3}$, $w_i=1$, $i=1,2$. With the PDA one obtains

$$B=\begin{pmatrix} 1 & 2 & b_{13} & b_{14} & b_{15} \\ 0 & 0 & b_{23} & b_{24} & 0 \\ 0 & \frac{2}{3} & b_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \begin{array}{lcl} b_{13} & = & b_{12}b_{21}-b_{11}b_{22}=0, \\ b_{23} & = & b_{12}b_{31}-b_{11}b_{32}=-\frac{2}{3}, \\ b_{14} & = & b_{13}b_{22}-b_{12}b_{23}=\frac{4}{3}, \\ & \vdots & \end{array}$$

Now, it follows that

$$c_1=2, \quad c_2=\frac{b_{13}}{b_{12}b_{11}}=0, \quad c_3=\frac{b_{14}}{b_{13}b_{12}}.$$

Here, the algorithm breaks down because of a division by zero. The same effect can be observed if one considers Gauss–Hermite quadrature, i.e. $f(e)=\exp(-e^2)$ in $(-\infty,\infty)$, with four moments.

The main reason for the failing of the PDA in these examples is the vanishing of a moment. If the internal coordinate is assumed to be non-negative, all moments are positive. However, due to discretization errors in solving the equations for the moments or because of round-off errors, computed moments might vanish or at least might become very small. In this case, a potential instability of the PDA cannot be excluded. The observation that the QMOM with the PDA might lead to an ill-conditioned problem is not new, e.g., see the discussion of this topic in Dorao and Jakobsen (2006), Upadhyay (2012). Further sources of instability of the PDA are highlighted in Section 3.

2.3. Long quotient-modified difference algorithm

The LQMDA was proposed by Sack and Donovan (1972). This algorithm can be applied for the problem of finding optimal abscissas and weights if the so-called modified moments:

$$\mu_l=\int_a^b P_l(x)f(x)\,dx, \quad l=0,1,\ldots$$

are given. In this formula, P_l are polynomials of degree l satisfying a three term recurrence relation with known coefficients:

$$xP_l(x)=a_lP_{l+1}(x)+b_lP_l(x)+c_lP_{l-1}, \quad l=0,1,\ldots.$$

The standard moments are recovered for $a_l=1$, $b_l=c_l=0$.

For the sake of brevity, only the algorithm for the standard moments will be presented here. The LQMDA is initialized by

$$s_{-1,j}=0, \quad s_{0,j}=\frac{m_j}{m_0}, \quad j=0,\ldots,2n-1.$$

Note that $s_{0j}>0$ if j is even, since $m_j>0$ for j even. Then, the following recursion can be computed

$$\sigma_i=s_{i,i+1}-s_{i-1,i}, \quad i=0,\ldots,n-1,$$

$$\rho_i=-\sigma_i s_{i,i+1}+s_{i,i+2}-s_{i-1,i+1}, \quad i=0,\ldots,n-2,$$

$$s_{i+1,j}=\rho_i^{-1}(-\sigma_i s_{i,j}+s_{i,j+1}-s_{i-1,j}), \quad \begin{array}{l} i=0,\ldots,n-2, \\ j=i+2,\ldots,2n-2-i. \end{array}$$

With these values, the coefficients of the matrix A_n are given by

$$\beta_i=\sigma_i, \quad i=0,\ldots,n-1, \quad \alpha_{i+1}=\sqrt{\rho_i}, \quad i=0,\ldots,n-2.$$

It was noted by Sack and Donovan (1972) that for the standard moments the LQMDA is essentially equivalent to an algorithm given already in Wall (1948). An alternative implementation of the LQMDA is presented by Wheeler (1974), see also Press et al. (1992). This implementation differs from the algorithm given above by considering intermediate quantities σ_{ij} instead of s_{ij}, where the relation between these quantities is given by $s_{ij}=\sigma_{ij}/\sigma_{ii}$. This so-called Wheeler algorithm was already used within the QMOM, e.g., in Fox (2009) and Yuan and Fox (2011). A slight modification of implementing the Wheeler algorithm is called Chebyshev algorithm, proposed by Upadhyay (2012), where the relation of intermediate quantities A_{ij} to s_{ij} is the same as in the Wheeler algorithm $s_{ij}=A_{ij}/A_{ii}$.

The LQMDA is based on a reformulation of the eigenvalue problem (8), see Sack and Donovan (1972). First, the recurrence of the orthogonal polynomials (6) is rewritten in the form:

$$ep_k(e)=\alpha_k p_{k+1}(e)+\beta_k p_k(e)+\alpha_{k-1}p_{k-1}(e), \quad k=0,1,\ldots,$$

where also a different normalization is used. Now, the entries of A_n can be expressed with the new set of orthogonal polynomials. With these expressions and some algebraic manipulations, a new eigenvalue problem is derived from (8). With a comparison of the trace of the matrix of the new eigenvalue problem with the trace of A_n, and the traces of the squares of both matrices, a recursion formula for the coefficients of A_n is derived.

The coefficients ρ_i in the LQMDA appear in the definition of a certain set of orthogonal polynomials which is connected to other sets of orthogonal polynomials, see Sack and Donovan (1972). A close inspection of the derivation reveals that, with appropriate normalizations of the former set of orthogonal polynomials, always $\rho_i>0$ holds. Hence, the LQMDA is well defined and there are no restrictions on the interval (a,b) and the values of the moments as for the PDA. For instance, straightforward calculations reveal that the abscissas and weights of the Gauss–Legendre quadrature in $(-1,1)$ and the Gauss–Hermite quadrature in $(-\infty,\infty)$ can be computed with the LQMDA. In this respect, the LQMDA is expected to be more stable than the PDA with respect to errors coming from numerical approximations and round-off errors.

2.4. The Golub–Welsch algorithm

The last algorithm which will be considered was proposed by Golub and Welsch (1969). This algorithm needs $(2n+1)$ moments for computing the weights and abscissas of (4).

The required moments are arranged in a matrix of the form:

$$M=\begin{pmatrix} m_0 & m_1 & m_2 & \ldots & m_n \\ m_1 & m_2 & & \iddots & \\ m_2 & & \iddots & & \vdots \\ \vdots & \iddots & & & \\ m_n & & \ldots & & m_{2n} \end{pmatrix} \in \mathbb{R}^{(2n+1)\times(2n+1)}. \tag{13}$$

This symmetric matrix is called Hankel matrix. Since M is the Gramian matrix of the inner product (5), it is even a positive definite matrix. A symmetric and positive definite matrix allows a Cholesky decomposition $M=R^TR$, where R is an upper triangular matrix with the entries:

$$r_{ii}=\left(M_{ii}-\sum_{k=1}^{i-1}r_{ki}^2\right)^{1/2}, \quad i=1,\ldots,n+1,$$

$$r_{ij}=\frac{M_{ij}-\sum_{k=1}^{i-1}r_{ki}r_{kj}}{r_{ii}}, \quad i<j,\ j=1,\ldots,n+1, \tag{14}$$

and $M_{ij}=m_{i+j-2}$ for $i,j=1,\ldots,n+1$. Given the Cholesky decomposition, one can compute the entries of the matrix A_n in (8) via

$$\beta_{j-1}=\frac{r_{j,j+1}}{r_{jj}}-\frac{r_{j-1,j}}{r_{j-1,j-1}}, \quad j=1,\ldots,n,$$

$$\alpha_j = \frac{r_{j+1,j+1}}{r_{jj}}, \quad j=1,\ldots,n-1, \tag{15}$$

with $r_{00}=1$ and $r_{01}=0$.

Since R is an upper triangular matrix, its inverse has the form:

$$R^{-1} = \begin{pmatrix} s_{11} & s_{12} & \ldots & s_{1,n+1} \\ 0 & s_{22} & \ldots & s_{2,n+1} \\ \vdots & & \ddots & \vdots \\ 0 & \ldots & 0 & s_{n+1,n+1} \end{pmatrix}.$$

For the derivation of the GWA in Golub and Welsch (1969), it was used that the polynomials

$$p_{j-1}(x) = \sum_{i=1}^{j} s_{ij} x^{i-1}, \quad j=1,\ldots,n+1,$$

form an orthonormal system and hence satisfy the three term recurrence relation:

$$xp_{j-1}(x) = \alpha_{j-1}p_{j-2}(x) + \beta_{j-1}p_{j-1}(x) + \alpha_j p_j(x), \quad j=1,\ldots,n, \tag{16}$$

with $p_{-1}(x)=0$ and $p_0(x)=1$. Comparing the coefficients of the two highest powers x^j and x^{j-1} on both sides of this identity results in

$$s_{jj} = \alpha_j s_{j+1,j+1}, \quad s_{j-1,j} = \beta_j s_{jj} + \alpha_j s_{j,j+1}, \quad j=1,\ldots,n.$$

It follows that

$$\alpha_j = \frac{s_{jj}}{s_{j+1,j+1}}, \quad \beta_j = \frac{s_{j-1,j}}{s_{jj}} - \frac{s_{j,j+1}}{s_{j+1,j+1}}, \quad j=1,\ldots,n.$$

One obtains (15) by expressing the entries of R^{-1} with the entries of R. These expressions can be calculated explicitly.

Note that there are close connections between Hankel matrices and continued fractions. The coefficients of a continued fraction can be determined via certain determinants of Hankel matrices, see Gordon (1968) and Wall (1948).

The additional moment which is needed in the GWA can be obtained in the QMOM in the following way. At the initial time, m_{2n} can be computed from the initial data. Then, the weights and abscissas for the initial time are computed. In all other discrete times, the weights w_i and the abscissas e_i, $i=1,\ldots,n$, from the previous discrete time are available. Since m_{2n} is defined as an integral on (a,b), it can be approximated by a quadrature rule as follows:

$$m_{2n} := \sum_{i=1}^{n} e_i^{2n} w_i. \tag{17}$$

With this approach, we could observe in numerical studies that the matrix M might be not positive definite. Concretely, the used MATLAB routine `chol` for the Cholesky decomposition returned a warning. However, the formulas (14) could still be used and the algorithm could be performed. At any rate, one should be aware of this potential instability of the GWA.

Another source of instability might be induced by the fact that in practice the moments are not computed from (5) but using some discretization of (2). Due to discretization errors, (5) and even the positive definiteness of the Hankel matrix M might be violated. This situation leads to a so-called non-realizable set of moments since it is known Curto and Fialkow (1991) that for a set of moments to be realizable the positive semi-definiteness of the Hankel matrix is a necessary condition.

3. Numerical studies

Two numerical studies on the efficiency of applying the PDA, LQMDA, and GWA for computing the weights and abscissas for the QMOM are presented. In the first study, seven problems are considered which were already used in the literature. The second study assesses also the robustness of the methods with respect to an increasing number of moments. In addition, the robustness in the situation that $m_0(t)\to 0$ as $t\to\infty$ is investigated.

With respect to the LQMDA, we implemented the variant presented in Section 2.3 and also the two variants called Wheeler and Chebyshev algorithm. All implementations proved to be equally robust and the computing times among these variants differed only marginally. For the sake of brevity, only results for the algorithm described in detail in Section 2.3 will be presented here. All conclusions for this algorithm can be transferred literally to the Wheeler and Chebyshev algorithm.

In the first numerical study, seven problems are studied for a fixed number of moments and a fixed time interval. For the sake of brevity, the used problems will be described only shortly. All problems are defined for the case of ideal mixing, i.e. the functions in (1) do not depend on $\mathbf{x}$. Problems I–III consider the growth of particles

$$\begin{cases} \dfrac{\partial f(t,e)}{\partial t} = -\dfrac{\partial}{\partial e}(\phi_i(e)f(t,e)), & (t,e)\in(0,T]\times(0,\infty), \\ f(0,e) = ae^2\exp(-be), & e\in(0,\infty), \end{cases} \tag{18}$$

with

$$\phi_1(e)=\beta, \quad \phi_2(e)=\beta e, \quad \phi_3(e)=\frac{\beta}{e}. \tag{19}$$

In particular, Problem III (diffusion-controlled growth) is discussed in detail in McGraw (1997). We used for (18) and (19) the same parameters as in McGraw (1997): $a=0.108$, $b=0.6$, $\beta=0.78$.

In Problems IV–VII, terms for the coalescence and breakage of particles appear in the equation for the particle size distribution:

$$\frac{\partial f(t,e)}{\partial t} = \frac{1}{2}\int_0^e f(t,e-e')f(t,e')\,de' - \int_0^\infty f(t,e)f(t,e')\,de' + 2\sigma\int_e^\infty f(t,e')\,de' - \sigma e f(t,e), \quad (t,e)\in(0,T]\times(0,\infty). \tag{20}$$

The following initial conditions were used:

$$f(0,e) = \begin{cases} \exp(-e), & \text{Problems IV}-\text{VI} \\ 4e\exp(-2e), & \text{Problem VII}, \end{cases} \quad e\in(0,\infty).$$

Problems of this type were studied in Patil and Andrews (1998), Lage (2002) and McCoy and Madras (2003). In (20), σ is the fragmentation rate. This rate is given by $\sigma=\Phi^2(\infty)/2$, where $\Phi(\infty)$ is a constant which represents the total number of particles in an asymptotic state of the system. The following situations were considered in our numerical studies:

$$\begin{cases} \text{Problem IV}: & \Phi(\infty)=0.1, & \text{number of particles decreases}, \\ \text{Problem V}: & \Phi(\infty)=5, & \text{number of particles increases}, \\ \text{Problem VI, VII}: & \Phi(\infty)=1, & \text{number of particles stays constant}. \end{cases}$$

The solution of all problems can be computed analytically such that the accuracy of numerical results can be assessed. For all

Table 1
Average computing times in seconds for performing the QMOM with different methods for computing the coefficients of A_n, $n=3$, $T=10$.

Problem	PDA	LQMDA	GWA
I	0.511	0.497	0.551
II	0.510	0.498	0.551
III	0.510	0.497	0.550
IV	0.431	0.419	0.475
V	0.434	0.420	0.474
VI	0.434	0.420	0.474
VII	0.433	0.419	0.474

Table 2
Average computing times in seconds for performing the QMOM with different methods for computing the coefficients of A_n for different values of n and different lengths of the time interval, Problem IV.

n	$T=0.1$			$T=1$			$T=100$		
	PDA	LQMDA	GWA	PDA	LQMDA	GWA	PDA	LQMDA	GWA
1	2.28e−3	2.18e−3	2.74e−3	2.23e−2	2.26e−2	2.92e−2	1.98	1.92	2.45
2	3.47e−3	3.36e−3	3.83e−3	3.67e−2	3.49e−2	3.92e−2	3.11	3.04	3.49
3	4.62e−3	4.53e−3	5.14e−3	4.72e−2	4.58e−2	5.15e−2	4.23	4.14	4.67
4	5.92e−3	5.78e−3	6.46e−3	6.03e−2	5.90e−2	6.60e−2	5.43	5.32	5.94
5	7.33e−3	7.15e−3	7.92e−3	7.58e−2	7.40e−2	8.21e−2	6.77	6.63	7.34
6	8.94e−3	8.76e−3	9.69e−3	9.24e−2	8.97e−2	9.93e−2	8.28	8.14	8.96
7	10.82e−3	10.64e−3	11.71e−3	11.15e−2	10.79e−2	11.96e−2	NaN	9.87	10.82
8	12.97e−3	12.73e−3	13.96e−3	NaN	13.01e−2	14.19e−2	NaN	11.86	12.93
9	NaN	15.22e−3	16.56e−3	NaN	15.40e−2	16.75e−2	NaN	14.16	15.34
10	NaN	18.09e−3	19.61e−3	NaN	17.83e−2	19.34e−2	NaN	16.72	18.07
12	NaN	24.87e−3	26.77e−3	NaN	24.65e−2	26.45e−2	NaN	23.16	24.84
14	NaN	33.83e−3	36.00e−3	NaN	33.06e−2	35.46e−2	NaN	31.24	33.33
16	NaN	44.48e−3	47.25e−3	NaN	43.15e−2	45.87e−2	NaN	41.44	43.91

problems, one obtains basically the same results for the weights and abscissas with all considered methods.

In all examples, the first six moments were used, i.e. $n=3$. For the temporal discretization, the classical explicit Runge–Kutta scheme of fourth order was applied. All examples were computed in the time interval [0,10] with a time step of length 0.01. The simulations were carried out with MATLAB, version 7.12.0 on a HP BL2x220c computer with Xeon 2933 MHz processors. The running times were measured with the MATLAB commands `tic` and `toc`. For each problem and for each method, 10 000 runs were performed and the execution times were averaged.

The results of the first computational study are presented in Table 1. It can be clearly seen that the QMOM with LQMDA is somewhat more efficient than the QMOM with PDA and that the slowest method is the QMOM with GWA. This behavior can be explained to some extend by the number of floating point operations which each method requires. Counting these operations, one finds that the LQMDA needs the smallest number among the considered methods. However, memory access is nowadays often more time-consuming than performing floating point operations. Nevertheless, since each floating point operation requires memory accesses, the number of floating point operations gives still a certain idea on the efficiency of a method. The relatively large computing times of the GWA seem to come from computing m_{2n} with (17) and from the larger number of square root evaluations compared with the PDA and the LQMDA.

The second numerical study considers the robustness of the methods. To this end, Problem IV is used. A main feature of this problem is that $m_0(t)$ decreases monotonically with $m_0(t)\to 0$ as $t\to\infty$. Since all considered methods fail in the case $m_0=0$, this behavior is a potential source of instability. Hence, Problem IV was simulated for different lengths of the time interval. Another potential source of instability is an increase of the number of moments, see Upadhyay (2012). In Upadhyay (2012), several examples are mentioned where it is essential to use a large number of moments for obtaining accurate results. Information on the success of the simulations and on the computing times are provided in Table 2. The length of the time step was chosen to be 0.01 in all simulations. In Table 2, the computing times are the averages of 1000 runs.

With respect to efficiency, the LQMDA was generally the best method. The slowest method was always the GWA. With respect to robustness, it can be seen that the PDA fails if the number of moments increases. In addition the admissible number of moments for the PDA decreases with an increasing length of the time interval. Detailed studies on the robustness of the PDA and the Chebyshev algorithm for problems coming from aerosol dynamics were published recently in Upadhyay (2012). The results presented in Table 2 supplement those studies very well. Upadhyay (2012) pointed out also the reason for the weak robustness of the PDA, namely that products of moments are computed, see (12), which easily lead to an overflow if a large number of moments are used. Altogether, the observations with respect to the robustness of the PDA and of the LQMDA (Chebyshev algorithm) from Upadhyay (2012) and of the studies presented here coincide. For the considered example, the GWA has shown to be equally robust as the LQMDA.

4. Summary

This note reviewed three numerical methods which can be applied for computing optimal weights and abscissas for the quadrature rule (4). These computations are of crucial importance for the QMOM.

It was shown that the traditionally used PDA is somewhat less efficient than the LQMDA. In addition, the PDA is less robust in some situation that might be important in applications. This observation is in agreement with recently published results (Upadhyay, 2012). The GWA is less efficient than the two other methods but in the considered example it was equally robust as the LQMDA.

In summary, based on the observations presented in this note and the results from Upadhyay (2012), we strongly recommend the use of the LQMDA, or one of its variants called Wheeler algorithm or Chebyshev algorithm, for computing the optimal weights and abscissas within the QMOM.

The case of multivariate PSDs has still to be studied.

Acknowledgments

We would like to acknowledge an unknown referee whose suggestions greatly helped to improve this note.

References

Alopaeus, V., Laakkonen, M., Aittamaa, J., 2008. Solution of population balances by high order moment-conserving method of classes: reconstruction of a non-negative density distribution. Chem. Eng. Sci. 63, 2741–2751.

Curto, R.E., Fialkow, L.A., 1991. Recursiveness, positivity, and truncated moment problems. Houston J. Math. 17 (4), 603–635.

de Souza, L.G.M., Janiga, G., John, V., Thévenin, D., 2010. Reconstruction of a distribution from a finite number of moments with an adaptive spline-based algorithm. Chem. Eng. Sci., 2741–2750.

Dorao, C.A., Jakobsen, H.A., 2006. Numerical calculation of the moments of the population balance equation. J. Comput. Appl. Math. 196 (2), 619–633.

Fox, R.O., 2009. Higher-order quadrature-based moment methods for kinetic equations. J. Comput. Phys. 228 (20), 7771–7791.
Golub, G.H., Welsch, J.H., 1969. Calculation of Gauss quadrature rules. Math. Comput. 23, 221–230.
Gordon, R.G., 1968. Error bounds in equilibrium statistical mechanics. J. Math. Phys. 9, 655–663.
Hackbusch, W., John, V., Khachatryan, A., Suciu, C. A numerical method for the simulation of an aggregation-driven population balance system. Int. J. Numer. Methods Fluids, http://dx.doi.org/10.1002/fld.2656, in press.
Hulburt, H.M., Katz, S., 1964. Some problems in particle technology—a statistical mechanical formulation. Chem. Eng. Sci. 19, 555–574.
John, V., Angelov, I., Öncül, A.A., Thévenin, D., 2007. Techniques for the reconstruction of a distribution from a finite number of its moments. Chem. Eng. Sci. 62, 2890–2904.
John, V., Roland, M., 2010. On the impact of the scheme for solving the higher dimensional equation in coupled population balance systems. Int. J. Numer. Methods Eng. 82, 1450–1474.
Kulikov, V., Briesen, H., Grosch, R., von Wedel, L., Yang, A., Marquardt, W., 2005. Modular dynamic simulation for integrated particulate processes by means of tool integration. Chem. Eng. Sci. 60, 2069–2083.
Lage, P.L.C., 2002. Comments on the "An analytical solution to continuous population balance model describing floc coalescence and breakage – a special case" by D.P. Patil and J.R.G. Andrews. Chem. Eng. Sci. 57, 4253–4254.
McCoy, B.J., Madras, G., 2003. Analytical solution for a population balance equation with aggregation and fragmentation. Chem. Eng. Sci. 58, 3049–3051.
McGraw, R., 1997. Description of aerosol dynamics by the quadrature method of moments. Aerosol Sci. Technol. 27, 255–265.
Nayak, A.K., Borka, Z., Patruno, L.E., Sporleder, F., Dorao, C.A., Jakobsen, H.A., 2011. A combined multifluid-population balance model for vertical gas-liquid bubble-driven flows considering bubble column operating conditions. Ind. Eng. Chem. Res. 50, 1786–1798.
Patil, D.P., Andrews, J.R.G., 1998. An analytical solution to continuous population balance model describing floc coalescence and breakage—a special case. Chem. Eng. Sci. 53, 599–601.
Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. The art of scientific computing, Numerical Recipes in C, second ed. Cambridge University Press, Cambridge.
Ramkrishna, D., 2000. Population Balances: Theory and Applications to Particulate Systems in Engineering. Academic Press, San Diego.
Rogers, R.R., Yau, M.K., 1996. Short course in cloud physics. In: International Series in Natural Philosophy, 3rd ed. Butterworth Heinemann.
Sack, R.A., Donovan, A.F., 1972. An algorithm for Gaussian quadrature given modified moments. Numer. Math. 18, 465–478.
Upadhyay, R.R., 2012. Evaluation of the use of the Chebyshev algorithm with the quadrature method of moments for simulating aerosol dynamics. Aerosol Sci. 44, 11–23.
Wall, H.S., 1948. Analytic Theory of Continued Fractions. D. Van Nostrand Company, Inc., New York, NY.
Wheeler, J.C., 1974. Modified moments and Gaussian quadratures. In: Proceedings of the International Conference of Padé Approximants, Continued Fractions and Related Topics (Univ. Colorado, Boulder, Colo., 1972; dedicated to the memory of H.S. Wall), vol. 4, pp. 287–296.
Wright Jr., D.L., 2007. Numerical advection of moments of the particle size distribution in Eulerian models. Aerosol Sci. 38, 352–369.
Yuan, C., Fox, R.O., 2011. Conditional quadrature method of moments for kinetic equations. J. Comput. Phys. 230 (22), 8216–8246.