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

Volker John\textsuperscript{a,b,\ast}, Ferdinand Thein\textsuperscript{c}

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

\textbf{A R T I C L E   I N F O}

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

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

\textbf{A B S T R A C T}

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.

\textcopyright{} 2012 Elsevier Ltd. All rights reserved.

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 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):

\[
\begin{align*}
\frac{\partial f(t,x,e)}{\partial t} + \nabla_x \cdot (u(t,x)f(t,x,e)) &= s(t,x,e) - \frac{\partial G(t,x,e)f(t,x,e)}{\partial e} \quad \text{in } (0,T) \times \Omega \times \Omega_e. \tag{1}
\end{align*}
\]

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, \( u(t,x) \) is a velocity field, \( s(t,x,e) \) is a source term, and \( G(t,x,e) \) 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)
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 \( \int_{a}^{b} e^{k} f(t,x,e) \, de \). This approach is called method of moments (MOM). Multiplying (1) by \( e^{k} \) and integrating over \( \Omega \) leads to

\[
\frac{\partial m_k(t,X)}{\partial t} + \nabla_x \cdot \mathbf{u}(t,X)m_k(t,X) = 0.
\]

(2)

\( k = 0, 1, 2, \ldots \) The derivation of (2) applied integration by parts for the growth term where \( \lim_{e \to a} a f(t,x,e) = \lim_{e \to b} b f(t,x,e) = 0 \) was assumed. The modified growth function is given by \( g(t,x,e) = e^{k-1} f(t,x,e) \). 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 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 (Alopaevs 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 McGraw (1997). The idea of QMOM consists in replacing the second term on the right hand side of (2) by a quadrature formula:

\[
\int_{a}^{b} g(t,x,e) f(t,x,e) \, de \approx \sum_{i=1}^{n} g(t,x,e_i) w_i(t,x),
\]

(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 \( x \) 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

\[
\int_{a}^{b} g(e) f(e) \, de \approx \sum_{i=1}^{n} g(e_i) w_i
\]

(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 \( p(e) = 0 \).

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 2\( n \) 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:

\[
\langle p, q \rangle := \int_{a}^{b} p(e) q(e) f(e) \, de.
\]

(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 \( p_0 = 1 \). 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 \( p_{n-1} \).

For this reason, an efficient way for computing the roots of \( p_{n} \) is necessary. To this end, the recursion property of orthogonal polynomials is used:

\[
p_{k+1}(e) = (e - \beta_k) p_k(e) - x_k^2 p_{k-1}(e), \quad k = 0, 1, \ldots
\]

(6)

with the coefficients

\[
\beta_k = \frac{\langle p_{k} p_{k} \rangle}{\langle p_{k-1} p_{k-1} \rangle}, \quad k \geq 0, \quad x_k^2 = \begin{cases} 1, & k = 0 \\ \frac{\langle p_{n} p_{n-1} \rangle}{\langle p_{n-1} p_{n-1} \rangle}, & k = 1, \ldots \end{cases}
\]

Note that, given \( p_0, \ldots, p_{n-1} \), the coefficients \( \beta_{n-1} \) and \( x_{n-1}^2 \) can be computed by knowing the first 2\( n \) 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:

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

\[
(\tilde{A}_n - eI) = \begin{pmatrix} \beta_0 & x_0 & 0 & \cdots & 0 \\ x_0 & \beta_1 & x_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \beta_{n-2} & x_{n-2} \\ 0 & 0 & \cdots & x_{n-2} & \beta_{n-1} \end{pmatrix}
\]
\[
A_n = \begin{pmatrix}
\beta_0 & 1 & 0 & \ldots & 0 \\
\alpha_1 & \beta_1 & 1 & \ldots & 0 \\
0 & \ldots & \alpha_{n-2} & \beta_{n-2} & 1 \\
0 & \ldots & 0 & \alpha_{n-1} & \beta_{n-1}
\end{pmatrix}.
\]

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 \( A_n \).

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 \( d_0 = 1 \), \( d_i = (x_1 \ldots x_i)^{-1} \), \( i = 1, \ldots, n-1 \), then (7) can be transformed for \( e = e_i \) into

\[
(A_n - eI)
\begin{pmatrix}
\hat{p}_0(e) \\
\vdots \\
\hat{p}_{n-1}(e)
\end{pmatrix} = 0
\]
and, using the definition of \( d_k \),

\[
\hat{p}_k(e) = d_k p_k(e) = \frac{\langle p_0 p_k \rangle}{\langle p_0 p_0 \rangle} e_i^{1/2}, \quad k = 0, \ldots, n-1.
\]

Let \( \{p_k\}_{k=0}^{n-1} \) be an orthogonal set of polynomials which are normalized such that \( \langle p_k p_0 \rangle = 1 \), \( k = 0, \ldots, n-1 \). Then it follows from the Christoffel–Darboux formula that the weights are given by

\[
w_i = \left( \sum_{k=0}^{n-1} p_k^2(e) \right)^{-1}, \quad i = 1, \ldots, n.
\]

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

\[
p_k(e) = \frac{\hat{p}_k(e)}{\langle p_k p_k \rangle^{1/2}} = \frac{\hat{p}_k(e)}{\langle p_0 p_0 \rangle e_i^{1/2}}.
\]

Let \( q = (q_0, \ldots, q_{n-1})^T \) be an eigenvector to the eigenvalue \( e_i \), which is, e.g., computed by a numerical method. Since \( p_0(e) = 1 \), it follows that the eigenvector \( (p_0(e), \ldots, p_{n-1}(e))^T \) is \( q_0^2 \) times \( q \).

Using (10) and (11), one obtains

\[
w_i = \left( \sum_{k=0}^{n-1} \frac{p_k^2(e)}{\langle p_0 p_0 \rangle} \right)^{-1} = \frac{\langle p_0 p_0 \rangle}{\langle p_0 p_0 \rangle} q_0 \frac{\left( \sum_{k=0}^{n-1} q_k^2 \right)^{-1}}{q_0^2}, \quad i = 1, \ldots, n
\]

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 MOM, 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 \( \{|m_i|_{i=0}^{2n}\} \) 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 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 \( p(e) = 1 \). Hence, it can be assumed even that \( m_0 > 0 \).

The product-difference algorithm

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

\[
b_{i1} = \delta_{i1}, \quad b_{i2} = (-1)^{i-1} m_{i-1}, \quad i = 1, \ldots, 2n,
\]

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

\[
b_{ij} = \begin{cases} b_{i-j-1} b_{i-j-2} - b_{i-j-2} b_{i-j-1}, & j = 3, \ldots, 2n+1, \\
0, & \text{else}.
\end{cases}
\]

The resulting matrix has the form

\[
B = \begin{pmatrix}
1 & m_0 & b_{13} & \ldots & b_{1,2n+1} \\
0 & -m_1 & b_{23} & \ldots & b_{2,2n} \\
0 & \ldots & \ldots & \ldots & 0 \\
0 & m_{2n-2} & b_{2n-1,3} & \ldots & 0 \\
0 & \ldots & \ldots & \ldots & 0
\end{pmatrix}
\]

In the next step, coefficients \( c_i \) are determined by

\[
c_i = \begin{cases} m_0, & i = 1, \\
\frac{b_{i+1,i+1}}{b_{i+1,i+1}}, & i = 2, \ldots, 2n.
\end{cases}
\]

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

\[
\beta_{i-1} = \begin{cases} c_2, & i = 1, \\
c_2 + c_{2i-1}, & i = 2, \ldots, n, \quad a_i = \sqrt{c_{2i-1} c_{2i}}, \quad i = 1, \ldots, n-1.
\end{cases}
\]

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

\[
l(z) = \int_0^\infty \frac{f(e)}{e + f} \, de,
\]
where \( f(z) \) is a weight function such that \( l(z) \) 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, 1974). In this way, one obtains for all \( z \) a quadrature formula for \( l(z) \) with weights and abscissas independent of \( z \):

\[
l(z) \approx \sum_{i=1}^n w_i \frac{1}{z + e_i}
\]

It can be concluded, using connections between Stieltjes transforms and optimal quadrature rules that one can use the same weights and abscissas if instead \((z+e)^{-1}\) any other function.
moments will be presented here. The LQMDA is initialized by Wheeler (1974), see also Press et al. (1992). This implementation differs from the algorithm given above by considering intermediate quantities $\sigma_m$ instead of $s_m$, where the relation between these quantities is given by $s_m = \sigma_m/\sigma_R$. 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 $A_j$ to $s_j$ is the same as in the Wheeler algorithm $s_j = A_j/A_{ii}$.

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:

$$ e_p(x) = \alpha_k p_{k+1}(x) + \beta_k p_k(x) + \gamma_k x p_{k-1}(x), \quad k = 0, 1, \ldots, $$

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 $\rho_i$ 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:

$$ M = \begin{pmatrix} m_0 & m_1 & m_2 & \cdots & m_n \\ m_1 & m_2 & m_3 & \cdots & m_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n-1} & m_n & \cdots & \cdots & m_{2n} \end{pmatrix} \in \mathbb{R}^{(2n+1) \times (2n+1)}. $$

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 $M = R^T R$, where $R$ is an upper triangular matrix with the entries:

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

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

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

$$ \beta_{j-1} = \frac{r_{ji-1}}{\beta_{ij} - \frac{r_{ij-1}}{r_{ij-2}}}, \quad j = 1, \ldots, n, $$
$y_j = \frac{r_j+1}{r_j}, \quad j = 1, \ldots, n-1,$ \hspace{1cm} (15)

with \( r_{00} = 1 \) and \( r_{01} = 0 \).

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

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

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:

$x p_{j-1}(x) = a_j x p_{j-2}(x) + \beta_j p_{j-1}(x) + c_j p_j(x), \quad j = 1, \ldots, n,$ \hspace{1cm} (16)

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

$s_{ij} = a_i s_{j-1,i-1}, \quad s_{i-1,j} = \beta_j s_{j-1,i-1} + s_{j,i} s_{j-1,i}, \quad j = 1, \ldots, n.$

It follows that

$y_j = \frac{s_{ij}}{s_{j-1,i-1}}, \quad \beta_j = \frac{s_{i-1,j}}{s_{j-1,i}} - \frac{s_{j,i} s_{j-1,i}}{s_{j-1,i-1}}, \quad j = 1, \ldots, n.$

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:

$m_{2n} = \sum_{i=1}^{n} e^{2n} w_i.$ \hspace{1cm} (17)

With this approach, we could observe in numerical studies that the matrix \( M \) might be not positive definite. Concretely, the used MATLAB routine \texttt{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 semidefiniteness 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 \( m_{2n}(t) \to 0 \) 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 \( x \). Problems I–III consider the growth of particles

$\frac{df(t,e)}{dt} = -\frac{\partial}{\partial e} (\phi_i e f(t,e)), \quad (t,e) \in (0,T) \times (0,\infty),$

$f(0,e) = ae^2 \exp(-be), \quad e \in (0,\infty),$

with

$\phi_1(e) = \beta, \quad \phi_2(e) = \beta e, \quad \phi_3(e) = \frac{\beta}{e}.$ \hspace{1cm} (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): \( \alpha = 0.108, \quad \beta = 0.6, \quad \beta = 0.78 \).

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

$\frac{df(t,e)}{dt} = \frac{1}{2} \int_0^t f(t,e-e') f(t,e) de' - \int_t^\infty f(t,e') f(t,e) de' + 2 \alpha \int_0^t f(t,e') de' - \sigma e f(t,e), \quad (t,e) \in (0,T) \times (0,\infty).$ \hspace{1cm} (20)

The following initial conditions were used:

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

Problems of this type were studied in Patil and Andrews (1998), Lage (2002) and McCoy and Madras (2003). In (20), \( \sigma \) is the fragmentation rate. This rate is given by \( \sigma = \Phi(\infty)/2 \), 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:

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

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

<table>
<thead>
<tr>
<th>Problem</th>
<th>PDA</th>
<th>LQMDA</th>
<th>GWA</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0.511</td>
<td>0.497</td>
<td>0.551</td>
</tr>
<tr>
<td>II</td>
<td>0.510</td>
<td>0.498</td>
<td>0.551</td>
</tr>
<tr>
<td>III</td>
<td>0.510</td>
<td>0.497</td>
<td>0.550</td>
</tr>
<tr>
<td>IV</td>
<td>0.431</td>
<td>0.419</td>
<td>0.475</td>
</tr>
<tr>
<td>V</td>
<td>0.434</td>
<td>0.420</td>
<td>0.474</td>
</tr>
<tr>
<td>VI</td>
<td>0.434</td>
<td>0.420</td>
<td>0.474</td>
</tr>
<tr>
<td>VII</td>
<td>0.433</td>
<td>0.419</td>
<td>0.474</td>
</tr>
</tbody>
</table>
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 step size 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 \( m_n\) 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 suggestions greatly helped to improve this note.

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


