Otto-von-Guericke-Universität Magdeburg
Fakultät für Mathematik
Institut für Analysis & Numerik

FAKULTÄT FÜR
MATHEMATIK

Weierstrass-Institut für Angewandte Analysis & Stochastik Berlin

Weierstraß-Institut für
Angewandte Analysis und Stochastik

On the Efficiency and Condition of the Core Routine of the Quadrature Methods of Moments (QMOM)

Diploma Thesis

Eingereicht von:	Ferdinand Thein
Betreuer/1. Gutachter:	Prof. Dr. Volker John
2. Gutachter:	Dr. Maren Hantke

Danksagung

An dieser Stelle möchte ich mich ausdrücklich bei Prof. Dr. John für die Betreuung dieser Arbeit bedanken.
Weiter gilt mein Dank all denen, die mich während der Erstellung dieser Arbeit und auch während des gesamten Studiums auf verschiedenen Wegen unterstützt und begleitet haben.
Insbesondere bedanke ich mich bei meinen Eltern, meinen Geschwistern und meiner Freundin.

Contents

1 Introduction **7**

2 Standard Moment Methods **9**
2.1 Method of Moments . . . 9
2.2 Quadrature Method of Moments . . . 11
2.3 Direct Quadrature Method of Moments . . . 13
2.3.1 Derivation With Distributions From Marchisio/Fox (2005) . . . 13
2.3.2 Derivation Without Distributions and Reformulation . . . 15
2.3.3 Multidimensional DQMOM . . . 17
2.4 Condition (QMOM & DQMOM) . . . 21

3 Algorithms for Gaussian Quadrature **25**
3.1 Gaussian Quadrature . . . 25
3.2 Product-Difference-Algorithm . . . 29
3.2.1 The Algorithm . . . 29
3.2.2 Proof of Correctness of the PDA . . . 31
3.3 Long Quotient-Modified Difference Algorithm . . . 35
3.3.1 The Algorithm . . . 35
3.3.2 Proof of Correctness of the LQMD . . . 37
3.4 Golub-Welsch Algorithm . . . 41
3.4.1 The Algorithm . . . 41
3.4.2 Proof of Correctness of the GWA . . . 41
3.5 Newton's Method . . . 43

4 Improvements to the DQMOM **47**
4.1 Approach With Universal Test Functions . . . 47
4.2 Finding Test Functions . . . 49

5 Numerical Results **53**
5.1 Analytical Solutions & Treatment of the Problems . . . 53
5.1.1 Problem I . . . 53
5.1.2 Problem II . . . 55
5.1.3 Problem III . . . 55
5.1.4 Problems IV – VII . . . 57
5.1.5 Approximation of the Source Terms . . . 61
5.2 Comparison of Quadrature - Algorithms . . . 63
5.3 Comparison of the Three Main Methods . . . 65

6 Conclusion **83**

1 Introduction

In this work moment based methods for the numerical treatment of a *Population Balance Equation*, *PBE*, are investigated. The methods that are treated in this work are the *Method of Moments (MOM)*, the *Quadrature Method of Moments (QMOM)* and the *Direct Quadrature Method of Moments (DQMOM)*. The methods are introduced in historical order and their key features and main differences are worked out.
The *PBEs* that are dealt with here arise for example in the field of aerosol dynamics. The equations describe a so called *Particle Size Distribution* f, *PSD*. This *PSD* depends on the time $t \in [0,T]$, the geometric space $x \in \Omega \subset \mathbb{R}^d$, $d \in \{1,2,3\}$, and additionally on an internal variable $\mathbf{e} \in \Omega_e \subset \mathbb{R}_+^N$. The complete problem for the *PSD* is

$$\begin{cases} \dfrac{\partial f(t,x,e)}{\partial t} + \nabla \cdot (uf(t,x,e)) - \nabla \cdot (D \nabla f(t,x,e)) = S(t,x,e), \quad (t,x,e) \in (0,T] \times \Omega \times \Omega_e, \\ f(t,x,e) = g(t,x,e), \quad (t,x,e) \in [0,T] \times \partial\Omega \times \Omega_e, \\ f(t,x,e) = 0, \quad (t,x,e) \in [0,T] \times \Omega \times \partial\Omega_e, \\ f_0(x,e) = f(0,x,e), \quad (x,e) \in \Omega \times \Omega_e. \end{cases} \tag{1.1}$$

Here we set $u := u(t,x)$ and $D := D(t,x)$. It is important to note that the source term $S(t,x,e)$ on the right-hand side will also depend on the *PSD* f, for example as in 2.1. As an example one can consider a precipitation process where the internal variable is the diameter of a particle, cf. [18]. Other applications may be found in [1] and [14]. The *PBE* is therefore often coupled to the *Navier–Stokes* equation via the velocity. Now the arising difficulty is that the dimension of the *PBE* is increased by N due to the appearance of the internal variable compared to the other system describing equations. There are different ways how to treat this difficulty. One can be seen in [18]. The key idea of moment based methods is not to solve the whole equation for the *PSD*. Instead the *PBE* is transformed and one solves a system of equations for the moments of the *PSD*. The dimension of these equations is now reduced by N. The first moments directly correspond to physical quantities of the system such as the number of particles, mass or the measure of the surface. For the moment transform one multiplies equation (1.1) with e^k for $k = 0,1,2,\dots$ and then integrates over Ω_e. The resulting equation is

$$\frac{\partial m_k(t,x)}{\partial t} + \nabla \cdot (u(t,x) m_k(t,x)) - \nabla \cdot (D(t,x) \nabla m_k(t,x)) = \int_{\Omega_e} e^k S(t,x,e) \,\mathrm{d}e, \quad k = 0,1,2,\dots \,. \tag{1.2}$$

Now the drawback is that f is not known in its entity anymore. To reconstruct f from a given set of moments is an ill–posed problem as shown in [16]. Furthermore one needs that system to be closed for a finite k. It is not obvious how many moments are needed to obtain satisfying results. In Section 2 the three methods are explained and a result for the condition number is given. In Section 3 we investigate algorithms that are needed for the *QMOM* to calculate weights and abscissas for the quadrature approximation. In Section 4 we suggest some improvements to the *DQMOM*. Finally we will give numerical results in Section 5. Therefore we will first treat several problems analytically and then give the numerical simulations.

2 Standard Moment Methods

2.1 Method of Moments

The *MOM* was introduced in 1964 by Hulburt and Katz in [8]. Since there are crucial restrictions to the problems which can be treated by the method, the *MOM* was not used very much, cf. [12]. In the following a simplified version of (1.1) shall be presented in order to illustrate the key points. Therefore we set $u \equiv 0$, $D \equiv 0$ and $S(t,x,e) = -\frac{\partial}{\partial e}\left(\phi(e) f(t,x,e)\right)$. The resulting equation is

$$\frac{\partial f(t,x,e)}{\partial t} = -\frac{\partial}{\partial e}\left(\phi(e) f(t,x,e)\right). \tag{2.1}$$

The function $\phi(e)$ is a growth function and describes the evolution of the internal variable. The shape of the right-hand side depends on the problem. When the moment transform is performed one gets

$$\frac{\partial m_k(t,x)}{\partial t} = k \int_{\Omega_e} e^{k-1} \phi(e) f(t,x,e)\,\mathrm{d}e, \quad k = 0,1,2,\dots\ . \tag{2.2}$$

Then one integrates by parts, the boundary terms vanish since we claim

$$\lim_{e\to 0} f(t,x,e) = 0 \quad \text{and} \quad \lim_{e\to\infty} f(t,x,e) = 0.$$

The remaining difficulty is the integral, since it still depends on the unknown function f. But if the growth function has a special shape, i.e. $\phi = \beta_0 + \beta_1 e$, one obtains

$$\frac{\partial m_k(t,x)}{\partial t} = k \int_{\Omega_e} e^{k-1} (\beta_0 + \beta_1 e) f(t,x,e)\,\mathrm{d}e, \quad k = 0,1,2,\dots\ . \tag{2.3}$$

This is equivalent to

$$\frac{\partial m_k(t,x)}{\partial t} = k\beta_0 m_{k-1}(t,x) + k\beta_1 m_k(t,x), \quad k = 0,1,2,\dots\ . \tag{2.4}$$

It is now obvious, that for this type of source term the number of particles stays constant. Therefore one looks at m_0. Furthermore one clearly sees, that the resulting equation would include higher order moments if the growth function would be of higher order.
To deal with other growth laws, Hulburt and Katz suggested to expand f in series with respect to the orthogonal *Laguerre* polynomials. Now one can also deal with growth laws like

$$\phi(e) = \frac{\beta}{e}, \quad e > 0. \tag{2.5}$$

For the first four moments one obtains analogous to (2.3) the system

$$\begin{aligned}
\tfrac{\partial m_0(t,x)}{\partial t} &= 0,\\
\tfrac{\partial m_1(t,x)}{\partial t} &= k\beta m_{-1}(t,x),\\
\tfrac{\partial m_2(t,x)}{\partial t} &= k\beta m_0(t,x),\\
\tfrac{\partial m_3(t,x)}{\partial t} &= k\beta m_1(t,x).
\end{aligned}$$

The moments of even order can be determined exactly. This is not possible for the remaining moments, since m_1 depends on m_{-1}. But if the series expansion of f is used, the moment m_{-1} can be expressed through the other moments [8]

$$m_{-1} = \frac{m_0^2 m_1}{2m_1^2 - m_0 m_2}.$$

Summing up, one can state that the range of the *MOM* is very restricted. If the growth law is not constant or linear, one has to use a suited approximation of the unknown function f. But one has to choose a good approximation. Hulburt and Katz suggested the *Laguerre* polynomials. Since the *Laguerre* polynomials are orthogonal with respect to the *gamma* distribution, one expects problems when f differs from that shape.

2.2 Quadrature Method of Moments

As shown above, there is a crucial restriction to the *MOM*. That is, if the growth term does not have a particular shape, one does not obtain a closed system of equations with respect to the moments. To circumvent this restriction McGraw introduced a new approach [12]. Instead of approximating f when the source term is to complicated, one approximates the integral through n-point Gaussian quadrature, i.e.

$$\int_{\Omega_e} g(e)f(t,x,e)\,\mathrm{d}e \approx \sum_{i=1}^{n} g(e_i)w_i(t,x) \tag{2.6}$$

where $g(e)$ is a given function. For $g(e) = e^k$, $k = 0, 1, 2, \dots$ one obtains from (2.6) the approximation for the moments m_k. For these moments of f one claims

$$m_k = \sum_{i=1}^{n} e_i^k w_i(t,x) \quad k = 0, 1, 2, \dots\ 2n-1. \tag{2.7}$$

Since there are $2n$ unknowns on the right-hand side, (2.7) implies exact integration of polynomials up to degree $2n-1$ if $2n$ moments are given, see Theorem 3.6 below. The transformed equation is (1.2)

$$\frac{\partial m_k(t,x)}{\partial t} + \nabla\cdot(u(t,x)m_k(t,x)) - \nabla\cdot(D(t,x)\nabla m_k(t,x)) = \int_{\Omega_e} e^k S(t,x,e)\,\mathrm{d}e, \quad k = 0, 1, 2, \dots\ .$$

The obtained system is now closed for all k when the integral is approximated using Gaussian quadrature. But one has to deal with $2n$ unknown weights and abscissas. It is now important to note that the $2n$ moments uniquely determine these weights and abscissas.
So the idea is to use the given moments in each time step to determine the corresponding weights and abscissas. Once these are obtained, one can approximate the integral containing the source term. How this can be done is shown in Section 5.
The unknown quantities can be calculated by solving the nonlinear system (2.7) involving the $2n$ moments. This system is $Ew = \mu$, with $w = (w_1, \dots, w_n)^T$, $\mu = (m_0, \dots, m_{2n-1})^T$ and

$$E := \begin{pmatrix} 1 & 1 & \dots & 1 \\ e_1 & e_2 & \dots & e_n \\ e_1^2 & e_2^2 & \dots & e_n^2 \\ e_1^3 & e_2^3 & \dots & e_n^3 \\ \vdots & \vdots & \dots & \vdots \\ e_1^{2n-1} & e_2^{2n-1} & \dots & e_n^{2n-1} \end{pmatrix} \in \mathbb{R}^{2n\times n}. \tag{2.8}$$

Keep in mind, that only μ is known. To emphasise this, we will write this system for $n = 1$ and $n = 2$ explicitly. For $n = 1$

$$\begin{aligned} m_0 &= w_1, \\ m_1 &= e_1 w_1. \end{aligned}$$

For $n = 2$ one has the system

$$\begin{aligned} m_0 &= w_1 + w_2, \\ m_1 &= e_1 w_1 + e_2 w_2, \\ m_2 &= e_1^2 w_1 + e_2^2 w_2, \\ m_3 &= e_1^3 w_1 + e_2^3 w_2. \end{aligned}$$

Once the weights and abscissas are determined, all the integrals can be approximated. When this is done you can calculate the next time step and start all over again.

Step 1 Calculate initial moments.

Step 2 Calculate weights and abscissas from the given moments.

Step 3 Approximate the integral containing the source term.

Step 4 Calculate the next time step for the moments

Step 5 Repeat *Step 2* to *Step 4* until T.

So now the missing step is the calculation of the weights and abscissas. For this, McGraw suggested the *Product-Difference-Algorithm*. We will discuss this one and other possible algorithms in a separate section.

2.3 Direct Quadrature Method of Moments

The *DQMOM* was introduced in 2005, in order to deal with problems including more than one internal variable, by Marchisio and Fox [10]. The main difference between the *QMOM* and the *DQMOM* is, that one does not solve a system for the moments but obtains equations for the weights and abscissas directly. At first we will derive the method as suggested in the original work. Then we will show an alternative way, which avoids the delta distribution.

2.3.1 Derivation With Distributions From Marchisio/Fox (2005)

The idea of the *QMOM* is to replace the integrals by Gaussian quadrature, therefore the weights and abscissas have to be determined. For the *DQMOM* the function f is approximated by a combination of delta distributions

$$f(t,x,e) \approx \sum_{i=1}^{n} w_i(t,x)\delta(e-e_i(t,x)) \tag{2.9}$$

where $\delta(.)$ is the delta distribution with

$$\delta(x) = \begin{cases} 0, & x \neq 0, \\ \infty, & x = 0, \end{cases}$$

and

$$\int_{-\infty}^{\infty} \delta(x)\,\mathrm{d}x = 1.$$

In (2.9) the e_i can be thought of as different particles in the phase space [10]. To derive this method one starts with inserting (2.9) into equation (1.1), multiplying with the test function e^k and integrating with respect to the internal variable. Therefore one obtains (summation over i)

$$\int_{\Omega_e} \Big\{ \frac{\partial w_i(t,x)\delta(e-e_i(t,x))}{\partial t} + \nabla\cdot(u(t,x)w_i(t,x)\delta(e-e_i(t,x))) - \nabla\cdot(D(t,x)\nabla(w_i(t,x)\delta(e-e_i(t,x)))) \Big\} e^k\,\mathrm{d}e = \int_{\Omega_e} S(t,x,e)e^k\,\mathrm{d}e\,. \tag{2.10}$$

We oppress the dependance of (t,x) in the following calculations. One gets

$$\begin{aligned} &\int_{\Omega_e} \Big\{ \delta(e-e_i)\frac{\partial w_i}{\partial t} - w_i\delta'(e-e_i)\frac{\partial e_i}{\partial t} + \delta(e-e_i)\nabla\cdot(uw_i) - w_i\delta'(e-e_i)u\cdot\nabla e_i \\ &- \delta(e-e_i)\nabla\cdot(D\nabla w_i) - Dw_i\delta''(e-e_i)(\nabla e_i)^2 + \delta'(e-e_i)\left(D\nabla w_i\cdot\nabla e_i + \nabla\cdot(Dw_i\nabla e_i)\right) \Big\} e^k\,\mathrm{d}e \\ &= \int_{\Omega_e} S(t,x,e)e^k\,\mathrm{d}e\,. \end{aligned} \tag{2.11}$$

Now the terms in this equation are sorted according to the derivatives of the delta distribution

$$\begin{aligned} &\int_{\Omega_e} \Big\{ \delta(e-e_i)\left\{ \frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i) \right\} \\ &- \delta'(e-e_i)\left\{ w_i\frac{\partial e_i}{\partial t} + w_iu\cdot\nabla e_i - (D\nabla w_i\cdot\nabla e_i + \nabla\cdot(Dw_i\nabla e_i)) \right\} \\ &- \delta''(e-e_i)\left\{ Dw_i(\nabla e_i)^2 \right\} \Big\} e^k\,\mathrm{d}e \\ &= \int_{\Omega_e} S(t,x,e)e^k\,\mathrm{d}e. \end{aligned}$$

Transforming the variables e_i to $\zeta_i = w_i e_i$ (*weighted abscissae*) the equations can be reformulated as follows

$$\begin{aligned}
&\int_{\Omega_e} \Big\{ \delta(e-e_i) \Big\{ \frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i) \Big\} \\
&- \delta'(e-e_i) \Big\{ \frac{\partial \zeta_i}{\partial t} + \nabla\cdot(u\zeta_i) - \nabla\cdot(D\nabla\zeta_i) - e_i \Big(\frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i) \Big) \Big\} \\
&- \delta''(e-e_i) \left\{ Dw_i(\nabla e_i)^2 \right\} \Big\} e^k \,\mathrm{d}e \\
&= \int_{\Omega_e} S(t,x,e) e^k \,\mathrm{d}e\,.
\end{aligned}$$

With the notation

$$\begin{aligned}
&\frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i) = \xi_i^{(1)}, \\
&\frac{\partial \zeta_i}{\partial t} + \nabla\cdot(u\zeta_i) - \nabla\cdot(D\nabla\zeta_i) = \xi_i^{(2)}, \\
&Dw_i(\nabla e_i)^2 = \xi_i^{(3)},
\end{aligned} \tag{2.12}$$

one gets another formulation of the k-th equation

$$\int_{\Omega_e} \left\{ \sum_{i=1}^n \delta(e-e_i)\xi_i^{(1)} - \sum_{i=1}^n \delta'(e-e_i)(\xi_i^{(2)} - e_i\xi_i^{(1)}) - \sum_{i=1}^n \delta''(e-e_i)\xi_i^{(3)} \right\} e^k \,\mathrm{d}e = \int_{\Omega_e} S(t,x,e)e^k \,\mathrm{d}e. \tag{2.13}$$

Recall the following for the delta distribution

$$\int_{\Omega_e} \delta(e-e_i)e^k \,\mathrm{d}e = e_i^k, \quad \int_{\Omega_e} \delta'(e-e_i)e^k \,\mathrm{d}e = -ke_i^{k-1}, \quad \int_{\Omega_e} \delta''(e-e_i)e^k \,\mathrm{d}e = k(k-1)e_i^{k-2}.$$

Inserting these expressions into (2.13) gives a linear system for the source terms $\xi_i^{(1)}$, $\xi_i^{(2)}$, $\xi_i^{(3)}$

$$(1-k)\sum_{i=1}^n e_i^k \xi_i^{(1)} + k\sum_{i=1}^n e_i^{k-1}\xi_i^{(2)} = k(k-1)\sum_{i=1}^n e_i^{k-2}\xi_i^{(3)} + \int_{\Omega_e} e^k S(t,x,e)\,\mathrm{d}e, \quad k = 0,1,2,\dots\,. \tag{2.14}$$

Defining the following matrices

$$A_1 := \begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ -e_1^2 & -e_2^2 & \dots & -e_n^2 \\ -2e_1^3 & -2e_2^3 & \dots & -2e_n^3 \\ \vdots & \vdots & \dots & \vdots \\ 2(1-n)e_1^{2n-1} & 2(1-n)e_2^{2n-1} & \dots & 2(1-n)e_n^{2n-1} \end{pmatrix} \in \mathbb{R}^{2n\times n}, \tag{2.15}$$

$$A_2 := \begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ 2e_1 & 2e_2 & \dots & 2e_n \\ 3e_1^2 & 3e_2^2 & \dots & 3e_n^2 \\ \vdots & \vdots & \dots & \vdots \\ (2n-1)e_1^{2(n-1)} & (2n-1)e_2^{2(n-1)} & \dots & (2n-1)e_n^{2(n-1)} \end{pmatrix} \in \mathbb{R}^{2n\times n}, \tag{2.16}$$

$$A_3 := \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 2 & 2 & \dots & 2 \\ 6e_1 & 6e_2 & \dots & 6e_n \\ \vdots & \vdots & \dots & \vdots \\ 2(n-1)(2n-1)e_1^{2n-3} & 2(n-1)(2n-1)e_2^{2n-3} & \dots & 2(n-1)(2n-1)e_n^{2n-3} \end{pmatrix} \in \mathbb{R}^{2n\times n} \tag{2.17}$$

and denote by

$$\begin{aligned} A = [A_1,\, A_2], \quad \xi = \left[\xi_1^{(1)}, \dots, \xi_n^{(1)}, \xi_1^{(2)}, \dots, \xi_n^{(2)}\right]^T, \xi^{(3)} = \left[\xi_1^{(3)}, \dots, \xi_n^{(3)}\right]^T, \\ \bar{S} = \left[\int_{\Omega_e} S(t,x,e)\,\mathrm{d}e, \dots, \int_{\Omega_e} e^{2n-1} S(t,x,e)\,\mathrm{d}e\right]^T, \quad d = A_3\xi^{(3)} + \bar{S}, \end{aligned} \tag{2.18}$$

one can write the system as $A\xi = d$. Now one has to perform the following steps

Step 1 Calculate initial moments.

Step 2 Calculate initial weights and abscissas from the given moments using one of the algorithms presented in Section 3.

Step 3 Approximate the integral containing the source term.

Step 4 Initialise and solve the linear system.

Step 5 Calculate the next time step for the weights and weighted abscissas.

Step 6 Optionally: Calculate the moments via (2.7).

Step 7 Repeat *Step 3* to *Step 6* until T.

It should be remarked, that the $\xi_i^{(3)}$ are directly calculated with the given quantities at the present time step. Now one can argue that there are some disadvantages. The first is the use of the delta distribution, (2.9) makes hardly sense when one multiplies with infinity and one can doubt whether the powers of e are the right test functions. Furthermore one can possibly face a situation where the weights are near to zero or the abscissas lie close to each other. In the first case one has to worry about the weighted abscissas and in the second case the matrix is close to be singular. These problems will be discussed below. Note that the test function e^k are not necessarily needed to introduce the moments. The moments can be obtained from the calculated weights and abscissas. These can be determined with any suited test function, as shown below in Section 4.

2.3.2 Derivation Without Distributions and Reformulation

Here we will present a way to circumvent the delta distribution. Furthermore this seems to clarify the key idea of the *DQMOM*. To do this, one inserts equation (2.7) directly into the system for the moments (1.2). The result is (again the dependance of (t,x) is oppressed)

$$\sum_{i=1}^{n} \left\{ \frac{\partial(w_i e_i^k)}{\partial t} + \nabla\cdot(u(t,x) w_i e_i^k) - \nabla\cdot(D\nabla(w_i e_i^k)) \right\} = \int_{\Omega_e} S(t,x,e) e^k \,\mathrm{d}e. \tag{2.19}$$

By differentiating, rearranging and introducing the variable $\zeta_i = w_i e_i$ (analogous to the original way) one obtains

$$\begin{aligned}
&\sum_{i=1}^{n}\Bigg\{ e_i^k \left\{ \frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i)\right\} \\
&\quad + ke_i^{k-1}\left\{ \frac{\partial \zeta_i}{\partial t} + \nabla\cdot(u\zeta_i) - \nabla\cdot(D\nabla\zeta_i) - e_i\left(\frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i)\right)\right\} \\
&\quad - k(k-1)e_i^{k-2}\left\{Dw_i(\nabla e_i)^2\right\}\Bigg\} \\
&\quad = \int_{\Omega_e} S(t,x,e)e^k \,\mathrm{d}e.
\end{aligned}$$

Again, with the source terms $\xi_i^{(1)}, \xi_i^{(2)}, \xi_i^{(3)}$ this results in (2.14)

$$(1-k)\sum_{i=1}^{n} e_i^k \xi_i^{(1)} + k\sum_{i=1}^{n} e_i^{k-1}\xi_i^{(2)} = k(k-1)\sum_{i=1}^{n} e_i^{k-2}\xi_i^{(3)} + \int_{\Omega_e} e^k S(t,x,e)\,\mathrm{d}e, \quad k = 0,1,2,\ldots.$$

So now consider the case of numerical difficulties. Sure one can exclude the distribution in the derivation of this method to be a reason for failing. Now it is also interesting to know, what happens if the variable ζ_i is not introduced and you define another system. Therefore review equation (2.19). After applying the product rule one gets

$$\begin{aligned}
&\sum_{i=1}^{n}\Bigg\{ e_i^k \left\{ \frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i)\right\} + kw_i e_i^{k-1}\left\{\frac{\partial e_i}{\partial t} + u\cdot\nabla e_i - D\Delta e_i\right\} \\
&\quad - ke_i^{k-1}\left\{D\nabla w_i\cdot\nabla e_i + \nabla e_i\cdot\nabla(Dw_i)\right\} - k(k-1)e_i^{k-2}\left\{Dw_i(\nabla e_i)^2\right\}\Bigg\} \\
&\quad = \int_{\Omega_e} S(t,x,e)e^k \,\mathrm{d}e.
\end{aligned}$$

Now one introduces four source terms

$$\begin{aligned}
&\frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i) = \xi_i^{(1)}, \\
&\frac{\partial e_i}{\partial t} + u\cdot\nabla e_i - D\Delta e_i = \xi_i^{(2)}, \\
&Dw_i(\nabla e_i)^2 = \xi_i^{(3)}, \\
&D\nabla w_i\cdot\nabla e_i + \nabla e_i\cdot\nabla(Dw_i) = \xi_i^{(4)}.
\end{aligned} \tag{2.20}$$

Again a linear system for the source terms (where the latter two can already be calculated with the initial data) is obtained

$$\begin{aligned}
&\sum_{i=1}^{n} e_i^k\xi_i^{(1)} + k\sum_{i=0}^{n} w_i e_i^{k-1}\xi_i^{(2)} \\
&= k(k-1)\sum_{i=1}^{n} e_i^{k-2}\xi_i^{(3)} + k\sum_{i=1}^{n} e_i^{k-1}\xi_i^{(3)} + \int_{\Omega_e} e^k S(t,x,e)\,\mathrm{d}e, \quad k = 0,1,2,\ldots\,.
\end{aligned} \tag{2.21}$$

With the matrices $B = A_3$ (2.17), $C = A_2$ (2.16),

$$A := \begin{pmatrix}
1 & \ldots & 1 & 0 & \ldots & 0 \\
e_1 & \ldots & e_n & w_1 & \ldots & w_n \\
e_1^2 & \ldots & e_n^2 & 2e_1w_1 & \ldots & 2e_nw_n \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
e_1^{2n-1} & \ldots & e_n^{2n-1} & (2n-1)e_1^{2n-2}w_1 & \ldots & (2n-1)e_n^{2n-2}w_n
\end{pmatrix} \in \mathbb{R}^{2n\times 2n} \tag{2.22}$$

and the vectors

$$\xi = \left[\xi_1^{(1)}, \ldots, \xi_n^{(1)}, \xi_1^{(2)}, \ldots, \xi_n^{(2)}\right]^T, \quad \xi^{(3)} = \left[\xi_1^{(3)}, \ldots, \xi_n^{(3)}\right]^T,$$
$$\xi^{(4)} = \left[\xi_1^{(4)}, \ldots, \xi_n^{(4)}\right]^T, \quad \bar{S} = \left[\int_{\Omega_e} S(t,x,e)\,\mathrm{d}e, \ldots, \int_{\Omega_e} e^{2n-1} S(t,x,e)\,\mathrm{d}e\right]^T,$$

one can write the system in the following form

$$A\xi = \underbrace{B\xi^{(3)} + C\xi^{(4)} + \bar{S}}_{=:d}.$$

It should be remarked, that the matrix A (2.22) is the Jacobian (2.26) that is obtained in sections 2.4 and 3.5 below. Now one performs the same steps like before. But this approach gives basically the same numerical results. This is, because one just shifted the difficulty that occurs for $w_i = 0$ from a division by zero to a singular system matrix (2.26). So there probably remains only one way for a possible improvement. One has to change the test functions. So the idea is to choose adequate test functions that improve the condition number of this problem.

2.3.3 Multidimensional DQMOM

An essential feature of the *DQMOM* is that it can be extended to the case of more than one internal variable. Therefore the *DQMOM* shall be derived for the multivariate case according to [10]. The delta distribution for the case of more than one dimension $x \in \mathbb{R}^m$ reads

$$\delta(\mathbf{x}) = \prod_{i=1}^{m} \delta(x_i).$$

Here one has $\mathbf{e} \in \Omega_e \subset \mathbb{R}^N$ with $\mathbf{e} = (e^{(1)}, \ldots, e^{(N)})$. The multidimensional moments are defined as

$$m_{l_1,\ldots,l_N} = \int_{\Omega_e} \prod_{\alpha=1}^{N} \left(e^{(\alpha)}\right)^{l_\alpha} f(t,x,\mathbf{e})\,\mathrm{d}\mathbf{e}.$$

Again the *PSD* is represented via a combination of delta distributions

$$f(t,x,\mathbf{e}) \approx \sum_{i=1}^{n} w_i(t,x)\delta(\mathbf{e} - \mathbf{e}_i(t,x)) = \sum_{i=1}^{n} w_i(t,x) \prod_{\alpha=1}^{N} \delta(e^{(\alpha)} - e_i^{(\alpha)}(t,x)). \tag{2.23}$$

This expression is now inserted into the *PBE* (1.1) and one obtains

$$\sum_{i=1}^{n} \Big\{ \frac{\partial w_i(t,x)\delta(\mathbf{e} - \mathbf{e}_i(t,x))}{\partial t} + \nabla \cdot (u(t,x) w_i(t,x)\delta(\mathbf{e} - \mathbf{e}_i(t,x))) \\ - \nabla \cdot (D(t,x)\nabla(w_i(t,x)\delta(\mathbf{e} - \mathbf{e}_i(t,x)))) \Big\} = S(t,x,\mathbf{e}).$$

In the following calculations we suppress the dependance on (t, x). Differentiating and sorting the terms yields

$$\begin{aligned}
&\sum_{i=1}^{n}\left\{\prod_{\alpha=1}^{N}\delta(e^{(\alpha)}-e_i^{(\alpha)})\left[\frac{\partial w_i}{\partial t}+\nabla\cdot(uw_i)-\nabla\cdot(D\nabla w_i)\right]\right.\\
&-\sum_{\alpha=1}^{N}\delta'(e^{(\alpha)}-e_i^{(\alpha)})\prod_{\beta=1,\beta\neq\alpha}^{N}\delta(e^{(\beta)}-e_i^{(\beta)})\left[w_i\frac{\partial e_i^{(\alpha)}}{\partial t}+(uw_i)\cdot\nabla e_i^{(\alpha)}-w_i\nabla D\cdot\nabla e_i^{(\alpha)}\right.\\
&\left.-\,2D\nabla w_i\cdot\nabla e_i^{(\alpha)}+w_iD\Delta e_i^{(\alpha)}\right]-\sum_{\alpha=1}^{N}\delta''(e^{(\alpha)}-e_i^{(\alpha)})\prod_{\beta=1,\beta\neq\alpha}^{N}\delta(e^{(\beta)}-e_i^{(\beta)})\left[w_iD\left(e_i^{(\alpha)}\right)^2\right]\\
&\left.-\sum_{\alpha=1}^{N}\sum_{\beta=1,\beta\neq\alpha}^{N}\delta'(e^{(\alpha)}-e_i^{(\alpha)})\delta'(e^{(\beta)}-e_i^{(\beta)})\prod_{\gamma=1,\gamma\neq\alpha,\beta}^{N}\delta(e^{(\gamma)}-e_i^{(\gamma)})\left[w_iD\nabla e_i^{(\alpha)}\nabla e_i^{(\beta)}\right]\right\}=S(t,x,\mathbf{e}).
\end{aligned}$$

When the *weighted abscissae* $\zeta_i^{(\alpha)}:=e_i^{(\alpha)}w_i$ is inserted one obtains

$$\begin{aligned}
&\sum_{i=1}^{n}\left\{\prod_{\alpha=1}^{N}\delta(e^{(\alpha)}-e_i^{(\alpha)})\left[\frac{\partial w_i}{\partial t}+\nabla\cdot(uw_i)-\nabla\cdot(D\nabla w_i)\right]\right.\\
&-\sum_{\alpha=1}^{N}\delta'(e^{(\alpha)}-e_i^{(\alpha)})\prod_{\beta=1,\beta\neq\alpha}^{N}\delta(e^{(\beta)}-e_i^{(\beta)})\left[\frac{\partial\zeta_i^{(\alpha)}}{\partial t}+\nabla\cdot(u\zeta_i^{(\alpha)})-\nabla\cdot(D\nabla\zeta_i^{(\alpha)})\right.\\
&\left.\left.-\,e_i^{(\alpha)}\left(\frac{\partial w_i}{\partial t}+\nabla\cdot(uw_i)-\nabla\cdot(D\nabla w_i)\right)\right]-\sum_{\alpha=1}^{N}\delta''(e^{(\alpha)}-e_i^{(\alpha)})\prod_{\beta=1,\beta\neq\alpha}^{N}\delta(e^{(\beta)}-e_i^{(\beta)})\left[w_iD\left(e_i^{(\alpha)}\right)^2\right]\right.\\
&\left.-\sum_{\alpha=1}^{N}\sum_{\beta=1,\beta\neq\alpha}^{N}\delta'(e^{(\alpha)}-e_i^{(\alpha)})\delta'(e^{(\beta)}-e_i^{(\beta)})\prod_{\gamma=1,\gamma\neq\alpha,\beta}^{N}\delta(e^{(\gamma)}-e_i^{(\gamma)})\left[w_iD\nabla e_i^{(\alpha)}\nabla e_i^{(\beta)}\right]\right\}=S(t,x,\mathbf{e}).
\end{aligned}$$

Again one introduces source terms for the different expressions, i.e.

$$\begin{aligned}
&\frac{\partial w_i}{\partial t}+\nabla\cdot(uw_i)-\nabla\cdot(D\nabla w_i)=\xi_i^{(1)},\\
&\frac{\partial\zeta_i^{(\alpha)}}{\partial t}+\nabla\cdot(u\zeta_i^{(\alpha)})-\nabla\cdot(D\nabla\zeta_i^{(\alpha)})=\xi_{i\alpha}^{(2)},\\
&w_iD\left(e_i^{(\alpha)}\right)^2=\xi_{i\alpha}^{(3)},\\
&w_iD\nabla e_i^{(\alpha)}\nabla e_i^{(\beta)}=\xi_{i\alpha\beta}^{(4)}.
\end{aligned}$$

This is a total of $n(N^2+N+1)$ source terms. But as in the mono variate case only $n(N+1)$ are unknown during the calculation. Inserting the source terms into the equation gives

$$\begin{aligned}
&\sum_{i=1}^{n}\left\{\prod_{\alpha=1}^{N}\delta(e^{(\alpha)}-e_i^{(\alpha)})\xi_i^{(1)}-\sum_{\alpha=1}^{N}\delta'(e^{(\alpha)}-e_i^{(\alpha)})\prod_{\beta=1,\beta\neq\alpha}^{N}\delta(e^{(\beta)}-e_i^{(\beta)})\left[\xi_{i\alpha}^{(2)}-e_i^{(\alpha)}\xi_i^{(1)}\right]\right.\\
&-\sum_{\alpha=1}^{N}\delta''(e^{(\alpha)}-e_i^{(\alpha)})\prod_{\beta=1,\beta\neq\alpha}^{N}\delta(e^{(\beta)}-e_i^{(\beta)})\xi_{i\alpha}^{(3)}\\
&\left.-\sum_{\alpha=1}^{N}\sum_{\beta=1,\beta\neq\alpha}^{N}\delta'(e^{(\alpha)}-e_i^{(\alpha)})\delta'(e^{(\beta)}-e_i^{(\beta)})\prod_{\gamma=1,\gamma\neq\alpha,\beta}^{N}\delta(e^{(\gamma)}-e_i^{(\gamma)})\xi_{i\alpha\beta}^{(4)}\right\}=S(t,x,\mathbf{e}).
\end{aligned}$$

Now one can perform the moment transform and this results in (summation over i)

$$\int_{\Omega_e} \prod_{\alpha=1}^{N} \left(e^{(\alpha)}\right)^{l_\alpha} \left\{ \prod_{\alpha=1}^{N} \delta(e^{(\alpha)} - e_i^{(\alpha)}) \xi_i^{(1)} - \sum_{\alpha=1}^{N} \delta'(e^{(\alpha)} - e_i^{(\alpha)}) \prod_{\substack{\beta=1,\\ \beta\neq\alpha}}^{N} \delta(e^{(\beta)} - e_i^{(\beta)}) \left[\xi_{i\alpha}^{(2)} - e_i^{(\alpha)} \xi_i^{(1)}\right] \right\} \mathrm{d}\mathbf{e}$$

$$= \int_{\Omega_e} \prod_{\alpha=1}^{N} \left(e^{(\alpha)}\right)^{l_\alpha} \sum_{\alpha=1}^{N} \delta''(e^{(\alpha)} - e_i^{(\alpha)}) \prod_{\beta=1,\beta\neq\alpha}^{N} \delta(e^{(\beta)} - e_i^{(\beta)}) \xi_{i\alpha}^{(3)} \mathrm{d}\mathbf{e}$$

$$+ \int_{\Omega_e} \prod_{\alpha=1}^{N} \left(e^{(\alpha)}\right)^{l_\alpha} \left\{ \sum_{\alpha=1}^{N} \sum_{\substack{\beta=1,\\ \beta\neq\alpha}}^{N} \delta'(e^{(\alpha)} - e_i^{(\alpha)}) \delta'(e^{(\beta)} - e_i^{(\beta)}) \prod_{\substack{\gamma=1,\\ \gamma\neq\alpha,\beta}}^{N} \delta(e^{(\gamma)} - e_i^{(\gamma)}) \xi_{i\alpha\beta}^{(4)} + S(t,x,\mathbf{e}) \right\} \mathrm{d}\mathbf{e}.$$

This simplifies analogous to the one dimensional case

$$\sum_{i=1}^{n} \left\{ \xi_i^{(1)} \prod_{\alpha=1}^{N} \left(e_i^{(\alpha)}\right)^{l_\alpha} + \sum_{\alpha=1}^{N} \left[\xi_{i\alpha}^{(2)} - e_i^{(\alpha)} \xi_i^{(1)}\right] l_\alpha \left(e_i^{(\alpha)}\right)^{l_\alpha - 1} \prod_{\beta=1,\beta\neq\alpha}^{N} \left(e_i^{(\beta)}\right)^{l_\beta} \right.$$

$$= \sum_{i=1}^{n} \sum_{\alpha=1}^{N} \xi_{i\alpha}^{(3)} l_\alpha (l_\alpha - 1) \left(e_i^{(\alpha)}\right)^{l_\alpha - 2} \prod_{\beta=1,\beta\neq\alpha}^{N} \left(e_i^{(\beta)}\right)^{l_\beta}$$

$$+ \sum_{i=1}^{n} \left\{ \sum_{\alpha=1}^{N} \sum_{\substack{\beta=1,\\ \beta\neq\alpha}}^{N} \xi_{i\alpha\beta}^{(4)} l_\alpha l\beta \left(e_i^{(\alpha)}\right)^{l_\alpha - 1} \left(e_i^{(\beta)}\right)^{l_\beta - 1} \prod_{\substack{\gamma=1,\\ \gamma\neq\alpha,\beta}}^{N} \left(e_i^{(\gamma)}\right)^{l_\gamma} + \int_{\Omega_e} \prod_{\alpha=1}^{N} \left(e^{(\alpha)}\right)^{l_\alpha} S(t,x,\mathbf{e}) \right\} \mathrm{d}\mathbf{e}.$$

For $N = 1$ one clearly sees that the mono variate case is included. In the mono variate case one has to choose $2n$ moments and therefore the exponents are $k = 0, \ldots, 2n-1$. This results in the well known linear system. The multivariate case is very different from that. The system matrices crucially depend on the choice of moments that is made. According to [10] we will present the bivariate case for $n = 1$ and $n = 2$. It is obvious that the number of given moments should not be smaller than the number of unknown source terms $n(N+1)$, i.e. 3 or 6 in the present cases. For $N = 2$ one obtains

$$\sum_{i=1}^{n} \left\{ \xi_i^{(1)} \left(e_i^{(1)}\right)^{l_1} \left(e_i^{(2)}\right)^{l_2} + \left[\xi_{i1}^{(2)} - e_i^{(1)} \xi_i^{(1)}\right] l_1 \left(e_i^{(1)}\right)^{l_1 - 1} \left(e_i^{(2)}\right)^{l_2} \right.$$

$$\left. + \left[\xi_{i2}^{(2)} - e_i^{(2)} \xi_i^{(1)}\right] l_2 \left(e_i^{(2)}\right)^{l_2 - 1} \left(e_i^{(1)}\right)^{l_1} \right\}$$

$$= \sum_{i=1}^{n} \left\{ \xi_{i1}^{(3)} l_1 (l_1 - 1) \left(e_i^{(1)}\right)^{l_1 - 2} \left(e_i^{(2)}\right)^{l_2} + \xi_{i2}^{(3)} l_2 (l_2 - 1) \left(e_i^{(2)}\right)^{l_2 - 2} \left(e_i^{(1)}\right)^{l_1} \right.$$

$$\left. + 2\xi_{i12}^{(4)} l_1 l_2 \left(e_i^{(1)}\right)^{l_1 - 1} \left(e_i^{(2)}\right)^{l_2 - 1} + \int_{\Omega_e} \prod_{\alpha=1}^{N} \left(e^{(\alpha)}\right)^{l_\alpha} S(t,x,\mathbf{e}) \, \mathrm{d}\mathbf{e} \right\}.$$

For $n = 1$ the three mixed moments m_{00}, m_{01} and m_{10} are chosen and hence the source term is

$$S_{00}^{(1)} = \int_{\Omega_e} S(t,x,e^{(1)},e^{(2)}) \, \mathrm{d}e^{(1)} \, \mathrm{d}e^{(2)},$$

$$S_{01}^{(1)} = \int_{\Omega_e} e^{(2)} S(t,x,e^{(1)},e^{(2)}) \, \mathrm{d}e^{(1)} \, \mathrm{d}e^{(2)},$$

$$S_{10}^{(1)} = \int_{\Omega_e} e^{(1)} S(t,x,e^{(1)},e^{(2)}) \, \mathrm{d}e^{(1)} \, \mathrm{d}e^{(2)}.$$

Altogether one obtains for the system of unknown source terms

$$\begin{aligned}
\frac{\partial w_i}{\partial t} + \nabla \cdot (u w_i) - \nabla \cdot (D \nabla w_i) &= S_{00}^{(1)}, \\
\frac{\partial \zeta_i^{(1)}}{\partial t} + \nabla \cdot (u \zeta_i^1) - \nabla \cdot (D \nabla \zeta_i^1) &= S_{01}^{(1)}, \\
\frac{\partial \zeta_i^{(2)}}{\partial t} + \nabla \cdot (u \zeta_i^2) - \nabla \cdot (D \nabla \zeta_i^2) &= S_{10}^{(1)}.
\end{aligned}$$

For $n = 2$ the six moments of lowest order $m_{00}, m_{01}, m_{10}, m_{11}, m_{02}$ and m_{20} are chosen. The vector of the unknown variables is in general

$$\xi = \left(\xi_1^{(1)}, \ldots, \xi_n^{(1)}, \xi_{11}^{(2)}, \ldots, \xi_{1N}^{(2)}, \xi_{21}^{(2)}, \ldots, \xi_{nN}^{(2)}\right)^T$$

this implies for the present case

$$\xi = \left(\xi_1^{(1)}, \xi_2^{(1)}, \xi_{11}^{(2)}, \xi_{12}^{(2)}, \xi_{21}^{(2)}, \xi_{22}^{(2)}\right)^T.$$

Therefore the system matrix is

$$A := \begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 \\
-\left(e_1^{(1)}\right)^2 & -\left(e_2^{(1)}\right)^2 & 2e_1^{(1)} & 2e_2^{(1)} & 0 & 0 \\
-e_1^{(1)} e_1^{(2)} & -e_2^{(1)} e_2^{(2)} & e_1^{(2)} & e_2^{(2)} & e_1^{(1)} & e_2^{(1)} \\
-\left(e_1^{(2)}\right)^2 & -\left(e_2^{(2)}\right)^2 & 0 & 0 & 2e_1^{(2)} & 2e_2^{(2)}
\end{pmatrix}.$$

It is shown in [10] that this matrix is singular. It can be turned into a regular one by replacing m_{11} by a higher order moment. For further information we refer to [10]. Again we remark that this result can be obtained by inserting

$$m_{l_1,\ldots,l_N} = \sum_{i=1}^{n} w_i(t,x) \prod_{\alpha=1}^{N} \left(e_i^{(\alpha)}(t,x)\right)^{l_\alpha}$$

directly into equation (1.1).

2.4 Condition (QMOM & DQMOM)

Now that the methods are introduced, we want to focus on the condition number of the latter two methods (since they are the most common ones). We will refer to a paper by Gautschi [5] and cite the result that is most interesting for us. As it was shown above, there is an analytical affinity between the *QMOM* and the *DQMOM* and we suggested to change the test function in the original derivation of the *DQMOM* in order to improve this method. These two aspects will be underlined by the following result.
It was explained in 2.2 that the solution to a nonlinear system is needed to obtain the weights and abscissas. This solution is obtained via a mapping G from the moment space Y to the space of weights and abscissas X

$$G : Y \to X.$$

These spaces are $2n$ dimensional real Euclidean spaces, i.e. $X = Y = \mathbb{R}^{2n}$. For a mapping from one normed space Y into another X the relative condition number of G in $y_0 \in Y$ is defined by

$$\kappa = \lim_{\delta \to 0} \max_{\|h\|=\delta} \frac{\|y_0\|}{\|G(y_0)\|} \frac{\|G(y_0+h) - G(y_0)\|}{\delta} = \|D_y G(y_0)\| \frac{\|y_0\|}{\|x_0\|}, \tag{2.24}$$

where differentiability (existence of the limit) is assumed and we set $x_0 = G(y_0)$. To determine G one looks at the mapping F

$$\begin{aligned} &F : X \to Y, \\ &F(w_1, \dots, w_n, e_1, \dots, e_n) = Ew = y_0 = (m_0, \dots, m_{2n-1})^T \end{aligned} \tag{2.25}$$

with the notation used in (2.8). If there is a unique solution for $2n$ given moments one can define the inverse mapping F^{-1} in a neighbourhood of the exact solution. This unique solution exists, because of Theorem 3.5 and Theorem 3.6 below and one has $G = F^{-1}$. Therefore the condition number (2.24) now changes to

$$\kappa = \|D_y G(y_0)\| \frac{\|y_0\|}{\|x_0\|} = \|(D_x F(x_0))^{-1}\| \frac{\|y_0\|}{\|x_0\|}$$

with $x_0 = (w_1, \dots, w_n, e_1, \dots, e_n)^T$. The Jacobian $D_x F(x_0)$ can be calculated to be

$$D_x F(x_0) := \begin{pmatrix} 1 & \dots & 1 & 0 & \dots & 0 \\ e_1 & \dots & e_n & w_1 & \dots & w_n \\ e_1^2 & \dots & e_n^2 & 2e_1 w_1 & \dots & 2e_n w_n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ e_1^{2n-1} & \dots & e_n^{2n-1} & (2n-1)e_1^{2n-2} w_1 & \dots & (2n-1)e_n^{2n-2} w_n \end{pmatrix} \in \mathbb{R}^{2n \times 2n}. \tag{2.26}$$

This is exactly the system matrix (2.22) obtained for the *DQMOM*. It can be written as a product of two matrices, i.e.

$$D_x F = \mathcal{E}\mathcal{W},$$

$$\mathcal{E} := \begin{pmatrix} 1 & \dots & 1 & 0 & \dots & 0 \\ e_1 & \dots & e_n & 1 & \dots & 1 \\ e_1^2 & \dots & e_n^2 & 2e_1 & \dots & 2e_n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ e_1^{2n-1} & \dots & e_n^{2n-1} & (2n-1)e_1^{2n-2} & \dots & (2n-1)e_n^{2n-2} \end{pmatrix} \in \mathbb{R}^{2n\times 2n}, \tag{2.27}$$

$$\mathcal{W} := \begin{pmatrix} 1 & 0 & & \dots & & 0 \\ 0 & 1 & & & & \\ & & \ddots & & & \\ \vdots & & & 1 & & \vdots \\ & & & & w_1 & \\ & & & & & \ddots \quad 0 \\ 0 & & & \dots & 0 & w_n \end{pmatrix} \in \mathbb{R}^{2n\times 2n}. \tag{2.28}$$

So the condition number is

$$\kappa = \|\mathcal{W}^{-1}\mathcal{E}^{-1}\| \frac{\|y_0\|}{\|x_0\|}. \tag{2.29}$$

The vector norm is chosen to be the maximum norm $\|x\| = \max_k |x_k|$ and hence the induced matrix norm is the maximum row sum norm

$$\|A\| = \max_i \sum_j |a_{ij}|.$$

In [5] the basic interval is $(0,1)$ and hence $e_i \in (0,1)$, for all $i = 1,\dots,n$. It is obvious that $\|y_0\| \geq m_0$. Furthermore one can conclude for the weights that if

$$\left.\begin{array}{l} w_i > 0, \text{ for all } i = 1,\dots,n \\ \displaystyle\sum_{i=1}^{n} w_i = m_0 \end{array}\right\} \Rightarrow w_i < m_0, \text{ for all } i = 1,\dots,n.$$

Altogether this fact implies

$$\|x_0\| = \|(w_1,\dots,w_n,e_1,\dots,e_n)\| = \max_{i=1,\dots,n}\{w_i, e_i\} < \max\{m_0, 1\}.$$

Since $\mathcal{W}$ is a (positive) diagonal matrix one has for the inverse

$$\|\mathcal{W}^{-1}\| = \max_{i=1,\dots,n}\left\{1, \frac{1}{w_i}\right\} \geq \max\left\{1, \frac{1}{m_0}\right\} \geq \min\left\{1, \frac{1}{m_0}\right\},$$

and one obtains for the product

$$\begin{aligned} \|\mathcal{W}^{-1}\mathcal{E}^{-1}\| &= \max_{i=1,\dots,2n} \sum_{j=1}^{2n} \left|\left(\mathcal{W}^{-1}\mathcal{E}^{-1}\right)_{ij}\right| \\ &= \max\left\{ \max_{i=1,\dots,n} \sum_{j=1}^{2n} \left|\left(\mathcal{E}^{-1}\right)_{ij}\right|, \max_{i=n+1,\dots,2n} \frac{1}{w_{i-n}} \sum_{j=1}^{2n} \left|\left(\mathcal{E}^{-1}\right)_{ij}\right| \right\} \\ &\geq \min\left\{1, \frac{1}{m_0}\right\} \|\mathcal{E}^{-1}\|. \end{aligned}$$

Combining these results with (2.29) leads to

$$\kappa = \|\mathcal{W}^{-1}\mathcal{E}^{-1}\|\frac{\|y_0\|}{\|x_0\|} > \frac{m_0 \min\left\{1, \frac{1}{m_0}\right\}}{\max\{1, m_0\}}\|\mathcal{E}^{-1}\| = \min\left\{m_0, \frac{1}{m_0}\right\}\|\mathcal{E}^{-1}\|. \tag{2.30}$$

It remains to determine a lower bound for $\|\mathcal{E}^{-1}\|$. In terms of Gautschi $\mathcal{E}$ is a *confluent Vandermonde* matrix [4] and the following theorem is applied.

Theorem 2.1 (Bounds for the Inverse of a Confluent Vandermonde Matrix)
Let $e_1, \dots, e_n$ be mutually distinct positive numbers and $\mathcal{E}$ be the matrix defined in (2.27). Then

$$u_1 \leq \|\mathcal{E}^{-1}\| \leq \max(u_1, u_2), \tag{2.31}$$

where the maximum row sum norm is used and for $l = 1, 2$

$$u_l = \max_{i=1,\dots,n} b_i^{(l)} \prod_{j=1;j\neq i}^{n} \left(\frac{1+e_j}{e_i - e_j}\right)^2, \tag{2.32}$$

$$b_i^{(1)} := 1 + e_i, \quad b_i^{(2)} := \left|1 + 2e_i \sum_{j=1;j\neq i}^{n} \frac{1}{e_i - e_j}\right| + 2\left|\sum_{j=1;j\neq i}^{n} \frac{1}{e_i - e_j}\right|.$$

Proof: Gautschi proved in [4] that

$$\mathcal{E}^{-1} = \begin{pmatrix} A \\ B \end{pmatrix},$$

where $A = (a_{ik})$, $B = (b_{ik})$ are $n \times 2n$-matrices satisfying

$$\sum_{k=1}^{2n} |a_{ik}| \leq b_i^{(2)} \prod_{j=1;j\neq i}^{n} \left(\frac{1+e_j}{e_i - e_j}\right)^2, \quad \sum_{k=1}^{2n} |b_{ik}| = b_i^{(1)} \prod_{j=1;j\neq i}^{n} \left(\frac{1+e_j}{e_i - e_j}\right)^2. \tag{2.33}$$

With

$$\alpha := \max_{i=1,\dots,n} \sum_{k=1}^{2n} |a_{ik}|, \quad \beta := \max_{i=1,\dots,n} \sum_{k=1}^{2n} |b_{ik}|,$$

(2.32) and (2.33) it follows that $\alpha \leq u_2$ and $\beta = u_1$. Now, if $\alpha \leq \beta$ it follows that $\|\mathcal{E}^{-1}\| = \beta = u_1$. If conversely $\alpha > \beta$ the result is $u_1 = \beta < \|\mathcal{E}^{-1}\| = \alpha \leq u_2$ and the theorem is proved. □
It should be remarked that Gautschi showed that there are cases where these bounds are attained by certain matrices [4]. Using Theorem 2.1 together with (2.30) one obtains the final result

$$\kappa > \min\left(m_0, \frac{1}{m_0}\right) \max_{i=1,\dots,n} \left[(1+e_i) \prod_{j=1;j\neq i}^{n} \left(\frac{1+e_j}{e_i - e_j}\right)^2\right]. \tag{2.34}$$

It is now obvious that if the abscissas lie close to each other the problem is badly conditioned. Furthermore Gautschi derived in [5] an approximate lower bound, i.e.

$$\kappa \gtrsim \min\left(m_0, \frac{1}{m_0}\right) \exp(3.5n). \tag{2.35}$$

So the condition number is already very large for small n, e.g. for $n = 3$ and $m_0 = 1$ one has $\kappa > 36\,315$. Therefore an alternative algorithm is needed to avoid a direct calculation of the solution to the nonlinear system introduced in 2.2.
How can this be applied to the *DQMOM*? It was previously shown that the matrix of the linear

system slightly changes if the *weighted abscissae* variable is not introduced. This matrix is exactly the Jacobian (2.26) and hence the linear system has the same bad condition number (2.34). Even if the *weighted abscissae* is introduced we obtain by analogous calculations the matrix

$$A := \begin{pmatrix} 1 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 1 \\ -e_1^2 & \dots & -e_n^2 & 2e_1 & \dots & 2e_n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -(2n-2)e_1^{2n-1} & \dots & -(2n-2)e_n^{2n-1} & (2n-1)e_1^{2n-2} & \dots & (2n-1)e_n^{2n-2} \end{pmatrix} \in \mathbb{R}^{2n\times 2n}. \tag{2.36}$$

It can be factorised into two matrices

$$A = \mathcal{E}\mathcal{V},$$

$$\mathcal{E} := \begin{pmatrix} 1 & \dots & 1 & 0 & \dots & 0 \\ e_1 & \dots & e_n & 1 & \dots & 1 \\ e_1^2 & \dots & e_n^2 & 2e_1 & \dots & 2e_n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ e_1^{2n-1} & \dots & e_n^{2n-1} & (2n-1)e_1^{2n-2} & \dots & (2n-1)e_n^{2n-2} \end{pmatrix} \in \mathbb{R}^{2n\times 2n},$$

$$\mathcal{V} := \begin{pmatrix} 1 & 0 & 0 & & & \dots & & 0 \\ 0 & 1 & 0 & & & \dots & & 0 \\ \vdots & \ddots & \ddots & & & & & \vdots \\ \vdots & & \ddots & \ddots & & & & \vdots \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ -e_1 & 0 & \dots & & 0 & 1 & 0 & \dots & 0 \\ 0 & -e_2 & 0 & & & & \ddots & & 0 \\ \vdots & \ddots & \ddots & & & & & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & & & \ddots & \vdots \\ 0 & \dots & \dots & 0 & -e_n & 0 & \dots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{2n\times 2n}. \tag{2.37}$$

Now we just have to exchange the lower bound of $\mathcal{W}^{-1}$ by the lower bound of $\mathcal{V}^{-1}$, i.e.

$$\|\mathcal{V}^{-1}\| = 1 + \max_{i=1,\dots,n} e_i > 1.$$

The condition number therefore changes to

$$\kappa > \min(m_0, 1) \max_{i=1,\dots,n} (1+e_i) \prod_{j=1;j\neq i}^{n} \left(\frac{1+e_j}{e_i - e_j}\right)^2. \tag{2.38}$$

In the case of $m_0 < 1$ the lower bounds for the condition numbers (2.34) and (2.38) are equal and in the other case $m_0 > 1$ (2.34) is by a factor $1/m_0$ smaller than (2.38). Since the linear system is an important part of the *DQMOM* one should try to improve the condition of this system.

3 Algorithms for Gaussian Quadrature

In this chapter we want to introduce four methods that can be used to calculate the weights and abscissas for the *QMOM*. The methods can basically be classified in two groups. The first three algorithms determine the coefficients of a recurrence relation for the orthogonal polynomials corresponding to the weight function $f(x)$. We use this notation, because the *PSD* that is given by (1.1) will be this weight function. With these coefficients the weights and abscissas can be obtained by solving an eigenvalue problem. The last method is the classical Newton iteration for a nonlinear system of equations. Since the algorithms use a lot of quadrature theory, the most important results will be briefly presented at the beginning.

3.1 Gaussian Quadrature

For the following results we refer to [3], but these can also be found in other numerical standard literature. The Gaussian quadrature tries to increase the order of the approximation

$$\int_a^b g(x)f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} g(x_i)w_i$$

by not choosing equidistant abscissas. One could also say that one tries to optimise the order of approximation by letting abscissas and weights be $2n$ degrees of freedom. At first we want to give a definition for the weight function.

Definition 3.1 (Weight Function)
A function f is called weight function on $[a,b] \subset \mathbb{R}$, if the following conditions are true

(i) *f must be measurable and non negative on $[a,b]$.*

(ii) *All moments*

$$m_k = \int_a^b x^k f(x)\,\mathrm{d}x, \quad k = 0, 1, \ldots$$

exist and are finite.

(iii) *For all polynomials $s(x) \geq 0$, for all $x \in [a,b]$ with*

$$\int_a^b s(x)f(x)\,\mathrm{d}x = 0$$

follows $s(x) \equiv 0$.

Remark 3.2
If $f \in C^0([a,b], \mathbb{R}_+)$, then the conditions in Definition 3.1 are met. Condition (iii) is equivalent to

$$0 < m_0 = \int_a^b f(x)\,\mathrm{d}x.$$

Since f is positive, one can define an inner product in

$$L^2_f([a,b]) := \left\{ g \in L^2([a,b]) \; : \quad \int_a^b g(x)^2 f(x)\,\mathrm{d}x < \infty \right\}.$$

Definition 3.3 (Inner Product)
Let f be a weight function as in Definition 3.1. For two functions g, $h \in L_f^2([a,b])$ the inner product is defined by

$$\langle g, h\rangle := \int_a^b g(x)h(x)f(x)\,\mathrm{d}x.$$

The following result is important for the algorithms in Sections 3.2, 3.3 and 3.4.

Theorem 3.4 (Unique System of Orthogonal Polynomials)
For $j = 0, 1, \ldots$ exist unique polynomials

$$p_j(x) = x^j + \sum_{l=0}^{j-1} a_{j-l}x^l \quad \text{with} \quad \langle p_i, p_k\rangle = 0, \quad i \neq k.$$

These polynomials satisfy the recursion

$$\begin{aligned} p_{-1}(x) &:= 0 \\ p_0(x) &:= 1 \\ p_{i+1}(x) &:= (x-\beta_i)p_i(x) - \alpha_i^2 p_{i-1}(x), \quad i = 0, 1, \ldots\ . \end{aligned} \tag{3.1}$$

One has for the coefficients

$$\beta_i = \frac{\langle xp_i, p_i\rangle}{\langle p_i, p_i\rangle},\ i \geq 0, \quad \alpha_i^2 = \begin{cases} 1, & i = 0, \\ \dfrac{\langle p_i, p_i\rangle}{\langle p_{i-1}, p_{i-1}\rangle}, & i = 1, \ldots\ . \end{cases}$$

Note that the uniqueness comes from the requirement that the coefficient of x^j in p_j is set to be one. Theorem 3.4 provides uniqueness of the orthogonal polynomials and hence also for coefficients in the recursion. Furthermore one clearly sees, that the square root α_i is well defined since the square is equal to one or a fraction consisting of positive definite inner products. Furthermore one can conclude that all polynomials up to degree $n-1$ are orthogonal to p_n, since they can be written as a linear combination of the p_j, $j = 0, 1, \ldots, n-1$. The next result is another step in proving uniqueness of the weights and abscissas in the quadrature rule.

Theorem 3.5 (Uniqueness of the Abscissas)
The roots x_i, $i = 1, \ldots, n$, of p_n are real, simple and are located in the open interval (a, b).

Now the next Theorem guaranties the uniqueness and positivity of the weights. The positivity was already used in Section 2.4.

Theorem 3.6 (Uniqueness & Positivity of the Weights)
(1) Let $x_1, \ldots, x_n$ be the roots of p_n and $w_1, \ldots, w_n$ the solution of the linear system

$$\sum_{i=1}^{n} p_k(x_i)w_i = \begin{cases} \langle p_0, p_0\rangle, & \text{if } k = 0, \\ 0, & \text{if } k = 1, 2, \ldots, n-1. \end{cases} \tag{3.2}$$

Note that this system is of full rank and there exists a unique solution. Then the weights are positive, i.e. $w_i > 0$ for $i = 1, \ldots, n$, as well as

$$\int_a^b p(x)f(x)\,\mathrm{d}x = \sum_{i=1}^{n} w_i p(x_i) \tag{3.3}$$

for all polynomials up to degree $2n-1$.

(2) If conversely (3.3) is true for certain real numbers w_i, x_i, $i = 1,\dots,n$ *and all polynomials up to degree* $2n-1$, *it follows that the* x_i *are the roots of* p_n *and the* w_i *solve the linear system (3.2).*

(3) There are no real numbers w_i, x_i, $i = 1,\dots,n$ *such that (3.3) is valid for all polynomials up to degree* $2n$.

The theory of orthogonal polynomials is connected to tridiagonal matrices. If one writes the coefficients of the relation (3.1) in the following way in a matrix

$$A_n = \begin{pmatrix} \beta_0 & \alpha_1 & 0 & \dots & \dots & 0 \\ \alpha_1 & \beta_1 & \alpha_2 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & \alpha_{n-2} & \beta_{n-2} & \alpha_{n-1} \\ 0 & \dots & \dots & 0 & \alpha_{n-1} & \beta_{n-1} \end{pmatrix}. \tag{3.4}$$

then the polynomials satisfy $p_i(x) \equiv \det(A_i - xI)$. We give the following very important theorem.

Theorem 3.7 (Correspondence to Tridiagonal Matrices)
The roots x_i, $i = 1,\dots,n$, *of the* n*-th orthogonal polynomial* p_n *are the eigenvalues of the matrix* A_n *(3.4). Furthermore it is*

$$w_k = (v_{k1})^2, \quad k = 1,\dots,n,$$

where v_{k1} *denotes the first component of the* k*-th eigenvector corresponding to the eigenvalue* x_k

$$A_n v_k = x_k v_k.$$

The eigenvector is normalised, such that

$$v_k^T v_k = \langle p_0, p_0 \rangle = \int_a^b f(x)\,\mathrm{d}x.$$

We close this rough presentation of results for Gaussian quadrature with a result for the approximation error.

Theorem 3.8 (Approximation Error)
For a function $g \in C^{2n}([a,b])$ *one has*

$$\int_a^b g(x)f(x)\,\mathrm{d}x - \sum_{i=1}^{n} w_i g(x_i) = \frac{g^{(2n)}(\xi)}{(2n)!}\langle p_n, p_n \rangle$$

with a $\xi \in (a,b)$.

3.2 Product-Difference-Algorithm

3.2.1 The Algorithm

The *Product-Difference-Algorithm* (*PDA*) was introduced in 1968 by Gordon [7]. We will present this algorithm and prove its correctness. The algorithm transforms a sequence of moments into coefficients of a continued fraction. These coefficients can be used to calculate the weights and abscissas via a corresponding eigenvalue problem. In the first step of the *PDA* a matrix $B = (b_{ij}) \in \mathbb{R}^{(2n+1)\times(2n+1)}$ is initialised. The elements of the first and second column are set as follows

$$\begin{aligned} b_{i1} &= \delta_{i1}, \quad i = 1, \dots, 2n+1, \\ b_{i2} &= (-1)^{i-1} m_{i-1}, \quad i = 1, \dots, 2n, \\ b_{2n+1,2} &= 0, \end{aligned}$$

where δ_{ij} is the Kronecker delta. It is possible to choose $m_0 = 1$ and rescale at the end of the algorithm. It is important that these moments are the moments of a weight function with compact support in the positive real axis. This algorithm will fail for example for the *Gauss Hermite quadrature* on $(-\infty, +\infty)$. The other components are obtained by applying the following rule

$$b_{ij} = \begin{cases} b_{1,j-1} b_{i+1,j-2} - b_{1,j-2} b_{i+1,j-1}, & j = 3, \dots, 2n+1,\ i = 1, \dots, 2n+2-j, \\ 0, \quad else. \end{cases} \tag{3.5}$$

Altogether the matrix looks like

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

In the next step the coefficients c_i are determined

$$c_i = \begin{cases} m_0, & i = 1, \\ \dfrac{b_{1,i+1}}{b_{1i} b_{1,i-1}}, & i = 2, \dots, 2n. \end{cases} \tag{3.6}$$

Now one can construct a symmetric tridiagonal matrix $A_n = (a_{ij}) \in \mathbb{R}^{n\times n}$. This is nearly matrix (3.4) mentioned before. The elements are given by

$$\begin{aligned} \beta_{i-1} &= \begin{cases} c_2, & i = 1, \\ c_{2i} + c_{2i-1}, & i = 2, \dots, n, \end{cases} \\ \alpha_i &= -\sqrt{c_{2i+1} c_{2i}}, \quad i = 1, \dots, n-1. \end{aligned} \tag{3.7}$$

The minus sign of the off-diagonal entries does not affect the eigenvalues, since the characteristic polynomial only depends on the squares of these elements. The weights and abscissas are now given by the eigenvectors and corresponding eigenvalues of the matrix ($Av_i = e_i v_i$), see Theorem 3.7. Specifically the weights are given by $w_i = m_0 v_{i1}^2$. Here v_{i1} denotes the first component of the i-th eigenvector.

3.2.2 Proof of Correctness of the PDA

To prove the correctness of the *PDA* an intensive use of the theory of continued fractions is necessary. Therefore we will refer to the book [19]. The idea behind this argumentation can be visualised by the following scheme

$$\begin{array}{c} \textit{Stieltjes Transform of } f \overset{(1)}{\leftrightarrow} \textit{Continued Fraction} \overset{(2)}{\leftrightarrow} \textit{Tridiagonal Matrix} \overset{(3)}{\leftrightarrow} \textit{Gaussian Quadrature}. \\ (4) \updownarrow \\ \textit{PDA Recursion (3.5)} \end{array} \tag{3.8}$$

The arrows just state that there is a connection between these topics and shall underline the idea behind the proof. We do not claim that these are strict logical equivalent connections.
For (1) we refer to [19] *Chapter XIII* and omit the details. It is shown that every suited function corresponding to a *positive definite* continued fraction can be expressed as a *Stieltjes Transform*, *Theorem 66.1* [19]. This is especially true for the approximants of certain *positive definite* continued fractions, (67.1) [19]. It is very important to talk about the condition *positive definite*. In [19] *Chapter IV* a continued fraction is said to be positive definite if a certain associated quadratic form is positive definite. We again omit the explicit details. But in our case this quadratic form is induced by the matrix A_n given by (3.7)

$$Q(\xi) := \xi^T A_n \xi, \quad \xi \in \mathbb{R}^n.$$

This quadratic form is positive definite if and only if the eigenvalues of this matrix are positive definite. Therefore the support of the weight function must lie in the positive real axis. This is a very important restriction to the *PDA*. The following two algorithms do not need this restriction. One could state that this is not important for the practical case since the *PSD* depends for example on the diameter and therefore the abscissas must be positive. But this is an important weak point in this algorithm, since the abscissas can become negative due to numerical errors.
Step (2) is a bit easier. It is shown in [7] and in [19] *Chapter XII* how a continued fraction corresponds to a tridiagonal matrix.
The third step, (3), was given above in Section 3.1. Theorem 3.7 states that the weights and abscissas can be obtained via an eigenvalue problem of a tridiagonal matrix. The entries of this matrix are the coefficients of the recurrence relation (3.1) for the system of orthogonal polynomials corresponding to the weight function.
In the *PDA* the coefficients of the matrix need to be calculated. Therefore the coefficients of the continued fraction are needed. Step (4) gives these coefficients via the recursion (3.5). This step will be explained in detail now. In the beginning we start with the integral

$$I(z) := \int_0^\infty \frac{f(\xi)}{z+\xi} \, \mathrm{d}\xi$$

since it corresponds to a certain type of continued fraction, [19]. Using the series expansion

$$\frac{1}{z+\xi} = \sum_{i=1}^{2n} \frac{(-1)^{i-1}\xi^{i-1}}{z^i} + \underbrace{\frac{\xi^{2n}}{z^{2n}(z+\xi)}}_{=:R_{2n}}$$

results in

$$\begin{aligned} I(z) &= \int_0^\infty f(\xi) \left(\sum_{i=1}^{2n} \frac{(-1)^{i-1}\xi^{i-1}}{z^i} + R_{2n} \right) \mathrm{d}\xi = \sum_{i=1}^{2n} \frac{(-1)^{i-1}}{z^i} \int_0^\infty \xi^{i-1} f(\xi) \, \mathrm{d}\xi + \int_0^\infty R_{2n} f(\xi) \, \mathrm{d}\xi \\ &= \sum_{i=1}^{2n} \frac{(-1)^{i-1} m_{i-1}}{z^i} + \int_0^\infty R_{2n} f(\xi) \, \mathrm{d}\xi. \end{aligned}$$

Here we used Definition 3.1 (ii). The first part of this shall now be expanded into a continued fraction. This is the first step. For this purpose it must be reformulated into a rational function with $\deg(\tilde{P}_1) = 2n$ and $\deg(\tilde{P}_2) = 2n-1$

$$C(z) = \sum_{i=1}^{2n} \frac{(-1)^{i-1} m_{i-1}}{z^i} = \frac{\sum_{i=1}^{2n} (-1)^{i-1} m_{i-1} z^{2n-i}}{z^{2n}} =: \frac{\tilde{P}_2(z)}{\tilde{P}_1(z)}.$$

In the next step we use a division procedure, i.e.

$$\tilde{P}_1(z) = r_1(z)\tilde{P}_2(z) + \tilde{P}_3(z).$$

This results in

$$\frac{\tilde{P}_2(z)}{\tilde{P}_1(z)} = \frac{1}{\frac{\tilde{P}_1(z)}{\tilde{P}_2(z)}} = \frac{1}{r_1(z) + \frac{\tilde{P}_3(z)}{\tilde{P}_2(z)}}.$$

In the first division the results are

$$r_1(z) = \frac{1}{m_0} z, \quad \tilde{P}_3(z) = \frac{1}{m_0} \sum_{i=1}^{2n-1} (-1)^{i+1} m_i z^{2n-i}.$$

So one only divides the terms of the highest power. The second division gives the following results

$$\begin{aligned} \tilde{P}_2(z) &= r_2(z)\tilde{P}_3(z) + \tilde{P}_4(z), \\ r_2(z) &= \frac{m_0^2}{m_1}, \quad \tilde{P}_4(z) = \sum_{i=1}^{2n-2} (-1)^i \left(m_i - \frac{m_0}{m_1} m_{i+1} \right) z^{2n-1-i} - m_{2n-1}. \end{aligned}$$

The degree is decreased at least by one in every second division and hence this process will terminate. In general, we define that the coefficients of each polynomial $\tilde{P}_j$ are denoted as $\tilde{b}_{ij}$, where $\tilde{b}_{1j}$ is the coefficient of the highest power of $\tilde{P}_j$. Furthermore we state for these coefficients

$$\tilde{b}_{ij} = 0, \quad \text{for all} \quad j = 3, \ldots, 2n+1,\ i = 2n+3-j, \ldots, 2n+1.$$

These polynomials satisfy the following relation by construction

$$\tilde{P}_{j-1}(z) = r_{j-1}(z)\tilde{P}_j(z) + \tilde{P}_{j+1}(z), \quad r_{j-1}(z) = \frac{\tilde{b}_{1,j-1}}{\tilde{b}_{1,j}} z^{deg(\tilde{P}_{j-1}) - deg(\tilde{P}_j)}.$$

One can explicitly write down the coefficients of $\tilde{P}_{j+1}$

$$\tilde{P}_{j+1}(z) = \tilde{P}_{j-1}(z) - r_{j-1}\tilde{P}_j(z) \quad \Leftrightarrow \quad \tilde{b}_{i,j+1} = \tilde{b}_{i+1,j-1} - \frac{\tilde{b}_{1,j-1}}{\tilde{b}_{1,j}} \tilde{b}_{i+1,j}. \tag{3.9}$$

The continued fraction for now looks like

$$C(z) = \cfrac{1}{r_1(z) + \cfrac{1}{r_2(z) + \cfrac{1}{r_3(z) + \ldots}}}.$$

In the next step (3.9) shall be modified, therefore we must expand the specific fraction with $\tilde{b}_{1j}$, i.e.

$$r_{j-1} + \frac{1}{\frac{\tilde{P}_j}{\tilde{P}_{j+1}}} = r_{j-1} + \frac{1}{\underbrace{\frac{\tilde{b}_{1,j}\tilde{P}_j}{\tilde{b}_{1,j}\tilde{P}_{j+1}}}_{=:P_{j+1}}}.$$

The recursion for the coefficients of the new polynomials P_{j+1} is (3.5)

$$b_{i,j+1} = b_{1,j}b_{i+1,j-1} - b_{1,j-1}b_{i+1,j}.$$

So we have derived the recursion (3.5) in order to construct the continued fraction. Now we normalise each r_{j-1} by setting the coefficient of the highest order to one and obtain

$$C(z) = \cfrac{\frac{b_{1,2}}{b_{1,1}}}{z + \cfrac{\frac{b_{1,3}}{b_{1,2}b_{1,1}}}{1 + \cfrac{\frac{b_{1,4}}{b_{1,3}b_{1,2}}}{z + \dots}}} = \cfrac{c_1}{z + \cfrac{c_2}{1 + \cfrac{c_3}{z + \dots}}} \quad .$$

From this calculation we obtain the first two formulae used in the *PDA*, i.e. (3.5) and (3.6). Now to the second step in this proof. It remains to deduce the eigenvalue problem, namely the tridiagonal matrix with the proper coefficients. Therefore one needs to define the even and odd part of a continued fraction. By [19] the even part is understood as the continued fraction whose sequence of approximants is the even sequence of approximants of the given continued fraction. Similarly for the odd part. If the approximants of $C(z)$ would be denoted with $C_1, C_2, C_3, \dots$ the approximants of C_{even} would be $C_2, C_4, \dots$ and analogously for C_{odd} $C_1, C_3, \dots$. We will give the first four approximants explicitly

$$C_1(z) = \frac{c_1}{z}, \quad C_2(z) = \cfrac{c_1}{z + c_2}$$
$$C_3(z) = \cfrac{c_1}{z + \cfrac{c_2}{1 + \cfrac{c_3}{z}}}, \quad C_4(z) = \cfrac{c_1}{z + \cfrac{c_2}{1 + \cfrac{c_3}{z + c_4}}}.$$

It is noted in [19] and also in [7] that the even part is a lower and the odd part an upper bound for the integral we started with. Furthermore it is shown in [7] that the following calculations can also be done with the odd part. The result will slightly differ in the coefficients that are needed. So we continue according to [7]. Taking the even approximants of this continued fraction one can write as in [7] and [19]

$$C_{even}(z) = \cfrac{c_1}{z + c_2 - \cfrac{c_2c_3}{z + c_3 + c_4 - \cfrac{c_4c_5}{z + c_5 + c_6 \dots}}} \quad .$$

Now to step (2) of (3.8). It is shown in [19] and [7] that C_{even} is the solution x_1 to the following problem

$$\begin{pmatrix} z + c_2 & -\sqrt{c_2c_3} & 0 & 0 & 0 & \dots \\ -\sqrt{c_2c_3} & z + c_3 + c_4 & -\sqrt{c_4c_5} & 0 & 0 & \dots \\ 0 & -\sqrt{c_4c_5} & z + c_5 + c_6 & -\sqrt{c_6c_7} & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} c_1 \\ 0 \\ 0 \\ \vdots \end{pmatrix}.$$

This equation can be written as

$$(zId + A_n)\mathbf{x} = c_1\mathbf{e}_1$$

and therefore the solution is

$$\mathbf{x} = c_1(zId + A_n)^{-1}\mathbf{e}_1.$$

We will give an example for $n = 2$. The system is

$$\begin{pmatrix} z + c_2 & -\sqrt{c_2c_3} \\ -\sqrt{c_2c_3} & z + c_3 + c_4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ 0 \end{pmatrix}.$$

Therefore one has the solution

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{1}{(z + c_2)(z + c_3 + c_4) - c_2c_3} \begin{pmatrix} z + c_3 + c_4 & \sqrt{c_2c_3} \\ \sqrt{c_2c_3} & z + c_2 \end{pmatrix} \begin{pmatrix} c_1 \\ 0 \end{pmatrix}.$$

So one obtains for x_1

$$x_1 = \frac{c_1(z + c_3 + c_4)}{(z + c_2)(z + c_3 + c_4) - c_2c_3} = \frac{c_1}{z + c_2 - \dfrac{c_2c_3}{z + c_3 + c_4}}$$

and this is the fraction C_{even} given above for $n = 2$. Since A_n is a symmetric tridiagonal matrix it can be transformed to a diagonal matrix Ξ by an orthogonal transformation matrix V and one gets

$$\begin{aligned} \mathbf{x} &= c_1VV^{-1}(zId + A_n)^{-1}VV^{-1}\mathbf{e}_1 \\ &= c_1V(V^{-1}(zId + A_n)V)^{-1}V^{-1}\mathbf{e}_1 \\ &= c_1V(zId + \Xi)^{-1}V^{-1}\mathbf{e}_1, \\ x_1 &= c_1 \sum_{i=1}^{n} \frac{1}{z + e_i} V_{1i}^2. \end{aligned}$$

In the last step we used the fact that V is an orthogonal transformation and denoted the eigenvalues of Ξ with e_i, $i = 1, \ldots, n$. Hence we have two representations for $x_1 = C_{even}$ and therefore step (3) of (3.8) is verified

$$I(z) = \int_0^\infty \frac{f(\xi)}{z + \xi}\, \mathrm{d}\xi \approx \sum_{i=1}^{n} \frac{c_1V_{1i}^2}{z + e_i}.$$

This is the n point Gaussian quadrature with abscissas e_i and weights

$$w_i = c_1V_{1i}^2 = m_0V_{1i}^2.$$

It is important to note that these quantities do not depend on z. Now one can write more general with Section 3.1

$$\int_0^\infty g(\xi)f(\xi)\, \mathrm{d}\xi \approx \sum_{i=1}^{n} w_ig(e_i).$$

The *PDA* needs $2n^2 - 1$ summations, $4n^2 + n - 2$ multiplications, $2n - 1$ divisions, $n - 1$ square roots and the solution of a n-dimensional eigenvalue problem.

3.3 Long Quotient-Modified Difference Algorithm

3.3.1 The Algorithm

The major part of the work with the *QMOM* used the *PDA* since it was suggested by McGraw. In the process of improving this method one should look for other possible algorithms which provide useful alternative features. The first algorithm which is discussed for this reason is the *Long Quotient - Modified Difference Algorithm* (*LQMD - Algorithm*). It was first discussed in 1972 by Sack and Donovan in [15]. There are two advantages of this method. The first one is that it can be directly applied to so-called modified moments which can increase the numerical stability. Second, when used for standard moments, i.e. powers of the internal variable e^k, the number of operations is decreased. Furthermore it can also be applied to quadratures with negative abscissas. We will present the complete algorithm and the special case, when applied to standard moments.
Consider the real weight function $f(x)$ and its modified moments

$$\nu_l = \int_a^b P_l(x) f(x)\,\mathrm{d}x, \quad l = 0, 1, \ldots, \tag{3.10}$$

where P_l are polynomials of degree l satisfying a three term recurrence relation with known coefficients

$$xP_l(x) = a_l P_{l+1}(x) + b_l P_l(x) + c_l P_{l-1}, \quad l = 0, 1, \ldots. \tag{3.11}$$

Again the tridiagonal matrix (3.4) is established from which the weights and abscissas can be calculated. Analogous to the matrix B in the *PDA*, a matrix $B \in \mathbb{R}^{(n+1)\times 2n}$ is derived. There are two rows given initially

$$b_{1,j} := s_{-1,j} = 0, \quad b_{2,j} := s_{0,j} = \frac{\nu_{j-1}}{\nu_0}, \quad j = 1, \ldots, 2n.$$

These can be used to calculate three coefficients

$$\begin{aligned}
\tau_i &= a_{i-1}, && i = 0, \ldots, n-2 \\
\sigma_i &= a_i s_{i,i+1} + b_i - a_{i-1} s_{i-1,i}, && i = 0, \ldots, n-1 \\
\rho_i &= (b_{i+1} - \sigma_i) s_{i,i+1} + a_{i+1} s_{i,i+2} - a_{i-1} s_{i-1,i+1} + c_{i+1}, && i = 0, \ldots, n-2.
\end{aligned} \tag{3.12}$$

Then the new row can be determined by

$$\begin{aligned}
s_{i+1,i+1} &= 1, & s_{i+1,j} &= \rho_i^{-1}[(b_j - \sigma_i) s_{i,j} + a_j s_{i,j+1} + c_j s_{i,j-1} - \tau_i s_{i-1,j}], \\
i &= 0, \ldots, n-2, & j &= i+2, \ldots, 2n-2-i
\end{aligned} \tag{3.13}$$

and all remaining values are set to zero. The fact that we set $s_{i+1,i+1}$ equal to one in (3.13) has purely computational reasons, since the coefficient ρ_i is chosen such that the result of the formula for $s_{i+1,j}$, $j = i+1$, would also be one. This can be seen in the proof below. With the new row one calculates new coefficients via (3.12) and a new row via (3.13) until B

$$B = \begin{pmatrix}
0 & & & & \cdots & & & 0 \\
1 & \frac{\nu_1}{\nu_0} & & & \cdots & & & \frac{\nu_{2n-1}}{\nu_0} \\
0 & 1 & s_{12} & & \cdots & & s_{1,2n-2} & 0 \\
\vdots & \ddots & \ddots & \ddots & & \iddots & \iddots & \vdots \\
0 & \cdots & 0 & 1 & s_{n-1,n} & 0 & \cdots & 0
\end{pmatrix}$$

is fully determined. We just introduced the matrix B to compare it with the matrix used in the *PDA*.
The $n \times n$ - tridiagonal matrix (3.4) has the elements

$$\begin{aligned} \beta_i &= \sigma_i, \quad i = 0, \ldots, n-1 \\ \alpha_{i+1}^2 &= a_{i-1}\rho_i = \rho_i \tau_{i+1}, \quad i = 0, \ldots, n-2. \end{aligned} \tag{3.14}$$

Now again, as in the *PDA*, the weights and abscissas can be obtained from the corresponding eigenvalues and eigenvectors.
For the standard moments the recurrence relation (3.11) has the coefficients $a_l = 1$, $b_l = c_l = 0$ and simplifies to

$$P_{l+1}(x) = xP_l(x).$$

Hence the formulae (3.12) and (3.13) simplify to

$$\begin{aligned} \tau_i &= 1, & & \\ \sigma_i &= s_{i,i+1} - s_{i-1,i}, & & i = 0, \ldots, n-1 \\ \rho_i &= -\sigma_i s_{i,i+1} + s_{i,i+2} - s_{i-1,i+1}, & & i = 0, \ldots, n-2 \\ s_{i+1,j} &= \rho_i^{-1}[-\sigma_i s_{i,j} + s_{i,j+1} - s_{i-1,j}], & & i = 0, \ldots, n-2 \\ & & & j = i+2, \ldots, 2n-2-i. \end{aligned}$$

and the matrix elements are

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

3.3.2 Proof of Correctness of the LQMD

To derive this procedure, Sack and Donovan ([15]) made use of the set of orthogonal polynomials associated to the weight function $f(x)$. These polynomials will be denoted with $T_i(x),\ i = -1, 0, 1, \dots$ and satisfy

$$xT_i(x) = \alpha_i T_{i+1}(x) + \beta_i T_i(x) + \gamma_i T_{i-1}(x), \quad i = 0, 1, \dots, \tag{3.15}$$

$$0 = \int_a^b T_i(x) T_j(x) f(x)\,\mathrm{d}x, \quad \text{for } i \neq j \tag{3.16}$$

with $T_{-1}(x) := 0$ and $T_0(x) := 1$. Since these polynomials can be scaled with an arbitrary multiplicative constant it is possible to obtain $\alpha_i = \gamma_{i+1}$ and the polynomials therefore satisfy

$$xT_i(x) = \alpha_i T_{i+1}(x) + \beta_i T_i(x) + \alpha_{i-1} T_{i-1}(x).$$

The following proof reformulates the eigenvalue problem for the matrix A_n (3.4) and derives the recursion formula in terms of the modified moments. Essentially a recurrence relation analogous to (3.11) for suited polynomials is established and it is shown that this is equivalent to the relation of the polynomials $T_i(x)$. Consider the eigenvalue problem

$$\chi_n(\lambda) = \det(A_n - \lambda Id) = 0. \tag{3.17}$$

The elements (i, j) of $A_n - \lambda Id$ can be written as

$$\int_a^b T_i(x) T_j(x)(x - \lambda) f(x)\,\mathrm{d}x,$$

this follows from the recursion (3.15) and property (3.16). Now each T_i can be written as a linear combination of the given polynomials P_l, $l \leq i$. Hence we find a infinite dimensional lower triangular constant matrix Q with non-zero elements such that

$$\mathbf{T} = Q\mathbf{P}.$$

Here $\mathbf{T}$ and $\mathbf{P}$ denote the coefficient vector of the corresponding set of polynomials, i.e.

$$\begin{aligned} T_i(x) &= \sum_{j=1}^{i} t_{ji} x^{j-1}, \quad i = 1, 2, \dots, \\ \mathbf{T} &= (t_{11}, t_{12}, t_{22}, t_{13}, t_{23}, \dots)^T, \\ P_i(x) &= \sum_{j=1}^{i} p_{ji} x^{j-1}, \quad i = 1, 2, \dots, \\ \mathbf{P} &= (p_{11}, p_{12}, p_{22}, p_{13}, p_{23}, \dots)^T. \end{aligned}$$

This equation still holds for a finite n. Now the eigenvalue problem (3.17) can be reformulated as

$$\det\left[Q_n(\Xi_n - \lambda N_n) Q_n^T\right] = 0. \tag{3.18}$$

The elements of Ξ_n and N_n are

$$\left.\begin{aligned} \xi_{ij} &= \int_a^b P_i(x) P_j(x) x f(x)\,\mathrm{d}x, \\ \nu_{ij} &= \int_a^b P_i(x) P_j(x) f(x)\,\mathrm{d}x, \end{aligned}\right\} \ i, j = 1, 2, \dots, n. \tag{3.19}$$

We will exemplarily calculate the element $(Q_n \Xi_n Q_n^T)_{ij}$

$$\begin{aligned}(Q_n \Xi_n Q_n^T)_{ij} &= \sum_{l=1}^{j} \sum_{k=1}^{i} q_{ik} \xi_{kl} q_{lj}^T \\ &= \sum_{l=1}^{j} \sum_{k=1}^{i} q_{ik} \xi_{kl} q_{jl} = \sum_{l=1}^{j} \sum_{k=1}^{i} q_{ik} \int_a^b P_k(x) P_l(x) x f(x)\,\mathrm{d}x\, q_{jl} \\ &= \sum_{l=1}^{j} \sum_{k=1}^{i} \int_a^b \left(\sum_{r=1}^{k} q_{ik} p_{rk} x^{j-1} \right) \left(\sum_{r=1}^{l} q_{jl} p_{rl} x^{j-1} \right) x f(x)\,\mathrm{d}x \\ &= \int_a^b T_i(x) T_j(x) x f(x)\,\mathrm{d}x.\end{aligned}$$

Since Q_n is non-singular (3.18) implies

$$\det[\Xi_n - \lambda N_n] = \det[N_n^{-1} \Xi_n - \lambda Id] = 0.$$

Hence the eigenvalues of A_n are equal to those of the asymmetric matrix $N_n^{-1}\Xi_n$. Why is this matrix asymmetric? Let X_∞ denote the infinite tridiagonal matrix corresponding to (3.11)

$$X_\infty = \begin{pmatrix} b_1 & c_2 & 0 & \dots & \dots & 0 \\ a_1 & b_2 & c_3 & 0 & \dots & \vdots \\ 0 & a_2 & b_3 & c_4 & & \vdots \\ 0 & & & \ddots & & \end{pmatrix}.$$

Now (3.11) and (3.19) imply

$$\Xi_\infty = N_\infty X_\infty = X_\infty^T N_\infty \tag{3.20}$$

and hence $N_n^{-1}\Xi_n$ is asymmetric. Again we will give a precise calculation

$$\begin{aligned}(N_\infty X_\infty)_{ij} &= \sum_{l=1}^{n} \nu_{il} x_{lj} \\ &= a_j \int_a^b P_i(x) P_{j+1}(x) f(x)\,\mathrm{d}x + b_j \int_a^b P_i(x) P_j(x) f(x)\,\mathrm{d}x + c_j \int_a^b P_i(x) P_{j-1}(x) f(x)\,\mathrm{d}x \\ &= \int_a^b P_i(x) [a_j P_{j+1}(x) + b_j P_j(x) + c_j P_{j-1}(x)] f(x)\,\mathrm{d}x \\ &= \int_a^b P_i(x) P_j(x) x f(x)\,\mathrm{d}x = \xi_{ij}.\end{aligned}$$

If the matrices are truncated for a finite n, the equation (3.20) is no longer true, since the element a_{n-1} is missing in X_n. Hence (3.20) is replaced by

$$\begin{aligned}\Xi_n &= N_n X_n + R_n, \\ N_n^{-1}\Xi_n &= X_n + Y_n.\end{aligned}$$

R_n and Y_n are matrices where only the last column is different form zero. The explicit elements of R_n are $a_{n-1}\nu_{in}$. Therefore the last column of Y_n, denoted by $\mathbf{y}^{(n)}$, is the solution to the following equation

$$\mathbf{r}^{(n)} = N_n \mathbf{y}^{(n)}. \tag{3.21}$$

So the eigenvalues of A_n are equal to those of $X_n + Y_n$ but the asymmetric form makes it expensive to diagonalise. Instead the elements of A_n are determined via recursion by the elements of $X_n + Y_n$. For this purpose the trace is used, since it is an invariant quantity

$$\begin{aligned}
\mathrm{Tr}(A_n) &= \sum_{k=1}^{n} \beta_k = \mathrm{Tr}(X_n + Y_n) = \sum_{k=1}^{n} b_k + y_n^{(n)}, \\
\mathrm{Tr}(A_n^2) &= \sum_{k=1}^{n} \beta_k^2 + 2\sum_{k=1}^{n-1} \alpha_k^2 \\
&= \sum_{k=1}^{n-1} b_k^2 + \left(b_n + y_n^{(n)}\right)^2 + 2\sum_{k=1}^{n-1} a_k c_{k+1} + 2a_{n-1} y_{n-1}^{(n)}.
\end{aligned} \tag{3.22}$$

Now if the β_k and α_{k-1}^2 are known for all $k < n$, β_n and α_{n-1}^2 can be obtained if $y_n^{(n)}$ and $y_{n-1}^{(n)}$ are known. By subtraction one can directly compute

$$\beta_i = \mathrm{Tr}(A_i) - \mathrm{Tr}(A_{i-1}) = b_i + y_i^{(i)} - y_{i-1}^{(i-1)}, \tag{3.23}$$

$$\begin{aligned}
\alpha_i^2 &= \frac{1}{2}\left[\mathrm{Tr}(A_{i+1}^2) - \mathrm{Tr}(A_i^2) - \beta_{i+1}^2\right] \\
&= a_i\left(c_{i+1} + y_i^{(i+1)}\right) + y_i^{(i+1)}\left(b_{i+1} - b_i + y_{i+1}^{(i+1)} - y_i^{(i)}\right) - a_{i-1} y_{i-1}^{(i)}.
\end{aligned} \tag{3.24}$$

Recall equation (3.21), $\mathbf{y}^{(n)}$ still remains a solution when this equation is multiplied with an arbitrary non-singular square matrix M_n

$$M_n \mathbf{r}^{(n)} = M_n N_n \mathbf{y}^{(n)} = S_n \mathbf{y}^{(n)}. \tag{3.25}$$

The multiplication means that multiple rows of N_n are added together. The elements s_{ij} of S_n are therefore given as integrals

$$s_{ij} = \int_a^b S_i(x) P_j(x) f(x)\,\mathrm{d}x, \tag{3.26}$$

where the functions $S_i(x)$ are polynomials of degree i (non-zero coefficient of x^i). Now Sack and Donovan ([15]) chose S_n to be the truncated form of an infinite upper triangular matrix with diagonal elements equal to unity

$$s_{ij} = 0,\ j < i, \quad s_{ii} = 1. \tag{3.27}$$

This implies, together with (3.26), that the polynomials $S_i(x)$ are orthogonal to all polynomials $P_j(x)$ with degree less than i. Since the polynomials $T_j(x)$ are linear combinations of all polynomials $P_l(x)$ up to degree $j < i$, these polynomials $S_i(x)$ are orthogonal to the $T_j(x)$ and hence they must be a constant multiple of $T_i(x)$. The elements of $M_n \mathbf{r}^{(n)}$ are given by $a_{n-1} s_{in}$, $i = 1, \ldots, n$. The $S_i(x)$ also satisfy a recurrence relation (since they are constant multiples) analogous to (3.15)

$$\begin{aligned}
x S_i(x) &= \rho_i S_{i+1}(x) + \sigma_i S_i(x) + \tau_i S_{i-1}(x) \\
\Leftrightarrow S_{i+1}(x) &= \frac{1}{\rho_i}[(x - \sigma_i) S_i(x) - \tau_i S_{i-1}(x)].
\end{aligned} \tag{3.28}$$

Taking (3.11) and (3.26) into account, equation (3.28) implies (3.13)

$$\begin{aligned}
s_{i+1,j} &= \int_a^b S_{i+1}(x)P_j(x)f(x)\,\mathrm{d}x \\
&= \int_a^b \frac{1}{\rho_i}[(x-\sigma_i)S_i(x) - \tau_i S_{i-1}(x)]P_j(x)f(x)\,\mathrm{d}x \\
&= \frac{1}{\rho_i}\left[\int_a^b xS_i(x)P_j(x)f(x)\,\mathrm{d}x - \sigma_i s_{ij} - \tau_i s_{i-1,j}\right] \\
&= \frac{1}{\rho_i}\left[\int_a^b S_i(x)\left(a_jP_{j+1}(x) + b_jP_j(x) + c_jP_{j-1}(x)\right)f(x)\,\mathrm{d}x - \sigma_i s_{ij} - \tau_i s_{i-1,j}\right] \\
&= \frac{1}{\rho_i}\left[(b_j-\sigma_i)s_{ij} + a_j s_{i,j+1} + c_j s_{i,j-1} - \tau_i s_{i-1,j}\right],\ i = 2,\dots,n.
\end{aligned}$$

Now the coefficients ρ_i, σ_i and τ_i have to be determined in a way that the shape of S_n given in (3.27) stays true, under the assumption that the previous row has the desired form. These conditions lead to (3.12)

$$\left.\begin{aligned}
s_{i+1,i-1} &= 0 \;\Rightarrow\; \tau_i &&= a_{i-1}, \\
s_{i+1,i} &= 0 \;\Rightarrow\; \sigma_i &&= a_i s_{i,i+1} + b_i - a_{i-1}s_{i-1,i}, \\
s_{i+1,i+1} &= 1 \;\Rightarrow\; \rho_i &&= (b_{i+1}-\sigma_i)s_{i,i+1} + a_{i+1}s_{i,i+2} - a_{i-1}s_{i-1,i+1} + c_{i+1},
\end{aligned}\right\}\ i = 0,\dots,n-2.$$

As already stated two initial rows are needed, these are given by

$$s_{-1,j} = 0, \quad s_{0,j} = \frac{\nu_{j-1}}{\nu_0}\ j = 1,\dots,2n.$$

It follows from (3.12) that the maximum value of j for which the elements s_{ij} are given through (3.13) is decreased by one in each step. That the moments must be known up to ν_{2n-1} is due to the fact that $y_{n-1}^{(n)}$ and hence $s_{n-1,n}$ must be known for the calculation. Now the two values of the Y_n that are needed can be expressed in terms of elements of S_n in view of (3.22), (3.25) and (3.27)

$$\begin{aligned}
y_{n-1}^{(n)} &= a_{n-1}s_{n-1,n}, \\
y_{n-2}^{(n)} &= a_{n-1}(s_{n-2,n} - s_{n-2,n-1}s_{n-1,n}).
\end{aligned}$$

In combination with (3.23),(3.24) and (3.12) this finally results in (3.14)

$$\begin{aligned}
\beta_i &= \sigma_i, \quad i = 1,\dots,n \\
\alpha_i^2 &= a_i\rho_i = \rho_i\tau_{i+1}, \quad i = 1,\dots,n-1.
\end{aligned}$$

Hence the equivalence of (3.15) and (3.28) is shown.
The unchanged case of the *LQMD-Algorithm* needs $4(n-1)^2 + 3(n-1) + 2n$ multiplications, $(n-1)^2 + 2n - 1$ divisions, $4(n-1)^2 + 6n - 4$ summations and $n-1$ square roots. The special case for the classical moments needs $(n-1)^2 + (n-1)$ multiplications, $(n-1)^2 + 2n - 1$ divisions, $2(n-1)^2 + 3(n-1) + 1$ summations and $n-1$ square roots. Both also need to solve a $n \times n$-eigenvalue problem.

3.4 Golub-Welsch Algorithm

3.4.1 The Algorithm

Golub and Welsch proposed another algorithm in [6]. This algorithm needs $2n+1$ moments and uses the Cholesky decomposition of a certain moment matrix M. To calculate the elements of the tridiagonal matrix one has to compute the elements of the Cholesky decomposition. With $M_{ij} = m_{i+j-2}$ for $i,j = 1,\dots,n+1$ these are given by

$$r_{ii} = \left(M_{ii} - \sum_{k=1}^{i-1} r_{ki}^2 \right)^{\frac{1}{2}}, \; i = 1,\dots,n+1,$$
$$r_{ij} = \frac{M_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}}{r_{ii}}, \; i < j, \; j = 1,\dots,n+1.$$

Given these elements one can compute the β_i and α_i via

$$\beta_{j-1} = \frac{r_{j,j+1}}{r_{j,j}} - \frac{r_{j-1,j}}{r_{j-1,j-1}}, \; j = 1,\dots,n,$$
$$\alpha_j = \frac{r_{j+1,j+1}}{r_{jj}}, \; j = 1,\dots,n-1,$$

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

3.4.2 Proof of Correctness of the GWA

As usual the moments are defined by

$$m_k = \int_a^b x^k f(x)\,\mathrm{d}x, \; k = 0,1,\dots,2n.$$

Now the matrix M is defined via

$$M = \left[\int_a^b x^{i+j-2} f(x)\,\mathrm{d}x \right]_{i,j=1,\dots,n+1},$$
$$M = \begin{pmatrix} m_0 & m_1 & m_2 & \dots & m_n \\ m_1 & m_2 & & \iddots & \\ m_2 & & \iddots & & \vdots \\ \vdots & \iddots & & & \\ m_n & & \dots & & m_{2n} \end{pmatrix}. \tag{3.29}$$

This is matrix is called *Hankel* matrix and it is also positive definite. In practice the moments are first obtained via the initial data and then from the solution of the next time step. It is known that a positive definite matrix is invertible and all principle minors are also positive definite. The Cholesky decomposition is based on the following theorem, again we refer to Stoer [3].

Theorem 3.9 (Cholesky Decomposition)
For every real positive $m \times m$ matrix M exists a unique real upper triangular $m \times m$ matrix R, $r_{ik} = 0$ for $k < i$, with $r_{ii} > 0$, $i = 1,2,\dots,m$, such that $M = R^T R$.

Let $M = R^T R$ be the Cholesky decomposition of M with

$$r_{ii} = \left(M_{ii} - \sum_{k=1}^{i-1} r_{ki}^2 \right)^{\frac{1}{2}}, \; i = 1, \ldots, n+1,$$
$$r_{ij} = \frac{M_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}}{r_{ii}}, \; i < j, j = 1, \ldots, n+1. \tag{3.30}$$

Since R is an upper triangular matrix, we can write for the inverse

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

Now Golub and Welsch state that the polynomials

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

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

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

with $p_{-1}(x) = 0$ and $p_0(x) = 1$, [6]. Comparing the coefficients of the two highest powers x^j and x^{j-1} on both sides of this identity results in

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

and so

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

Now, with

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

a straightforward computation shows

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

Inserting this in the equation for the coefficients of the recurrence relation gives

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

with $r_{00} = 1$ and $r_{01} = 0$. These are again exactly the coefficients for the tridiagonal matrix (3.4). It clearly seems that there are some connections to the formula used in the *PDA* and this is no surprise, since the coefficients of the continued fraction can be determined via certain determinants of Hankel matrices, cf. [7] and [19].
The algorithm proposed by Golub and Welsch needs $n(n+1)/2 + (n^3 - n)/6$ multiplications, $3n - 1 + n(n+1)/2$ divisions, $n(n+1)/2 + (n^3 - n)/6 + n$ summations and $n+1$ square roots. Furthermore the solution to the eigenvalue problem is needed.

3.5 Newton's Method

The last alternative algorithm which should be presented here is Newton's method. For a given function $F : \mathbb{R}^m \to \mathbb{R}^m$ it calculates the root x^* of $F(x^*) = 0$. Therefore F must be differentiable in an appropriate neighbourhood of x^* and the Jacobian of F must not be singular. By iterating

$$x^{(k+1)} = x^{(k)} - DF\left(x^{(k)}\right)^{-1} \cdot F\left(x^{(k)}\right), \tag{3.31}$$

the root can the be obtained with the desired accuracy. With $DF \in \mathbb{R}^{m\times m}$ we denote the Jacobian. In our case we have $m = 2n$ and $F : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is given by

$$F(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n}) := \begin{pmatrix} \sum_{i=1}^{n} x_i - m_0 \\ \sum_{i=1}^{n} x_{n+i}x_i - m_1 \\ \vdots \\ \sum_{i=1}^{n} x_{n+i}^{2n-1}x_i - m_{2n-1} \end{pmatrix}.$$

Therefore the Jacobian is

$$DF(x) := \begin{pmatrix} 1 & \ldots & 1 & 0 & \ldots & 0 \\ x_{n+1} & \ldots & x_{2n} & x_1 & \ldots & x_n \\ x_{n+1}^2 & \ldots & x_{2n}^2 & 2x_{n+1}x_1 & \ldots & 2x_{2n}x_n \\ \vdots & & \vdots & \vdots & & \vdots \\ x_{n+1}^{2n-1} & \ldots & x_{2n}^{2n-1} & (2n-1)x_{n+1}^{2n-2}x_1 & \ldots & (2n-1)x_{2n}^{2n-2}x_n \end{pmatrix}.$$

In view of (2.7) we have for $x^* = (w_1, \ldots, w_n, e_1, \ldots, e_n)$

$$F(x^*) = 0,$$

$$DF(x^*) := \begin{pmatrix} 1 & \ldots & 1 & 0 & \ldots & 0 \\ e_1 & \ldots & e_n & w_1 & \ldots & w_n \\ e_1^2 & \ldots & e_n^2 & 2e_1w_1 & \ldots & 2e_nw_n \\ \vdots & & \vdots & \vdots & & \vdots \\ e_1^{2n-1} & \ldots & e_n^{2n-1} & (2n-1)e_1^{2n-2}w_1 & \ldots & (2n-1)e_n^{2n-2}w_n \end{pmatrix}.$$

This matrix is exactly the matrix (2.26) we obtained in Section 2.4 in order to investigate the condition number of the *QMOM*. So we cannot expect the condition number to be too good. In Section 2.4 we investigated the condition number of the solution to the nonlinear problem (2.8) $Ew = \mu$. Now we focus on $Ew - \mu = 0$. The matrix (2.26) is clearly non-singular if $e_i \neq e_j$ for $i \neq j$ and $w_i \neq 0$ for all i is true. In practice one is interested in the question of convergence. We therefore refer to [3] (Theorem 5.3.4, p. 299) for the following result.

Theorem 3.10 (Newton-Kantorovich)
Let $F : \Omega \to \mathbb{R}^n$ be continuous differentiable on the convex set $\Omega \subseteq \mathbb{R}^n$ with the Jacobian matrix $DF(x_0)$, non-singular in x_0. Furthermore there are positive constants α, β and γ such that the following conditions are met

(a) $\|DF(x) - DF(y)\| \leq \gamma\|x - y\|$ *for all* $x, y \in \Omega$,

(b) $\|DF(x_0)^{-1}\| \leq \beta$,

(c) $\|DF(x_0)^{-1}F(x_0)\| \leq \alpha$.

With the constants

$$h := \alpha\beta\gamma, \quad r_{1,2} := \alpha\frac{1 \mp \sqrt{1-2h}}{h}$$

the following is true:
If $h \leq 1/2$ *and* $\overline{B_{r_1}(x_0)} \subset \Omega$, $F(x)$ *has exactly one root* x^* *in* $\Omega \cap B_{r_2}(x_0)$, *then the sequence* $(x_k)_{k\in\mathbb{N}}$,

$$x^{(k+1)} = x^{(k)} - DF\left(x^{(k)}\right)^{-1} \cdot F\left(x^{(k)}\right),\ k = 0, 1, \ldots,$$

stays in $B_{r_1}(x_0)$ *and converges to* x^*.

Now we want to apply this theorem to our case. In the following calculations we scale the quantities such that $m_0 = 1$ and the basic interval for the abscissas is $(0, 1)$. Hence we have $\Omega = (0, 1)^{2n}$. Finding a suited x_0 is very difficult but we can state that in view of the Jacobian the components $x_0^{(i)}$ are non-zero for $i = 1, \ldots, n$ and mutually distinct for $i = n+1, \ldots, 2n$. Therefore we can apply Theorem 2.1 and obtain

$$\begin{aligned}
\|DF(x_0)^{-1}\| &\leq \max(u_1, u_2) =: \beta \\
u_l &= \max_{i=1,\ldots,n} b_i^l \prod_{j=1;j\neq i}^{n} \left(\frac{1 + x_0^{(j)}}{x_0^{(i)} - x_0^{(j)}}\right)^2, \\
b_i^{(1)} &:= 1 + x_0^{(i)}, \\
b_i^{(2)} &:= \left|1 + 2x_0^{(i)} \sum_{j=1;j\neq i}^{n} \frac{1}{x_0^{(i)} - x_0^{(j)}}\right| + 2\left|\sum_{j=1;j\neq i}^{n} \frac{1}{x_0^{(i)} - x_0^{(j)}}\right|.
\end{aligned}$$

In the next step a Lipschitz constant γ shall be derived, again we are using the row sum norm according to Section 2.4

$$\begin{aligned}
&\|DF(x) - DF(y)\| = \max_{i=1,\ldots,2n} \sum_{j=1}^{n} \left\{|x_{n+j}^{i-1} - y_{n+j}^{i-1}| + (i-1)|x_{n+j}^{i-2}x_j - y_{n+j}^{i-2}y_j|\right\} \\
&\leq \max_{i=1,\ldots,2n} \sum_{j=1}^{n} \left\{\sup_{\xi\in(0,1)} (i-1)\xi^{i-2}|x_{n+j} - y_{n+j}| + (i-1)|x_{n+j}^{i-2}x_j - y_{n+j}^{i-2}x_j + y_{n+j}^{i-2}x_j - y_{n+j}^{i-2}y_j|\right\} \\
&\leq \max_{i=1,\ldots,2n} (i-1) \sum_{j=1}^{n} \left\{\sup_{\xi\in(0,1)} \xi^{i-2}|x_{n+j} - y_{n+j}| + |x_{n+j}^{i-2}x_j - y_{n+j}^{i-2}x_j| + |y_{n+j}^{i-2}x_j - y_{n+j}^{i-2}y_j|\right\} \\
&\leq \max_{i=1,\ldots,2n} (i-1) \sum_{j=1}^{n} \left\{\sup_{\xi\in(0,1)} \xi^{i-2}|x_{n+j} - y_{n+j}| + \sup_{\xi\in(0,1)} (i-2)\xi^{i-3}|x_j||x_{n+j} - y_{n+j}| + |y_{n+j}^{i-2}||x_j - y_j|\right\} \\
&= \max_{i=1,\ldots,2n} (i-1) \sum_{j=1}^{n} \left\{|y_{n+j}^{i-2}||x_j - y_j| + \left(\sup_{\xi\in(0,1)} \xi^{i-2} + \sup_{\xi\in(0,1)} (i-2)\xi^{i-3}|x_j|\right)|x_{n+j} - y_{n+j}|\right\} \\
&\leq \max_{i=2,\ldots,2n} (i-1) \sum_{j=1}^{n} \{|x_j - y_j| + (i-1)|x_{n+j} - y_{n+j}|\} \\
&\leq \max_{i=2,\ldots,2n} n(i-1)i \max_{j=1,\ldots,2n} |x_j - y_j| = 2n^2(2n-1) \max_{j=1,\ldots,2n} |x_j - y_j| = \gamma\|x - y\|.
\end{aligned}$$

Here we have used the Lipschitz inequality for differentiable functions

$$|g(x) - g(y)| \leq \sup_{\xi\in(a,b)} |g'(\xi)||x - y|$$

to estimate the terms $|x_{n+j}^{i-1} - y_{n+j}^{i-1}|$ and $|x_{n+j}^{i-2} - y_{n+j}^{i-2}|$.
Now Theorem 3.10 states that h should be smaller than $1/2$, that leads to

$$\alpha \leq \frac{1}{2\beta\gamma}.$$

Since Gautschi derived an approximate lower bound (2.35) we conclude

$$\alpha \leq \frac{1}{2\exp(3.5n)2n^2(2n-1)}.$$

Here we have assumed that m_0 is normalised to one.
Considering (c) we have

$$\|DF(x_0)^{-1}F(x_0)\| \leq \|DF(x_0)^{-1}\|\|F(x_0)\| \leq \beta\|F(x_0)\| \overset{!}{\leq} \alpha$$

and hence

$$\|F(x_0)\| \leq \frac{\alpha}{\beta} \leq \frac{1}{2\beta^2\gamma} \leq \frac{1}{2\exp(7n)2n^2(2n-1)}.$$

That means for $n = 1$ that $\|F(x_0)\| \leq 0.227970 \cdot 10^{-3}$ and for $n = 2$ already $\|F(x_0)\| \leq 0.000017 \cdot 10^{-3}$. So the starting value must be very close to the actual zero to guarantee convergence of Newton's method and therefore this approach is not recommended from a theoretical point of view. For this reason, it is not included into the numerical studies in Section 5.

4 Improvements to the DQMOM

We have shown in Section 2.3 that the one chance to improve the bad condition (2.38) of the *DQMOM* seems to be the change of the test functions. We will still need the weights and abscissas but we will be able to improve the condition of the linear system that is needed for the source terms, cf. (2.14).

4.1 Approach With Universal Test Functions

We will go the same way as *Marchisio* and *Fox* did, just with universal test function φ_k. For now we will leave them unspecified and just assume enough differentiability for our needs. We again start with the following equation

$$\begin{aligned}\int_{\Omega_e} \Big\{ &\frac{\partial w_i(t,x)\delta(e-e_i(t,x))}{\partial t} + \nabla\cdot(u(t,x)w_i(t,x)\delta(e-e_i(t,x))) \\ &- \nabla\cdot(D(t,x)\nabla(w_i(t,x)\delta(e-e_i(t,x)))) \Big\}\varphi_k(e)\,\mathrm{d}e \\ &= \int_{\Omega_e} S(t,x,e)\varphi_k(e)\,\mathrm{d}e. \end{aligned} \tag{4.1}$$

Now we rearrange the left-hand side, at first we differentiate

$$\begin{aligned}\int_{\Omega_e} \Big\{ &\delta(e-e_i)\frac{\partial w_i}{\partial t} - w_i\frac{\partial e_i}{\partial t}\frac{\partial\delta(e-e_i)}{\partial e} + \delta(e-e_i)\nabla\cdot(uw_i) - w_iu\cdot\nabla e_i\frac{\partial\delta(e-e_i)}{\partial e} \\ &- \delta(e-e_i)\nabla\cdot(D\nabla w_i) + D\nabla w_i\cdot\nabla e_i\frac{\partial\delta(e-e_i)}{\partial e} + \nabla\cdot(Dw_i\nabla e_i) \\ &- Dw_i(\nabla e_i)^2\frac{\partial^2\delta(e-e_i)}{\partial e^2}\Big\}\varphi_k(e)\,\mathrm{d}e = \int_{\Omega_e} S(t,x,e)\varphi_k(e)\,\mathrm{d}e.\end{aligned}$$

By sorting the terms we obtain

$$\begin{aligned}&\int_{\Omega_e}\left\{\frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i)\right\}\delta(e-e_i)\varphi_k(e)\,\mathrm{d}e \\ -&\int_{\Omega_e}\left\{w_i\frac{\partial e_i}{\partial t} + w_iu\cdot\nabla e_i - (D\nabla w_i\cdot\nabla e_i + \nabla\cdot(Dw_i\nabla e_i))\right\}\frac{\partial\delta(e-e_i)}{\partial e}\varphi_k(e)\,\mathrm{d}e \\ -&\int_{\Omega_e}\left\{Dw_i(\nabla e_i)^2\right\}\frac{\partial^2\delta(e-e_i)}{\partial e^2}\varphi_k(e)\,\mathrm{d}e \\ =&\int_{\Omega_e} S(t,x,e)\varphi_k(e)\,\mathrm{d}e.\end{aligned}$$

The *PDEs* including the w_i and e_i are extracted from the integrals. Then we integrate by parts and obtain

$$\begin{aligned}\sum_{i=1}^{n}\Big[&\left\{\frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i)\right\}\varphi_k(e_i) \\ &+\left\{w_i\frac{\partial e_i}{\partial t} + w_iu\cdot\nabla e_i - (D\nabla w_i\cdot\nabla e_i + \nabla\cdot(Dw_i\nabla e_i))\right\}\varphi_k'(e_i) \\ &- Dw_i(\nabla e_i)^2\varphi_k''(e_i)\Big] = \int_{\Omega_e} S(t,x,e)\varphi_k(e)\,\mathrm{d}e.\end{aligned}$$

As we have seen in Section 2.3.2 and Section 2.4, the variable $\zeta_i := w_i e_i$ makes no difference, analytically and numerically. So again we introduce this variable since it makes the equation more convenient to read and work with. We obtain

$$\begin{aligned}\sum_{i=1}^{n}\Bigg[&\left\{\frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i)\right\}\varphi_k(e_i)\\ &+\left\{\frac{\partial \zeta_i}{\partial t} + \nabla\cdot(u\zeta_i) - \nabla\cdot(D\nabla \zeta_i) - e_i\left(\frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i)\right)\right\}\varphi_k'(e_i)\\ &- Dw_i(\nabla e_i)^2\varphi_k''(e_i)\Bigg] = \int_{\Omega_e} S(t,x,e)\varphi_k(e)\,\mathrm{d}e.\end{aligned}$$

As in (2.12) we set

$$\begin{aligned}&\frac{\partial w_i}{\partial t} + \nabla\cdot(uw_i) - \nabla\cdot(D\nabla w_i) = \xi_i^{(1)},\\ &\frac{\partial \zeta_i}{\partial t} + \nabla\cdot(u\zeta_i) - \nabla\cdot(D\nabla \zeta_i) = \xi_i^{(2)},\\ &Dw_i(\nabla e_i)^2 = \xi_i^{(3)}.\end{aligned}$$

The following equation is obtained

$$\sum_{i=1}^{n}\left\{\xi_i^{(1)}\varphi_k(e_i) + (\xi_i^{(2)} - e_i\xi_i^{(1)})\varphi_k'(e_i) - \xi_i^{(3)}\varphi_k''(e_i)\right) = \int_{\Omega_e} S(t,x,e)\varphi_k(e)\,\mathrm{d}e$$

and by rearranging

$$\sum_{i=1}^{n}\left\{\xi_i^{(1)}(\varphi_k(e_i) - e_i\varphi_k'(e_i)) + \xi_i^{(2)}\varphi_k'(e_i)\right\} = \sum_{i=1}^{n}\xi_i^{(3)}\varphi_k''(e_i) + \int_{\Omega_e} S(t,x,e)\varphi_k(e)\,\mathrm{d}e.$$

Now with $2n$ suited test functions $\varphi_1,\dots,\varphi_{2n}$ we obtain analogous to (2.15), (2.16) and (2.17) the matrices

$$M_1 := \begin{pmatrix} \varphi_1(e_1) - e_1\varphi_1'(e_1) & \dots & \varphi_1(e_n) - e_n\varphi_1'(e_n)\\ \varphi_2(e_1) - e_1\varphi_2'(e_1) & \dots & \varphi_2(e_n) - e_n\varphi_2'(e_n)\\ \vdots & \dots & \vdots\\ \varphi_{2n}(e_1) - e_1\varphi_{2n}'(e_1) & \dots & \varphi_{2n}(e_n) - e_n\varphi_{2n}'(e_n)\end{pmatrix}, \tag{4.2}$$

$$M_2 := \begin{pmatrix} \varphi_1'(e_1) & \dots & \varphi_1'(e_n)\\ \varphi_2'(e_1) & \dots & \varphi_2'(e_n)\\ \vdots & \dots & \vdots\\ \varphi_{2n}'(e_1) & \dots & \varphi_{2n}'(e_n)\end{pmatrix}, \quad M_3 := \begin{pmatrix} \varphi_1''(e_1) & \dots & \varphi_1''(e_n)\\ \varphi_2''(e_1) & \dots & \varphi_2''(e_n)\\ \vdots & \dots & \vdots\\ \varphi_{2n}''(e_1) & \dots & \varphi_{2n}''(e_n)\end{pmatrix}. \tag{4.3}$$

Therefore we can write the system compact as follows

$$M\xi = \underbrace{M_3\xi^{(3)} + \bar{S}}_{=:d}.$$

Here we set $M = [M_1, M_2]$ and defined ξ, ξ_3 and $\bar{S}$ analogous to (2.18). If one would choose $\varphi_k(e) = e^{k-1}$, one would obtain the standard *DQMOM*. Now the aim is to choose the test functions such

that the matrix M suffice some desired conditions (full rank etc.). If one takes a closer look at M, one can see that it can be written as a product of two matrices

$$P := \begin{pmatrix} \varphi_1(e_1) & \dots & \varphi_1(e_n) & \varphi_1'(e_1) & \dots & \varphi_1'(e_n) \\ \vdots & & \vdots & \vdots & & \vdots \\ \varphi_{2n}(e_1) & \dots & \varphi_{2n}(e_n) & \varphi_{2n}'(e_1) & \dots & \varphi_{2n}'(e_n) \end{pmatrix} \in \mathbb{R}^{2n\times 2n} \tag{4.4}$$

and

$$Q := \begin{pmatrix} 1 & 0 & 0 & & & & \dots & & & 0 \\ 0 & 1 & 0 & & & & \dots & & & 0 \\ \vdots & \ddots & \ddots & & & & & & & \vdots \\ \vdots & & \ddots & \ddots & & & & & & \vdots \\ 0 & 0 & \dots & 0 & 1 & 0 & & \dots & \dots & 0 \\ -e_1 & 0 & \dots & & 0 & 1 & 0 & & \dots & 0 \\ 0 & -e_2 & 0 & & & & \ddots & & & 0 \\ \vdots & \ddots & \ddots & & & & & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & & & & & \ddots & \vdots \\ 0 & \dots & \dots & 0 & -e_n & 0 & \dots & \dots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{2n\times 2n}. \tag{4.5}$$

Therefore the determinant is

$$\det M = \det P \underbrace{\det Q}_{=1} = \det P$$

and for the condition number (we claim $\det M \neq 0$)

$$\begin{aligned} \kappa(M) &= \|M\|\|M^{-1}\| = \|P\cdot Q\|\|Q^{-1}\cdot P^{-1}\| \\ &\leq \|P\|\|Q\|\|Q^{-1}\|\|P^{-1}\| = \kappa(P)\kappa(Q). \end{aligned}$$

If one would choose $\|.\| = \|.\|_\infty$, the result is

$$\kappa(M) \leq \kappa(P)\left(\max_{i=1,\dots,n}\{|e_i|+1\}\right)^2.$$

4.2 Finding Test Functions

Since the test functions $\varphi_1,\dots,\varphi_{2n}$ are not specified yet, one could claim any desired condition number for M, e.g. if $P = Q^{-1}$ then $cond(M) = 1$. However the choice $P = Q^{-1}$ is infeasible.
For this reason one has to prescribe the values $\varphi_k^{(j)}(e_i)$ for $j = 0, 1$, $k = 1,\dots,2n$ and $i = 1,\dots,n$. But it is important to notice that the test functions also occur on the right hand side of the system. The idea is now to look at this as an interpolation problem. Since we prescribe values for the function and its first derivative we will use the *Hermite interpolation*. An advantage of this interpolation is that the interpolation polynomials can be written down explicitly and one does not have to solve another linear system. So now we have to formulate and solve an interpolation problem for each test function φ_k. We have n real numbers $e_1 < \dots < e_n$ and $2n$ prescribed values

$$\varphi_k^{(j)}(e_i) = \begin{cases} \delta_{ki}, & j = 0, \\ \delta_{k,i+n}, & j = 1. \end{cases} \tag{4.6}$$

That means we go for the best that is possible, P shall be the identity. It is well known that there exists a unique polynomial P_k with degree $2n-1$ such that $P_k^{(j)}(e_i) = \varphi_k^{(j)}(e_i)$, cf. [3]. The polynomials are given through

$$P_k(x) = \sum_{i=1}^{n} \sum_{j=0}^{1} \varphi_k^{(j)}(e_i) L_{ij}(x). \tag{4.7}$$

The $L_{ij}(x)$ denote the generalized *Lagrange* polynomials. Consider the polynomials

$$l_{ij}(x) := (x-e_i)^j \prod_{r=1, r\neq i}^{n} \left(\frac{x-e_r}{e_i-e_r}\right)^2$$

for $i = 1, \ldots, n$ and $j = 0, 1$. Then the L_{ij} are defined via

$$L_{i1}(x) := l_{i1}(x), \quad L_{i0}(x) := l_{i0}(x) - l'_{i0}(e_i) l_{i1}(x)$$

and therefore they have the degree $2n-1$. Altogether the polynomials $P_k(x)$ are

$$\begin{aligned} P_k(x) &= \sum_{i=1}^{n} \sum_{j=0}^{1} \varphi_k^{(j)}(e_i) L_{ij}(x) = \sum_{i=1}^{n} \left[\delta_{ki}(l_{i0}(x) - l'_{i0}(e_i) l_{i1}(x)) + \delta_{k,i+n} l_{i1}(x)\right] \\ &= \begin{cases} l_{k0}(x) - l'_{k0}(e_k) l_{k1}(x), & k = 1, \ldots, n, \\ l_{k-n,1}(x), & k = n+1, \ldots, 2n. \end{cases} \end{aligned}$$

For $l'_{k0}(e_k)$ one obtains

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x} l_{k0}(x)\Big|_{x=e_k} &= \frac{\mathrm{d}}{\mathrm{d}x} \prod_{r=1, r\neq k}^{n} \left(\frac{x-e_r}{e_k-e_r}\right)^2 \Bigg|_{x=e_k} \\ &= 2 \sum_{s=1, s\neq k}^{n} \left[\frac{x-e_s}{(e_k-e_s)^2} \prod_{r=1, r\neq k, s}^{n} \left(\frac{x-e_r}{e_k-e_r}\right)^2\right] \Bigg|_{x=e_k} \\ &= 2 \sum_{s=1, s\neq k}^{n} \frac{1}{e_k-e_s}. \end{aligned}$$

That finally gives

$$P_k(x) = \begin{cases} \prod\limits_{r=1, r\neq k}^{n} \left(\frac{x-e_r}{e_k-e_r}\right)^2 \left[1 - 2(x-e_k) \sum\limits_{s=1, s\neq k}^{n} \frac{1}{e_k-e_s}\right], & k = 1, \ldots, n, \\ (x-e_{k-n}) \prod\limits_{r=1, r\neq k-n}^{n} \left(\frac{x-e_r}{e_{k-n}-e_r}\right)^2, & k = n+1, \ldots, 2n. \end{cases} \tag{4.8}$$

Now we want to determine the first and second derivative of $P_k(x)$. The second derivative is needed for matrix M_3 (4.3). We start with the case $k = 1, \ldots, n$ and obtain for $P'_k(x)$

$$\begin{aligned} P'_k(x) = 2 &\sum_{s=1, s\neq k}^{n} \left[\frac{x-e_s}{(e_k-e_s)^2} \prod_{r=1, r\neq k, s}^{n} \left(\frac{x-e_r}{e_k-e_r}\right)^2\right] \left[1 - 2(x-e_k) \sum_{s=1, s\neq k}^{n} \frac{1}{e_k-e_s}\right] \\ &- \left(2 \sum_{s=1, s\neq k}^{n} \frac{1}{e_k-e_s}\right) \prod_{r=1, r\neq k}^{n} \left(\frac{x-e_r}{e_k-e_r}\right)^2. \end{aligned}$$

It follows for $P_k''(x)$

$$\begin{aligned}P_k''(x) = 2 \sum_{s=1,s\neq k}^{n} \Bigg[& \frac{1}{(e_k-e_s)^2} \prod_{r=1,r\neq k,s}^{n} \left(\frac{x-e_r}{e_k-e_r}\right)^2 \\ & +2\frac{x-e_s}{(e_k-e_s)^2} \sum_{i=1,i\neq k,s}^{n} \left[\frac{x-e_i}{(e_k-e_i)^2} \prod_{r=1,r\neq k,s,i}^{n} \left(\frac{x-e_r}{e_k-e_r}\right)^2 \right] \Bigg] \left[1-2(x-e_k) \sum_{s=1,s\neq k}^{n} \frac{1}{e_k-e_s} \right] \\ & -8 \left(\sum_{s=1,s\neq k}^{n} \frac{1}{e_k-e_s} \right) \sum_{s=1,s\neq k}^{n} \left[\frac{x-e_s}{(e_k-e_s)^2} \prod_{r=1,r\neq k,s}^{n} \left(\frac{x-e_r}{e_k-e_r}\right)^2 \right].\end{aligned}$$

We evaluate the second derivative for matrix M_3 (4.3)

$$P_k''(e_l) = \begin{cases} \frac{2}{(e_k-e_l)^2} \prod_{r=1,r\neq k,l}^{n} \left(\frac{e_l-e_r}{e_k-e_r}\right)^2 \left[1-2 \sum_{s=1,s\neq k}^{n} \frac{e_l-e_k}{(e_k-e_s)^2} \right], & l\neq k, \\ 2 \sum_{s=1,s\neq k}^{n} \left[\frac{1}{(e_k-e_s)^2} + \frac{2}{e_k-e_s} \sum_{i=1,i\neq k,s}^{n} \frac{1}{e_k-e_i} \right] - 8 \left(\sum_{s=1,s\neq k}^{n} \frac{1}{e_k-e_s} \right)^2, & l=k. \end{cases}$$

It remains to do the same calculations for the case $k = n+1,\dots,2n$. The polynomials $P_k(x)$ are given through (4.8). Hence the first derivative is

$$P_k'(x) = \prod_{r=1,r\neq k-n}^{n} \left(\frac{x-e_r}{e_{k-n}-e_r}\right)^2 + 2(x-e_{k-n}) \sum_{s=1,s\neq k-n}^{n} \frac{x-e_s}{(e_{k-n}-e_s)^2} \prod_{r=1,r\neq k-n,s}^{n} \left(\frac{x-e_r}{e_{k-n}-e_r}\right)^2.$$

Now the second derivative can be obtained

$$\begin{aligned}P_k''(x) = 4 & \sum_{s=1,s\neq k-n}^{n} \frac{x-e_s}{(e_{k-n}-e_s)^2} \prod_{r=1,r\neq k-n,s}^{n} \left(\frac{x-e_r}{e_{k-n}-e_r}\right)^2 \\ & + 2(x-e_{k-n}) \sum_{s=1,s\neq k-n}^{n} \left[\frac{1}{(e_{k-n}-e_s)^2} \prod_{r=1,r\neq k-n,s}^{n} \left(\frac{x-e_r}{e_{k-n}-e_r}\right)^2 \right. \\ & \left. +2\frac{x-e_s}{(e_{k-n}-e_s)^2} \sum_{i=1,i\neq k-n,s}^{n} \left[\frac{x-e_i}{(e_{k-n}-e_i)^2} \prod_{r=1,r\neq k-n,s,i}^{n} \left(\frac{x-e_r}{e_{k-n}-e_r}\right)^2 \right] \right].\end{aligned}$$

Again we evaluate the polynomials and obtain

$$P_k''(e_l) = \begin{cases} -\frac{2}{e_{k-n}-e_l} \prod_{r=1,r\neq k-n,l}^{n} \left(\frac{e_l-e_r}{e_{k-n}-e_r}\right)^2, & l\neq k-n, \\ 4 \sum_{s=1,s\neq k}^{n} \frac{1}{e_{k-n}-e_s}, & l=k-n. \end{cases}$$

What was basically done here can be compared to the approach of [15] or [5]. But they suggested orthogonal polynomials, whereas we can use any suited test function. Furthermore there are more benefits than the improvement of the condition number. To highlight these advantages we will consider the example (5.1)

$$\begin{cases} \dfrac{\partial f(t,e)}{\partial t} = -\dfrac{\partial}{\partial e}\left(\phi(e) f(t,e)\right), & (t,e) \in (0,T]\times(0,\infty), \\ f_0(e) = f(0,e) = ae^2\exp(-be), & e \in (0,\infty), \end{cases}$$

which will also be treated in Section 5. Transforming the source term according to the previous calculation one obtains

$$-\int_0^\infty \frac{\partial}{\partial e}\left(\phi(e)f(t,e)\right)\varphi_k(e)\,\mathrm{d}e = \int_0^\infty \phi(e)f(t,e)\varphi_k'(e)\,\mathrm{d}e \quad k=1,\dots,2n.$$

With (4.6) the approximation therefore simplifies to

$$\int_0^\infty \phi(e)f(t,e)\varphi_k'(e)\,\mathrm{d}e \approx \begin{cases} 0, & k=1,\dots,n \\ \phi(e_{k-n})w_{k-n}, & k=n+1,\dots,2n. \end{cases} \tag{4.9}$$

For the next improvement one looks at the powers of the internal variable e^l for $l=0,\dots,2n-1$ and $\Omega_e=(0,\infty)$. These powers will increase rapidly during the calculation for our example and therefore the values for the source term will do the same. As shown above there are no more powers of the internal variable in the approximated source term. The polynomials will still grow as e grows but not as fast as the powers. This is shown in the following Figure 1 which was done for the given initial data of *Problem I* and $n=2$ at the beginning of a calculation.
Furthermore we will give the corresponding matrix with the second derivates, which occurs on the right-hand side of the system

$$M_3 = \begin{pmatrix} -0.1350 & 0.0315 \\ 0.0585 & -0.1350 \\ -0.6000 & 0.3000 \\ -0.3000 & 0.6000 \end{pmatrix}.$$

For comparison we will give the analogue matrix obtained in the *DQMOM*

$$M_3 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 2 & 2 \\ 20.0014 & 60.0031 \end{pmatrix}.$$

It is important to note that, as mentioned above, $P=Q^{-1}$ is not a good choice for this approach. In our tests *Matlab* failed to compute results. But one clearly sees the advantage to the standard powers of the internal variable, the values are smaller. For example for $n=2$, $t=0$ and $e=20$ the polynomial P_4 is smaller than 70, whereas the third power of e would be 20^3.

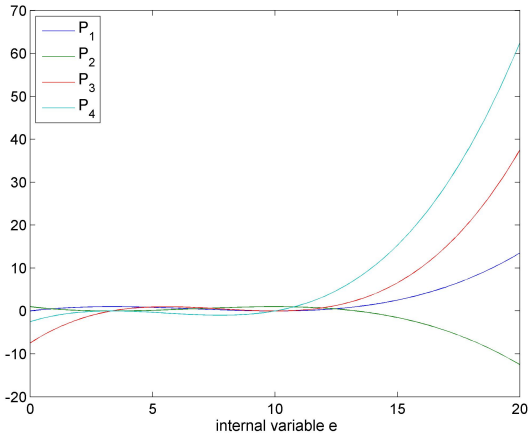


Fig. 1: Test Functions for Problem I with $n=2$ at $t=0$.

5 Numerical Results

In this section we want to discuss the presented methods. Therefore we will present *PBEs* which can be solved analytically, so that the numerical results can be compared to an exact solution. Then we want to compare the algorithms from Sections 3.2, 3.3 and 3.4 which compute the weights and abscissas for a given set of moments. The focus will lie on the time that is needed for a calculation. In the end we will compare the *QMOM*, *DQMOM* and the improved *DQMOM* for several problems.

5.1 Analytical Solutions & Treatment of the Problems

In this section several test problems are solved analytically in order to compare the numerical results with them. After this is done the approximated source terms are derived. The first three problems are of the form

$$\begin{cases} \dfrac{\partial f(t,e)}{\partial t} = -\dfrac{\partial}{\partial e}\left(\phi_i(e) f(t,e)\right), & (t,e) \in (0,T] \times (0,\infty), \\ f_0(e) = f(0,e) = ae^2 \exp(-be), & e \in (0,\infty), \end{cases} \tag{5.1}$$

with

$$\phi_1(e) = \beta, \tag{5.2}$$
$$\phi_2(e) = \beta e, \tag{5.3}$$
$$\phi_3(e) = \beta / e. \tag{5.4}$$

We will refer to these problems as *Problem I – III*. These three problems are solved by applying the *Method of Characteristics*, cf. [2]. These problems were also treated in [12]. The last four problems are of the form

$$\begin{aligned} \frac{\partial f(t,e)}{\partial t} = \frac{1}{2}\int_0^e C(e-e',e') f(t,e-e') f(t,e') \,\mathrm{d}e' - \int_0^\infty C(e,e') f(t,e) f(t,e') \,\mathrm{d}e' \\ + \int_e^\infty M(e') b(e,e') \sigma(e') f(t,e') \,\mathrm{d}e' - \sigma(e) f(t,e). \end{aligned} \tag{5.5}$$

We will use different initial data and integral kernels for this equation. This type of problem was treated in [13] with some remarks in [9] and in a generalized way in [11]. We will refer to these problems as *Problem IV – VII*.

5.1.1 Problem I

With the *Method of Characteristics* one obtains the following system of ordinary differential equations for the source term (5.2)

$$\begin{aligned} \dot{x}_1(s) &= 1, & x_1(0) &= 0, \\ \dot{x}_2(s) &= \beta, & x_2(0) &= \xi, \\ \dot{z}(s) &\overset{(5.1)}{=} 0, & z(0) &= a\xi^2 \exp(-b\xi). \end{aligned} \tag{5.6}$$

Where x_1 corresponds to t, x_2 corresponds to e and z corresponds to f. This is just a homogeneous scalar transport equation. Hence the well known solution is $f(t,e) = f_0(e - \beta t)$

$$f(t,e) = \begin{cases} a(e-\beta t)^2 \exp(-b(e-\beta t)) & , \quad e - \beta t \geq 0, \\ 0 & , \quad e - \beta t < 0. \end{cases}$$

Now the moments are calculated for $k = 0, 1, \ldots$

$$\begin{aligned} m_k(t) &= a \int_{\beta t}^{\infty} e^k (e - \beta t)^2 \exp(-b(e - \beta t)) \, \mathrm{d}e \\ &= a \int_{\beta t}^{\infty} \sum_{i=0}^{2} \binom{2}{i} (-\beta t)^{2-i} e^{i+k} \exp(-b(e - \beta t)) \, \mathrm{d}e \\ &= a \exp(b\beta t) \sum_{i=0}^{2} \binom{2}{i} (-\beta t)^{2-i} \int_{\beta t}^{\infty} e^{i+k} \exp(-be) \, \mathrm{d}e \\ &= a \sum_{i=0}^{2} \binom{2}{i} (-1)^{2-i} \sum_{j=0}^{k+i} \frac{(\beta t)^{2+k-j}}{b^{j+1}} \prod_{l=0}^{j-1} (k + i - l) \quad k = 0, 1, 2, \ldots \, . \end{aligned}$$

If the moment transform is performed one yields the following equations

$$\frac{\partial m_k}{\partial t} = - \int_0^{\infty} \frac{\partial}{\partial e} \left(\phi_1(e) f(t, e) \right) e^k \, \mathrm{d}e = k \cdot \beta m_{k-1}, \quad k = 0, 1, 2, \ldots \, . \tag{5.7}$$

To see that the moments full fill this equations one substitutes $v = b(e - \beta t)$. One therefore obtains

$$m_k(t) = \frac{a}{b^3} \int_0^{\infty} \left(\frac{v}{b} + \beta t \right)^k v^2 \exp(-v) \, \mathrm{d}v$$

and now one clearly sees that the moments satisfy (5.7).

5.1.2 Problem II

With the *Method of Characteristics* one obtains the following system of ordinary differential equations for the source term (5.3)

$$\begin{array}{rcll} \dot{x}_1(s) & = & 1, & x_1(0) = 0, \\ \dot{x}_2(s) & = & \beta x_2(s), & x_2(0) = \xi, \\ \dot{z}(s) & \overset{(5.1)}{=} & -\beta z(s), & z(0) = a\xi^2 \exp(-b\xi). \end{array} \tag{5.8}$$

When (5.8) is solved and transformed back, one yields the solution

$$f(t,e) = ae^2 \exp(-(be\exp(-\beta t) + 3\beta t)).$$

Now the moments can be determined exactly for all $k = 0, 1, \ldots$

$$\begin{aligned} m_k(t) &= \int_0^\infty e^k f(t,e)\,\mathrm{d}e \\ &= a\exp(-3\beta t)\int_0^\infty e^{k+2}\exp(-be\exp(-\beta t))\,\mathrm{d}e \quad k = 0,1,2,\ldots\ . \end{aligned}$$

To clarify this integral we set $\tilde{a} := \tilde{a}(t) := a\exp(-3\beta t)$ and $\tilde{b} := \tilde{b}(t) := b\exp(-\beta t)$. Then the moments can be obtained by integrating by parts

$$m_k(t) = \tilde{a}\int_0^\infty e^{k+2}\exp(-\tilde{b}e)\,\mathrm{d}e = (k+2)!\frac{\tilde{a}}{\tilde{b}^{k+3}} = (k+2)!\frac{a}{b^{k+3}}\exp(k\beta t), \quad k = 0,1,2,\ldots\ .$$

When the moment transform is performed one obtains

$$\frac{\partial m_k}{\partial t} = -\int_0^\infty \frac{\partial}{\partial e}\left(\phi_2(e)f(t,e)\right)e^k\,\mathrm{d}e = k\cdot\beta m_k, \quad k = 0,1,2,\ldots\ . \tag{5.9}$$

One clearly sees that the calculated moments satisfy these equations.

5.1.3 Problem III

As before one obtains for source term (5.4) the following system of characteristic *ODEs*

$$\begin{array}{rcll} \dot{x}_1(s) & = & 1, & x_1(0) = 0, \\ \dot{x}_2(s) & = & \frac{\beta}{x_2(s)}, & x_2(0) = \xi, \\ \dot{z}(s) & \overset{(5.1)}{=} & \frac{\beta}{x_2(s)^2}z(s), & z(0) = a\xi^2\exp(-b\xi). \end{array} \tag{5.10}$$

If (5.10) is solved and transformed back, one obtains the solution

$$f(t,e) = \begin{cases} ae\sqrt{e^2-2\beta t}\exp(-b\sqrt{e^2-2\beta t}) & , \quad e^2-2\beta t \geq 0, \\ 0 & , \quad e^2-2\beta t < 0. \end{cases}$$

For this function one can only determine the moments of even order in a closed form. The remaining moments are treated separately. In general one has

$$m_k(t) = \int_0^\infty e^k f(t,e)\,\mathrm{d}e \tag{5.11}$$

$$= a\int_0^\infty e^{k+1}\sqrt{e^2-2\beta t}\exp(-b\sqrt{e^2-2\beta t})\,\mathrm{d}e. \tag{5.12}$$

For $k = 0, 2, 4, \dots$. one obtains by substituting $v = \sqrt{e^2 - 2\beta t}$

$$\begin{aligned} m_k(t) &= a \int_0^\infty \left(v^2 + 2\beta t\right)^{k/2} v^2 \exp(-bv)\,\mathrm{d}v \\ &= a \int_0^\infty \sum_{i=0}^{k/2} \binom{k/2}{i} (2\beta t)^{k/2-i} v^{2i+2} \exp(-bv)\,\mathrm{d}v \\ &= a \sum_{i=0}^{k/2} \binom{k/2}{i} (2\beta t)^{k/2-i} \int_0^\infty v^{2i+2} \exp(-bv)\,\mathrm{d}v \\ &= \frac{a}{b^3} \sum_{i=0}^{k/2} \binom{k/2}{i} (2\beta t)^{k/2-i} \frac{(2i+2)!}{b^{2i}} \quad k = 0, 2, 4, \dots \ . \end{aligned}$$

When the moment transform is performed on obtains

$$\frac{\partial m_k}{\partial t} = -\int_0^\infty \frac{\partial}{\partial e} \left(\phi_2(e) f(t,e)\right) e^k \,\mathrm{d}e = k \cdot \beta m_{k-2}, \quad k = 0, 1, 2, \dots \ . \tag{5.13}$$

This system was already mentioned here (2.5). Again one can directly verify that the moments satisfy these equations. For the moments of uneven order one can perform a similar substitution, i.e. $v = b\sqrt{e^2 - 2\beta t}$ and therefore on obtains

$$m_k(t) = \frac{a}{b^2} \int_0^\infty \left(\frac{v^2}{b^2} + 2\beta t\right)^{k/2} v^2 \exp(-v)\,\mathrm{d}v.$$

Since these moments cannot be calculated analytically one could apply the *Gauss-Laguerre quadratur*

$$m_k(t) = \frac{a}{b^2} \int_0^\infty \left(\frac{v^2}{b^2} + 2\beta t\right)^{k/2} v^2 \exp(-v)\,\mathrm{d}v \approx \frac{a}{b^2} \sum_{i=1}^{\nu} \left(\frac{\xi_i^2}{b^2} + 2\beta t\right)^{k/2} \xi_i^2 \omega_i.$$

For the approximation error we refer to Theorem 3.8. To calculate the weights and abscissas for this specific quadrature one can use algorithms 3.2 or 3.3 with the standard moments for $\mu_k = k!$ for $k = 0, 1, 2, \dots, 2\nu - 1$ to obtain the weights and abscissas.

5.1.4 Problems IV – VII

In this section equation (5.5)

$$\frac{\partial f(t,e)}{\partial t} = \frac{1}{2}\int_0^e C(e-e',e')f(t,e-e')f(t,e')\,\mathrm{d}e' - \int_0^\infty C(e,e')f(t,e)f(t,e')\,\mathrm{d}e' + \int_e^\infty M(e')b(e,e')\sigma(e')f(t,e')\,\mathrm{d}e' - \sigma(e)f(t,e).$$

shall be solved analytically. For more information to the following quantities and assumptions we refer to [13], [9] and [11]. At first the integral kernels and initial conditions have to be specified. For a precise interpretation of those quantities we refer to suited literature. The expression $C(e,e')$ represents the aggregation rate and is set to one, $C(e,e') = 1$. The term $M(e') = 2$ models binary breakage. $b(e,e') = 1/e'$ is a probability density that measures the probability that the breakage of a particle of size e' produces a particle of size e. Obviously one should state $b(e,e') = 0$ if $e \geq e'$. The quantity $\sigma(e)$ models the fragmentation rate and is set proportional to the particle size, $\sigma(e) = \sigma e$. Finally the equation reads

$$\frac{\partial f(t,e)}{\partial t} = \frac{1}{2}\int_0^e f(t,e-e')f(t,e')\,\mathrm{d}e' - \int_0^\infty f(t,e)f(t,e')\,\mathrm{d}e' + 2\sigma\int_e^\infty f(t,e')\,\mathrm{d}e' - \sigma e f(t,e). \tag{5.14}$$

Two initial conditions are used for this problem. These are

$$f(0,e) = f_0(e) = \begin{cases} \exp(-e), & \text{Problems IV, V \& VI} \\ 4e\exp(-2e), & \text{Problem VII} \end{cases}. \tag{5.15}$$

To solve this equation one first applies the *Laplace Transform* to the internal variable

$$G(t,p) = \mathcal{L}(g(t,e)) = \int_0^\infty g(t,e)\exp(-pe)\,\mathrm{d}e.$$

The result is a partial differential equation which can be solved by the *Method of Characteristics* which means that an *ODE* of the *Riccati Type* has to be solved. When the *Laplace Transform* is applied one obtains

$$\begin{aligned}
\frac{\partial F(t,p)}{\partial t} &= \frac{1}{2}\int_0^\infty\int_0^e f(t,e-e')f(t,e')\,\mathrm{d}e'\,\exp(-pe)\,\mathrm{d}e - \int_0^\infty \underbrace{\int_0^\infty f(t,e')\,\mathrm{d}e'}_{=:\Phi(t)} f(t,e)\exp(-pe)\,\mathrm{d}e \\
&\quad + 2\sigma\int_0^\infty\int_e^\infty f(t,e')\,\mathrm{d}e'\,\exp(-pe)\,\mathrm{d}e - \sigma\int_0^\infty e f(t,e)\exp(-pe)\,\mathrm{d}e \\
&= \frac{1}{2}F(t,p)^2 - \Phi(t)F(t,p) + 2\sigma\int_0^\infty\left[\int_0^\infty f(t,e')\,\mathrm{d}e' - \int_0^e f(t,e')\,\mathrm{d}e'\right]\exp(-pe)\,\mathrm{d}e \\
&\quad - \sigma\frac{\partial F(t,p)}{\partial p} \\
&= \frac{1}{2}F(t,p)^2 - \Phi(t)F(t,p) + \frac{2\sigma}{p}\left[\Phi(t) - F(t,p)\right] + \sigma\frac{\partial F(t,p)}{\partial p}.
\end{aligned}$$

This is a quasi linear first order *PDE*

$$\frac{\partial F(t,p)}{\partial t} - \sigma\frac{\partial F(t,p)}{\partial p} = \frac{1}{2}F(t,p)^2 - F(t,p)\left[\Phi(t) + \frac{2\sigma}{p}\right] + \frac{2\sigma\Phi(t)}{p}. \tag{5.16}$$

Therefore one can use the *Method of Characteristics* to solve this equation. The resulting system is

$$\begin{array}{rcll} \dot{x}_1(s) & = & 1, & x_1(0) = 0, \\ \dot{x}_2(s) & = & -\sigma, & x_2(0) = \xi, \\ \dot{z}(s) & \overset{(5.16)}{=} & \frac{1}{2}z(s)^2 - z(s)\left[\Phi(s) + \frac{2\sigma}{x_2(s)}\right] + \frac{2\sigma\Phi(s)}{x_2(s)}, & z(0) = z_0. \end{array} \tag{5.17}$$

The initial data is obtained by transforming (5.15). Hence one obtains

$$z(\xi, 0) = z_0(\xi) = \begin{cases} \dfrac{1}{\xi + 1}, & \text{Problems IV V \& VI} \\ \dfrac{4}{(\xi + 2)^2}, & \text{Problem VII} \end{cases}. \tag{5.18}$$

The first two equations of (5.17) can be solved directly and one obtains

$$x_1(s) = s, \quad x_2(s) = -\sigma s + \xi.$$

Altogether one has to solve the following problem

$$\dot{z}(s) = \frac{1}{2}z(s)^2 - z(s)\left[\Phi(s) + \frac{2\sigma}{\xi - \sigma s}\right] + \frac{2\sigma\Phi(s)}{\xi - \sigma s}, \tag{5.19}$$

$$z(0) = \begin{cases} \dfrac{1}{\xi + 1}, & \text{Problems IV V\& VI} \\ \dfrac{4}{(\xi + 2)^2}, & \text{Problem VII} \end{cases}. \tag{5.20}$$

Now one would usually try to guess a special solution and then transform this *ODE* into an *ODE* of *Bernoulli Type*. Unfortunately the function $\Phi(s)$ is unknown. To obtain a unique solution for z one has to define Φ through another equation. Since $\Phi(s)$ represents the total number of particles this will directly affect the system. In [13] Φ was chosen to be constant $\Phi = 1$, [9] adopted this choice. A more general choice was made in [11]. The total number of particles is described by the *ODE*

$$\dot{\Phi}(s) = \frac{\Phi(\infty)^2 - \Phi(s)^2}{2}. \tag{5.21}$$

The initial condition is given by the zero order moment of the initial distribution given in (5.15), i.e. $\Phi(0) = 1$. Here $\Phi(\infty)$ denotes a constant which represents an asymptotic state of the system and the following relation holds

$$\sigma = \frac{1}{2}\Phi(\infty)^2.$$

One clearly sees that the case $\Phi(s) = 1$ treated in [13] is included in this equation. For the first initial condition used in *Problem IV – VI* the solution is

$$z(s) = \frac{\Phi(s)^2}{\xi - \sigma s + \Phi(s)}. \tag{5.22}$$

This is, because $z(s)$ satisfies the initial condition and Φ satisfies (5.21). The solution of (5.21) is according to [11]

$$\Phi(s) = \Phi(\infty)\frac{1 + \Phi(\infty)\tanh(\Phi(\infty)s/2)}{\Phi(\infty) + \tanh(\Phi(\infty)s/2)}. \tag{5.23}$$

So by the *Method of Characteristics* the solution is

$$F(t,p) = \frac{\Phi(t)^2}{p+\Phi(t)}.$$

Now $F(t,p)$ has to be transformed back, because of the simple shape this is no difficulty and one obtains

$$f(t,e) = \Phi(t)^2 \exp(-\Phi(t)e).$$

Having determined $f(t,e)$ one can calculate the moments

$$\begin{aligned} m_k(t) &= \int_0^\infty e^k f(t,e)\,\mathrm{d}e = \Phi(t)^2 \int_0^\infty e^k \exp(-\Phi(t)e)\,\mathrm{d}e = k!\frac{\Phi(t)^2}{\Phi(t)^{k+1}} \\ &= k!\Phi(t)^{1-k} = k!\left(\frac{\Phi(\infty)+\tanh(\Phi(\infty)s/2)}{\Phi(\infty)(1+\Phi(\infty)\tanh(\Phi(\infty)s/2))}\right)^{k-1}, \quad k=0,1,2\dots\,. \end{aligned} \tag{5.24}$$

Now the difference between the problems is the different choice of $\Phi(\infty)$ in the initial condition. In *Problem IV* $\Phi(\infty)$ is chosen to be smaller than one, that means in view of (5.21) that the number of particles is decreasing, i.e. aggregation. Whereas in *Problem V* $\Phi(\infty)$ is chosen to be larger than one and hence the particle number is increasing, i.e. fragmentation. For *Problem VI* we choose $\Phi(\infty)=1$ and therefore one obtains a steady state solution. For *Problem VII* we choose the second initial condition $f_0(e) = 4e\exp(-2e)$ and $\Phi(\infty)=1$, which implies that the total number of particle stays constant. We will directly give the solution to this problem. For the derivation of this solution we again refer to [13] and [9]. The solution is

$$f(t,e) = \sum_{i=1}^{2} \frac{K_1(t)+p_i(t)K_2(t)}{L(t)+4p_i(t)} \exp(p_i e) \tag{5.25}$$

for all $t>0$. The quantities are as follows

$$\begin{aligned} K_1(t) &= 7+t+\exp(-t), \quad & K_2(t) &= 2-2\exp(-t), \\ L(t) &= 9+t-\exp(-t), & p_{1/2} &= \frac{1}{4}(\exp(-t)-t-9) \pm \frac{1}{4}\sqrt{d(t)}, \\ d(t) &= t^2+(10-2\exp(-t))t+25-26\exp(-t)+\exp(-2t). \end{aligned}$$

One has to verify that the solution converges to the initial data as t converges to zero. We will give a rough presentation of this calculation. At first one writes the solution (5.25) as one fraction

$$f(t,e) = \frac{(K_1+p_1K_2)(L+4p_2)\exp(p_1e)+(K_1+p_2K_2)(L+4p_1)\exp(p_2e)}{(L+4p_1)(L+4p_2)}.$$

Now one can apply *l'Hospital's* rule to this fraction. After that an important step is to split the p_i' into two summands, i.e.

$$p_{1/2}' = \underbrace{-\frac{1}{4}(\exp(-t)+1)}_{=:A} \pm \underbrace{\frac{d'(t)}{8\sqrt{d(t)}}}_{=:B}.$$

Now a carefully examination finally shows

$$\lim_{t\searrow 0} f(t,e) = \frac{-64e\exp(-2e)}{-16} = 4e\exp(-2e) = f_0(e).$$

To verify that the moments are finite one has to proof that the p_i are negative and hence that $d(t)$ is positive

$$\begin{aligned}
p_1(t) &= \frac{1}{4}(\exp(-t) - t - 9) + \frac{1}{4}\sqrt{d(t)} \\
&= \frac{1}{4}(\exp(-t) - t - 9) + \frac{1}{4}\sqrt{t^2 + (10 - 2\exp(-t))t + 25 - 26\exp(-t) + \exp(-2t)} \\
&\leq -2 - \frac{1}{4}t + \frac{1}{4}\sqrt{t^2 + 10t + 25 + 1} \\
&= -2 - \frac{1}{4}t + \frac{1}{4}\sqrt{(t+5)^2 + 1} \\
&\leq -2 - \frac{1}{4}t + \frac{1}{4}(t+5) + \frac{1}{4} = -\frac{1}{2} < 0, \\
p_2(t) &= \frac{1}{4}(\exp(-t) - t - 9) - \frac{1}{4}\sqrt{d(t)} \\
&\leq -2 - \frac{1}{4}\sqrt{t^2 + (10 - 2\exp(-t))t + 25 - 26\exp(-t) + \exp(-2t)} < 0.
\end{aligned}$$

For $d(t)$ one sees $d(0) = 0$ and

$$\begin{aligned}
\dot{d}(t) &= 2t + (10 + 2\exp(-t))t + 10 - 2\exp(-t) + 26\exp(-t) - 2\exp(-2t) \\
&= 12t + 2\exp(-t)(t + 25) - 2\exp(-2t) + 10 \\
&\geq 12t + 2\exp(-t)(t + 25) + 8 > 0,
\end{aligned}$$

so clearly $d(t)$ is positive for all $t > 0$. Therefore the moments are easily calculated to be

$$\begin{aligned}
m_k(t) &= \int_0^\infty e^k f(t, e)\,\mathrm{d}e = \sum_{i=1}^{2} \frac{K_1(t) + p_i(t)K_2(t)}{L(t) + 4p_i(t)} \int_0^\infty e^k \exp(p_i e)\,\mathrm{d}e \\
&= k! \sum_{i=1}^{2} (-p_i(t))^{-(k+1)} \frac{K_1(t) + p_i(t)K_2(t)}{L(t) + 4p_i(t)}.
\end{aligned} \tag{5.26}$$

5.1.5 Approximation of the Source Terms

In the previous section the analytical solutions to the problems were shown. Now the methods shall be applied to the problems and therefore one has to approximate the source term. For the *QMOM* and *DQMOM* the approximation of the source term is the same. One has for the problems I – III

$$\int_0^\infty e^k S(t,e)\,\mathrm{d}e \approx \sum_{i=1}^{n} k\phi(e_i)e_i^{k-1}w_i, \quad k = 0,1,2\ldots 2n-1\ . \tag{5.27}$$

For the *improved DQMOM* the approximation reduces to

$$\begin{aligned} &-\int_0^\infty \frac{\partial}{\partial e}\left(\phi(e)f(t,e)\right)\varphi_k(e)\,\mathrm{d}e = \int_0^\infty \phi(e)f(t,e)\varphi_k'(e)\,\mathrm{d}e \\ &= \int_0^\infty \phi(e)f(t,e)\varphi_k'(e)\,\mathrm{d}e \approx \begin{cases} 0, & k = 1,\ldots,n, \\ \phi(e_{k-n})w_{k-n}, & k = n+1,\ldots,2n. \end{cases} \end{aligned} \tag{5.28}$$

For the problems IV – VII the situation is a bit more complicated, especially for the *improved DQMOM*. Here one has to use $b(e,e') = 0$ for $e \geq e'$ and $f(t,e) = 0$ for $e \leq 0$. One obtains for the *QMOM* and *DQMOM* for $k = 0,1,\ldots,2n-1$

$$\begin{aligned} &\int_0^\infty e^k S(t,e)\,\mathrm{d}e = \frac{1}{2}\int_0^\infty e^k \int_0^e C(e-e',e')f(t,e-e')f(t,e')\,\mathrm{d}e'\,\mathrm{d}e - \int_0^\infty e^k\sigma(e)f(t,e)\,\mathrm{d}e \\ &-\int_0^\infty e^k \int_0^\infty C(e,e')f(t,e)f(t,e')\,\mathrm{d}e'\,\mathrm{d}e + \int_0^\infty e^k\int_e^\infty M(e')b(e,e')\sigma(e')f(t,e')\,\mathrm{d}e'\,\mathrm{d}e \\ &\approx \frac{1}{2}\int_0^\infty e^k \sum_{j=1}^{n} C(e-e_j,e_j)f(t,e-e_j)w_j\,\mathrm{d}e - \sum_{i=1}^{n} e_i^k\sigma(e_i)w_i \\ &-\int_0^\infty e^k\sum_{j=1}^{n} C(e,e_j)w_j f(t,e)\,\mathrm{d}e + \int_0^\infty e^k \sum_{i=1}^{n} M(e_i)b(e,e_i)\sigma(e_i)w_i\,\mathrm{d}e \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(e_i+e_j)^k C(e_i,e_j)w_j w_i - \sum_{i=1}^{n}\sum_{j=1}^{n} e_i^k C(e_i,e_j)w_j w_i \\ &+\sum_{i=1}^{n}\left[M(e_i)\int_0^\infty e^k b(e,e_i)\,\mathrm{d}e - e_i^k\right]\sigma(e_i)w_i \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[(e_i+e_j)^k - e_i^k\right]C(e_i,e_j)w_j w_i + \sum_{i=1}^{n}\left[M(e_i)\int_0^{e_i} e^k b(e,e_i)\,\mathrm{d}e - e_i^k\right]\sigma(e_i)w_i. \end{aligned}$$

Now one inserts the quantities and finally yields for $k = 0,1,2,\ldots,2n-1$

$$\begin{aligned} \int_0^\infty e^k S(t,e)\,\mathrm{d}e \approx &\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[(e_i+e_j)^k - e_i^k\right]w_j w_i \\ &+ \frac{1}{2}\Phi(\infty)\sum_{i=1}^{n}\left[\frac{2}{k+1} - 1\right]e_i^{k+1}w_i. \end{aligned} \tag{5.29}$$

For the *improved DQMOM* one basically does the same calculations and obtains for $k = 1,\ldots,2n$

$$\begin{aligned} \int_0^\infty P_k(e)S(t,e)\,\mathrm{d}e \approx &\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[P_k(e_i+e_j) - P_k(e_i)\right]w_j w_i \\ &+ \frac{1}{2}\Phi(\infty)\sum_{i=1}^{n}\left[2\frac{1}{e_i}\int_0^{e_i} P_k(e)\,\mathrm{d}e - P_k(e_i)\right]e_i w_i. \end{aligned}$$

The polynomials P_k are by construction of degree $2n-1$. Therefore one can use the n-point *Gauss–Legendre quadrature* for the exact integration of the integral in the second summand. After transforming the interval one obtains

$$\int_0^{e_i} P_k(e)\,\mathrm{d}e = \frac{e_i}{2}\sum_{l=1}^{n} P_k\left(\frac{e_i}{2}(\xi_l+1)\right)\omega_l.$$

To obtain the weights and abscissas for this specific quadrature one can use one of the algorithms introduced in Section 3.3 or 3.4 with the moments

$$\mu_k = \begin{cases} \dfrac{2}{k+1}, & k \text{ even} \\ 0, & k \text{ uneven} \end{cases}.$$

It is not possible to use the *PDA* since negative abscissas are involved and the *PDA* therefore would fail.
This calculation has only to be performed once at the beginning of the simulations.

5.2 Comparison of Quadrature - Algorithms

In this section the three algorithms 3.2, 3.3 and 3.4 discussed in Section 3 are compared when they are used in the *QMOM*. Since all of the moment equations are reduced to exclusively time dependent equations one can use *Runge Kutta Methods* for solving these equations. We have used the standard fourth order *Runge Kutta Method*, i.e. written in a *Butcher Tableau*

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array} .$$

Given the same set of moments the algorithms basically give the same results. Therefore we will not present any results of the calculated moments, but we will focus on the time that is needed for a calculation. We recall the results for the number of operations needed in these algorithms

	PDA 3.2	*LQMD* 3.3	*GWA* 3.4
Summations	$2n^2-1$	$2(n-1)^2+3(n-1)+1$	$\frac{n(n+1)}{2}+\frac{n^3-n}{6}+n$
Multiplications	$4n^2+n-2$	$(n-1)^2+(n-1)$	$\frac{n(n+1)}{2}+\frac{n^3-n}{6}$
Divisions	$2n-1$	$(n-1)^2+2n-1$	$3n-1+\frac{n(n+1)}{2}$
Square Roots	$n-1$	$n-1$	$n+1$
Eigenvalue Problem	1	1	1

and specifically for $n=3$

	PDA 3.2	*LQMD* 3.3	*GWA* 3.4
Summations	17	15	13
Multiplications	37	6	10
Divisions	5	9	14
Square Roots	2	2	4
Eigenvalue Problem	1	1	1

The second algorithm 3.3 was used for the standard moments. The third algorithm needs an extra moment m_{2n}. The first one is calculated from the initial data and the following ones are calculated from the obtained weights and abscissas using the given quadrature rule

$$m_{2n} \approx \sum_{i=1}^{n} e_i^{2n} w_i.$$

A problem that occurs is, that this value is not the exact value.
Nicht im Original enthalten: Using Theorem 3.8 and 3.4 one obtaines for the approximation error

$$\int_0^\infty e^{2n} f(t,e)\,\mathrm{d}e - \sum_{i=1}^{n} e_i^{2n} w_i = \frac{\mathrm{d}^{(2n)}}{\mathrm{d}e^{(2n)}} e^{2n}\Big|_{e=\xi} \langle p_n, p_n\rangle = \langle p_n, p_n\rangle = \prod_{i=1}^{n} \alpha_i^2.$$

As a consequence of this it may happen that the matrix (3.29) is not positive definite. Therefore the Cholesky decomposition might fail. We observed that the *Matlab* procedure *chol* returned an error because of that. But when the formula (3.30) are used, one can still perform this algorithm. We guess that the use of the approximated moment makes the matrix analytically not positive definite.

But the error seems to be small enough so that the method remains stable. As mentioned before we did not run tests with *Newton's Method* 3.5, because of the analytical results obtained for the convergence theorem 3.10. Another disadvantage is the fact that all of the first three algorithms perform a finite number of steps until the result is obtained. Whereas *Newton's Method* is iterative and therefore it is difficult to predict the number of steps that are needed for a certain accuracy. All calculations were performed in *Matlab 7.4*. For all the Problems we have chosen $T = 10$, $dt = 0.01$ and $n = 3$. The times were measured using the *Matlab* commands *tic* and *toc*. The time values are given in seconds.

	PDA 3.2	*LQMD* 3.3	*GWA* 3.4
Problem I	0.5741	0.5481	0.6095
Problem II	0.5784	0.5471	0.6125
Problem III	0.5608	0.5454	0.6116
Problem IV	0.4728	0.4586	0.5162
Problem V	0.4763	0.4533	0.4774
Problem VI	0.4758	0.4539	0.5004
Problem VII	0.4709	0.4539	0.5

The important outcome of this is, that the second method, the *Long Quotient Modified Difference Algorithm* is the fastest. This is consistent with the number of operations given in the table above. The precise times may vary from system to system. Here the time differences are not that large but we only simulated simple problems for only one location. In more difficult *CFD* computations one needs to perform this computations in much more than one location. Even the size of a time step may be decreased which leads to more iterations until the final time and therefore one expects larger time differences between the algorithms.

5.3 Comparison of the Three Main Methods

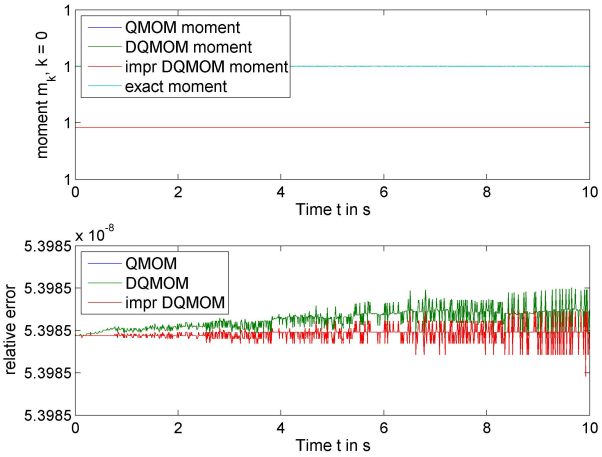
We finally want to compare the three Methods *QMOM*, *DQMOM* and the *improved DQMOM* for the seven introduced problems. Again we have used the standard fourth order *Runge Kutta Method*. It is important to note that this is a relevant topic for itself. The moments should always be the moments of a positive weight function. Therefore they have to satisfy certain conditions. A good indication for failure are negative abscissas during the calculations. A work that deals with this topic was recently published by *Vikas et al.* [17]. For the problems treated here the standard method just worked fine and we were not concerned with this topic.
If the problems would be space dependent one could use the *Method of Lines* which means that at first the space variable is discretised and then the time integration is applied to the obtained system. Taking into account the results from the previous Section 5.2 all of the calculations used the *Long Quotient Modified Difference Algorithm* 3.3. As before we have chosen $T = 10$ and $dt = 0.01$ for all problems. It is possible to choose a bigger dt for some problems. The initial moments were calculated using a standard (left) rectangle rule on the interval $[0, 100]$ with a step size $h_e = 0.1$

$$m_k(0) = \sum_{i=1}^{1000} ((i-1)h_e)^k f_0((i-1)h_e)h_e.$$

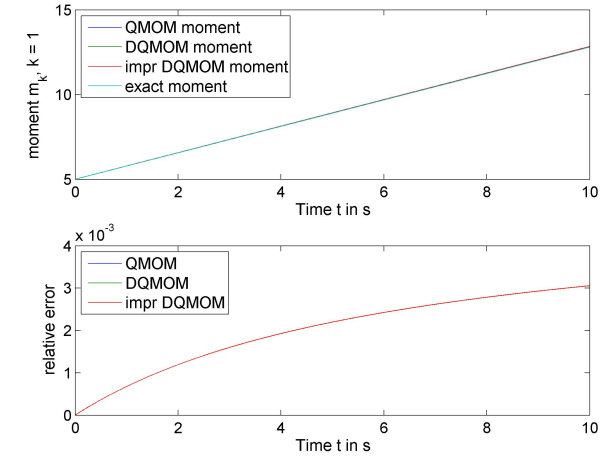
For the first three problems we have chosen the following constants according to [12]

$$a = 0.108, \quad b = 0.6 \quad \text{and} \quad \beta = 0.78.$$

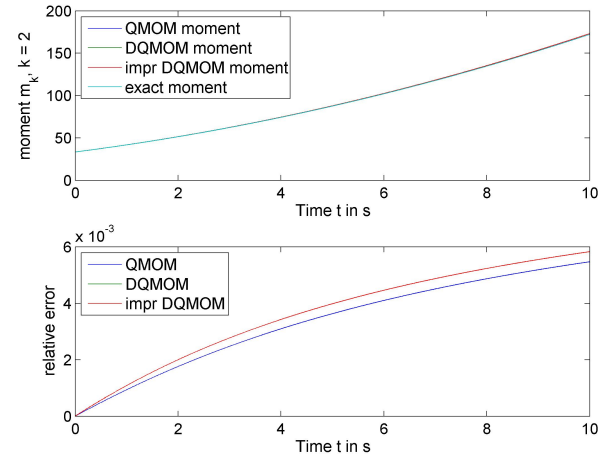
The *Problems I – III* all model growth laws with a constant number of particles, normalised to one. *Problem I* is presented in Figure 2 and Figure 3. It describes particle growth in a free-molecular size regime, see [12].
Problem II models the growth of solution droplets for sulfric acid-water droplets under certain quasi-equilibrium conditions, see [12] and is presented in Figure 4 and Figure 5.
In Figure 7 and Figure 8 we display the results for *Problem III*. This problem describes diffusion controlled growth, see [12].
The results for *Problem IV* are shown in Figure 9 and Figure 10. For this problem we have used $\Phi(\infty) = 0.1$.
Figure 11 and Figure 12 display the results for *Problem V*. Here we have used $\Phi(\infty) = 5$.
The steady state result for *Problem VI* with $\Phi(\infty) = 1$ is presented in Figure 13 and Figure 14.
Finally Figure 15 and Figure 16 show the results for *Problem VII*.



(a) m_0

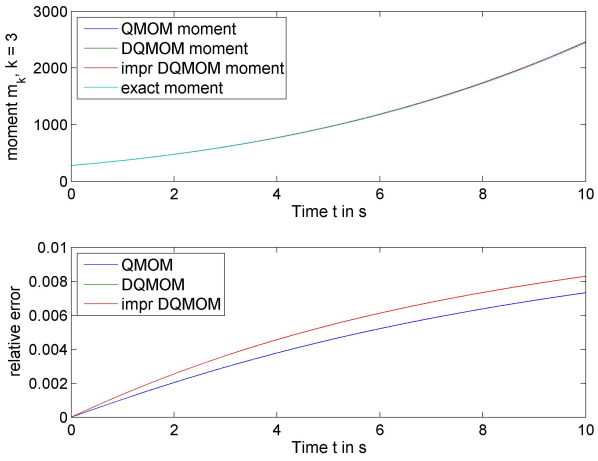


(b) m_1

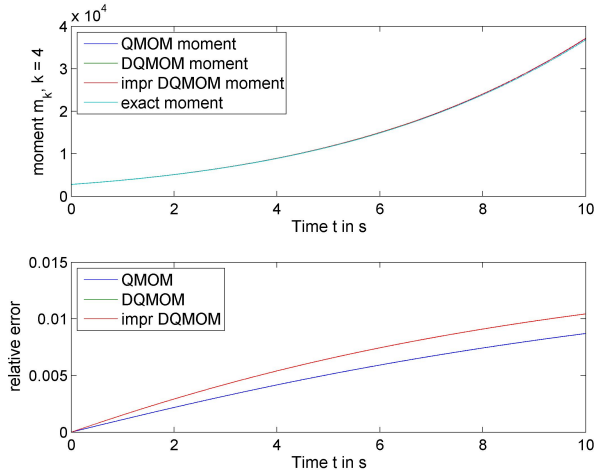


(c) m_2

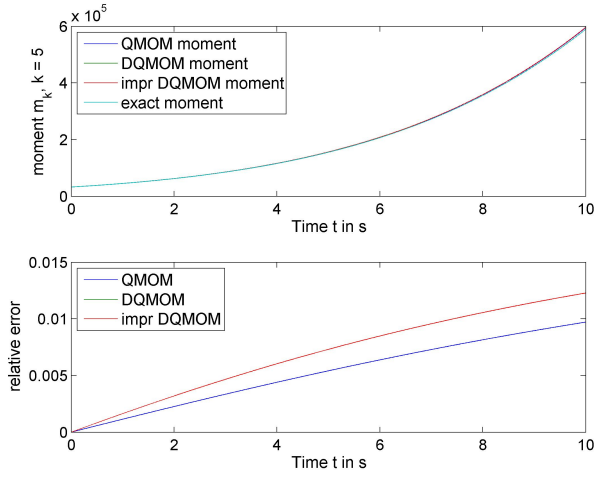
Fig. 2: Problem I, calculated moments m_0, m_1, m_2 and the relative error



(a) m_3

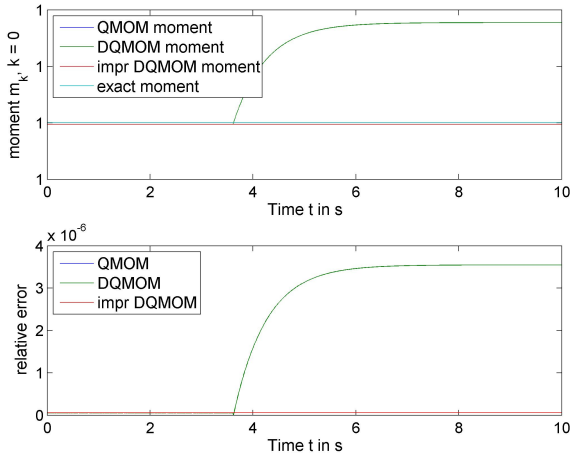


(b) m_4

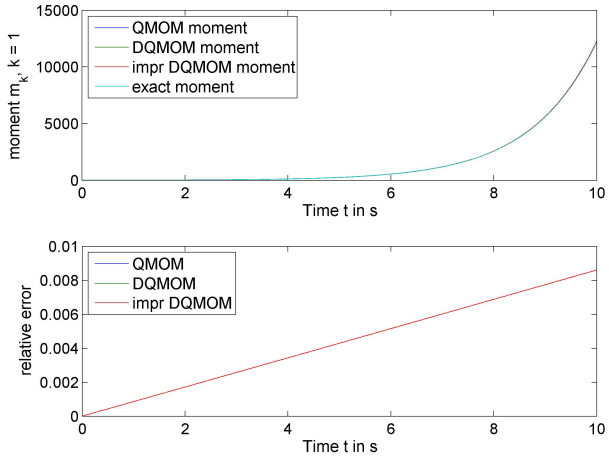


(c) m_5

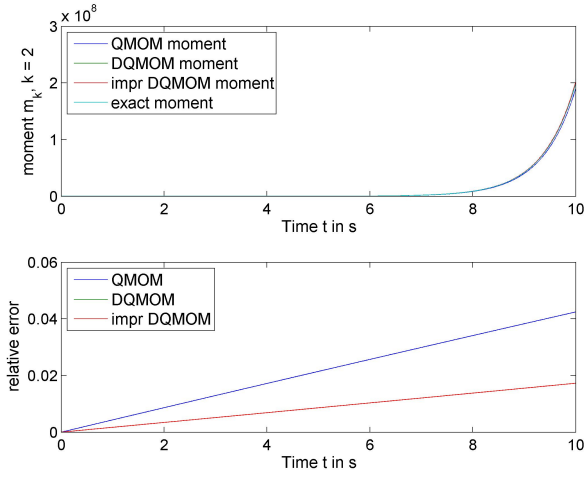
Fig. 3: Problem I, calculated moments m_3, m_4, m_5 and the relative error



(a) m_0

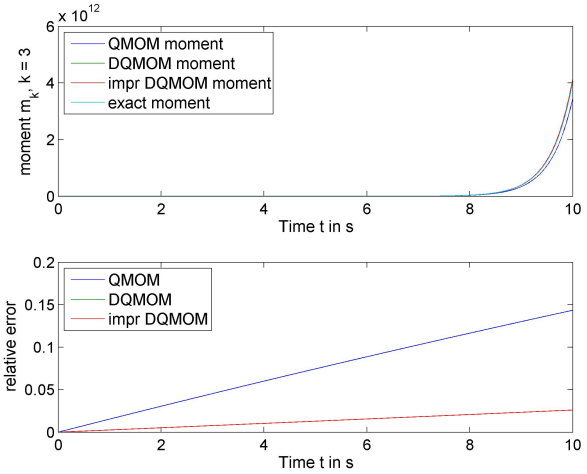


(b) m_1

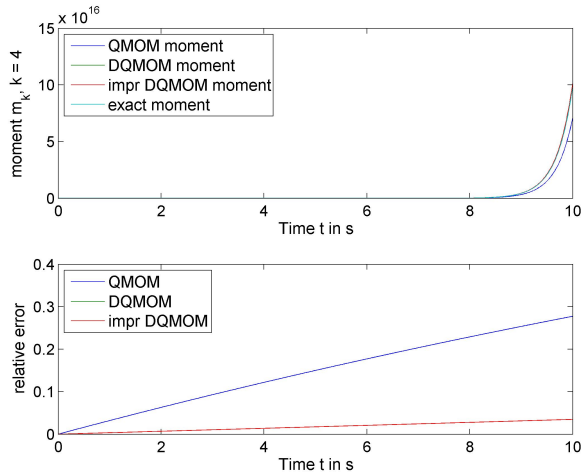


(c) m_2

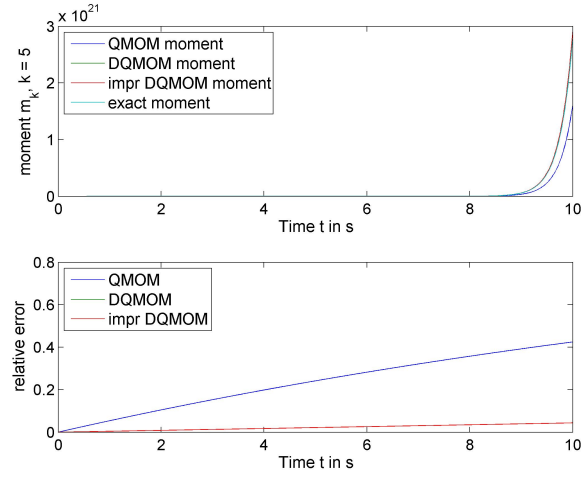
Fig. 4: Problem II, calculated moments m_0, m_1, m_2 and the relative error



(a) m_3



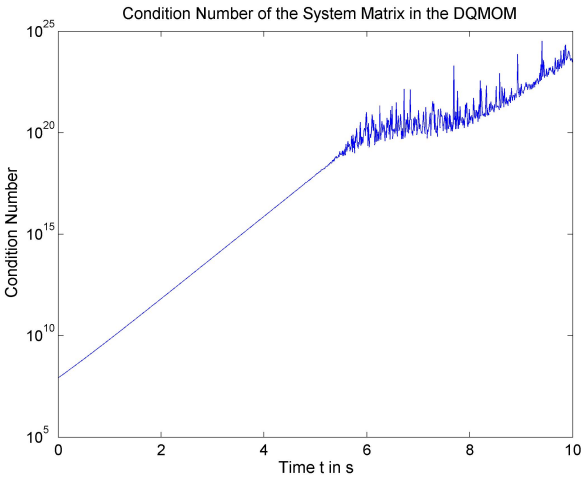
(b) m_4



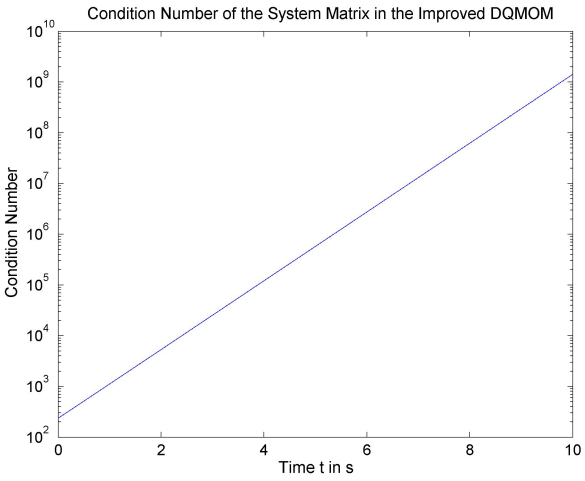
(c) m_5

Fig. 5: Problem II, calculated moments m_3, m_4, m_5 and the relative error

We will now compare the condition numbers of the linear systems in the *DQMOM* and *improved DQMOM* for *Problem II*. We used the same parameters as before. In Figure 6 on can clearly see the improvement due to the test functions. That the condition number is still that big is because of the fact that the value for the largest abscissa is $e_n \approx 3.7626\,10^4$. The growth of the condition number in time is due to the problem. The moments of this problem grow and become very large, e.g. $m_5 \sim 10^{21}$.

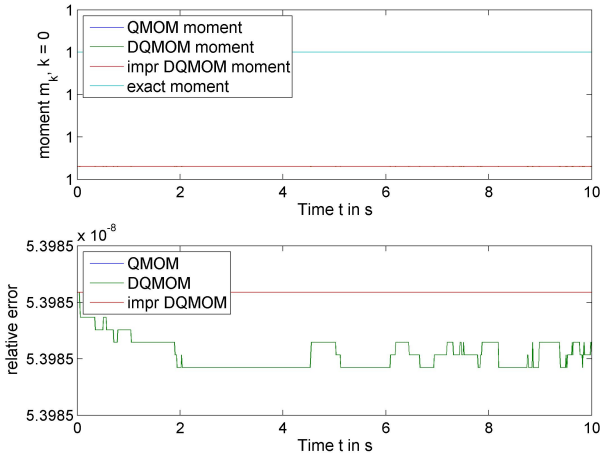


(a) *DQMOM*

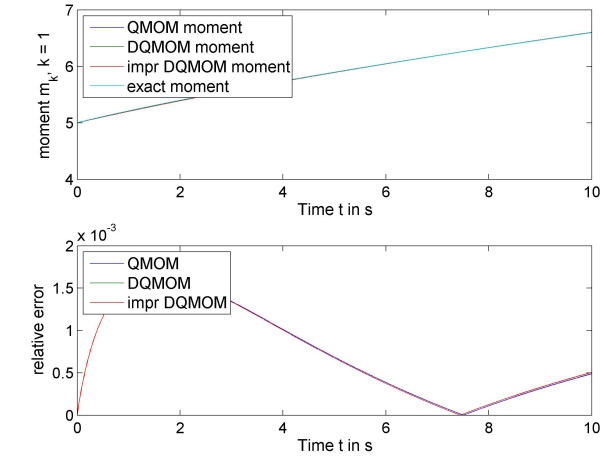


(b) *improved DQMOM*

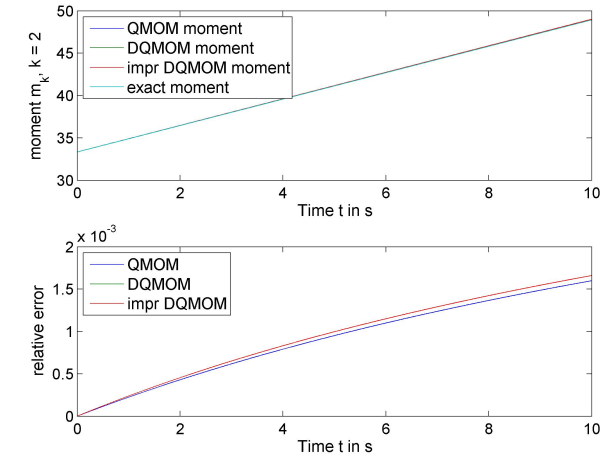
Fig. 6: Condition number for the linear system in the DQMOM and improved DQMOM



(a) m_0

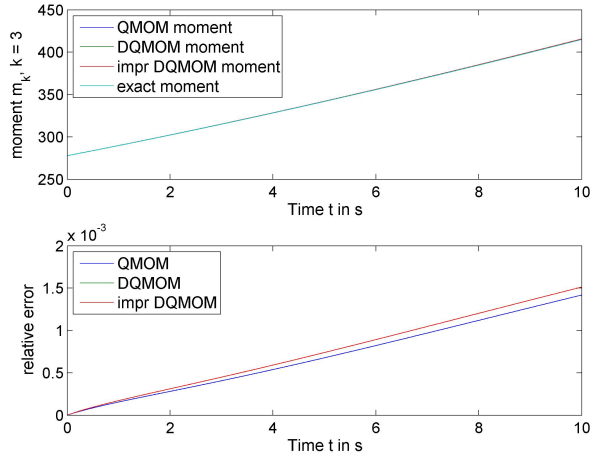


(b) m_1

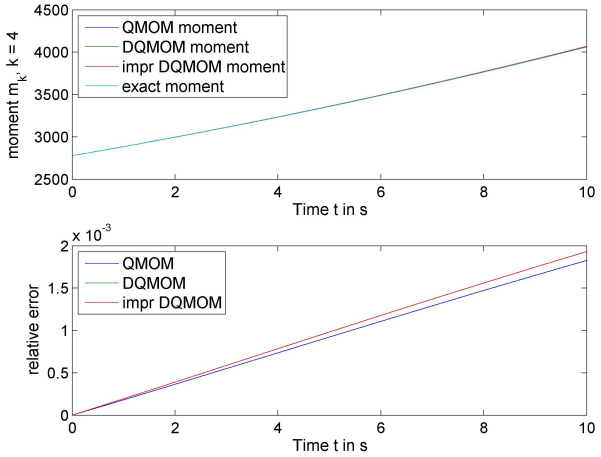


(c) m_2

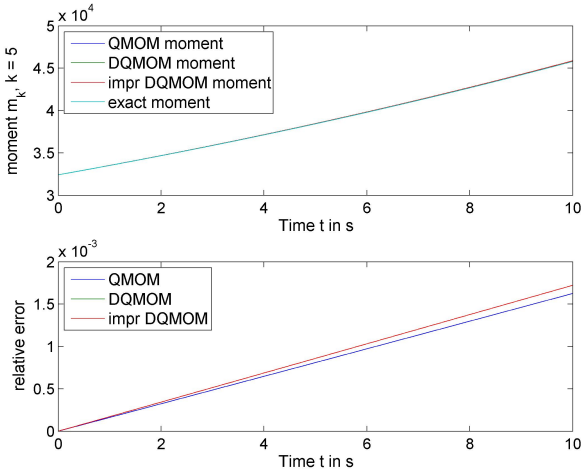
Fig. 7: Problem III, calculated moments m_0, m_1, m_2 and the relative error



(a) m_3

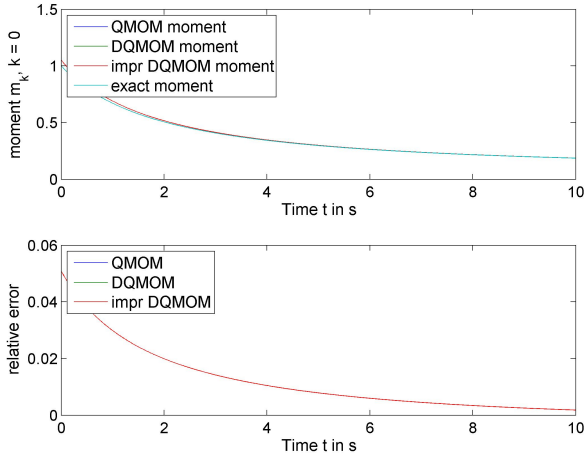


(b) m_4

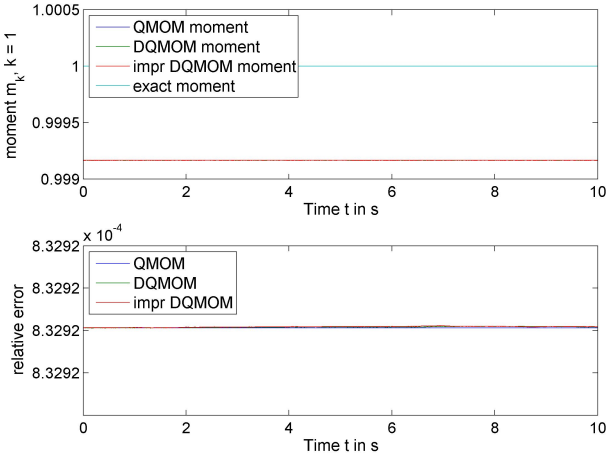


(c) m_5

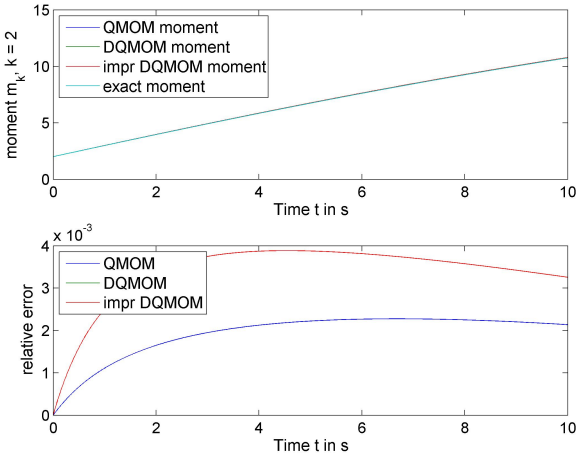
Fig. 8: Problem III, calculated moments m_3, m_4, m_5 and the relative error



(a) m_0

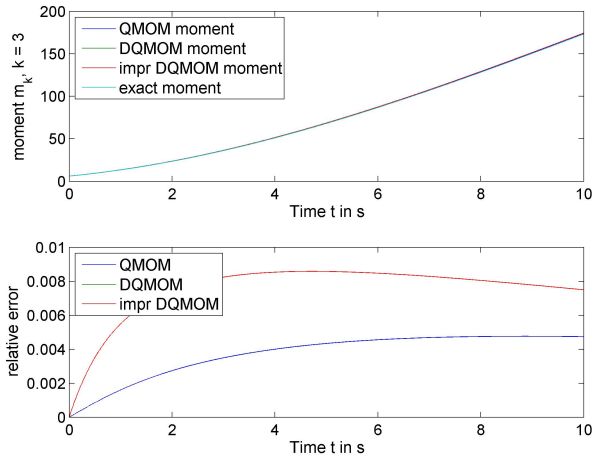


(b) m_1

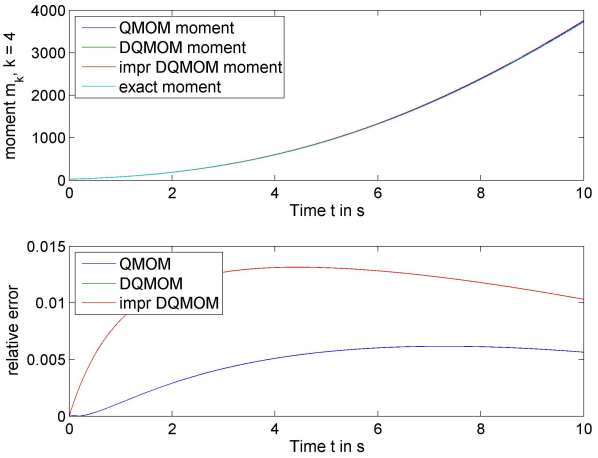


(c) m_2

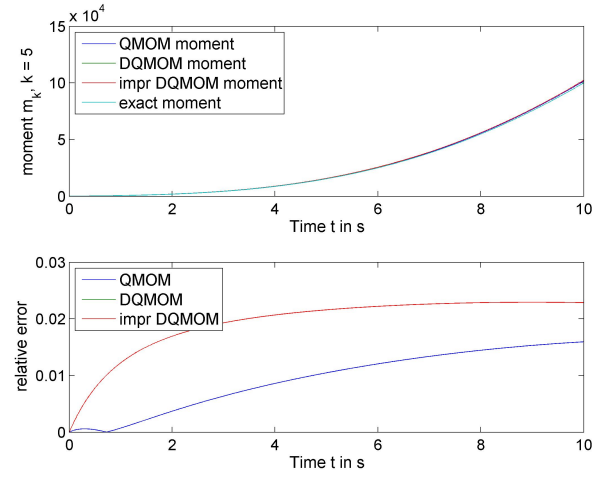
Fig. 9: Problem IV, $\Phi(\infty) = 0.1$, calculated moments m_0, m_1, m_2 and the relative error



(a) m_3

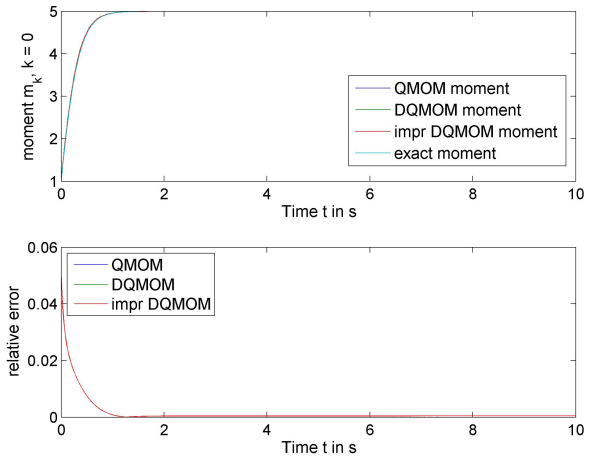


(b) m_4

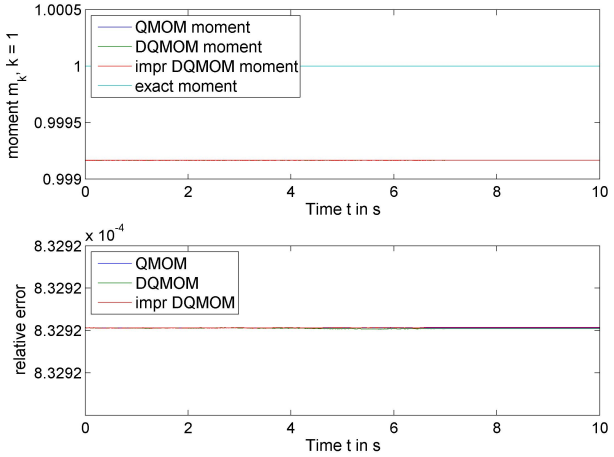


(c) m_5

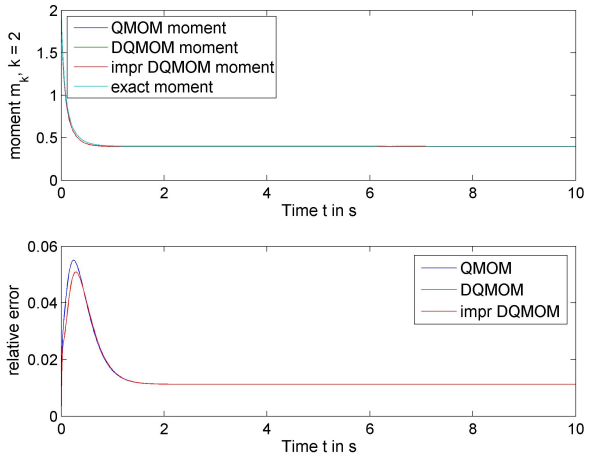
Fig. 10: Problem IV, $\Phi(\infty) = 0.1$, calculated moments m_3, m_4, m_5 and the relative error



(a) m_0

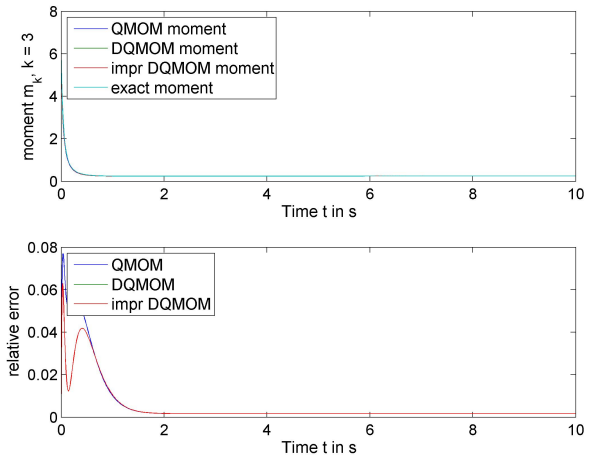


(b) m_1

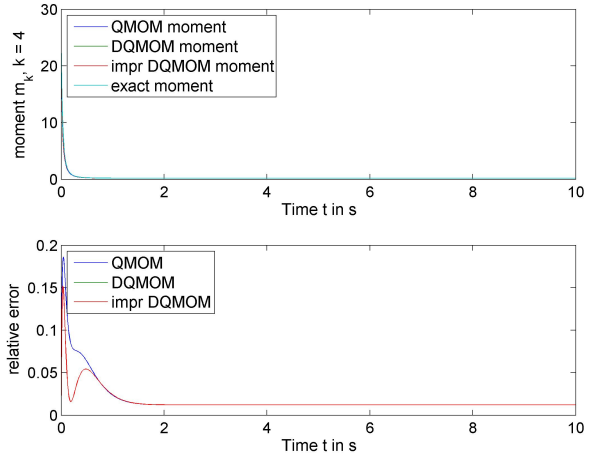


(c) m_2

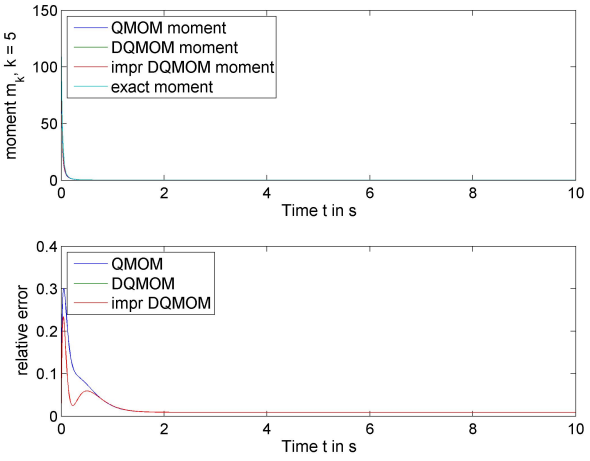
Fig. 11: Problem V, $\Phi(\infty) = 5$, calculated moments m_0, m_1, m_2 and the relative error



(a) m_3

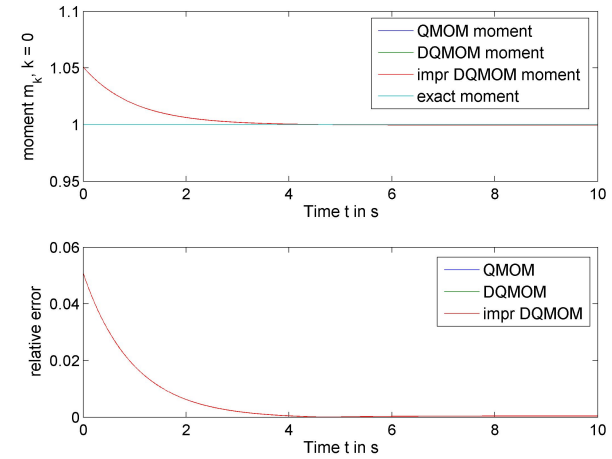


(b) m_4

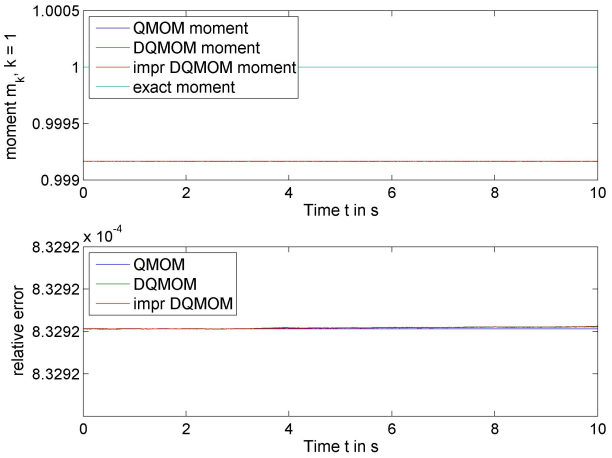


(c) m_5

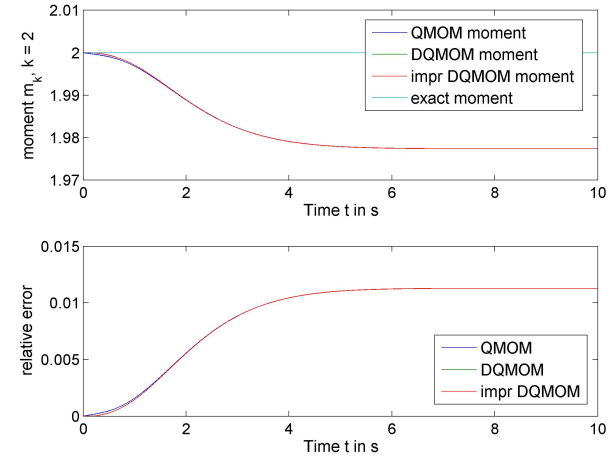
Fig. 12: Problem V, $\Phi(\infty) = 5$, calculated moments m_3, m_4, m_5 and the relative error



(a) m_0

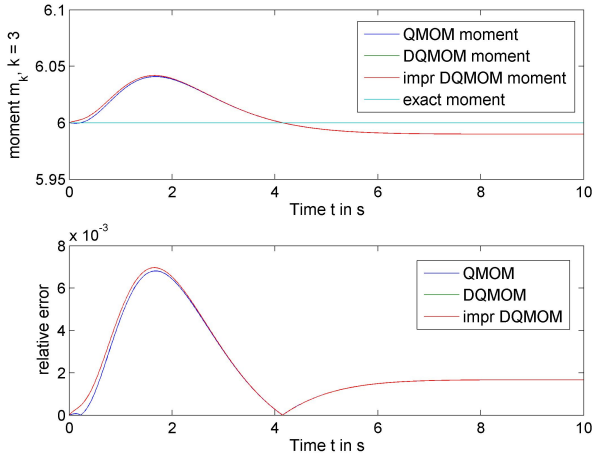


(b) m_1

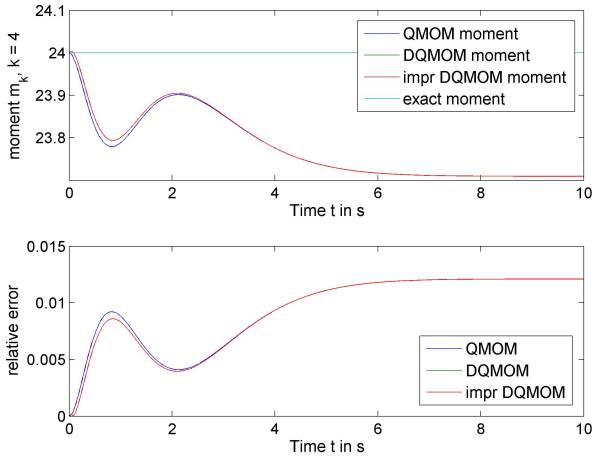


(c) m_2

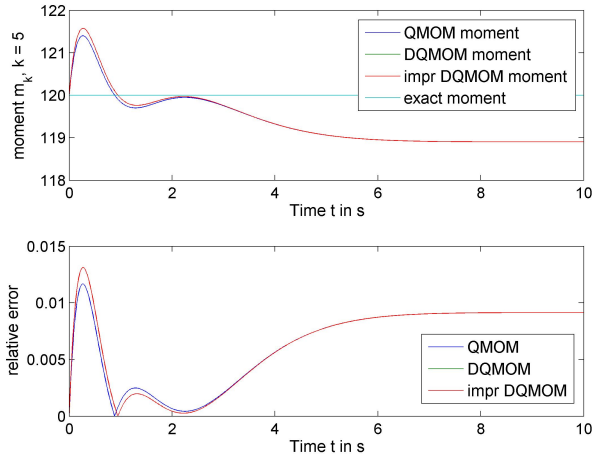
Fig. 13: Problem VI, $\Phi(\infty) = 1$, calculated moments m_0, m_1, m_2 and the relative error



(a) m_3

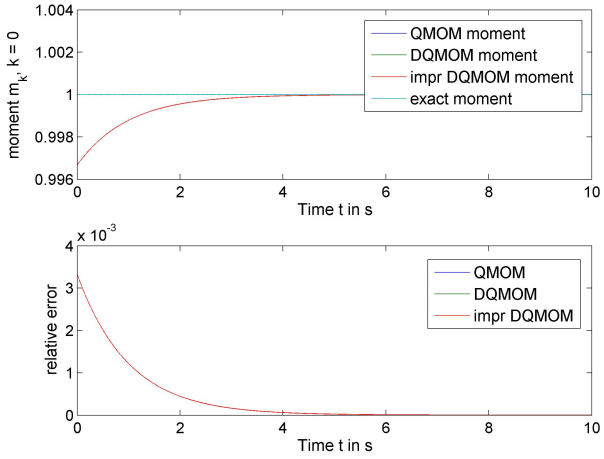


(b) m_4

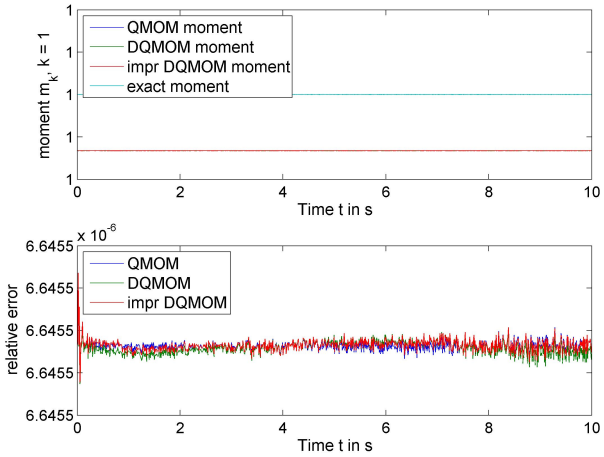


(c) m_5

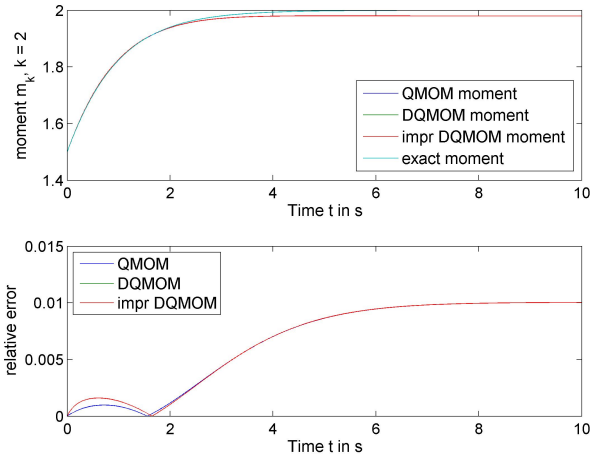
Fig. 14: Problem VI, $\Phi(\infty) = 1$, calculated moments m_3, m_4, m_5 and the relative error



(a) m_0

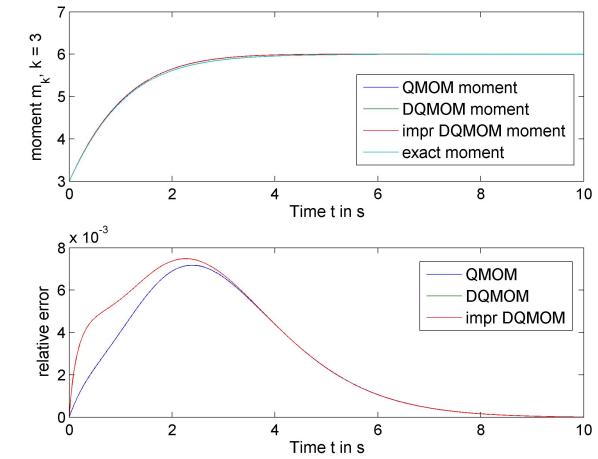


(b) m_1

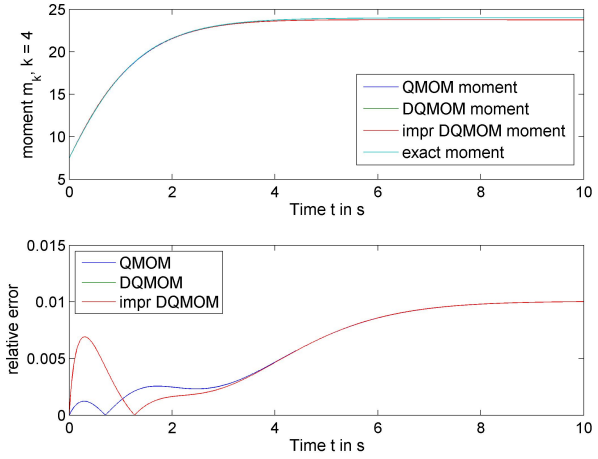


(c) m_2

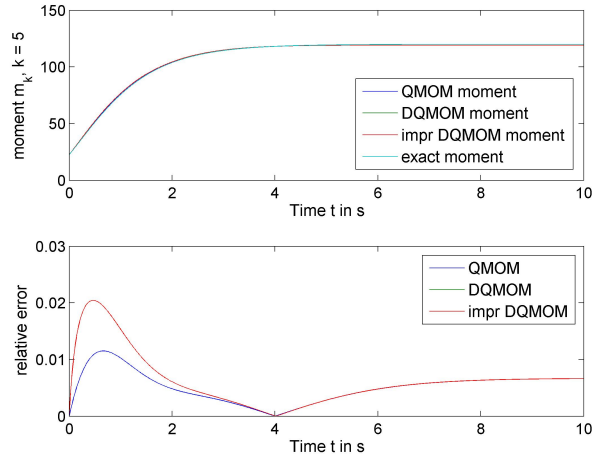
Fig. 15: Problem VII, calculated moments m_0, m_1, m_2 and the relative error



(a) m_3



(b) m_4



(c) m_5

Fig. 16: Problem VII, calculated moments m_3, m_4, m_5 and the relative error

These results can be interpreted as follows. The three methods basically give the same results for the case of one internal variable. The qualitative behaviour of the relative error does not differ very much. In fact in many figures one sees overlapping lines due to the same results. For the first three problems the results can also be compared to those in [12] and one will see the consistency. It should be remarked that the polynomials obtained in Section 4 seem to fit very good to the problems one to three, but not that good to the remaining problems. The improved *DQMOM* simplifies the calculation of some source terms drastically, as shown in (4.9). This then can be seen for example in Figure 4 and 5 for *Problem II*. This underlines the fact that one has to find different test functions for different types of problems. We therefore suggest to use the *QMOM* combined with the *LQMD* for problems with one internal variable, if the test functions do not improve the computation. For the case of more then one internal variable further work will be required. The fact that the relative error seems to be quite large in the beginning, for example for m_0 of *Problem IV* Figure 9, is due to the approximation error of the initial moments. If the accuracy would be higher the relative error will decrease in the beginning. This does not affect the qualitative behaviour of the relative error.

6 Conclusion

In this work we have presented moment based methods for the numerical treatment of *Population Balance Equations*. The *MOM* was discussed as it highlights the key idea of using the moments. Furthermore we have shown the main disadvantage that led to the formulation of the *QMOM*.
For the *QMOM* we discussed four methods to obtain the quadrature weights and abscissas. The *Product Difference Algorithm* 3.2 was first discussed, since it was suggested in [12] and therefore it became the commonly used one. A comprehensive proof of correctness was given for this algorithm. The understanding of such algorithms is of importance when such methods as the *QMOM* and their potential failures are discussed.
As the second algorithm we introduced the *Long Quotient Modified Difference Algorithm* 3.3. To the authors knowledge, this algorithm has not been used in the *QMOM* yet. We have shown that this algorithm needs less operations than the *PDA* when used for the standard moments. Furthermore it can be applied to modified moments which may increase the stability of the whole method. Considering the results in section 5.2 we suggest to use this algorithm for calculations where such algorithms play a crucial role in the process.
The third algorithm 3.4 presented by *Golub* and *Welsch* was discussed as another alternative. Although it is not recommended for the use in the *QMOM* for one internal variable, we suggest to investigate this algorithm for multidimensional quadrature. In the original work [6] the moment matrix (3.29) was derived for multidimensional moments and even the result that the columns of the inverse matrix form an orthogonal system of polynomials is given for more than one dimension. Therefore we assume that the remaining part can be extended to multivariate case, at least for certain Ω_e. That would give the opportunity to easily extend the *QMOM* to multivariate cases as an alternative to the *DQMOM*.
The last algorithm that was discussed was *Newton's Method*. We have shown theoretical worst case estimates for the convergence of this method. This led to the conclusion that this approach is very expensive an therefore it is not recommended. It still may be that there is a feasible practical approach to use this method.
The next method that was introduced is the *DQMOM*. For this method we discussed the standard derivation as given in [10] and an approach without using distributions. Furthermore we derived the multidimensional *DQMOM*. A result obtained in [5] was given in Section 2.4. This result is used to estimate the condition number of the nonlinear system (2.8) and the system matrix of the linear system (2.14) from below. To the authors knowledge this was not done before. Given the estimated condition number we thought of improving the *DQMOM*. In Section 4 we made our suggestions. We derived a formulation that makes it possible to choose any suited test function to work with in the *DQMOM*. We used polynomials that were obtained by *Hermite interpolation*. That might not be the optimal choice, but we surely improved the condition number of the linear system. With this new approach one can investigate the underlying problem and then choose the right set of test functions that for example reduce computational time or increase the stability of the calculations. Our approach was only discussed for the mono variate case and it includes the standard *DQMOM*. It therefore has to be investigated in which way this can be extended to the multivariate case.
Finally we discussed seven different problems in order to compare the numerical with the analytical results. These have shown that the three compared methods are nearly giving the same results.
The last thing we would like to remark is concerned with the reconstruction of the *Particle Size Distribution*. The first three algorithms discussed in Section 3 all calculate certain coefficients β_i and α_i. These are the coefficients of the recurrence relation for the orthogonal polynomials corresponding to the *PSD*. Given $2n$ moments it is therefore possible to determine these polynomials up to p_n. Now one can think of expanding the *PSD* in a series of its orthogonal polynomials. Perhaps this will give a good approximation. Of course one has to increase n. We have done calculations with $n = 10$ and obtained satisfying results for all the moments. Another possible approach can be the idea which

was used to prove the correctness of the *Product Difference Algorithm*. One knows the moments and therefore one can calculate an approximation of the *Stieltjes Transform* of the *PSD*. It remains to invert the transform to obtain an approximation of the *PSD* itself.

References

[1] Claire Laurent, Gerard Lavergne and Philippe Villedieu. Quadrature method of moments for modeling multi–component spray vaporization. *International Journal of Multiphase Flow*, 36:51–59, 2010.

[2] Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, 1998.

[3] Roland W. Freund and Roland H.W. Hoppe. *Stoer/Bulirsch: Numerische Mathematik I.* Springer–Verlag Berlin Heidelberg, 10 edition, 2007.

[4] Walter Gautschi. On inverses of Vandermonde and confluent Vandermonde matrices. ii. *Numerische Mathematik*, 5:425–430, 1963.

[5] Walter Gautschi. Construction of Gauss-Christoffel quadrature formulas. *Mathematics of Computation*, 22(102):251–270, 1968.

[6] Gene H. Golub and John H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230, 1969.

[7] Roy G. Gordon. Error bounds in equilibrium statistical mechanics. *Journal of Mathematical Physics*, 9:655 – 663, 1968.

[8] H.M. Hulburt and S. Katz. Some problems in particle technology - a statistical mechanical formulation. *Chemical Engineering Science*, 19:555 – 574, 1964.

[9] P.L.C. Lage. 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. *Chemical Engineering Science*, 57:4253 – 4254, 2002.

[10] D.L. Marchisio and R.O. Fox. Solution of population balance equations using the direct quadrature method of moments. *Journal of Aerosol Science*, 36:43–73, 2005.

[11] Benjamin J. McCoy and Giridhar Madras. Analytical solution for a population balance equation with aggregation and fragmentation. *Chemical Engineering Science*, 58:3049 – 3051, 2003.

[12] Robert McGraw. Description of aerosol dynamics by the quadrature method of moments. *Aerosol Science and Technology*, 27:255 – 265, 1997.

[13] D.P. Patil and J.R.G. Andrews. An analytical solution to continuous population balance model describing floc coalescence and breakage – a special case. *Chemical Engineering Science*, 53(3):599–601, 1998.

[14] Rong Fan, Daniele L. Marchisio and Rodney Fox. Application of the direct quadrature method of moments to polydisperse gas–solid fluidized beds. *Power Technology*, 139:7–20, 2004.

[15] R.A. Sack and A.F. Donovan. An algorithm for Gaussian quadratur given modified moments. *Numer. Math., Springer*, 18:465–478, 1972.

[16] V. John, I. Angelov, A.A. Öncül and D. Thévenin. Techniques for the reconstruction of a distribution from a finite number of its moments. *Chemical Engineering Science*, 62:2890–2904, 2007.

[17] V. Vikas, Z.J. Wang, A. Passalacqua and R.O. Fox. Realizable high–order finite–volume schemes for quadrature–based moment methods. *Journal of Computational Physics*, 230:5328–5352, 2011.

[18] Volker John, Teodora Mitkova, Michael Roland, Kai Sundmacher, Lutz Tobiska and Andreas Voigt. Simulations of population balance systems with one internal coordinate using finite element methods. *Chemical Engineering Science*, 64:733–741, 2009.

[19] H.S. Wall. *Analytic Theory of Continued Fractions*, volume 1 of *The University Series In Higher Mathematics*. D. Van Nostrand Company, Inc., 1967.

Erklärung

Ich, Ferdinand Thein, versichere hiermit gemäß §9 der Prüfungsordnung für den Diplomstudiengang Mathematik, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen Hilfsmittel und Quellen außer den Angegebenen verwendet habe. Diese Arbeit wurde bisher noch keiner Prüfungsbehörde vorgelegt und ist noch nicht veröffentlicht.

_____

Ort, Datum, Unterschrift