AIChE

Automatic Differentiation-Based Quadrature Method of Moments for Solving Population Balance Equations

Vinay Kariwala
School of Chemical & Biomedical Engineering, Nanyang Technological University, Singapore 637459

Yi Cao
School of Engineering, Cranfield University, Cranfield, Bedford MK43 0AL, U.K.

Zoltan K. Nagy
Dept. of Chemical Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, U.K.

DOI 10.1002/aic.12613
Published online May 18, 2011 in Wiley Online Library (wileyonlinelibrary.com).

The quadrature method of moments (QMOM) is a promising tool for the solution of population balance equations. QMOM requires solving differential algebraic equations (DAEs) consisting of ordinary differential equations related to the evolution of moments and nonlinear algebraic equations resulting from the quadrature approximation of moments. The available techniques for QMOM are computationally expensive and are able to solve for only a few moments due to numerical robustness deficiencies. In this article, the use of automatic differentiation (AD) is proposed for solution of DAEs arising in QMOM. In the proposed method, the variables of interest are approximated using high-order Taylor series. The use of AD and Taylor series gives rise to algebraic equations, which can be solved sequentially to obtain high-fidelity solution of the DAEs. Benchmark examples involving different mechanisms are used to demonstrate the superior accuracy, computational advantage, and robustness of AD-QMOM over the existing state-of-the-art technique, that is, DAE-QMOM. © 2011 American Institute of Chemical Engineers *AIChE J,* 58: 842–854, 2012
Keywords: automatic differentiation, dynamic simulation, particulate processes, population balance equations, quadrature method of moments

Introduction

Population balance models have been widely used for modelling particulate, droplet, or bubble dynamics in single or multiphase processes.[1–3] The solution of a population balance equation (PBE) usually requires complex numerical techniques. The variety of solution approaches proposed for the PBE includes the standard method of moments (MOM),[4] the quadrature method of moments (QMOM),[5] the method of characteristics,[6–8] method of weighted residuals,[9] direct numerical solution techniques, such as finite element scheme,[10] fixed pivot method,[11] high-resolution finite volume methods,[12–14] weighted essentially nonoscillatory methods[15–17] and lattice Boltzmann method,[18,19] and kinetic Monte Carlo simulation approaches.[20,21] The numerical robustness and computational efficiency of the solution methods are of significant importance, especially in the cases of model-based control and optimization, as well as in coupled computational fluid dynamics and PBE applications.[22,23]

A preliminary version of this work was presented at 9th International Symposium on Dynamics and Control of Process Systems held in Leuven, Brussels in July 2010.
Correspondence concerning this article should be addressed to V. Kariwala at vinay@ntu.edu.sg.

The QMOM proposed by McGraw[5] is one of the most efficient approaches for solving generic PBEs with growth, nucleation, aggregation/coalescence, and breakage mechanisms. The QMOM utilizes the quadrature theory to avoid the closure problem encountered in the case of standard MOM simulations. The PD-QMOM by McGraw[5] is based on the product difference algorithm (PD) of Gordon.[24] Application of the PD-QMOM has been extended to aggregation, coagulation, and breakage systems.[25–27] However, the PD algorithm is not always the best approach for computing the quadrature points from the moments of the particle size distribution, because for a larger number of moments, the method is sensitive to small errors. Therefore, the applicability of PD-QMOM is limited to no more than six quadrature points[5] and generally even fewer for more complex processes, such as diffusion-controlled growth with secondary nucleation.[28]

Several variants of the QMOM methods have been developed recently, such as Jacobian matrix transformation (JMT) method,[29] direct QMOM (DQMOM),[30] and fixed QMOM (FQMOM).[31] Similar to the PD-QMOM, the JMT method suffers from ill-conditioning requiring the number of moments to be small.[29] For one-dimensional (1D) PBEs, as considered in this article, DQMOM has been shown to provide the same level of accuracy as PD-QMOM. Although DQMOM overcomes some of the limitations of PD-QMOM, it can still be ill-conditioned for higher number of moments.[32] The FQMOM method requires finding an optimized arbitrary constant, which may require many trial and error iterations for complex systems.[28] Recently, an alternate solution technique for the QMOM was introduced based on the simultaneous solution of the moment equations and quadrature approximation as a semiexplicit differential-algebraic equation (DAE) system.[28,33] The DAE-QMOM[28] showed increased robustness and significantly better computational efficiency than PD-QMOM. These advantages, however, can only be achieved with the analytical computation of the Jacobian matrix of the DAE system, which is not readily possible for complex mechanisms.

In this article, a novel methodology is proposed for the solution of 1D PBEs based on an automatic differentiation (AD) algorithm. AD lies in a class of computational techniques used to evaluate the derivatives of functions defined in computer programs.[34,35] Such programs consist of a sequence of elementary operations, whose derivatives are well known. By numerically applying the chain rule to these arithmetic sequences, not only AD can deliver derivatives, which are free of truncation error, hence superior to finite difference approximation, but also avoids code growth, which is a common issue associated with symbolic differentiation approaches. In AD, high-order Taylor coefficients of continuous functions can be recursively obtained. Recursive algorithms for solving general differential equations and DAE problems using high-order Taylor series have also been developed.[36,37] Recently, the superior computational efficiency and numerical accuracy of this method for solving various dynamic optimization problems have been demonstrated.[38–40]

In this work, the AD-based recursive Taylor approach is applied to solve the DAEs arising in QMOM (AD-QMOM) with a focus to tackle the numerical robustness associated with the large number of moments and quadrature points. The numerical difficulty of the QMOM formulation is mainly due to the corresponding algebraic equations, which are the same for all applications. For these equations, the recursive algorithm derived in this work shows that the Taylor coefficients of the algebraic variables can be efficiently obtained by solving a set of linear equations. This result makes the AD-based approach much more efficient and accurate than other methods. The AD-QMOM approach is evaluated for various mechanisms using benchmark examples taken from literature. It is shown that AD-QMOM provides increased robustness and computational efficiency compared with the DAE-QMOM.[28] In particular, AD-QMOM is shown to be 2–10 times faster than DAE-QMOM with similar levels of accuracy. Furthermore, AD-QMOM allows us to track the evolution of 14–20 moments easily, which is not possible using the existing QMOM techniques. We point out that although in practice only the first few lower order moments are physically meaningful, solving for a higher number of moments is still beneficial as it reduces the error associated with the lower order moments arising due to quadrature approximation as well as facilitates the accurate reconstruction of the particle size distribution from the calculated moments.[41]

The rest of this article is organized as follows: first, the DAEs arising in QMOM for solving 1D PBEs are presented, and the AD-QMOM for solving these DAEs is derived. Subsequently, numerical examples involving different mechanisms are used to demonstrate that the AD-QMOM can provide the same level of accuracy as DAE-QMOM, when used for solving for low number of moments. Finally, the computational efficiency and robustness of AD-QMOM are evaluated, and the article is concluded.

Quadrature Method of Moments

The monovariate PBE for a closed well-mixed system can be written with diameter as the internal coordinate as:[31,42]

$$\frac{\partial n(L)}{\partial t} = \underbrace{\int_L^\infty b(L,\lambda)a(\lambda)n(\lambda)d\lambda}_{\text{birth due to breakage}} + \underbrace{\frac{L^2}{2}\int_0^L \frac{\beta((L^3-\lambda^3)^{1/3},\lambda)n((L^3+\lambda^3)^{1/3},\lambda)n(\lambda)}{(L^3-\lambda^3)^{2/3}}d\lambda}_{\text{birth due to coalescence or aggregation}} - \underbrace{a(L)n(L)}_{\text{death due to breakage}} - \underbrace{n(L)\int_0^\infty \beta(L,\lambda)n(\lambda)d\lambda}_{\text{death due to coalescence or aggregation}} - \underbrace{\frac{\partial(G(L)n(L))}{\partial L}}_{\text{growth}} + \underbrace{\delta(L-L_0)B}_{\text{nucleation}} \quad (1)$$

where n is the particle size distribution, and β, a, G, B, b, and L_0 are the aggregation kernel, breakage kernel, growth rate, nucleation rate, the daughter particle size distribution,

and the size of the nuclei, respectively, whereas both L and λ are the particle characteristic length. In Eq. 1, $\delta(L - L_0)$ represents the Dirac delta function, which is zero for all values of L, except when $L = L_0$, and its integral over L equals 1. The PBE in Eq. 1 can be simplified using a moment transformation, where the rth moment of the distribution, μ_r, is given by:

$$\mu_r = \int_0^\infty n(L) L^r dL. \tag{2}$$

The lower order moments (i.e., zeroth to third) are related to the physical description of the particle size distribution; that is, μ_0 is the total number of particles, and μ_1, μ_2, and μ_3 are proportional to the total diameter, total surface area, and total volume of the particles, respectively. After the moment transformation, the PBE in Eq. 1 is represented by a set of ordinary differential equations (ODEs) in terms of the moments:

$$\begin{aligned}\frac{d\mu_r}{dt} = & \underbrace{\int_0^\infty L^r \int_0^\infty a(\lambda) b(L,\lambda) n(\lambda) d\lambda dL}_{\text{birth due to breakage}} \\ & + \underbrace{\frac{1}{2}\int_0^\infty n(\lambda) \int_0^\infty \beta(L,\lambda)(L^3+\lambda^3)^{r/3} n(L) dL d\lambda}_{\text{birth due to coalescence or aggregation}} \\ & - \underbrace{\int_0^\infty L^r a(L) n(L) dL}_{\text{death due to breakage}} - \underbrace{\int_0^\infty L^r n(L) \int_0^\infty \beta(L,\lambda) n(\lambda) d\lambda dL}_{\text{death due to coalescence or aggregation}} \\ & - \underbrace{\int_0^\infty r L^{r-1} G(L) n(L) dL}_{\text{growth}} + \underbrace{BL_0^r}_{\text{nucleation}}. \end{aligned} \tag{3}$$

The moment equations represented by Eq. 3 are solvable for constant growth and nucleation problems using the standard MOM technique. This approach, however, cannot handle more complex mechanisms, such as breakage and coalescence, due to the closure problem, because the integrations cannot be written in term of the moments. Therefore, Eq. 3 needs to be transformed again into a QMOM formulation to eliminate the closure problem. The essence of the quadrature closure is to consider the number density $n(L)$ as a general weight function and to approximate the integrals that appear during the transformation of the PBE to moment equations in terms of a set of abscissas and weights. The QMOM use a quadrature approximation[5]:

$$\mu_r = \int_0^{+\infty} n(L) L^r dL \approx \sum_{\ell=1}^{N} w_\ell L_\ell^r; \quad r = 0, 1, \cdots, 2N-1 \tag{4}$$

where w_ℓ are the weights, L_ℓ are the abscissas, and N is the number of quadrature points. This quadrature approximation is exact if the function in Eq. 4 is a polynomial up to the order $2N - 1$. After applying the quadrature rule, the moment-transformed PBE can be written as[31,42]:

$$\begin{aligned}\frac{d\mu_r}{dt} = & \underbrace{\sum_{\ell=1}^{N} w_\ell a(L_\ell) b(r, L_\ell)}_{\text{birth due to breakage}} \\ & + \underbrace{\frac{1}{2}\sum_{\ell=1}^{N} w_\ell \sum_{m=1}^{N} w_m \beta(L_\ell, L_m)(L_\ell^3 + L_m^3)^{r/3}}_{\text{birth due to coalescence or aggregation}} \\ & - \underbrace{\sum_{\ell=1}^{N} w_\ell a(L_\ell) L_\ell^r}_{\text{death due to breakage}} - \underbrace{\sum_{\ell=1}^{N} w_\ell L_\ell^r \sum_{m=1}^{N} w_m \beta(L_\ell, L_m)}_{\text{death due to coalescence or aggregation}} \\ & - \underbrace{r\sum_{\ell=1}^{N} w_\ell L_\ell^{r-1} G(L_\ell)}_{\text{growth}} + \underbrace{BL_0^r}_{\text{nucleation}}. \end{aligned} \tag{5}$$

Now the closure problem has been eliminated, and hence, the PBE in (5) is solvable by QMOM approach by following the evolution of w_ℓ, L_ℓ, and μ_r. The moments are nonlinearly related to the weights and abscissas by (4).

The QMOM requires integration of the ODEs in (5) generated from the moment equations for $r = 0,1,\cdots,2N - 1$, alongside the solution of nonlinear algebraic equations in (4) obtained from the quadrature rule. Equations 4 and 5 together represent a semiexplicit index-1 DAE system, which can be solved using standard DAE solution techniques and software. The DAE method is attractive for using the QMOM, because it arises naturally from the mathematical formulation of the QMOM approximation problem.[28,33] For increased robustness, the Jacobian of the DAE system should be computed analytically and used in the numerical integration of the system.[28] Although previous studies have shown that the DAE-QMOM method provides a computationally more efficient and robust approach for solving PBEs than other QMOM-based approaches, this technique is still limited to a small number of quadrature points (typically not more than 5–6) and relies on the computation of the analytical Jacobian of the DAE system, which is not always possible. Thus, there is a strong need to develop alternate methods, which are more robust and computationally more efficient than DAE-QMOM. Such a method is presented in the next section.

Automatic Differentiation-Based QMOM

In this section, we present the AD-QMOM technique for solving the DAEs arising in the QMOM formulation. Before presenting the detailed derivation of the proposed technique, some general principles of AD and the use of high-order Taylor series expansion for solving the DAEs associated with the QMOM are presented. The general AD principles can be found in the monograph by Griewank.[35] Approaches using high-order Taylor series expansion for solving general DAEs have been proposed earlier in the literature.[36,37] For the DAEs associated with QMOM, however, the differential (μ_r) and algebraic (w_ℓ and L_ℓ) variables can be easily identified, which allows the derivation of a more efficient algorithm than the algorithms available for solving general DAEs.

Table 1. Useful Identities for Propagation of Taylor Series Coefficients[35]

	z	$z^{[k]}$
1	cx	$cx^{[k]}$
2	$x + y$	$x^{[k]} + y^{[k]}$
3	xy	$\sum_{j=0}^{k} x^{[k-j]} y^{[j]}$
4	x^r	$\frac{1}{k x^{[0]}} \left(r \sum_{j=1}^{k} j z^{[k-j]} x^{[j]} - \sum_{j=1}^{k-1} j x^{[k-j]} z^{[j]} \right)$

General principle

Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a d-time continuously differentiable function and $x(t) \in \mathbb{R}^n$ be given as a truncated Taylor series:

$$x(t) = x^{[0]} + x^{[1]} t + \cdots + x^{[d]} t^d = \sum_{k=0}^{d} x^{[k]} t^k \tag{6}$$

where $x^{[k]}$ is the kth Taylor coefficient of $x(t)$. Then, the Taylor coefficients of

$$z(t) = f(x(t)) = \sum_{k=0}^{d} z^{[k]} t^k \tag{7}$$

can be recursively calculated through AD based on the chain rule,

$$z^{[k]} \equiv f^{[k]}(x^{[0]}, \ldots, x^{[k]}). \tag{8}$$

A few examples of such recursive Taylor coefficient formulations are given in Table 1. The derivatives of $z^{[k]}$ satisfy:[43]

$$\frac{\partial z^{[k]}}{\partial x^{[j]}} = \frac{\partial z^{[k-j]}}{\partial x^{[0]}} = A^{[k-j]}(x^{[0]}, \ldots, x^{[k-j]}) \tag{9}$$

where $A^{[k-j]}$ is the $(k-j)^{\text{th}}$ Taylor coefficient matrix of the Jacobian matrix, $\frac{\partial f}{\partial x}$, which depends on $x^{[0]}, x^{[1]}, \cdots, x^{[k-j]}$. Based on Eq. 9, it can be readily proven that $f^{[k]}$ is linear with respect to $x^{[j]}$ for $k/2 < j \leq k$ and satisfies:[35]

$$f^{[k]}(x^{[0]}, \ldots, x^{[k]}) = f^{[k]}\big|_{x^{[j]} = \cdots = x^{[k]} = 0} + \sum_{i=j}^{k} A^{[k-i]}(x^{[0]}, \ldots, x^{[k-i]}) x^{[i]}. \tag{10}$$

In Eq. 10, $f^{[k]}|_{x^{[j]} = x^{[j+1]} = \ldots = x^{[k]} = 0} = f^{[k]}(x^{[0]}, \ldots, x^{[j-1]}, 0, \ldots, 0)$ is referred to as the partial Taylor coefficient with $x^{[j]}, x^{[j+1]}, \ldots, x^{[k]}$ set to zero. The linearity is based on the fact that $A^{[k-i]}$ is independent of $x^{[i]}$, whenever $i > k - i$. A special case of Eq. 10 for $j = k$ with $j,k \geq 1$ will be used to derive the solution of a DAE later.

Consider a DAE problem given as

$$\frac{dx}{dt} = \dot{x} = f(x, y) \tag{11}$$

$$0 = g(x, y) \tag{12}$$

where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ are differential and algebraic variables, respectively. Based on Eq. 6, $\dot{x}^{[k]} = (k+1) x^{[k+1]}$ and thus

$$x^{[k+1]} = \frac{1}{k+1} f^{[k]}(x^{[0]}, \ldots, x^{[k]}, y^{[0]}, \ldots, y^{[k]}) \tag{13}$$

For the algebraic equation in Eq. 12, according to Eq. 10 with $j = k$, $g^{[k]}$ is linear with respect to $y^{[k]}$ for $k > 0$, that is,

$$g^{[k]}(x^{[0]}, \ldots, x^{[k]}, y^{[0]}, \ldots, y^{[k]}) = g^{[k]}\big|_{y^{[k]}=0} + A y^{[k]} \tag{14}$$

where $g^{[k]}|_{y^{[k]}=0} = g^{[k]}(x^{[0]}, \ldots, x^{[k]}, y^{[0]}, \ldots, y^{[k-1]}, 0)$ is the partial Taylor coefficient with $y^{[k]} = 0$ and $A = \partial g^{[0]}(x^{[0]}, y^{[0]}) / \partial y^{[0]}$ is an $m \times m$ matrix, which depends only on $x^{[0]}$ and $y^{[0]}$. For an index 1 problem, A is invertible. Therefore, $y^{[k]}$ can be recursively obtained from the condition $g^{[k]} \equiv 0$ as

$$y^{[k]} = -A^{-1} g^{[k]}\big|_{y^{[k]}=0}. \tag{15}$$

Alternatively, by applying Eqs. 13 and 15, the Taylor coefficients of the solution to the DAE given in Eqs. 11 and 12 can be recursively obtained.

The efficiency of the AD method greatly relies on error control to adopt the integration step as large as possible. The maximum integration step, h to control the local error within the specified tolerance, ζ has been derived by Cao[38] and is briefly discussed next. Assume that $\tilde{x}(t) = [x^T(t) \quad y^T(t)]^T$ at the next integration time $t = t_0 + h$ be $\tilde{x}(t_0 + h) = \sum_{k=0}^{d} \tilde{x}^{[k]}(t_0) h^k + \varepsilon(h, d)$, where $\varepsilon(h,d)$ is the truncation error. Then,

$$\epsilon(h, d) \approx C(h/r)^{d+1} \tag{16}$$

where r is the radius of convergence, and C is a constant. For a sufficiently large d,

$$r \approx r_d := \frac{\| \tilde{x}^{[d-1]} \|_\infty}{\| \tilde{x}^{[d]} \|_\infty}. \tag{17}$$

As, $\varepsilon(h, d-1) \approx \varepsilon(h, d)(r_d/h) \approx \varepsilon(h, d) + \|\tilde{x}^{[d]}\|_\infty h^d$, it leads to the following estimation of the truncation error:

$$\epsilon(h, d) \approx \frac{h^{d+1} \| \tilde{x}^{[d]} \|_\infty^2}{\| \tilde{x}^{[d-1]} \|_\infty - h \| x^{[d]} \|_\infty} \tag{18}$$

Therefore, for specified d and error tolerance ζ, the integration step can be estimated to satisfy $\varepsilon(h,d) \leq \zeta$. For $h > 1$, it leads to[38]

$$h \leq \left(\frac{\zeta \| \tilde{x}^{[d-1]} \|_\infty}{\| \tilde{x}^{[d]} \|_\infty^2} \right)^{1/(d+1)}. \tag{19}$$

For improving numerical stability, a time-scaling factor, τ, can be introduced in Eq. 13. Let the scaled time be $t_s = t/\tau$. Then, Eq. 11 can be expressed as $dx/dt_s = (dx/dt)(dt/dt_s) = \tau dx/dt = \tau f(x,y)$ and similar to (13), we have

$$x^{[k+1]} = \frac{\tau}{k+1} f^{[k]}(x^{[0]}, \ldots, x^{[k]}, y^{[0]}, \ldots, y^{[k]}). \tag{20}$$

Usually τ can be chosen to be unity, but the numerical stability can be improved by properly selecting τ. This

happens as the numerical property of calculation involving extremely small and extremely large numbers is very poor due to the limited precision of floating point computation. In the Taylor approach, the truncation error mainly depends on the norm of the high-order term of the Taylor series, $\|x^{[d]}h^d\|$. When $h \gg 1$ or $h \ll 1$, for the same error tolerance, $\|x^{[d]}\|$ is extremely small or large, respectively. This causes the numerical property of the iterative Taylor coefficient calculation to be very poor. When the scaling factor τ is used, the actual integration step is $\tilde{h} = \tau h$, where h is determined in Eq. 19. Ideally, τ should be chosen as $\tau = \tilde{h}$ so that $h = 1$. However, h is determined using Eq. 19 after $x^{[k]}$ corresponding to a given τ is calculated. To avoid recalculation of $x^{[k]}$, we select τ to be the previous time step $\tilde{h}$. Assume in step j, the scaling factor is τ_j, and the scaled integration length determined in (19) is h_j. Then, the true integration length is $\tau_j h_j$, which can be used as the scaling factor for step $j + 1$, i.e. $\tau_{j+1} = \tau_j h_j$.

AD-QMOM

As shown in last section, QMOM requires solving the following DAE system:

$$\frac{d\mu_r}{dt} = f_r(\mu, w, L, \gamma) \tag{21}$$

$$g_r = \mu_r - \sum_{\ell=1}^{N} w_\ell L_\ell^r = 0; \quad r = 0, 1, \cdots, 2N-1 \tag{22}$$

where $\gamma = [\beta, a, G, B, b, L_0]$. Equation 21 depends on the system type, whereas Eq. 22 is the same for every system.

To solve the DAE system in Eqs. 21 and 22 using the AD method, the variables of interest, namely μ, w, and L, are approximated using dth order Taylor series,

$$\mu_r(t) = \mu_r^{[0]} + \mu_r^{[1]} t + \cdots + \mu_r^{[d]} t^d = \sum_{k=0}^{d} \mu_r^{[k]} t^k \tag{23}$$

$$w_\ell(t) = w_\ell^{[0]} + w_\ell^{[1]} t + \cdots + w_\ell^{[d]} t^d = \sum_{k=0}^{d} w_\ell^{[k]} t^k \tag{24}$$

$$L_\ell(t) = L_\ell^{[0]} + L_\ell^{[1]} t + \cdots + L_\ell^{[d]} t^d = \sum_{k=0}^{d} L_\ell^{[k]} t^k \tag{25}$$

where $\mu_r^{[k]}$, $w_\ell^{[k]}$, and $L_\ell^{[k]}$ are the kth Taylor coefficients of $\mu_r(t)$, $w_\ell(t)$, and $L_\ell(t)$, respectively. These coefficients can be recursively obtained through AD.

To solve the ODEs in Eq. 21, let

$$f_r(t) = \sum_{k=0}^{d} f_r^{[k]} t^k \tag{26}$$

where $f_r^{[k]}$ is the kth Taylor coefficients of $f_r(t)$. It follows from Eq. 21 that

$$\mu_r^{[k+1]} = \frac{\tau}{k+1} f_r^{[k]}; \quad k = 1, 2, \cdots, d-1 \tag{27}$$

where τ is a time scaling factor as explained earlier.

Clearly, $f_r^{[k]}$ depends on the Taylor coefficients of w, L, and μ, which is problem specific. These relationships for specific examples are derived in the next section. The focus of the rest of this section is on solving the algebraic equations in Eq. 22. Note that the zeroth-order Taylor coefficients of w_ℓ and L_ℓ are inherited from the previous step. According to the principle of AD for general algebraic equations shown in (14), the higher order Taylor coefficients of w_ℓ and L_ℓ are derived by solving a set of linear equations, as shown next.

For convenience of notation, we define $v_{r\ell} = w_\ell L_\ell^r$. It follows that $v_{(r+1)\ell} = v_{r\ell} L_\ell$. Then, the Taylor coefficients of $v_{r\ell}$ can be recursively derived as follows

$$v_{r\ell}^{[k]} = \begin{cases} w_\ell^{[k]}; & r = 0 \\ \sum_{j=0}^{k} v_{(r-1)\ell}^{[j]} L_\ell^{[k-j]}; & r \geq 1. \end{cases} \tag{28}$$

Using (14), we have

$$v_{r\ell}^{[k]} = v_{r\ell}^{[k]}\Big|_{w_\ell^{[k]}=L_\ell^{[k]}=0} + \frac{\partial v_{r\ell}^{[0]}}{\partial w_\ell^{[0]}} w_\ell^{[k]} + \frac{\partial v_{r\ell}^{[0]}}{\partial L_\ell^{[0]}} L_\ell^{[k]}. \tag{29}$$

Therefore, the Taylor coefficients of the rth algebraic equation g_r can be represented as:

$$g_r^{[k]} = \mu_r^{[k]} - \sum_{\ell=1}^{N} v_{r\ell}^{[k]} \tag{30}$$

$$= \mu_r^{[k]} - \sum_{\ell=1}^{N} \left(v_{r\ell}^{[k]}\Big|_{w_\ell^{[k]}=L_\ell^{[k]}=0} + \frac{\partial v_{r\ell}^{[0]}}{\partial w_\ell^{[0]}} w_\ell^{[k]} + \frac{\partial v_{r\ell}^{[0]}}{\partial L_\ell^{[0]}} L_\ell^{[k]} \right) \tag{31}$$

Based on Eq. 28, it follows that

$$v_{r\ell}^{[k]}\Big|_{w_\ell^{[k]}=L_\ell^{[k]}=0} = \begin{cases} 0; & r = 0 \\ \sum_{j=1}^{k} v_{(r-1)\ell}^{[j]}\Big|_{w_\ell^{[k]}=L_\ell^{[k]}=0} L_\ell^{[k-j]}; & r \geq 1 \end{cases} \tag{32}$$

Note that $v_{r\ell}^{[j]}$ is independent of $w_\ell^{[k]}$ and $L_\ell^{[k]}$, when $j < k$. Therefore, $v_{r\ell}^{[j]}\big|_{w_\ell^{[k]}=L_\ell^{[k]}=0} = v_{r\ell}^{[j]}$ for $j < k$.

As $v_{r\ell}^{[0]} = w_\ell^{[0]} (L_\ell^{[0]})^r$ and $g_r^{[0]} = \sum_{\ell=1}^{N} v_{r\ell}^{[0]}$, the sensitivities of $g_r^{[0]}$ against $w_\ell^{[0]}$ and $L_\ell^{[0]}$ are

$$\frac{\partial g_r^{[0]}}{\partial w_\ell^{[0]}} = \frac{\partial v_{r\ell}^{[0]}}{\partial w_\ell^{[0]}} = (L_\ell^{[0]})^r \tag{33}$$

$$\frac{\partial g_r^{[0]}}{\partial L_\ell^{[0]}} = \frac{\partial v_{r\ell}^{[0]}}{\partial L_\ell^{[0]}} = r w_\ell^{[0]} (L_\ell^{[0]})^{r-1}. \tag{34}$$

Now, the expression for $g_0^{[k]}$ can be simplified as

$$g_0^{[k]} = \mu_0^{[k]} - \sum_{\ell=1}^{N} w_\ell^{[k]}. \tag{35}$$

Similarly, for $r \geq 1$, we have

$$g_r^{[k]} = \underbrace{\mu_r^{[k]} - \sum_{\ell=1}^{N} v_{r\ell}^{[k]}\Big|_{w_\ell^{[k]}=L_\ell^{[k]}=0}}_{\tilde{\mu}_r^{[k]}} - \sum_{\ell=1}^{N}\left((L_\ell^{[0]})^r w_\ell^{[k]} + r w_\ell^{[0]}(L_\ell^{[0]})^{r-1} L_\ell^{[k]}\right) \quad (36)$$

Equation 36 is linear with respect to $w_\ell^{[k]}$ and $L_\ell^{[k]}$. Therefore, the kth Taylor coefficients of w_ℓ and L_ℓ can be obtained by solving the following $2N$ linear equations derived by letting $g_r^{[k]} = 0$, for $r = 0,\ldots,2N-1$,

$$Ap = q \quad (37)$$

where

$$A = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ L_1^{[0]} & \cdots & L_N^{[0]} & w_1^{[0]} & \cdots & w_N^{[0]} \\ (L_1^{[0]})^2 & \cdots & (L_N^{[0]})^2 & 2w_1^{[0]}L_1^{[0]} & \cdots & 2w_N^{[0]}L_N^{[0]} \\ (L_1^{[0]})^3 & \cdots & (L_N^{[0]})^3 & 3w_1^{[0]}(L_1^{[0]})^2 & \cdots & 3w_N^{[0]}(L_N^{[0]})^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (L_1^{[0]})^{2N-1} & \cdots & (L_N^{[0]})^{2N-1} & (2N-1)w_1^{[0]}(L_1^{[0]})^{2N-2} & \cdots & (2N-1)w_N^{[0]}(L_N^{[0]})^{2N-2} \end{bmatrix} \quad (38)$$

$$p = \begin{bmatrix} w_1^{[k]} & \cdots & w_N^{[k]} & L_1^{[k]} & \cdots & L_N^{[k]} \end{bmatrix}^T \quad (39)$$

$$q = \begin{bmatrix} \mu_0^{[k]} & \tilde{\mu}_1^{[k]} & \tilde{\mu}_2^{[k]} & \cdots & \cdots & \tilde{\mu}_{2N-1}^{[k]} \end{bmatrix}^T \quad (40)$$

Note that the matrix A depends on the zeroth-order Taylor coefficients only and, thus, needs to be inverted only once for all k. The same result has been derived through a different approach by Kariwala et al.[44] The procedure for AD-QMOM is given in Algorithm 1.

Algorithm 1 Initially, compute $\mu_r(0)$, $r = 0,1,\ldots 2N-1$, based on the given initial distribution. Solve the nonlinear equations in Eq. 22 to obtain $w_\ell(0)$ and $L_\ell(0)$, $\ell = 1,2,\ldots,N$. Choose d and initial τ. While $t < t_f$:

(a) Set $\mu_r^{[0]} = \mu_r(t)$, $w_\ell^{[0]} = w_\ell(t)$, and $L_\ell^{[0]} = L_\ell(t)$.
(b) Compute A in (38) and its inverse. Set $v_{r\ell}^{[0]} = w_\ell^{[0]}(L_\ell^{[0]})^r$.
(c) For $k = 0$ to d
  (i) Compute the partial Taylor coefficient, $v_{r\ell}^{[k]}\big|_{w_\ell^{[k]}=L_\ell^{[k]}=0}$. Compute $\tilde{\mu}_r^{[k]}$ and solve linear equations in (37) to get $w_\ell^{[k]}$ and $L_\ell^{[k]}$. Update $v_{r\ell}^{[k]}$ using (29).
  (ii) Compute $f_r^{[k]}$ and set $\mu_r^{[k+1]} = \frac{\tau}{k+1} f_r^{[k]}$.
(d) Select h to satisfy (19) with $\tilde{x}(t) = [\mu^T(t) \quad w^T(t) \quad L^T(t)]^T$.
(e) Update μ_r, w_ℓ, and L_ℓ using (23)–(25).
(f) Set $t = t + \tau h$ and update $\tau = \tau h$. Go to step (a).

Validation of AD-QMOM

In this section, we validate AD-QMOM by comparing its accuracy with DAE-QMOM using benchmark examples involving different mechanisms, namely growth, nucleation, aggregation, and breakage. All of these examples were also considered by Gimbun et al.[28] to show the application of DAE-QMOM. For all the examples, the final simulation time is $t_f = 100$ s. DAE-QMOM is applied using the Matlab routine *ode15s* with relative and absolute tolerances set to 10^{-12} and 10^{-10}, respectively. For AD-QMOM, we set $d = 20$ and initially $\tau = 0.15$, while the tolerance level ζ is determined such that the percentage errors in μ is comparable with DAE-QMOM. Here, the percentage error is defined as

$$\%\,\text{error} = \frac{\mu_{\text{analytical}} - \mu_{\text{calculated}}}{\mu_{\text{analytical}}} \times 100. \quad (41)$$

For ease of comparison of the accuracy, the percentage error values are calculated for both the methods at the same time points given as $t = 0,1,2,\cdots,100$ s.

Diffusion-controlled growth

First, a crystallization process is considered only with growth mechanism, which is entirely diffusion-controlled. The general equation for the diffusion-controlled linear growth rate under constant supersaturation conditions can be written as[45]

$$G = \frac{G_0}{L} \quad (42)$$

In this article, we consider that $G_0 = 0.01$ m^2/s. For this process, based on Eq. 3, the moment equations are:

$$\frac{d\mu_r}{dt} = r G_0 \mu_{r-2}; \quad r = 1,2,\cdots,2N-1 \quad (43)$$

These moment equations are not closed for any N, as μ_1 depends on $\mu_{-1} = \int_0^\infty (1/L) n(L) dL$.[5] Using the quadrature approximation, the moment equations can be derived to be:

$$\frac{d\mu_r}{dt} = f_r(w_\ell, L_\ell, G_0) = r G_0 \sum_{\ell=1}^{N} w_\ell L_\ell^{r-2} \quad (44)$$

The initial distribution is given as

$$n(0,L) = 3L^2 \frac{\mu_0(0)}{v_0} e^{-L^3/v_0} \quad (45)$$

with $\mu_0(0) = 1$ m^{-3} and $v_0 = 1$ m^3. The moments at $t = 0$ are obtained by analytically calculating $\int_0^\infty n(0,L) L^r dL$ for

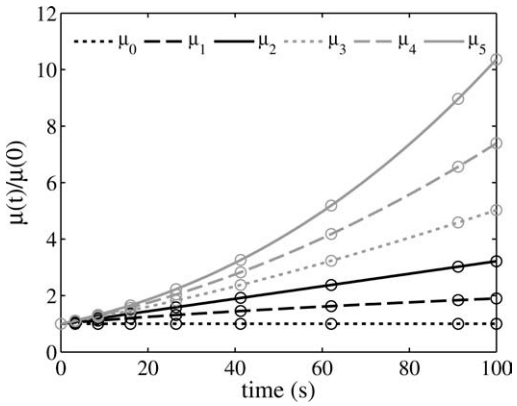

Figure 1. Evolution of Moments for diffusion-controlled growth; analytical solution (lines), AD-QMOM (circles); For odd moments, analytical solution is not available and numerical integration of Eq. 49 is instead used.

$r = 0,1,\cdots,2N-1$, where $N = 3$. The weights w and sizes L at $t = 0$ are obtained by solving the nonlinear algebraic equations numerically.

The analytical solution is available only for the even moments and is given as[5]

$$\mu_0(t) = \mu_0(0) \tag{46}$$

$$\mu_2(t) = 2G_0\mu_0(t)t + \mu_2(0) \tag{47}$$

$$\mu_4(t) = 4G_0^2\mu_0(t)t^2 + 4G_0\mu_2(0)t + \mu_4(0). \tag{48}$$

The analytical size distribution at time t is given as[31]

$$n(L,t) = \frac{3\mu_0(0)}{v_0} L\sqrt{L^2 - 2G_0t}\, e^{-(L^2-2G_0t)^{1.5}/v_0} \tag{49}$$

which is numerically integrated using Matlab routine *quadl* to obtain the the odd moments. Note that the analytical integration of Eq. 49 is difficult due to the presence of discontinuity at $L = 2G_0t$.[28]

To apply AD-QMOM to solve this problem, we need to find the Taylor coefficients of f_r. Based on Eq. 44 and Table 1, the Taylor coefficients of f_r are derived as follows:

$$f_r^{[k]} = rG_0 \sum_{\ell=1}^{N} \sum_{j=0}^{k} w_\ell^{[k-j]} \left(L_\ell^{r-2}\right)^{[j]} \tag{50}$$

where $(L_\ell^{r-2})^{[j]}$ can be found using Identity 4 in Table 1.

AD-QMOM is applied with $\zeta = 10^{-10}$, and the evolution of the moments is shown in Figure 1. As expected, the number of particles (given by μ_0) remains constant, whereas the other moments increase with time, as the particles grow. The % errors for the even moments (for which analytical solution is available) are shown in Figure 2. Both the methods provide similar accuracies with the errors for μ_2 being slightly lower for AD-QMOM. Although the % error in μ_4 obtained using AD-QMOM increases with time, the absolute % error for this moment does not exceed 4×10^{-9} for long-time simulation. Overall, this example demonstrates the potential of AD method to provide accurate solutions for QMOM.

Constant growth with nucleation

Next, we consider an example with constant growth and nucleation rates, where $G = 0.01$ m/s and $B_0(t) = 0.1$ m^{-3}/s. The initial distribution is given by (45). It is assumed that the size of nuclei L_0 is zero. Under this assumption, the moment equations are

$$\frac{d\mu_r}{dt} = \begin{cases} B_0; & r = 0 \\ rG\mu_{r-1}; & r \geq 1 \end{cases}, \tag{51}$$

which are closed, and standard MOM can be used.[28] Nevertheless, this example is useful to demonstrate the ability of AD-QMOM to handle nucleation. With quadrature approximation, the moment equations are

$$\frac{d\mu_r}{dt} = f_r(w_\ell, L_\ell, G, B_0) = \begin{cases} B_0; & r = 0 \\ rG\sum_{\ell=1}^{N} w_\ell L_\ell^{r-1}; & r \geq 1 \end{cases} \tag{52}$$

On the basis of Eq. 51, we note that

$$\mu_r(t) = rG\int_0^t \mu_{r-1}(t)dt + \mu_r(0); \quad r \geq 1 \tag{53}$$

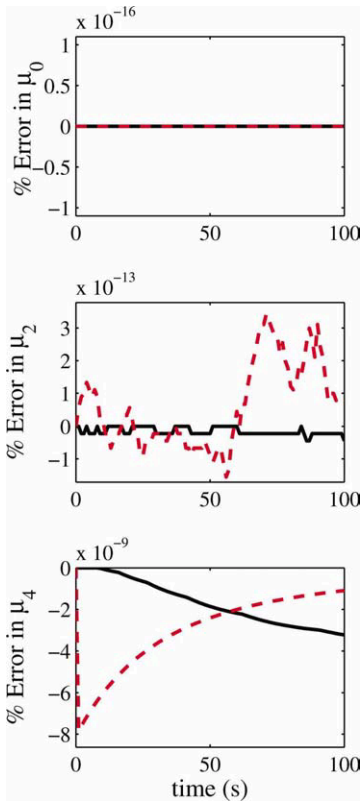

Figure 2. Comparison of % errors in moments for diffusion-controlled growth using AD-QMOM (continuous line) and DAE-QMOM (dashed line).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

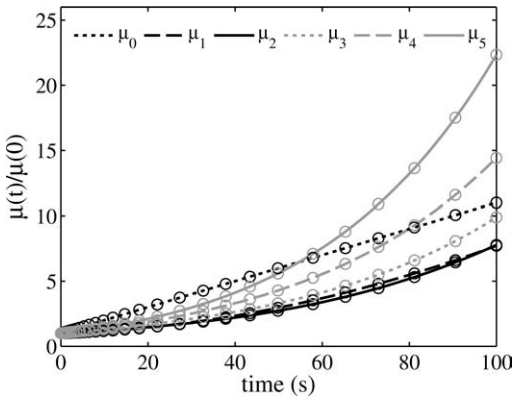

Figure 3. Evolution of Moments for Constant Growth with Nucleation; analytical solution (lines), AD-QMOM (circles).

which can be used to compute analytical solutions for any number of moments. For example, the analytical solution for the first three moments is given as:[28]

$$\mu_0(t) = B_0 t + \mu_0(0) \tag{54}$$

$$\mu_1(t) = B_0 G \frac{t^2}{2} + G\mu_0(0)t + \mu_1(0) \tag{55}$$

$$\mu_2(t) = B_0 G^2 \frac{t^3}{3} + 2G^2\mu_0(0)\frac{t^2}{2} + 2G\mu_1(0)t + \mu_2(0). \tag{56}$$

The application of AD method for $r \geq 1$ is the same as the previous example, but some care is needed for $r = 0$. As B_0 does not change with time, we need to use

$$f_0^{[k]} = \begin{cases} B_0; & k = 0 \\ 0; & k \geq 1 \end{cases}. \tag{57}$$

The AD method is applied with $\zeta = 10^{-15}$. The evolution of the first six moments and their errors are shown in Figures 3 and 4, respectively. Similar to the previous example, it can be noted that the accuracies of both the methods are similar with the errors for μ_4 and μ_5 being marginally lower for AD-QMOM.

Aggregation with constant kernel

To illustrate the applicability of AD-QMOM method to processes with aggregation, we consider an example with $\beta(L_\ell,L_m) = 1$, while the growth, nucleation, and breakage phenomena are not present. In this case, the moment equations are given as

$$\frac{d\mu_r}{dt} = f_r(w_\ell, L_\ell) = \frac{1}{2}\sum_{\ell=1}^{N} w_\ell \sum_{m=1}^{N} w_m (L_\ell^3 + L_m^3)^{r/3} - \sum_{\ell=1}^{N} w_\ell L_\ell^r \sum_{m=1}^{N} w_m. \tag{58}$$

The initial distribution is given by (45) with $\mu_0(0) = 1\ \mathrm{m}^{-3}$ and $v_0 = 0.001\ \mathrm{m}^3$. The analytical solution is given as[46]

$$\mu_r(t) = \mu_r(0)\left(\frac{2}{2 + \mu_0(0)\beta(L_\ell, L_m)t}\right)^{1-r/3}. \tag{59}$$

Before implementation of the AD method, the moment equations are simplified to avoid redundant calculations. Let $\mathcal{L}$ denote the ensemble of all possible pairs (with repetition) among N indices, 1,…,N, where the cardinality of $\mathcal{L}$ is $|\mathcal{L}| = N(N+1)/2$. For example, for $N = 3$, $\mathcal{L} = \{(1,1), (1,2), (1,3), (2,2), (2,3), (3,3)\}$. With this notation,

$$f_r(w_\ell, L_\ell) = \sum_{(\ell,m)\in\mathcal{L}} w_\ell w_m (L_\ell^3 + L_m^3)^{r/3}(1 - 0.5\tilde{\delta}(\ell - m)) - \sum_{m=1}^{N} w_m \sum_{\ell=1}^{N} w_\ell L_\ell^r \tag{60}$$

where $\tilde{\delta}$ represents the Kronecker delta function, which takes a value of one, if its argument is zero, and is zero otherwise. To derive an explicit expression for $f_r^{[k]}$, the following intermediate variables are introduced:

$$u_i = w_\ell w_m (1 - 0.5\tilde{\delta}(\ell - m)) \tag{61}$$

$$v_i = L_\ell^3 + L_m^3 \tag{62}$$

$$X_{ir} = v_i^{r/3} \tag{63}$$

$$Y_{\ell r} = w_\ell L_\ell^r \tag{64}$$

$$z_r = \sum_{\ell=1}^{N} Y_{\ell r} \tag{65}$$

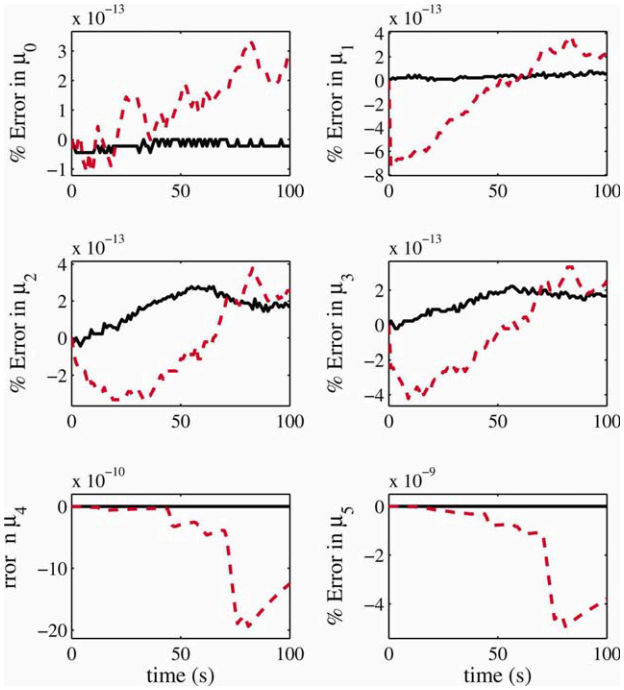

Figure 4. Comparison of errors for Constant Growth with Nucleation using AD-QMOM (continuous line) and DAE-QMOM (dashed line).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

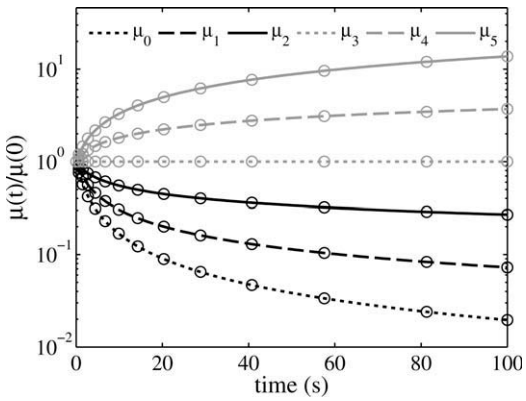

Figure 5. Evolution of Moments for Aggregation with constant kernel; analytical solution (lines), AD-QMOM (circles).

where $\ell = 1,2,\cdots,N$ and $i = 1,2,\cdots,N(N+1)/2$. Thus, we have

$$f_r(w_\ell, L_\ell) = \sum_{i=1}^{N(N+1)/2} u_i X_{ir} - z_0 z_r \tag{66}$$

and

$$f_r^{[k]}(w_\ell, L_\ell) = \sum_{i=1}^{N(N+1)/2} \sum_{j=0}^{k} u_i^{[k-j]} X_{ir}^{[j]} - \sum_{j=0}^{k} z_0^{[k-j]} z_r^{[j]} \tag{67}$$

where the Taylor coefficients of intermediate variables are found using the identities in Table 1. Although the procedure for finding $f_r^{[k]}(w_\ell,L_\ell)$ may seem complicated, it only involves breaking down the overall calculation into basic calculation, such as summation, binary product, and power, through introduction of intermediate variables.

AD-QMOM is applied with $\zeta = 10^{-10}$, and the evolution of the first six moments and their % errors are shown in Figures 7 and 8, respectively. It can be noted that both the methods conserve μ_3 closely showing that the volume of the particles remain unchanged. The accuracy of the two methods for other moments is nearly identical, as is clear from the overlapping curves in Figure 6. It can be noted that as compared with the earlier examples, the errors in the moments is much larger. When AD-QMOM is used to solve the model equations for larger N, these errors reduce significantly. For example, the maximum absolute % errors in μ_4 and μ_5 are 2.30×10^{-7} and 3.26×10^{-7}, respectively, for $N = 9$. This implies that the large errors are mainly due to the quadrature approximation, which is beyond the control of the AD algorithm.

Volume breakage kernel

Finally, we consider a breakage example, where the breakage kernel and probability of breakage are given as[47]

$$a(L) = L^3; \quad b(L,\lambda) = \frac{6L^2}{\lambda^3}, \tag{68}$$

which imply breakage of larger particles into smaller particles with equal volume. The moment equations can be derived to be

$$\frac{d\mu_r}{dt} = \sum_{\ell=1}^{N} w_\ell a(L_\ell) b(r, L_\ell) - \sum_{\ell=1}^{N} w_\ell a(L_\ell) L_\ell^r. \tag{69}$$

For $b(L, \lambda)$ in Eq. 68, $b(r, L_\ell) = \frac{6L_\ell^r}{r+3}$. Thus, the moment equations can be simplified as

$$\frac{d\mu_r}{dt} = f_r(w_\ell, L_\ell) = \frac{3-r}{3+r} \sum_{\ell=1}^{N} w_\ell L_\ell^{r+3} \tag{70}$$

The initial distribution is given by Eq. 45. The analytical solution for the size distribution at any time t is given as[47]

$$n(L,t) = 3L^2(1+t)^2 \exp(-L^3(1+t)), \tag{71}$$

which can be integrated to find the analytical solution for moments.

For application of AD-QMOM, the Taylor coefficients of f_r can be found as follows:

$$f_r^{[k]} = \frac{3-r}{3+r} \sum_{\ell=1}^{N} \sum_{j=0}^{k} w_\ell^{[k-j]} \left(L_\ell^{r+3}\right)^{[j]} \tag{72}$$

where $(L_\ell^{r+3})^{[j]}$ can be found using Identity 4 in Table 1.

AD-QMOM is applied with $\zeta = 10^{-16}$, and the evolution of the first six moments and their % errors are shown in Figures 7 and 8, respectively. Similar to the aggregation example, both the methods conserve μ_3 precisely and provide the same level of accuracy for other moments. The large errors seen in the moments are due to the quadrature

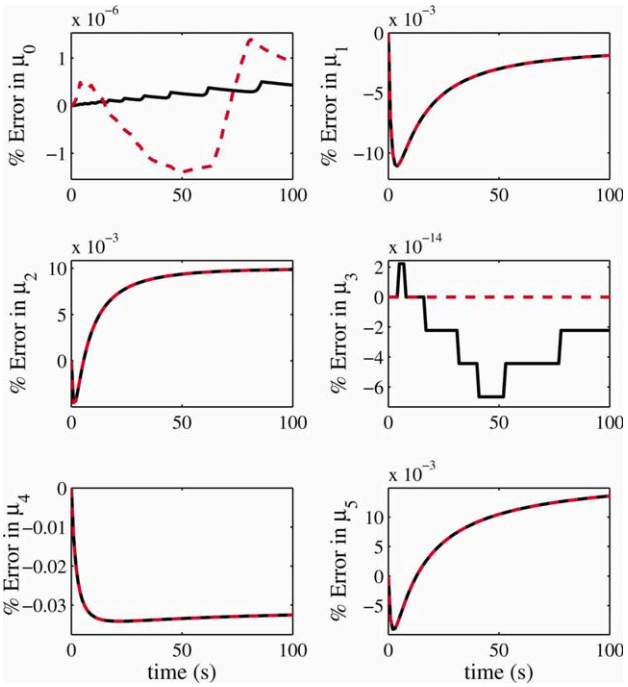

Figure 6. Comparison of errors for Aggregation with constant kernel using AD-QMOM (continuous line) and DAE-QMOM (dashed line).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

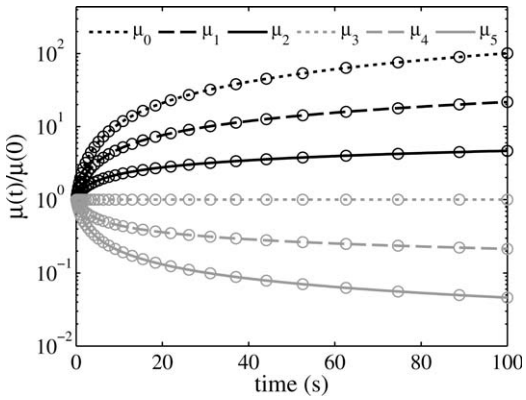

Figure 7. Evolution of moments for Volume Breakage Kernel; analytical solution (lines), AD-QMOM (circles).

approximation, which reduces significantly when AD-QMOM is applied with larger N; see next section for details.

Computational Efficiency and Robustness

The results of the last section demonstrate that the proposed AD-QMOM can provide the same level of accuracy as the DAE-QMOM. In this section, the performances of these algorithms are compared further in terms of computational efficiency and robustness. For this purpose, the different examples are solved with both the methods with the same parameter values, that is, relative and absolute tolerances for DAE-QMOM and, d and ζ for AD-QMOM, respectively, as used in the last section. To obtain a reliable estimate of the solution time, each problem is solved 100 times, and the median value of the CPU time is reported. All computations are carried out on a Windows Vista PC with an Intel®Core™ 2 Duo Processor E6750 (2.66 GHz, 4-MB RAM) using MATLAB® R2010a.

The CPU times needed by both methods for different examples are shown in Table 2. When used to solve for six moments ($N = 3$), AD-QMOM requires lower computation time than DAE-QMOM for all the examples. The computational efficiency of AD-QMOM is most prominent for diffusion-controlled growth, for which AD-QMOM requires an order of magnitude lower computation time than DAE-QMOM. This computational advantage is a result of the large time steps (h) taken by AD-QMOM; hence requiring only seven iterations to solve the model until $t_f = 100$ s. Similarly, for the case of volume breakage kernel, AD-QMOM is about six times faster than DAE-QMOM. For aggregation with constant kernel, AD-QMOM requires much larger number of floating point operations but is still about four times faster than DAE-QMOM. For nucleation with constant growth, AD-QMOM is seen to be only twice as fast as DAE-QMOM. In this case, the Taylor coefficients of w and L do not decay to zero fast enough and thus to maintain the required level of accuracy, the AD-QMOM algorithm needs to take smaller time steps toward the beginning of the simulation. As mentioned earlier, the moment equations for this example are closed and standard MOM formulation can be used. When used to solve the ODE model in Eq. 51, AD-QMOM is able to provide very high levels of accuracy (order 10^{-14}) in only 0.0312 s. This confirms our earlier argument that the lower computational advantage of AD-QMOM for the case of nucleation with constant growth is due to the need to maintain low error in w and L.

Both the methods are also used to solve the DAE model in Eqs. 21 and 22 for larger number of moments, where $N > 3$. Here, N is increased sequentially until the method fails to solve the model until the final simulation time of $t_f = 100$ s. DAE-QMOM is only able to handle cases until $N = 3$ for diffusion-controlled growth and breakage examples, and $N = 5$ for nucleation with constant growth and aggregation examples. For higher values of N, Matlab routine *ode15s* fails to solve the DAEs until $t_f = 100$ s with desired accuracy. In comparison, AD-QMOM is able to solve for much larger number of moments showing robustness. For diffusion-controlled growth until $N = 7$ and, aggregation and breakage examples until $N = 9$, the CPU time increases almost linearly with N. For larger values of N for these examples, the A matrix in Eq. 38 becomes nearly singular, which causes the integration step h to be very small, and AD-QMOM is not able to provide the solution in reasonable time. The increase in CPU time with N for the case of nucleation with constant growth is faster than other examples but is still quadratic. Based on these observations, it can be concluded that the AD-QMOM scales well with problem dimensions.

It is worth pointing out that the AD-QMOM maintains high levels of accuracy, even when used to solve for a large number of moments. To illustrate this, the average % error of the moments obtained for different values of N for the case of volume breakage kernel are shown in Table 3. For this example, for $N = 3$, both the methods show comparable but large errors in the moments due to the quadrature

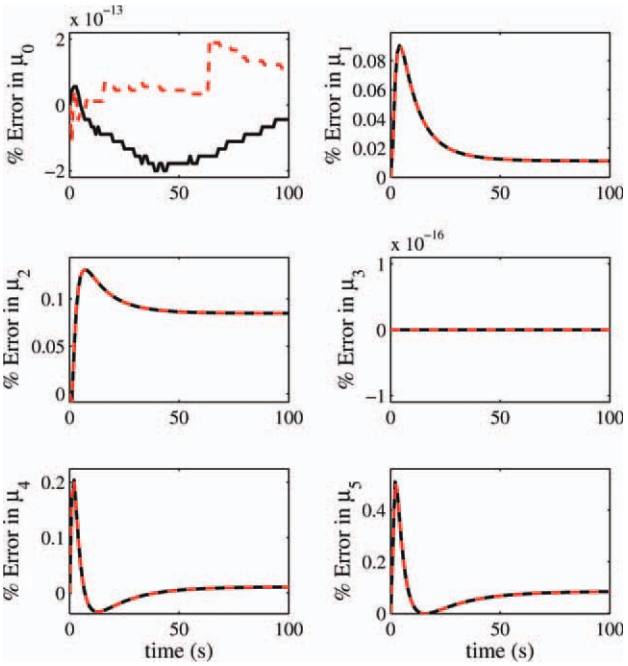

Figure 8. Comparison of errors for Volume Breakage Kernel using AD-QMOM (continuous line) and DAE-QMOM (dashed line).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 2. Comparison of CPU Times (s) for DAE-QMOM and AD-QMOM

Method	N	Diffusion-Controlled Growth	Constant Growth with Nucleation	Aggregation with Constant Kernel	Volume Breakage Kernel
DAE-QMOM	3	0.3588	0.2808	0.2964	0.7488
	4	–	0.3432	0.3588	–
	5	–	0.4992	0.4524	–
AD-QMOM	3	0.0312	0.1404	0.0780	0.1248
	4	0.0468	0.2184	0.1092	0.1560
	5	0.0624	0.2808	0.1248	0.2184
	6	0.0780	0.3744	0.1404	0.2808
	7	0.0936	0.4680	0.1716	0.3588
	8	–	0.6084	0.1872	0.4680
	9	–	0.7176	0.2964	0.5928
	10	–	0.8892	–	–

approximation; see also Figure 8. DAE-QMOM fails for N larger than 3, whereas AD-QMOM is able to solve the model equations successfully until $N = 9$. The error in moments decreases monotonically, as N is increased. The average errors in lower order moments ($\mu_0 - \mu_5$) are about two to three orders of magnitude lower for $N = 9$ when compared with $N = 3$. For $N = 9$, the average error in higher order moments obtained using AD-QMOM is reasonable (order 10^{-2}). Interestingly, the time taken by AD-QMOM to solve for 18 moments ($N = 9$) is lower than the time taken by DAE-QMOM to solve for six moments ($N = 3$); see Table 2. Similar trends are also seen for aggregation with constant kernel. Overall, in comparison with DAE-QMOM, AD-QMOM provides higher level of accuracy for a given computational time or in other words and requires lower computational time for providing the same level of accuracy.

The use of AD-QMOM to solve for a given number of moments of the PBE requires the selection of two parameters, that is, tolerance level (ζ) and the number of Taylor series coefficients (d). Among these parameters, the choice of ζ is more critical to obtain a trade-off between the accuracy and solution time. Although reducing ζ monotonically improves the accuracy at the expense of larger solution time, it is worth pointing out that ζ can only affect the numerical error in μ but not the error arising due to quadrature approximation. Thus, in general, the observed error in μ can differ significantly from the chosen ζ. On the other hand, the accuracy and solution time are relatively insensitive to the choice of d. To demonstrate this, the case of volume breakage kernel is considered with $N = 9$ and $\zeta = 10^{-16}$. This example is solved for d ranging from 10 to 50. For every value of d, the errors in all the moments are nearly identical to the values shown in Table 3, and the CPU time required by the algorithm is shown in Figure 9. It can be noted that the CPU time initially decreases, as the use of a larger d allows larger time steps (h). For larger values of d, the increase in Δt is balanced by the increased computational load for additional Taylor coefficients. Thus, the CPU time for $18 \leq d \leq 50$ only differs by approximately $\pm 15\%$ in comparison with the case of $d = 20$ (used in this article). This observation highlights that the accuracy and solution time of AD-QMOM are not very sensitive to the choice of d, as long as d is chosen to be sufficiently large.

Conclusions

Solving PBEs accurately and efficiently represents a computational challenge. In the article, an AD-based approach is

Table 3. Average % Error for Volume Breakage Kernel

	DAE-QMOM	AD-QMOM			
	$N = 3$	$N = 3$	$N = 5$	$N = 7$	$N = 9$
μ_0	7.99×10^{-14}	1.16×10^{-13}	2.49×10^{-13}	5.73×10^{-14}	8.77×10^{-13}
μ_1	0.0214	0.0214	3.03×10^{-3}	2.65×10^{-4}	6.83×10^{-5}
μ_2	0.0903	0.0903	2.67×10^{-3}	3.26×10^{-4}	8.74×10^{-5}
μ_3	0	0	0	0	0
μ_4	0.0172	0.0172	3.10×10^{-3}	1.04×10^{-3}	2.24×10^{-4}
μ_5	0.0764	0.0764	8.17×10^{-3}	2.65×10^{-3}	6.71×10^{-4}
μ_6	*0.0835*	*0.0835*	0.0148	4.81×10^{-3}	1.45×10^{-3}
μ_7	*0.1063*	*0.1063*	0.0231	7.50×10^{-3}	2.51×10^{-3}
μ_8	*0.1577*	*0.1577*	0.0334	0.0107	3.81×10^{-3}
μ_9	*0.7219*	*0.7219*	0.0454	0.0144	5.32×10^{-3}
μ_{10}	*2.1949*	*2.1949*	*0.0592*	0.0187	7.01×10^{-3}
μ_{11}	*4.8330*	*4.8330*	*0.0753*	0.0233	8.88×10^{-3}
μ_{12}	*8.7990*	*8.7990*	*0.0930*	0.0290	0.0110
μ_{13}	*14.0421*	*14.0421*	*0.1184*	0.0354	0.0133
μ_{14}	*20.3901*	*20.3901*	*0.1689*	*0.0429*	0.0160
μ_{15}	*27.5707*	*27.5707*	*0.2765*	*0.0505*	0.0191
μ_{16}	*35.2612*	*35.2612*	*0.5002*	*0.0595*	0.0225
μ_{17}	*43.1326*	*43.1326*	*0.9766*	*0.0693*	0.0262

Numbers in italics denote the average % error for the moments extrapolated using w_ℓ and L_ℓ based on Eq. 22.

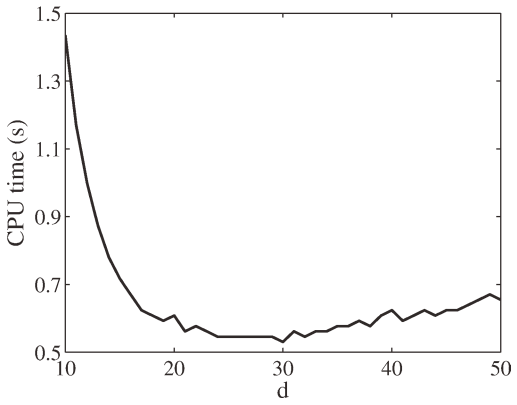

Figure 9. Variation of CPU time with *d* for volume breakage kernel with $N = 9$.

proposed to solve PBE using the QMOM. The AD-QMOM uses high-order Taylor expansions to obtain accurate solution of the DAEs arising from the quadrature approximation. Examples involving different mechanisms show that for the same accuracy level, AD-QMOM is 2–10 times faster than conventional DAE solvers. The AD-QMOM is also more robust and is able to handle a much higher number of moments than conventional DAE solvers. In summary, this article establishes AD-QMOM as a promising approach for solving PBEs. The superior efficiency and robustness of AD-QMOM make the approach useful for parameter estimation, process optimization, online model-based control, and coupled CFD-PBE simulations.

Acknowledgments

This work was supported by the Royal Society, UK through grant no. IV0871568.

Literature Cited

1. Ramkrishna D. *Population Balances: Theory and Applications to Particulate Systems in Engineering*. San Diego, CA: Acadamic Press, 2000.
2. Moilanen P, Laakkonen M, Visuri O, Alopaeus V, Aittamaa J. Modelling mass transfer in an aerated 0.2 m^3 vessel agitated by Rushton, Phasejet and Combijet impellers. *Chem Eng J*. 2008;142:95–108.
3. Gimbun J, Rielly C, Nagy Z. Modelling of mass transfer in gas–liquid stirred tanks agitated by Rushton turbine and CD-6 impeller: a scale-up study. *Chem Eng Res Des*. 2009;87:437–451.
4. Hulburt HM, Katz S. Some problems in particle technology: a statistical mechanical formulation. *Chem Eng Sci*. 1964;19:555–574.
5. McGraw R. Description of aerosol dynamics by the quadrature method of moments. *Aerosol Sci Technol*. 1997;27:255–265.
6. Kumar S, Ramkrishna D. On the solution of population balance equations by discretization–III. Nucleation, growth and aggregation of particles. *Chem Eng Sci*. 1997;52:4659–4679.
7. Pilon L, Viskanta R. Modified method of characteristics for solving population balance equations. *Int Numerical Methods Fluids*. 2003;42:1211–1236.
8. Aamir E, Nagy ZK, Rielly CD, Kleinert T, Judat B. Combined quadrature method of moments and method of characteristics approach for efficient solution of population balance models for dynamic modeling and crystal size distribution control of crystallization processes. *Ind Eng Chem Res*. 2009;48:8575–8584.
9. Singh P, Ramkrishna D. Solution of population balance equations by MWR. *Comput Chem Eng*. 1977;1:23–31.
10. Nicmanis M, Hounslow M. Finite-element methods for steady-state population balance equations. *AIChE J*. 1998;44:2258–2272.
11. Kumar S, Ramkrishna D. On the solution of population balance equations by discretization. I. A fixed pivot technique. *Chem Eng Sci*. 1996;51:1311–1332.
12. Gunawan R, Fusman I, Braatz RD. High resolution algorithms for multidimensional population balance equations. *AIChE J*. 2004; 50:2738–2749.
13. Qamar S, Elsner MP, Angelov IA, Warnecke G, Seidel-Morgenstern A. A comparative study of high resolution schemes for solving population balances in crystallization. *Comput Chem Eng*. 2006;30: 1119–1131.
14. Majumder A, Kariwala V, Ansumali S, Rajendran A. Fast high-resolution method for solving multidimensional population balances in crystallization. *Ind Eng Chem Res*. 2010;49:3682–3782.
15. Lim Y, Le Lann J, Meyer X, Joulia X, Lee G, Yoon E. On the solution of population balance equations (PBE) with accurate front tracking methods in practical crystallization processes. *Chem Eng Sci*. 2002;57:3715–3732.
16. Hermanto M, Braatz R, Chiu M. High-order simulation of polymorphic crystallization using weighted essentially nonoscillatory methods. *AIChE J*. 2008;55:122–131.
17. Bouaswaig A, Engell S. WENO scheme with static grid adaptation for tracking steep moving fronts. *Chem Eng Sci*. 2009;64:3214–3226.
18. Majumder A, Kariwala V, Ansumali S, Rajendran A. Entropic lattice Boltzmann method for crystallization processes. *Chem Eng Sci*. 2010;65:3928–3936.
19. Majumder A, Kariwala V, Ansumali S, Rajendran A. Lattice Boltzmann method for multi-dimensional population balance equations. In: *Proceedings of the Population Balance Modelling*. Berlin, Germany, 2010.
20. Rosner D, McGraw R, Tandon P. Multivariate population balances via moment and Monte Carlo simulation methods. *Ind Eng Chem Res*. 2003;42:2699–2711.
21. Zhao H, Maisels A, Matsoukas T, Zheng C. Analysis of four Monte Carlo methods for the solution of population balances in dispersed systems. *Powder Technol*. 2007;173:38–50.
22. Woo XY, Tan RBH, Chow PS, Braatz RD. Simulation of mixing effects in antisolvent crystallization using a coupled CFD-PDF-PBE approach. *Cryst Growth Des*. 2006;6:1291–1303.
23. Kulikov V, Briesen H, Marquardt W. A framework for the simulation of mass crystallization considering the effect of fluid dynamics. *Chem Eng Process*. 2006;45:886–899.
24. Gordon R. Error bounds in equilibrium statistical mechanics. *J Math Phys*. 1968;9:655–663.
25. Wright D, McGraw R, Rosner D. Bivariate extension of the quadrature method of moments for modeling simultaneous coagulation and sintering of particle populations. *J Colloid Interface Sci*. 2001;236:242–251.
26. Rosner D, Pyykonen J. Bivariate moment simulation of coagulating and sintering nanoparticles in flames. *AIChE J*. 2002;48:476–491.
27. Marchisio D, Vigil R, Fox R. Quadrature method of moments for aggregation–breakage processes. *J Colloid Interface Sci*. 2003;258: 322–334.
28. Gimbun J, Nagy Z, Rielly C. Simultaneous quadrature method of moments for the solution of population balance equations, using a differential algebraic equation framework. *Ind Eng Chem Res*. 2009;48:7798–7812.
29. McGraw R, Wright D. Chemically resolved aerosol dynamics for internal mixtures by the quadrature method of moments. *J Aerosol Sci*. 2003;34:189–209.
30. Fan R, Marchisio D, Fox R. Application of the direct quadrature method of moments to polydisperse gas–solid fluidized beds. *Powder Technol*. 2004;139:7–20.
31. Alopaeus V, Laakkonen M, Aittamaa J. Numerical solution of moment-transformed population balance equation with fixed quadrature points. *Chem Eng Sci*. 2006;61:4919–4929.
32. Dorao C, Jakobsen H. The quadrature method of moments and its relationship with the method of weighted residuals. *Chem Eng Sci*. 2006;61:7795–7804.
33. Grosch R, Briesen H, Marquardt W, Wulkow M. Generalization and numerical investigation of QMOM. *AIChE J*. 2006;53:207–227.
34. Tolsma J, Barton P. On computational differentiation. *Comput Chem Eng*. 1998;22:475–490.
35. Griewank A. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2000.

36. Pryce J. Solving high-index DAEs by Taylor series. *Numer Algorithm*. 1998;19:195–211.
37. Barrio R. Performance of the Taylor series method for ODEs/DAEs. *Appl Math Comput*. 2005;163:525–545.
38. Cao Y. A formulation of nonlinear model predictive control using automatic differentiation. *J Process Control*. 2005;15:851–858.
39. Al-Seyab R, Cao Y. Differential recurrent neural network based predictive control. *Comput Chem Eng*. 2008;32:1533–1545.
40. Al-Seyab R, Cao Y. Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation. *J Process Control*. 2008;18:568–581.
41. Diemer R, Ehrman S. Pipeline agglomerator design as a model test case. *Powder Technol*. 2005;156:129–145.
42. Marchisio DL, Pikturna JT, Fox RO, Vigil RD, Barresi AA. Quadrature method of moments for population-balance equations. *AIChE J*. 2003;49:1266–1276.
43. Christianson B. Reverse accumulation and accurate rounding error estimates for Taylor series. *Opt Methods Software*. 1992;1:81–94.
44. Kariwala V, Cao Y, Nagy Z. Automatic Differentiation based QMOM for population balance equations. In: Mayuresh Kothare, Moses Tade, Alain Vande Wouwer, Ilse Smets, Editors, *Proceeding of the 9th International Symposium on Dynamics and Control of Process Systems*. Leuven, Belgium, 2010:383–388.
45. Mullin JW. *Crystallization*, 4th ed. Oxford, UK: Butterworth-Heinemann, 2001.
46. Gelbard F, Seinfeld J. Numerical solution of the dynamic equation for particulate systems. *J Comput Phys*. 1978;28:357–375.
47. Hounslow M, Pearson J, Instone T. Tracer studies of high-shear granulation. II. Population balance modeling. *AIChE J*. 2001;47:1984–1999.

Manuscript received Sept. 14, 2010, and revision received Feb. 16, 2011.