# Application of the QMOM in Research on the Behavior of Solid-liquid Suspensions

**M. Lemanowicz,[a,*] M. H. Al-Rashed,[b] A. T. Gierczycki,[a] and J. Kocurek[a]**

[a]Department of Chemical and Process Engineering,
Faculty of Chemistry Silesian University of Technology,
ul. Ks. M. Strzody 7, 44-100 Gliwice, Poland

[b]The Public Authority for Applied Education and Training,
College of Technological Studies,
Department of Chemical Engineering, Kuwait

Original scientific paper
Received: December 13, 2007
Accepted: August 25, 2008

The population balance equation (PBE) is a continuity statement written in terms of the number density function. It describes, among others, the aggregation-breakage, nucleation and particle growth phenomena in solid-liquid suspensions. Fast and computationally inexpensive methods for solving the PBE are very much in demand today for computational fluid dynamics (CFD) simulations, industrial plants designs or scientific research. The Class Method (CM), Monte Carlo method (MC), method of moments (MOM) and its derivatives are the most popular solutions. In this work application of the Mathcad software for the quadrature method of moments (QMOM) is shown. The Mathcad allows one to create a simple and clear algorithm which can be easily changed according to the user needs, e.g. in modification of an aggregation kernel mathematical formula. The program written can be readily run on a standard PC computer. To verify a correctness of calculations the appropriate measurements were made and an efficiency of the program created was compared with another one written in the Turbo Pascal language.

*Key words:*

Aggregation, breakage, population balance equation (PBE), particle size distribution (PSD), quadrature method of moments (QMOM), Mathcad

## Introduction

Modern chemical engineering is focused, among others, in research on the behavior of solid-liquid suspensions. Processes taking place in such systems are usually very complex. Therefore, simulation and scale-up are very difficult because of various interactions between such processes as mixing in different scales, nucleation, crystal growth, aggregation and breakage.[1,2] There are many methods allowing one to estimate the evolution of particle size distribution (PSD). They can be divided into four groups:[3] zero-order methods, high-order methods, stochastic methods and internally inconsistent methods. In the first case, the aggregate sizes are divided into a finite number of classes assuming one constant value in each class. The first approach in this group was proposed by Hounslow *et al.*, as the class method.[4] Its idea is to set the size classes of the PSD in such a way that a volume of the aggregate in the $i^{\text{th}}+1$ class is two times larger than the volume of the aggregate in the class $i^{\text{th}}$. Another powerful methods are so called fixed-pivot technique[5] and moving-pivot technique.[6] In the first case the population is concentrated in a given size range at a single point (pivot) by means of the Dirac delta function. In the second case the pivots are dynamic and they change their values to represent the change in the average number distribution. In high-order methods the approximation of the distribution function is usually done by a set of linearly independent functions of the order larger than zero using the finite element method. Here the Galerkin methods[7,8] and orthogonal collocation method[7] should be mentioned. Another example in this group is the least squares method[9] which basic idea is to minimize the integral of the square of residual over the computational domain. In stochastic methods, the system behavior is simulated by the generation of random numbers used for the identification of probability functions governing the system behavior.[10,11] For example in the Monte Carlo simulation method a “sample path” of the process can be created by artificially generating random variables that satisfy the specified probability laws of change. By generating numerous such sample paths, the expected or mean behavior of the system can be calculated by averaging all of sample paths.[10] In the last group of solutions, i.e. internally inconsistent methods, the wavelet-based method proposed by Lu and Cameron[12] is worthwhile to mention. It is used to discretize the

[*] Corresponding author; fax: +48322371461;
e-mail: Marcin.Lemanowicz@polsl.pl

PBE by approximating the average number concentration and the coalescence frequency using a series of wavelets. The quadrature method of moments proposed by McGraw[13] is outside this classification. It will be presented in details in the further part of the paper.

Some of the PBE solving methods, like QMOM or its derivative DQMOM proposed by Marchisio and Fox,[14] are very efficient and can be used in CFD codes.[15,16,17,18] Others are simple but computationally expensive. As an example one can mention such methods like the CM or Monte Carlo simulations. Their application in CFD codes is ineffective due to a large number of scalars and long calculation time. But these methods can be used in "stand-alone programs",[19,20,21] e.g. in calculations of the kinetic coefficients in aggregation and breakage kernels.

The aim of this work was to create a computer program which would allow one to verify different aggregation and breakage kernels mathematical formulae with experimental data as well as to calculate the kinetic coefficients for a chosen aggregation-breakage model. The main guidelines were simplicity and flexibility.

Programming languages can display difficulties for standard computer users. Creating a program one should have not only large amount of chemical engineering knowledge but also experience in using advanced scripts. This problem can be solved by using more user-friendly software. For utility assessment, the Mathcad was selected. It presents large capabilities and is enough clear and understandable. In the Mathcad program, mathematical formulae can be written in unchanged forms. Also, calculating more complex problems like eigenvalues and eigenvectors is very simple (see appendix A), unlike e.g. in the Turbo Pascal language.

The population balance equations are solved using quadrature method of moments (QMOM). It is fast, effective and moreover, it enables one to verify effectiveness of build-in Mathcad functions. To verify a correctness of calculations the appropriate measurements were made in a laboratory setup consisted of the vibrating mixer and the laser particle size analyzer. The efficiency of the created program is compared with another one written in the Turbo Pascal language.[19,20,21]

## Population balance equation

The population balance equation (PBE) is a continuity statement written in terms of the number density function. The macro-distributed PBE has a form[22,23]

$$\frac{\partial n(v,t)}{\partial t} = \nabla[u\, n(v,t)] - n(v,t)\frac{\mathrm{d}(\log V)}{\mathrm{d}t} + + B - D - \sum_i \frac{Q_i n_i}{V} \tag{1}$$

Considering the aggregation and breakage only with no inlet and outlet streams, and omitting the nucleation, molecular growth, creation and fading of the aggregates, eq. (1) can be simplified to the general aggregation-breakage equation

$$\frac{\partial n(v,t)}{\partial t} = B - D \tag{2}$$

Terms "*B*" and "*D*" in eq. (2) correspond to birth and death of an aggregate of the volume "*v*" due to aggregation-breakage process. They can be described by four equations:

– rate of birth of particles due to aggregation of smaller particles

$$B_{ag} = \frac{1}{2}\int_0^y \beta_{ag}(v-u,u)\, n(u,t)\, n(v-u,t)\,\mathrm{d}u \tag{3}$$

– rate of death of particles due to aggregation of particles

$$D_{ag} = \int_0^\infty \beta_{ag}(v,u)\, n(v,t)\, n(u,t)\,\mathrm{d}u \tag{4}$$

– rate of birth of particles due to breakage of larger particles

$$B_{br} = \int_v^\infty b(v,u)\,\beta_{br}(u)\, n(u,t)\,\mathrm{d}u \tag{5}$$

– rate of death of particles due to breakage of particles

$$D_{br} = \beta_{br}(v)\, n(v,t) \tag{6}$$

For calculation purposes eqs. (2)–(6) are written in terms of the particle length, instead of the particle volume.[15] The new equations take the following forms, respectively

$$\frac{\partial n(L,t)}{\partial t} = B - D \tag{7}$$

$$B_{ag}(L,t) = \frac{L^2}{2}\int_0^L \frac{\beta_{ag}((L^3-\lambda^3)^{1/3},\,\lambda)}{(L^3-\lambda^3)^{2/3}} \cdot \cdot\, n((L^3-\lambda^3)^{1/3},t)\, n(\lambda,t)\,\mathrm{d}\lambda \tag{8}$$

$$D_{ag}(L,t) = n(L,t)\int_0^\infty \beta_{ag}(L,\lambda)\, n(\lambda,t)\,\mathrm{d}\lambda \tag{9}$$

$$B_{br}(L,t) = \int_L^\infty b(L,\lambda)\,\beta_{br}(\lambda)\,n(\lambda,t)\,\mathrm{d}\lambda \tag{10}$$

$$D_{br}(L,t) = \beta_{br}(L)\,n(L,t) \tag{11}$$

### Quadrature method of moments (QMOM)

Tracing time evolution of the PBE is computational expensive because of large number of scalars. Therefore, the method of moments was proposed where only low order moments of the number density function are tracked.[24] This considerably reduces the number of scalars allowing one to determine the particle size distribution. The $k^{\mathrm{th}}$ moment of the number density function "*n*" can be described as follows

$$\mu_k(t) = \int_0^\infty L^k\,n(L,t)\,\mathrm{d}L \tag{12}$$

Modifying eqs. (7)–(11) according to eq. (12) and connecting them together one receives

$$\begin{aligned}\frac{\mathrm{d}\mu_k}{\mathrm{d}t} = {} & \frac{1}{2}\int_0^\infty n(z,t)\int_0^\infty \beta_{ag}(z,\lambda)(z^3+\lambda^3)^{\frac{k}{3}}\,n(z,t)\,\mathrm{d}z\,\mathrm{d}\lambda - \\ & -\int_0^\infty L^k\,n(L,t)\int_0^\infty \beta_{ag}(L,\lambda)\,n(\lambda,t)\,\mathrm{d}\lambda\,\mathrm{d}\lambda + \ldots \\ & \ldots + \int_0^\infty L^k \int_0^\infty \beta_{br}(\lambda)\,b(L,\lambda)\,n(\lambda,t)\,\mathrm{d}\lambda\,\mathrm{d}L - \\ & -\int_0^\infty L^k\,\beta_{br}(L)\,n(L,t)\,\mathrm{d}L\end{aligned} \tag{13}$$

where

$$z = (L^3 - \lambda^3)^{\frac{1}{3}} \tag{14}$$

Then, the integrals in eq. (13) can be approximated by means of *N*-point Gaussian quadrature according to the equation

$$\mu_k(t) = \int_0^\infty L^k\,n(L,t)\,\mathrm{d}L \approx \sum_{i=1}^{N} w_i L_i^k \tag{15}$$

which gives the following closed transport equation

$$\begin{aligned}\frac{\mathrm{d}\mu_k}{\mathrm{d}t} = {} & \frac{1}{2}\sum_{i=1}^{n} w_i \sum_{j=1}^{N} w_j (L_i^3 + L_j^3)^{\frac{k}{3}}\,\beta_{ag}(L_i,L_j) - \\ & -\sum_{i=1}^{n} L_i^k w_i \sum_{j=1}^{N} w_j \beta_{ag}(L_i,L_j) + \ldots \\ & \ldots + \sum_{i=1}^{N} \beta_{br}(L_i)\,\bar{b}_i^{(k)} w_i - \sum_{i=1}^{N} L_i^k \beta_{br}(L_i)\,w_i\end{aligned} \tag{16}$$

The main problem regarding QMOM is related to the use of the power moments for characterizing the number density function.[25] Grosch *et al.*[26] proved that QMOM should be used for the investigated models with non-local, smooth phenomena. For models with phenomena that act strictly locally on a given particle size range it tends to oscillate around the true solutions for increasing number of nodes. Therefore, Grosch *et al.* recommend this method as a model reduction technique for process control and optimization for simple cases.

Another disadvantage of the QMOM is the number of used moment paradox.[27] The nature of the breakage kernel functions determine the number of moments that should be used in the case of this method for reaching a certain accuracy. The higher the polynomial order of the kernel functions, the higher the number of moments required for getting reliable predictions. On the other hand, the PD algorithm is a numerical ill-conditioned method for computing the Gauss quadrature rule and it fails with the number of moments larger than 14.

In the calculations the set of 6 moments are used which results with 3 pairs of weights and abscissas obtained through the PD algorithm. Therefore, it should be the fastest[26] and the optimal[15] solution.

### Theoretical research

A first step of research was a comparison of the results obtained using the program with analytical solutions.[28] The same procedure was made by Marchisio *et al.*[15] Five cases were considered (Table 1). In the first case (Fig. 1) constant aggregation and breakage kernels were used with the symmetric fragmentation fragment-distribution function

$$\beta_{ag}(L_i,L_j) = \beta_{ag,0} \tag{17}$$

$$\beta_{br}(L_i) = \beta_{br,0} \tag{18}$$

$$\bar{b}_i^{(k)} = 2^{\frac{3-k}{3}} L_i^k \tag{19}$$

The symmetric fragmentation fragment-distribution function was also used in the second case (Fig. 2). Here and also in all other cases the hydrodynamic aggregation function was employed

$$\beta_{ag}(L_i,L_j) = \beta_{ag,0}(L_i + L_j)^3 \tag{20}$$

In cases 2–4 (Figs. 2–4) a breakage kernel has an exponential form

$$\beta_{br}(L_i) = \beta_{br,0}\exp(\delta L_i^3) \tag{21}$$

In the last case (Fig. 5) a power law breakage kernel was used

Table 1 – *Theoretical cases*

| No | Initial set of moments | Aggregation kernel | $\beta_{ag,0}$ | Breakage kernel | $\beta_{br,0}$ | Fragment distribution function | Analytical solution $\mu_0$ | Mathcad solution $\mu_0$ | Time *t*/s |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $\mu_k = 1$ for $k = 0 \div 5$ | constant | 1 | constant | 0.02 | sym. frag. | 0.035 | 0.039 | 200 |
| 2 | | hydrodyn. | 1 | exponential $\delta = 0.01$ | 0.1 | sym. frag. | 0.004 | 0.004 | 10 |
| 3 | | hydrodyn. | 1 | exponential $\delta = 0.01$ | 0.1 | mass ratio $\zeta = 1:4$ | 0.005 | 0.005 | 10 |
| 4 | | hydrodyn. | 1 | exponential $\delta = 0.01$ | 0.1 | erosion | 0.273 | 0.259 | 10 |
| 5 | | hydrodyn. | 1 | power law $\alpha = 6$ | 0.01 | erosion | 0.298 | 0.296 | 10 |

$$\beta_{br}(L_i) = \beta_{br,0} L_i^{\alpha} \tag{22}$$

Two remaining fragment distribution functions, i.e. mass ratio 1 : 4 and erosion took a following form respectively

$$\bar{b}_i^{(k)} = L_i^k \frac{4^{\frac{k}{3}} + 1}{5^{\frac{k}{3}}} \tag{23}$$

$$\bar{b}_i^{(k)} = L_0^k + (L_i^3 - L_0^3)^{\frac{k}{3}} \tag{24}$$

Figs. 1–5 present dimensionless moments time evolution. The dimensionless moments were defined as:

$$\mu_k = \frac{\mu_k(t)}{\mu_k(t=0)} \tag{25}$$

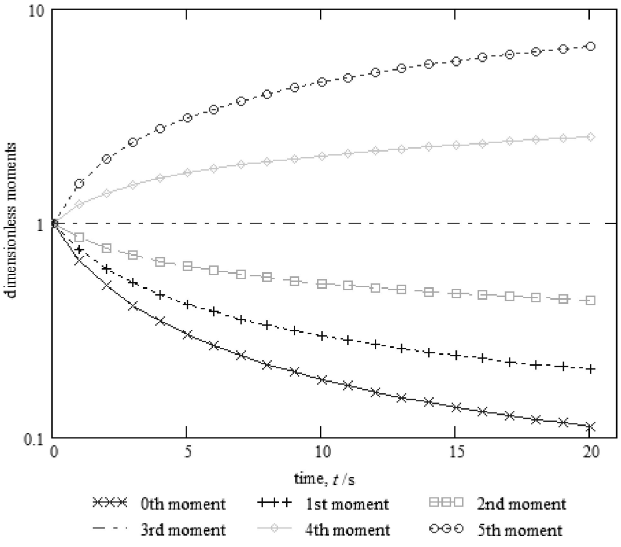


Fig. 1 – *Dimensionless moments time evolution for the first case*

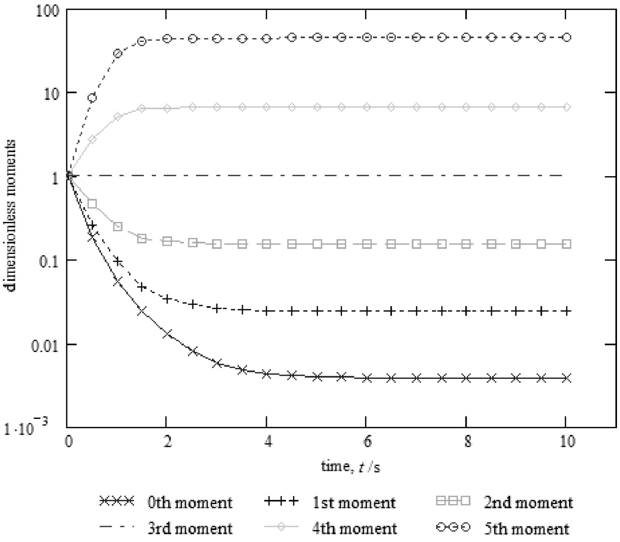


Fig. 2 – *Dimensionless moments time evolution for the second case*

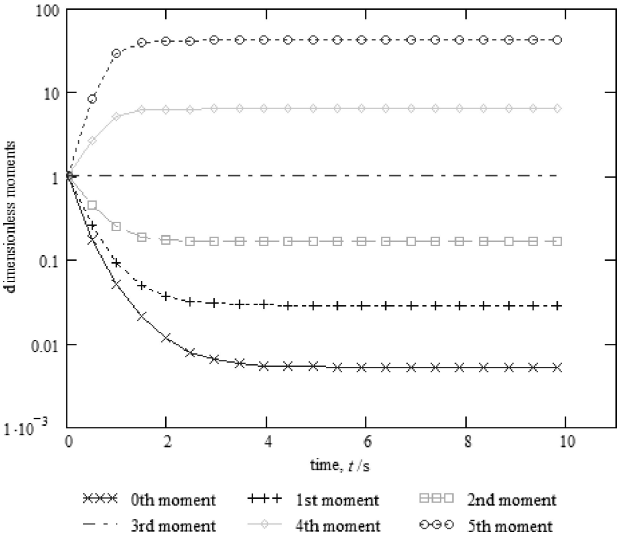


Fig. 3 – *Dimensionless moments time evolution for the third case*

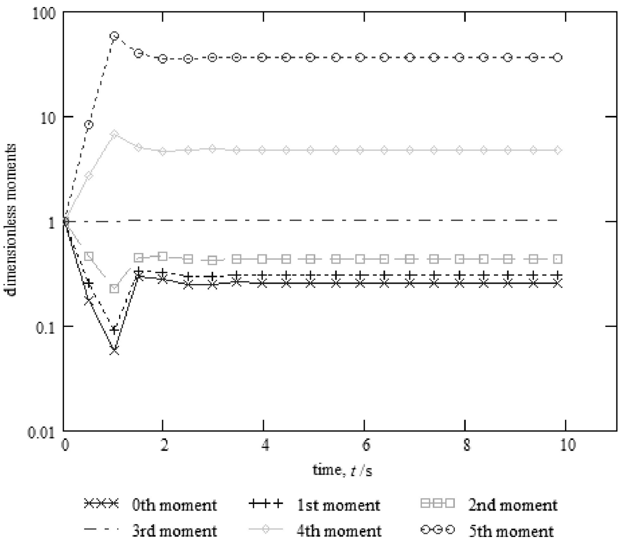


Fig. 4 – *Dimensionless moments time evolution for the fourth case*

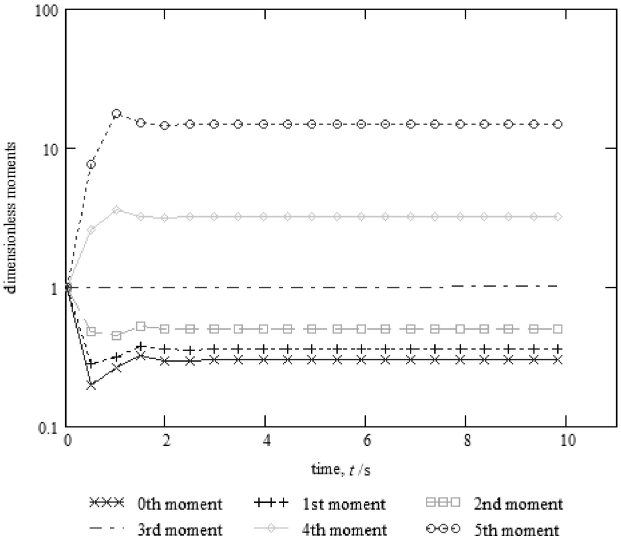


Fig. 5 – *Dimensionless moments time evolution for the fifth case*

## Experiments

The next step was application of the program created in calculation of kinetic coefficients for a chosen aggregation-breakage model based on experimental data. The experiments were carried out using commercial chalk particles ($d_{32} \approx 2$ μm Fig. 6) suspended in distilled water in the presence of commercial flocculant ZETAG 63. ZETAG 63 was a synthetic, high molecular mass cationic polyacrylamide of medium charge, used mainly in sewage treatment processes. The laboratory setup consisted of the mixing tank equipped with the vibrating impeller, laser particle size analyzer Fritsch Analysette 22[29] and peristaltic pump (Fig. 7). Measurements were made in 1, 3, 5, 10, 15 and 20 minutes after injection of the flocculant. The suspension was gained though the suction tube and it was transported to the measuring cell where measurements were taken. Then the suspension was transported through the peristaltic pump back to the tank. The experiments were run at the following parameters:

- Volume of suspension in the mixer – $4.8 \cdot 10^{-3}$ m$^3$
- Concentration of chalk particles – 0.375 kg m$^{-3}$
- Mass fraction of ZETAG 63 flocculant – $4.44 \cdot 10^{-3}$ kg kg$^{-1}$ (with respect to solid phase)
- Mixing disc frequency – 2.83 s$^{-1}$

The initial chalk PSD fluctuations did not have any influence on the final results. Measurements were repeated at least 3 times to eliminate random errors. The results were verified by the statistical test. According to the Analysette 22 documentation[29] the laser particle size analyzer fulfils ISO 9001 and ISO 13320-1 standards. The experimental procedure is also described in detail elsewhere.[19,20,21]

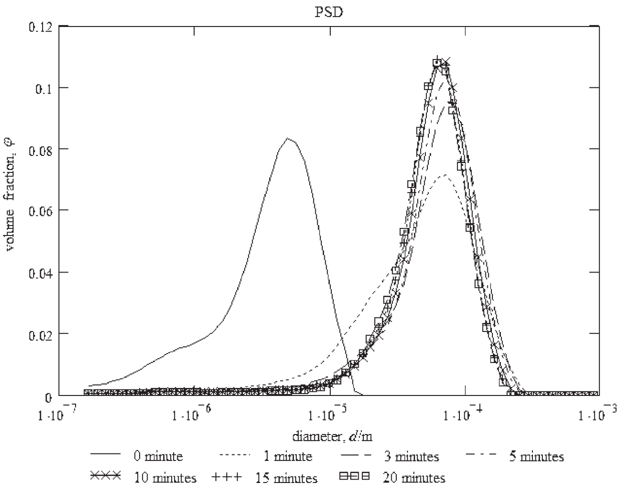


Fig. 6 – *The PSD obtained in experiments*

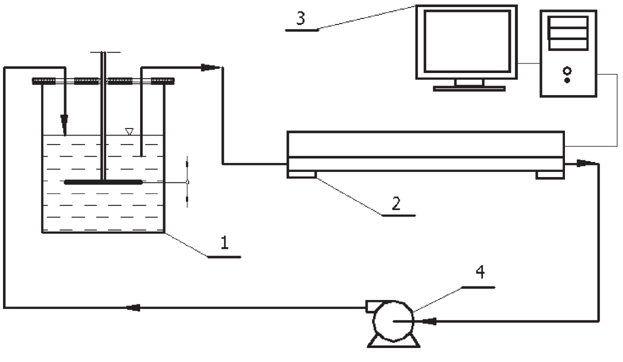


Fig. 7 – *Laboratory setup scheme: 1) mixing tank, 2) laser particle size analyzer, 3) computer, 4) peristaltic pump*

## Results and discussion

The Mathcad calculations were performed for the length-based size-dependent hydrodynamic aggregation kernel and length-based size-depended power law breakage kernel with $\alpha = 6$ (square of the particle volume), $\alpha = 3$ (particle volume) and $\alpha = 1$ (particle length). Symmetric fragmentation as the fragment distribution function was chosen, the constant porosity $\vartheta = 0.38$[30] and shape factor for chalk particles $k_v = 2.28$ according to Heywood[31] was assumed. These aggregation-breakage parameters were selected on the basis of research results taken from previous works.[19,20,21] Values of $\beta_{ag,0}$ and $\beta_{br,0}$ were found through a minimizing objective function which was the sum of squares of differences of moments obtained from experiments and moments obtained from the model. The algorithm used the minimizing function build in the Mathcad.

$$\mathrm{ObF}(\beta_{ag,0},\beta_{br,0}) = \\ = \sum_i ((\mu_{i-\mathrm{exp}} - \mu_{i-\mathrm{math}}(\beta_{ag,0},\beta_{br,0}))a_i)^2 \quad (26)$$

$$\begin{pmatrix}\beta_{ag,0}\\ \beta_{br,0}\end{pmatrix} = \mathrm{Minimize}(\mathrm{ObF},\beta_{ag,0},\beta_{br,0}) \quad (27)$$

The program could not find the solution trying to match all values of moments from two time points, so a new objective function was created. Only 0th moment values were taken into consideration. Solution was found but as one can expect obtained set of moments did not match the set of moments from experiments (Figs. 8–10).

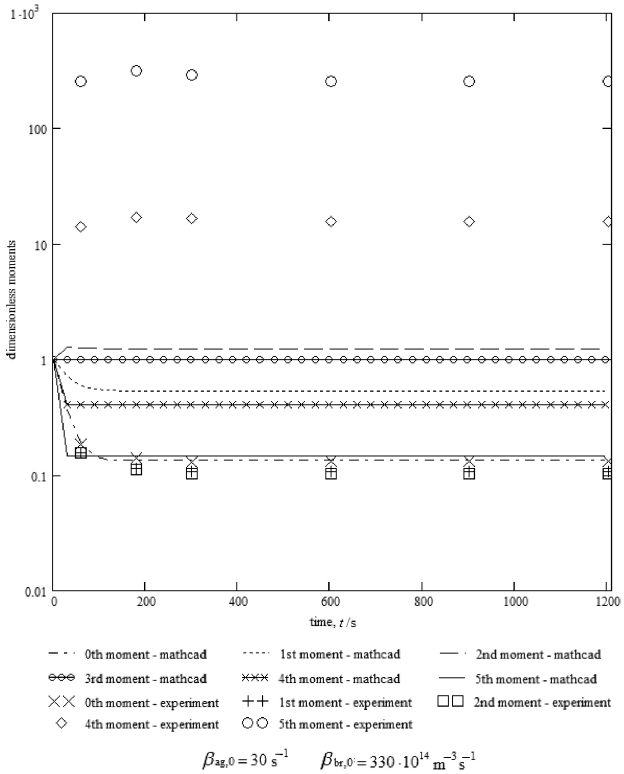


Fig. 9 – *Dimensionless moments time evolution* $\alpha = 3$

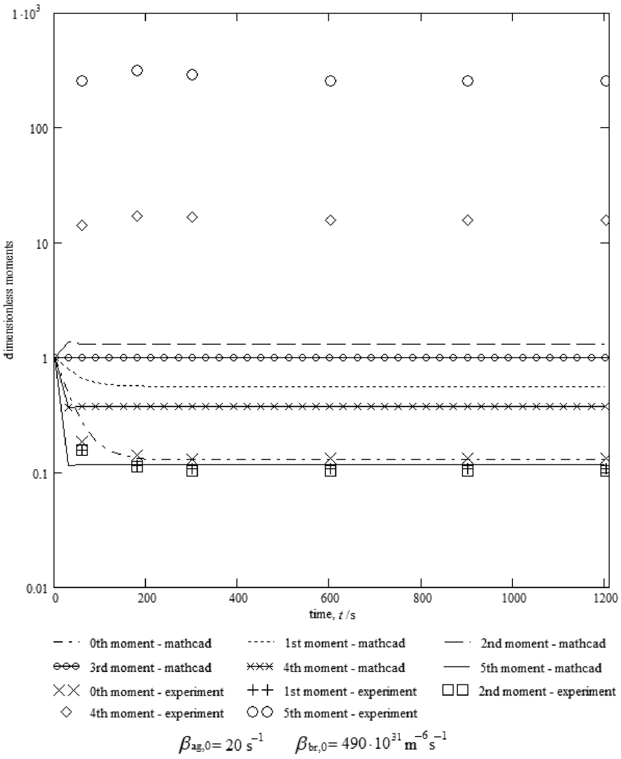


Fig. 8 – *Dimensionless moments time evolution* $\alpha = 6$

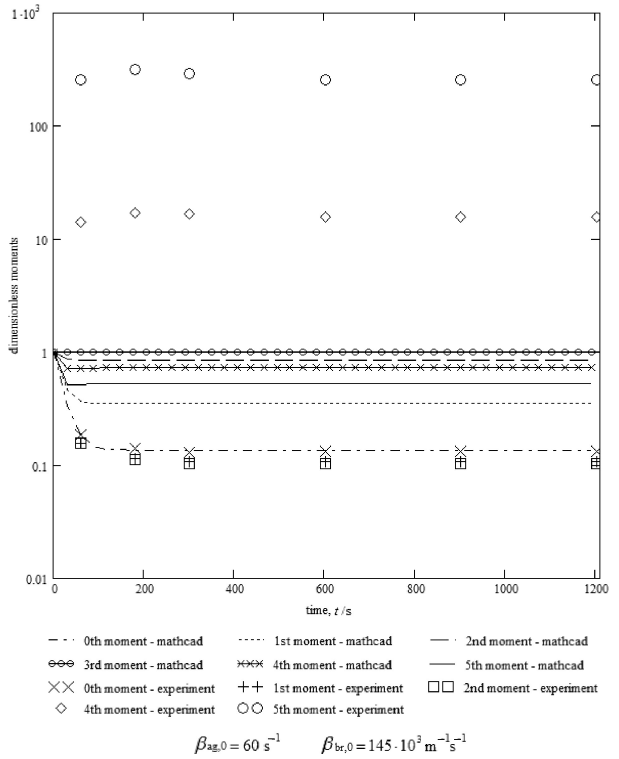


Fig. 10 – *Dimensionless moments time evolution* $\alpha = 1$

The next step was to find a value of $\alpha$ which enabled one to match the whole set of moments. The optimal value was obtained through series of calculations for the breakage kernel mathematical formulae $\beta_{br,0}$ $L^{1/N}$, where $N$ was successive natural numbers. The best convergence was found for $N$ = 8 (Figs. 11–12). As it is shown, only the second moment was not predicted correctly. The large values of the number density function at the lowest size classes are responsible for the moments inversion (difference between the lowest size classes and new ones is of $10^7$ order of magnitude). If one takes $d_{10}$ into consideration (i.e. mean particle size term of the particle number), it decreased during the experiment, although the particles aggregated. This results with unusual moment's order: 2nd, 1st, 0 instead 0, 1st, 2nd. Authors believe that the incorrect evolution of the 2nd, 4th and 5th moments in Figs. 8–10 is caused by the unusual shape of the initial number density function (broad range) and calculation errors of the PD algorithm. The second issue is connected with the mathematical formulae of breakage process.[27] For a constant number of moments used in calculations, the convergence was better with the lower polynomial order of the breakage kernel. The further research on the impact of this fact is needed. The obtained results allowed one to predict properly the mean diameter in the terms of particle numbers $d_{10}$ and mean diameter in the terms of volumetric fraction $d_{43}$.

The last step was a comparison of the Mathcad program and the Turbo Pascal program. The Turbo Pascal program using the Class Method estimated $\beta_{ag,0}$ and $\beta_{br,0}$ values in such a way that square of the sum of deviations between the experimental volume concentration and those obtained from the model had the least values (for details see[19,20,21]). Result after 10 minutes is shown in Fig. 13.

The Turbo Pascal program calculates the number of particles in newly defined size classes with better convergence that the QMOM algorithm. It also allows one to present the final PSD. There was no large difference in the computational time. Unfortunately, the Turbo Pascal does not allow for a simple instant change of the aggregation-breakage parameters like aggregation and breakage kernels or fragment distribution function.

## Conclusion

The Mathcad software can be used to create the QMOM algorithm. Its simplicity and flexibility allows inexperienced users a fast and easy creation of an appropriate algorithm. Also finding the solution does not cause a problem due to the ready to use

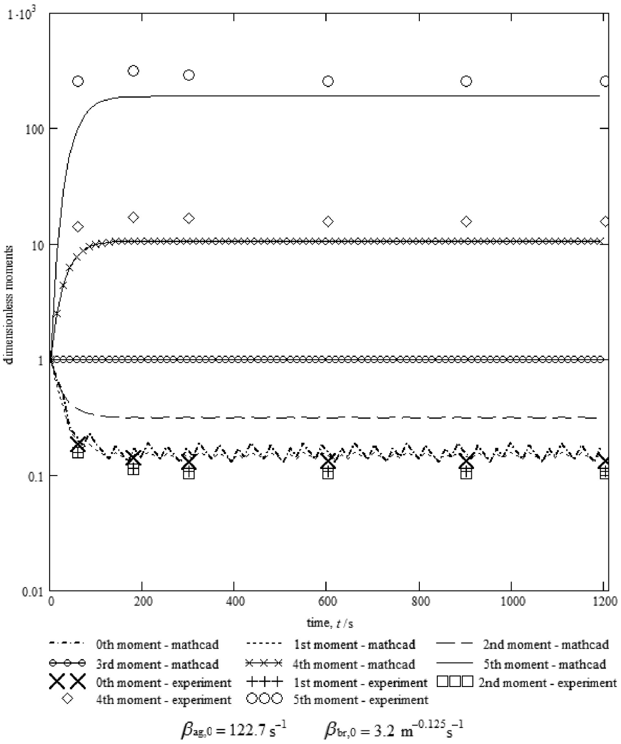


Fig. 11 – *Dimensionless moments time evolution* $\alpha = 1/8$

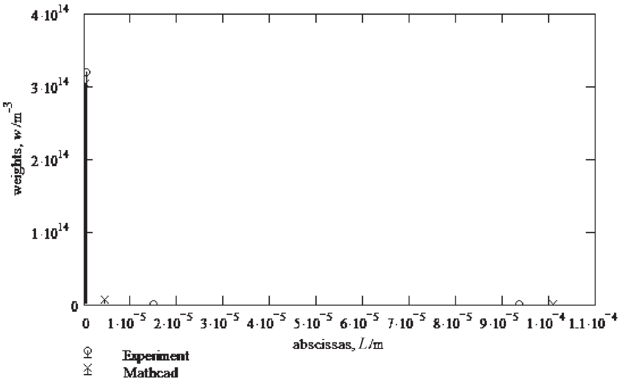


Fig. 12 – *Weights and abscissas in 10 minute* $\alpha = 1/8$

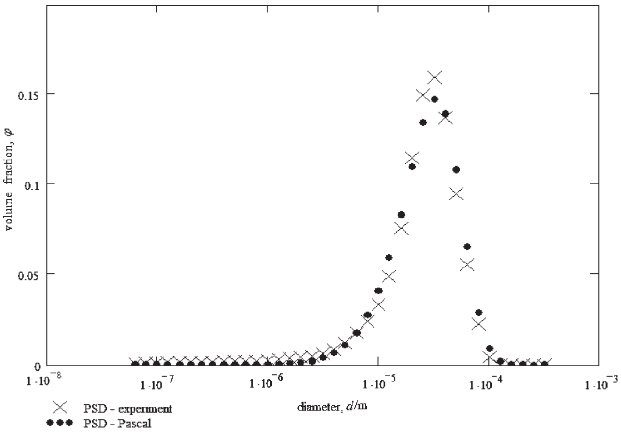


Fig. 13 – *The PSD after 10 minutes – experiments and the Turbo Pascal program calculation results*

optimizing functions. The source code is clear and can be modified according to the user's needs.

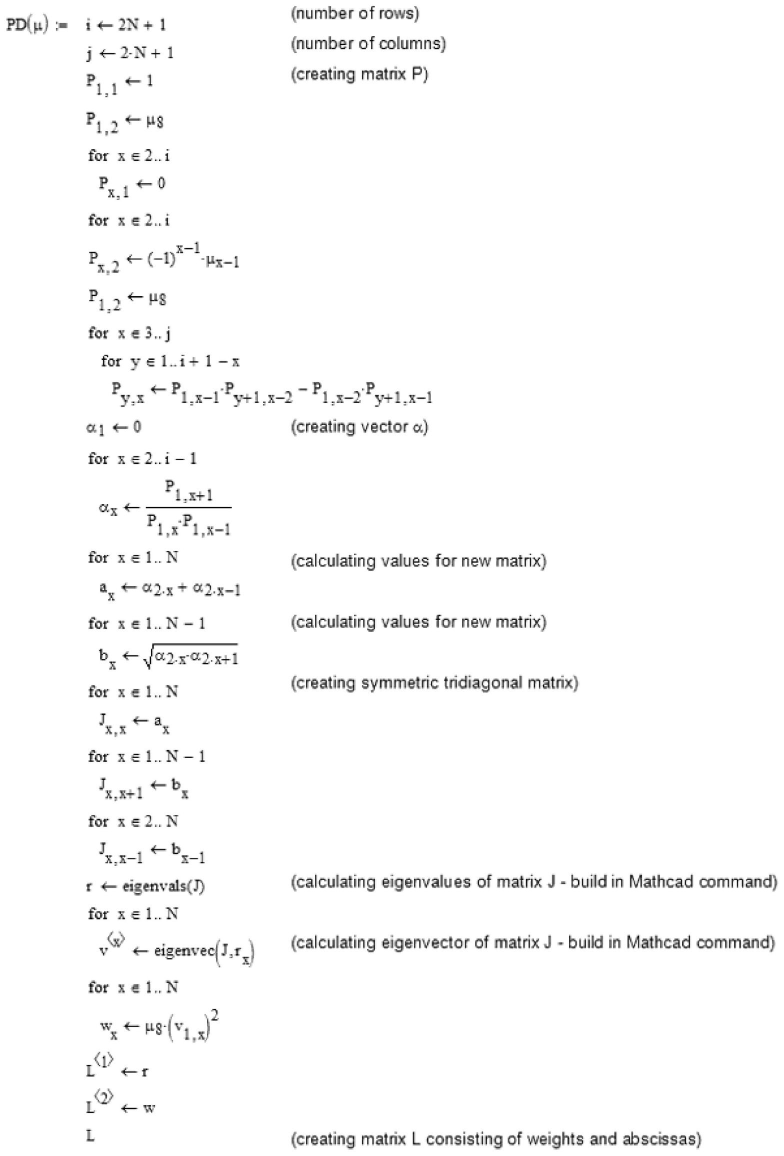
The created program allows one to simulate solid-liquid suspension behavior for any aggregation and breakage kernels and fragment distribution function. It is fast and precise. Any desirable results can be presented, e.g. values of moments, weights and abscissas values or time derivatives of moments. The Mathcad algorithm can be implemented to a larger program.

Theoretical calculations were verified by analytical solutions with fairly good result. Five cases were shown in theoretical section.

Unfortunately, the experimental results could not be reconstructed from moment simulation with a satisfying convergence. There are two possible sources of this failure. The first one is the lack of information about the PSD under 0.15 μm which is crucial for calculations. The transformation of the volume-based PSD into a number density function shows that the lowest class is the most numerous one. The second reason is caused by large numbers used in calculations. A difference between number of particles in the lowest size class and in new size classes created in aggregation is in the $10^7$ order of magnitude. Therefore paradoxically, although the aggregation phenomenon occurs, the particle number based mean diameter decreases. This results with the moment inversion. These two problems require further examination.

## Appendix A: PD algorithm written in the Mathcad software

In PD algorithm 0th moment is assigned to 8th place in the matrix.



```
$PD(\mu) :=$  $i \leftarrow 2N + 1$                          (number of rows)
              $j \leftarrow 2 \cdot N + 1$                   (number of columns)
              $P_{1,1} \leftarrow 1$                          (creating matrix P)
              $P_{1,2} \leftarrow \mu_8$
              for $x \in 2..i$
                $P_{x,1} \leftarrow 0$
              for $x \in 2..i$
              $P_{x,2} \leftarrow (-1)^{x-1} \cdot \mu_{x-1}$
              $P_{1,2} \leftarrow \mu_8$
              for $x \in 3..j$
                for $y \in 1..i + 1 - x$
                  $P_{y,x} \leftarrow P_{1,x-1} \cdot P_{y+1,x-2} - P_{1,x-2} \cdot P_{y+1,x-1}$
              $\alpha_1 \leftarrow 0$                          (creating vector α)
              for $x \in 2..i - 1$
                $\alpha_x \leftarrow \frac{P_{1,x+1}}{P_{1,x} \cdot P_{1,x-1}}$
              for $x \in 1..N$                                  (calculating values for new matrix)
                $a_x \leftarrow \alpha_{2 \cdot x} + \alpha_{2 \cdot x-1}$
              for $x \in 1..N - 1$                              (calculating values for new matrix)
                $b_x \leftarrow \sqrt{\alpha_{2 \cdot x} \cdot \alpha_{2 \cdot x+1}}$
                                                                (creating symmetric tridiagonal matrix)
              for $x \in 1..N$
                $J_{x,x} \leftarrow a_x$
              for $x \in 1..N - 1$
                $J_{x,x+1} \leftarrow b_x$
              for $x \in 2..N$
                $J_{x,x-1} \leftarrow b_{x-1}$
              $r \leftarrow$ eigenvals(J)                      (calculating eigenvalues of matrix J - build in Mathcad command)
              for $x \in 1..N$
                $v^{\langle x \rangle} \leftarrow$ eigenvec$(J, r_x)$   (calculating eigenvector of matrix J - build in Mathcad command)
              for $x \in 1..N$
                $w_x \leftarrow \mu_8 \cdot (v_{1,x})^2$
              $L^{\langle 1 \rangle} \leftarrow r$
              $L^{\langle 2 \rangle} \leftarrow w$
              L                                                 (creating matrix L consisting of weights and abscissas)
```

Fig. 14 – *PD algorithm in the Mathcad*

## Notation

| | |
|---|---|
| $b$ | – volume- or length-based fragment distribution function, $m^{-3}$, $m^{-1}$ |
| $\bar{b}^{(k)}$ | – length-based fragment distribution function after moment transformation |
| $B$ | – birth terms, $m^{-6}s^{-1}$, $m^{-4}s^{-1}$ |
| $d$ | – diameter, μm, m |
| $D$ | – death terms, $m^{-6}s^{-1}$, $m^{-4}s^{-1}$ |
| $k$ | – order of the moment |
| $k_v$ | – shape factor for chalk particles, – |
| $L$ | – particle length or abscissa, m |
| $n$ | – volume- or length-based number density function, $m^{-6}$, $m^{-4}$ |
| $N$ | – number of nodes in quadrature approximation |
| $Q$ | – volumetric flow rate, $m^3s^{-1}$ |
| $t$ | – time, s |
| $u$ | – particle volume, $m^3$ |
| $w$ | – weight moment, $m^{-3}$ |
| $v$ | – particle volume, $m^3$ |
| $V$ | – system volume, $m^3$ |
| $z$ | – new variable defined in eq. (14), m |

## Greek letters

| | |
|---|---|
| $\alpha$ | – geometry factor |
| $\beta_{ag}$ | – volume- or length-based aggregation kernel, $m^3$ $s^{-1}$ |
| $\beta_{ag.0}$ | – kinetic coefficient in volume- or length-based aggregation kernel, depends on a model |
| $\beta_{br}$ | – volume- or length-based breakage kernel, $s^{-1}$ |
| $\beta_{br,0}$ | – kinetic coefficient in volume- or length-based breakage kernel, depends on a model |
| $\delta$ | – parameter in eq. (21) |
| $\zeta$ | – mass ratio |
| $\lambda$ | – particle length, m |
| $\mu$ | – moment |
| $\varphi$ | – volume fraction |

## References

1. *Hill, P. J., Ng, K. M*, Chem. Eng. Sci. **57** (2002) 2125.
2. *Maximova, N., Dahl, O.,* Current Opinion in Colloid & Interface Science **11** (2006) 246.
3. *Attarakih, M. M., Bart, H. J., Faqir, N. M.*, Chem. Eng. Sci. **59** (2004) 2567.
4. *Hounslow, M. J., Ryall, R. L., Marshall, V. R.*, AIChE J. **34** (11) (1988) 1821.
5. *Kumar, S., Ramkrishna, D.*, Chem. Eng. Sci. **51** (1996) 1311.
6. *Kumar, S., Ramkrishna, D.*, Chem. Eng. Sci. **51** (1996) 1333.
7. *Nicmanis, M., Hounslow, M. J.*, AIChE. J. **44** (1998) 2258.
8. *Wulkow, M., Gerstlauer, A., Nieken, U.*, Chem. Eng. Sci. **56** (2001) 2575.
9. *Dorao, C. A., Jakobsen, H. A.*, Computers Chem. Eng. **30** (2006) 535.
10. *Ramkrishna, D.*, Population Balances. Theory and Applications to Particulate Systems in Engineering, Academic Press Inc., San Diego, 2000.
11. *Smith, M., Matsoukas, T.*, Chem. Eng. Sci. **53** (1998) 1777.
12. *Liu, Y., Cameron, I. T.*, Chem. Eng. Sci. **56** (2001) 5283.
13. *McGraw, R.*, Aerosol Sci. Tech. **27** (1997) 255.
14. *Marchisio, D. L., Fox, R. O.,* J. Aerosol Sci. **36** (2005) 43.
15. *Marchisio, D. L., Vigil, D. L., Fox, R. O.*, J. Coll. Int. Sci. **258** (2003) 322.
16. *Marchisio, D. L., Vigil, D. L., Fox, R. O.*, Chem. Eng. Sci. **58** (2003) 3337.
17. *Marchisio, D. L., Pikturna, J. T., Fox, R. O., Vigil, R. D., Barresi, A. A.*, AIChE J. **49** (5) (2003) 1266.
18. *Baldyga, J., Orciuch, W., Makowski, L., Malasik-Brodzicki, M., Malik, K.*, Chem. Eng. Proc. **46** (2007) 851.
19. *Gierczycki, A. T.*, Chem. Eng. Comm. **192** (2005) 1361.
20. *Gierczycki, A. T.*, Chem. Proc. Eng. **28** (2007) 629.
21. *Gierczycki, A. T., Mohsen H. Al-Rashed*, Chem. Eng. Comm. **195** (2008) 427.
22. *Randolph, A. D., Larson, M. A.*, Theory of particulate Processes, Academic Press Inc., New York, 1988.
23. *Nicmanis, M., Hounslow, M. J.*, Computers Chem. Eng. **20** (1996) 261.
24. *Hulburt, H. M., Katz, S.*, Chem. Eng. Sci. **19** (1964) 555.
25. *Dorao, C. A., Jakobsen, H. A.*, Chem. Eng. Sci. **61** (2006) 7795.
26. *Grosch, R., Briesen, H., Marquardt, W.*, AIChE J. **53** (1) (2007) 207.
27. *Dorao, C. A., Jakobsen, H. A.*, J. of Computational and Applied Mathematics **193** (2006) 619.
28. *Vani, M.,* J. Coll. Int. Sci. **221** (2000) 143.
29. www.fritsch.de
30. *Bos, A. S., Zuiderweg, F. J.,* Chem. Eng. Res. Des. **65** (1987) 187.
31. *Koch, R., Noworyta, A.,* Mechanical processes in chemical engineering (in Polish), WNT, Warsaw, 1995.