

Simulation and Control of Drop Size Distributions in Stirred Liquid/Liquid Systems

J. Heiland¹, M. Baumann¹, A. Walle², V. Mehrmann¹, M. Schäfer²

¹*Department of Mathematics, TU Berlin, Germany*

²*Institute of Numerical Methods in Mechanical Engineering, TU Darmstadt, Germany*

Techniques to control the particle distribution may significantly improve the performances of liquid/liquid mixing processes in many industrial applications. Currently there are no mathematical tools available that can be used to control the transient processes in stirred liquid/liquid systems in CFD simulations. This paper addresses the simulation of dispersions using population balance equations and describes the coupling of the flow solver FASTEST 3D, that simulates the dispersion in stirred tanks, with MATLAB. This coupling makes a vast variety of control routines from the MATLAB *Control Toolbox* directly available in the stirrer simulations. Also an exemplary control design, using an identified model and a linear quadratic regulator is presented and discussed.

Keywords – Liquid/liquid dispersion, Control interface, Control design, DQMOM, CFD

1 Introduction

Stirred liquid/liquid systems can be found in numerous technical processes, especially in chemical, oil and gas, pharmaceutical and food industries. In particular the mixing of immiscible fluids, that lead to dispersions of fluids in a continuous phase, are of major importance. The properties of these dispersions are essential for the technical process, e.g. chemical reactions. Hence, it is desirable to be able to accomplish a requested particle size distribution (PSD).

Since experimental investigations are very costly and time-consuming, the advantages of computational fluid dynamics (CFD) should be utilised also for mixing processes. CFD has already become the standard tool for predicting the flow in turbulent fields and it's application to stirred tanks shows good results [7, 19]. For the simulation of the particle size distribution, in a mixing-process several numerical techniques have been developed, verified and partially implemented into CFD-software packages, c.f. [2, 14, 20].

The control of the PSD has been addressed during the last two decades. In particular for experimental setups of polymerisation reactors there exists a vast record of investigations, c.f [22] and the references therein. In the most cases the PSD is influenced by changing parameters that immediately act on the particle size distribution like feed rate and temperature.

For the control of the PSD in the reactor by influencing the flow field, however, no mathematical methods are available. As a first step, the investigations described in [17] showed that such a joint system of flow and population balance equations (PBE) has the potential to allow control.

We present a coupling of the combined flow/PBE solver to MATLAB which makes the problem available to the methods implemented in the MATLAB *Control Toolbox*. This

includes heuristic feedback control like PID controllers, optimal control on the base of identified models, and model reduction techniques.

The basis for the CFD in a stirred tank is the in-house flow solver FASTEST 3D [4], which is based on a fully conservative finite volume method (FVM) for the solution of the incompressible Navier-Stokes equations on a non-staggered, cell-centered, blockstructured, boundary-fitted grid.

To compute the particle size distribution, the flowfield of the mixture, treated as single fluid, is solved and used as an input for the suitably modelled population balance equation.

The quantities of the solver are then transferred to MATLAB where a control algorithm computes the inputs for the solver for the next time step.

The work presented is funded by DFG and is part of a joint project dedicated to the control of particle size distributions in stirred liquid/liquid dispersions. At every stage, the numerical results are compared with experimental findings, which are supplied by our partners from the working group of M. Kraume at TU Berlin, Department of Process Engineering.

2 The Model

To reduce the overall complexity of the numerical solution of the flow system, the mixture is treated as a single fluid and the dispersion is defined via a separate population balance equation. Thereto one calls on the basic assumptions:

1. The volume fraction ϕ of the dispersed phase is small, i.e. $\phi \sim 0.1$. This means that the mixture is mainly made up by the so-called continuous phase, whereas the second phase occupies not more than 10% of the overall volume.
2. The physical properties density and viscosity of the fluids are similar, such that e.g. buoyancy and inertia effects can be neglected.

And concludes that:

3. The formation of the dispersed phase is defined by but does not influence the flow field. This also includes that the drops of the dispersed phase simply move with the continuous phase.

For the investigated setup of water and toluene with $\phi = 0.1$ these assumptions are acceptable. Thus we model the system by incompressible Reynolds averaged Navier-Stokes equations (RANS)

$$\frac{\partial \rho v}{\partial t} + \text{div}(\rho v v) - \text{div} T^* - \rho f = 0 \quad (1a)$$

$$\text{div} v = 0, \quad (1b)$$

to determine the flow of the mixture and track the dispersion via the population balance equation

$$\frac{\partial n}{\partial t} + \text{div}(v n) + \text{div}(\Gamma_t \nabla n) - S = 0. \quad (1c)$$

Here v denotes the mean velocity and ρ is the density of the mixture, T^* is the turbulent stress tensor and f denotes a volume force acting on the flow. The dispersion is described by the number density function n and Γ_t denotes a turbulent diffusion coefficient, c.f [17]. The source terms, modelling the evolution of the dispersion, are pooled in the function S .

The above equations are posed within the spatial domain occupied by the stirrer for time t varying in a finite time interval $(0, T]$ and are to be completed by suitable initial and boundary conditions. The number density function and hence equation (1c) additionally depend on internal variables. In this investigation we assume that the dispersion is defined only by the diameter of the drops, i.e. the diameter d is the only internal variable. Note that the PBE is influenced by the flow field, since the velocity v appears in the advection term and also the sources S depend on variables extracted from the flow field, while the reverse coupling is not considered in this model.

For the numerical simulation the turbulent stresses are approximated using the k - ε model and the number density function is approximated by a set of its first moments.

2.1 Numerical Simulation of the Flow in Stirred Tanks

In order to ensure the comparability of simulation and measurements, the numerical setup is modelled after the experimental arrangement, c.f. [5]. The matter of interest is a Rushton turbine (see Figure 1-(a)) with water as continuous phase and toluene as dispersed phase, whereas we confine the following treatise to the simulation of the continuous phase. The rotational speed ω is used as a parameter for the active control of the drop size distribution. Since the flow is expected to be turbulent, the simulations are performed with the Reynolds averaged Navier-Stokes (RANS) equations. Therefore the standard k - ε -model with wallfunction is applied. For the correction of velocity and pressure the SIMPLE algorithm is implemented in the used flow solver FASTEST 3D and the resulting linear system is calculated with an iterative ILU solver.

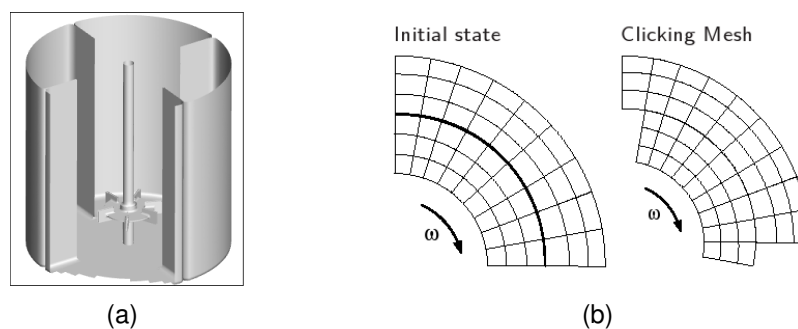


Figure 1: (a) Stirrer geometry created with grid generation tool and (b) clicking mesh technique for grid movement

For the spatial discretisation of the mixer the algebraic grid-generation-tool GRIDGEN is used, which was especially designed for the grid generation of various stirrer types by the group of M. Schäfer at TU Darmstadt. GRIDGEN has the advantage of a short computation time and results in a very well adapted, hexaedron, blockstructured, boundary fitted grid, which fulfills all requirements.

A stirred vessel involves a time dependent flow geometry and can be considered as a rotor-stator configuration. The fluid structure interaction caused by the existence of

rotating and non-rotating parts requires a special approach. Clicking grids, Sliding grids and Deforming grids are three general methods for realizing the time dependent geometry changes of rotor-stator configurations. In FASTEST 3D the clicking grid method is implemented, which is based on a partitioning of the whole fluid region into an inner rotating and an outer stationary region.

For realizing the rotation of the stirrer, the inner region rotates and "clicks" into a new position for every time step. The next position of the inner region is defined such that the grid lines are exactly connected to a new partner, so the boundary grid cells find a new neighbour in radial direction. The main advantage is that no grid deformation is needed and for every time step the finite volume method can be applied as usual. On the other hand the number of possible rotation positions is restricted by the number of grid segments in azimuthal direction. The time step size depends on the grid and the given rotational speed, because during one time step the next clicking position has to be reached. As there are restrictions for the time step size with respect to the simulation of turbulent flow, the rotational speed is restricted as well.

Considering that the fluid elements on the clicking interface change their positions and get new neighbour cells in every time step, a data transfer has to be performed in every time step. Velocities and mass fluxes are shifted to their original place. Additionally for the data transfer at the clicking interface a coordinate transformation according to the different frames of reference has to be realised.

2.2 Approximate Solution of the PBE Using DQMOM

Instead of solving the PBE (1c) directly, one can, as firstly proposed in [9], compute a certain number of moments to approximately reconstruct the density function n . Following the investigations by Marchisio [13] we solve the transport equations for N weights w_α and weighted abscissae $\zeta_\alpha = w_\alpha \xi_\alpha$

$$\frac{\partial w_\alpha}{\partial t} + \text{div}(vw_\alpha) - \text{div}(\Gamma_t \nabla w_\alpha) = a_\alpha \quad (2a)$$

$$\frac{\partial \zeta_\alpha}{\partial t} + \text{div}(v\zeta_\alpha) - \text{div}(\Gamma_t \nabla \zeta_\alpha) = b_\alpha, \quad (2b)$$

where the source terms a_α and b_α are determined by the linear equation system

$$(1 - k) \sum_{\alpha=1}^N \xi_\alpha^k a_\alpha + k \sum_{\alpha=1}^N \xi_\alpha^{k-1} b_\alpha = r_k^{(N)}, \quad \text{for } k = 0, \dots, 2N - 1 \quad (2c)$$

with the right hand side

$$r_k^{(N)} := \int_{-\infty}^{\infty} d^k s(d) \mathbf{d} \mathbf{d} + k(k-1) \sum_{\alpha=1}^N \xi_\alpha^{k-2} w_\alpha \Gamma_t \|\nabla \xi_\alpha\|_2^2.$$

consisting of the moment transformed source terms and a diffusive correction term. The turbulent diffusivity is computed via $\Gamma_t = 0.09 \cdot \frac{k^2}{\varepsilon}$. The quadrature formula, the weights and abscissae are belonging to, is defined such that the first N moments of n can be computed exactly via

$$m_k := \int_0^\infty d^k n(t, x; d) \mathbf{d} \mathbf{d} = \sum_{\alpha=1}^N w_\alpha \xi_\alpha^k \quad \text{for } k = 0, 1, \dots, N - 1$$

where the order of approximation N can be chosen arbitrarily high.

Formally, system (2a) can be solved for w_α and ξ_α , $\alpha = 1, \dots, N$ if the coefficient matrix in (2c) is regular, which is the case for distinct abscissae ξ_α and, if needed, can be assured by regularisations as described e.g. in [12].

2.3 Discretisation of the Transport Equations in Space and Time

For the spatial discretisation the same grid is used as in the FVM simulation in the flow solver, i.e. the flow variables like v, ε and Γ_t are directly available at the considered discrete points. For the selected segments of the cylindrical stirrer domain, c.f. Figure 5, that do not include the origin, parameterizing the grid in cylindrical coordinates gives a uniform distribution of the cell nodes in each dimension.

To illustrate the method we consider (2a-b) as special cases of equations

$$\begin{aligned} \frac{\partial \varphi}{\partial t} + \operatorname{div}(v\varphi) - \operatorname{div}(\Gamma \nabla \varphi) &= f(\varphi) \quad \text{in } (0, T] \times \Omega, \\ \varphi|_{t=0} &= \varphi_0 \quad \text{and} \quad \varphi|_{\partial\Omega} = g, \end{aligned} \quad (3)$$

that determine a scalar function φ for a given right-hand side f and parameters v and Γ . Using the formulation of the differentiation operators div and ∇ in cylindrical coordinates, equation (3) is discretised by means of central finite differences for the convection and forward-backward differences for the diffusion term. Thus we obtain the linear system of ordinary differential equations

$$\dot{\varphi}_h + A\varphi_h = f_h(\varphi_h) \quad \text{in } (0, T] \quad \text{and} \quad \varphi_h|_{t=0} = \varphi_{h,0} \quad (4)$$

where matrix A contains the discretised differential operators and the boundary condition g and the vectors φ_h and f_h the approximate discrete values of φ and f , respectively. For the time numerical integration of the spatially discretised convection diffusion equation (4) one can use common numerical schemes for initial value problems.

In our concrete case we will use a combination of implicit and explicit schemes, as for example suggested in [6]. In particular the treatment of the right hand side will be explicit since the inhomogeneity is highly nonlinear and not directly available but defined via an equation system, c.f. Section 2.2.

Thus using, for the sake of illustration, explicit Euler for the $f_h(\varphi_h)$ and implicit Euler for $A_h\varphi_h$ the update of the solution from time level ϕ_h^k to ϕ_h^{k+1} with a stepsize τ^k is obtained by the solution of

$$[I + \tau^k A]\varphi_h^{k+1} = \varphi_h^k + \tau^k f_h(\varphi_h^k). \quad (5)$$

If the trapezoid rule is used for the linear part, one has to solve

$$[I + \frac{\tau^k}{2} A]\varphi_h^{k+1} = [I - \frac{\tau^k}{2} A]\varphi_h^k + \tau^k f_h(\varphi_h^k).$$

Analogue formulations can be derived for all sorts of one-step methods. Since the time step is varying by default, the application of multistep methods is not taken into account. To obtain an approximate solution for the weights and abscissae, equations (2a-b) are brought into the fully discretised form (5) giving the system

$$\begin{aligned} [I + \tau^k A]w_{\alpha,h}^{k+1} &= w_{\alpha,h}^k + \tau^k a_{\alpha,h}(\xi_{1,h}^k, \dots, \xi_{N,h}^k) \\ [I + \tau^k A]w_{\alpha,h}^{k+1} \xi_{\alpha,h}^{k+1} &= w_{\alpha,h}^k \xi_{\alpha,h}^k + \tau^k b_{\alpha,h}(\xi_{1,h}^k, \dots, \xi_{N,h}^k), \end{aligned}$$

$\alpha = 1, \dots, N$, that updates the discretised weights and weighted abscissae. The right hand sides $a_{\alpha,h}$ and $b_{\alpha,h}$, $\alpha = 1, \dots, N$ are vectors containing the values of a_α or b_α at every point of the discretisation. They are obtained by solving the $2N \times 2N$ linear equation system defined by (2c) and the discrete abscissae and flow variables of the previous time step at every node of the discretisation.

2.4 Modelling of the Source Terms

In liquid/liquid dispersions the dominating phenomena influencing the drop size distribution are breakage and coalescence, c.f. [5]. Thus the source term in the PBE (1c) takes the form

$$S = S^{\text{break}} - D^{\text{break}} + S^{\text{coal}} - D^{\text{coal}}.$$

Since the moment transform is a linear operation we obtain for the transformed source terms

$$S_k = S_k^{\text{break}} - D_k^{\text{break}} + S_k^{\text{coal}} - D_k^{\text{coal}}$$

with e.g.

$$S_k^{\text{break}}(x, t) = \int_0^\infty \xi^k S^{\text{break}}(\xi; x, t) d\xi.$$

Again, the transformed quantities do not depend on the internal coordinate ξ but still on space and time. The latter connection can be both explicit and implicit, for example through the variables of the flowfield like the dissipation rate ε . However, for the sake of a clear notation the dependency on time and space of the involved functions is skipped. Taking the drop diameter d as the internal coordinate one can derive general mathematical formulations for the source terms. A detailed description of the derivation and interpretation of the source terms is provided in [17].

The breakage birth term can be written as

$$S^{\text{break}}(d) = \int_d^{d_{\text{max}}} m(d')\beta(d, d')g(d')n(d')dd'$$

where n is the number density function, g is the drop breakage rate, β is the daughter size distribution and m denotes the number of daughter drops arising from the breakage of one mother drop. In case of binary breakage one has $m = 2$.

The associated breakage death term is

$$D^{\text{break}}(d) = g(d)n(d).$$

The source due to coalescence is formulated as

$$S^{\text{coal}}(d) = \int_0^d F(d', d'')n(d')n(d'')dd'$$

with $d'' = d - d'$ and F denoting the drop coalescence rate for two coalescing drops. The same function is used to model the death term due to coalescence to get

$$D^{\text{coal}}(d) = n(d) \int_0^{d_{\text{max}}-d} F(d, d')n(d')dd'.$$

For the numerical treatment all integrals containing the drop size distribution n are approximated by quadrature using the weights w_α and abscissae ξ_α . Also we introduce the cutting function

$$\chi(t, s) = \begin{cases} 0 & \text{if } t \leq s \\ 1 & \text{if } t > s \end{cases} \quad (6)$$

to simplify the mathematical expressions.

Thus, for the breakage source term we get

$$\begin{aligned} S_k^{\text{break}} &= \int_0^{d_{\max}} d^k \int_d^{d_{\max}} m(d')\beta(d, d')g(d')n(d')\mathbf{d}d'\mathbf{d}d \\ &= \int_0^{d_{\max}} d^k \int_0^{d_{\max}} \chi(d', d)m(d')\beta(d, d')g(d')n(d')\mathbf{d}d'\mathbf{d}d \\ &\approx \int_0^{d_{\max}} d^k \left[\sum_{\alpha=1}^N \chi(\xi_\alpha, d)w_\alpha m(\xi_\alpha)\beta(d, \xi_\alpha)g(\xi_\alpha) \right] \mathbf{d}d \\ &\approx \sum_{\alpha=1}^N w_\alpha m(\xi_\alpha)g(\xi_\alpha) \int_0^{d_{\max}} \chi(\xi_\alpha, d)d^k \beta(d, \xi_\alpha) \mathbf{d}d. \end{aligned}$$

Given a specific model for the involved functions it remains to evaluate the integrals

$$b_{k,\alpha} := \int_0^{d_{\max}} \chi(d, \xi_\alpha)d^k \beta(d, \xi_\alpha) \mathbf{d}d.$$

Analogously we obtain for the connected sink term

$$D_k^{\text{break}} \approx \sum_{\alpha=1}^N w_\alpha \xi_\alpha^k g(\xi_\alpha).$$

Using again the cutting function (6) and the relation $d'' = d - d'$ we derive for the coalescence source term

$$\begin{aligned} S_k^{\text{coal}}(d) &= \int_0^{d_{\max}} d^k \int_0^{d_{\max}} \chi(d, d')F(d', d - d')n(d')n(d - d')\mathbf{d}d'\mathbf{d}d \\ &\approx \int_0^{d_{\max}} d^k \left[\sum_{\alpha=1}^N \chi(d, \xi_\alpha)w_\alpha F(\xi_\alpha, d - \xi_\alpha)n(d - \xi_\alpha) \right] \mathbf{d}d \\ &\approx \sum_{\alpha=1}^N w_\alpha \int_0^{d_{\max}} \chi(d, \xi_\alpha)d^k F(\xi_\alpha, d - \xi_\alpha) \sum_{\gamma=1}^N w_\gamma \delta(d - \xi_\alpha - \xi_\gamma) \mathbf{d}d \\ &\approx \sum_{\alpha=1}^N \sum_{\gamma=1}^N \chi(d_{\max}, \xi_\alpha + \xi_\gamma)w_\alpha w_\gamma (\xi_\alpha + \xi_\gamma)^k F(\xi_\alpha, \xi_\gamma), \end{aligned}$$

Note that the cutting function χ has been adapted to restrict the sum $\xi_\alpha + \xi_\gamma$ to the feasible range. The restriction used in the derivation cancels out, since $\chi(\xi_\alpha + \xi_\gamma, \xi_\alpha) \equiv 1$.

Similarly the moment approximation of the corresponding death term is derived:

$$\begin{aligned} D_k^{\text{coal}}(d) &= \int_0^{d_{\max}} d^k n(d) \int_0^{d_{\max}} \chi(d', d - d_{\max})F(d, d')n(d')\mathbf{d}d'\mathbf{d}d \\ &\approx \sum_{\alpha=1}^N \sum_{\gamma=1}^N \chi(\xi_\alpha, d_{\max} - \xi_\gamma)w_\alpha w_\gamma \xi_\gamma^k F(\xi_\gamma, \xi_\alpha). \end{aligned}$$

Besides the approaches considered and described in this work there exist a wide range of models for the breakage and coalescence phenomena in liquid/liquid dispersions, we refer to [5, 17] for references and a detailed discussion of the most common models. The most widely used model is the one proposed by Coualoglou and Tavlarides [3]. Here, as in many other common models breakage is assumed to happen due to collisions of droplets and eddies of the flow field. In particular this models assumes that breakage takes place if the kinetic energy of the eddy is larger than the surface energy of the droplet. Hence, the breakage rate is modelled by

$$g(d) = c_{1,b} \frac{\varepsilon^{\frac{1}{3}}}{(1 + \phi)d^{\frac{2}{3}}} \exp\left(-c_{2,b} \frac{s(1 + \phi)^2}{\rho\varepsilon^{\frac{2}{3}}d^{\frac{5}{3}}}\right).$$

The constants $c_{1,b}, c_{2,b}$ are free parameters, ε, ϕ, s and ρ denote the turbulent dissipation, disperse phase rate, interfacial tension and density, respectively.

The associated coalescence rate by Coualoglou and Tavlarides [3] is

$$F(d', d'') = \frac{c_{1,c}\varepsilon^{\frac{1}{3}}}{(1 + \phi)} (d' + d'')^2 (d'^{\frac{2}{3}} + d''^{\frac{2}{3}})^{\frac{1}{2}} \exp\left(\frac{-c_{2,c}\mu\rho\varepsilon}{s^2(1 + \phi)^3} \left(\frac{d'd''}{d' + d''}\right)^4\right).$$

Besides others, this model contains two free parameter $c_{1,c}, c_{2,c}$ and the dynamic viscosity μ of the continuous phase. The free parameters can be used to fit the model to the given setup. For this investigation the values, listed in Table 1, were taken from [5].

$c_{1,b}$	$6.14 \cdot 10^{-4}$
$c_{2,b}$	$5.7 \cdot 10^{-2}$
$c_{1,c}$	$1.5 \cdot 10^{-4}$
$c_{2,c}$	$2.56 \cdot 10^{12} m^{-2}$

Table 1: Values of the model parameters

It remains to quantify the breakup. In this model we assume that eventually a drop breaks up into two daughter drops of random but normally distributed size with a maximum likelihood of forming to equal sized drops. Thus we take $m(d) = 2$ and, adapting the model by Coualoglou and Tavlarides [3] to the drop diameter and defining the standard deviation $\sigma := \pi d^3/36$,

$$\beta(d, d') = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\frac{\pi^2}{36}(\frac{1}{2}d' - d)^2}{2\sigma^2}\right)$$

to describe the probability that the breakup of a drop of size d' leads to a drop of size d .

2.5 Initial and Boundary Conditions

The abscissae and weights of the quadrature method can be derived from the moments by means of the PD-algorithm, see e.g. [15]. If the distribution is known the computation of the moments is straight forward. Hence, a general approach is to initialize the distribution either arbitrarily for general investigations [13, 24] or problem-specific [17] using measured values if a certain setup is simulated. Another approach is to use arbitrary initial values for the weights and abscissae and to hope for a convergence to meaningful values after some simulation time by physical convection and diffusion [18].

The handling of boundary conditions is not discussed by the authors mentioned above. The simulations described in [17] use a space-independent formulation of the PBE. For their investigations the authors of [20] propose Dirichlet conditions at the inlet and zero gradients elsewhere.

For our investigation, where only a subdomain of the fluid phase is considered, we will define initial and Dirichlet boundary values on the base of measured and computed quantities.

In particular we will use the density function

$$n_b(\xi) := \frac{m_0}{B(p, q)} \frac{\xi^{p-1}(\xi_{max} - \xi)^{q-1}}{(\xi_{max})^{p+q-1}}, \quad (7)$$

with $B(p, q) = \int_0^1 s^{p-1}(1-s)^{q-1}ds$ and problem specific chosen parameters m_0, p and q . Function (7) is the density function of a nonstandard Beta-distribution on $[0, \xi_{max}]$ multiplied with m_0 and comes with the known expressions for the moments (c.f. [10])

$$m_r = m_0 \xi_{max}^r \frac{p^{[r]}}{(p+q)^{[r]}}, \quad \text{for } r = 1, 2, \dots \quad (8)$$

where $y^{[r]} = y(y+1) \cdots (y+r-1)$ is the ascending factorial.

To adjust the distribution to the given setup we formulate it for $\xi = d$ and impose the three requirements

$$\text{a.) } \frac{m_3}{m_2} = d_{32}, \quad \text{b.) } \frac{m_2}{m_0} - \frac{m_1^2}{m_0^2} = \sigma^2 \quad \text{and} \quad \text{c.) } m_3 = \frac{6}{\pi} \phi. \quad (9)$$

to obtain three nonlinear equations for the three parameters m_0, p and q .

Conditions (9a) and (9b) are independent of m_0 and demand that the distribution defined via (7) matches a given Sauter diameter and standard deviation, respectively. The corresponding equations read:

$$d_{32} = \frac{m_3}{m_2} = d_{max} \frac{p^{[3]}(p+q)^{[2]}}{p^{[2]}(p+q)^{[3]}} = d_{max} \frac{p+2}{p+q+2} \quad (10a)$$

and

$$\sigma_m^2 = \frac{m_2}{m_0} - \frac{m_1^2}{m_0^2} = d_{max}^2 \frac{qp}{(p+q)^2(p+q+1)} \quad (10b)$$

Equation (9c) bases on the assumption that the drops are spherical, i.e. that for the volume of the drops holds $V = d^3\pi/6$, and ensures that the distribution fits the predefined volume fraction ϕ of the dispersed and continuous phase. In fact one has

$$\phi = \int_0^{d_{max}} V(d)n_b(d)dd = \int_0^{d_{max}} \frac{\pi}{6} d^3 n_b(d)dd = \frac{\pi}{6} m_3$$

Knowing p and q the third constraint gives

$$m_0 = \frac{6\phi}{\pi d_{max}^3} \frac{(p+q)(p+q+1)(p+q+2)}{p(p+1)(p+2)} \quad (10c)$$

The roots of (10a-b) with the parameters d_{32} and σ^2 can be obtained by means of computer algebra tools. However, one still has to pick out the feasible solutions, i.e. $p, q > 0$ in general and $p, q \geq 1$ in the case of decaying density towards the margins. If there are several feasible solutions one can use the following decision rules to determine the most appropriate choice:

1. $p \approx q$ gives a symmetric distribution
2. $p \gg q$ gives high density values in the upper range of d
3. $p \ll q$ gives high density values in the lower range of d
4. p or $q \approx 1$ gives a high gradient of the density at the corresponding borders
5. $p, q \gg 1$ means a smooth transition towards the borders (bell-shape curve)

In this survey the roots of (10a-b) computed numerically for given d_{32} and σ_m^2 . A feasible solutions is determined by picking out a pair $(\Re(p), \Re(q))$ with $\Re(p), \Re(q) > 1$ and $|\Re(p)/\Im(p)|, |\Re(q)/\Im(q)| > 10^8$.

This procedure is tested against measured data by [5] with satisfactory results depicted in Figure 2. The Beta-distribution is preferable since it can be adjusted to the interval of interest, in contrast to a normal distribution that is defined on the whole real axis.

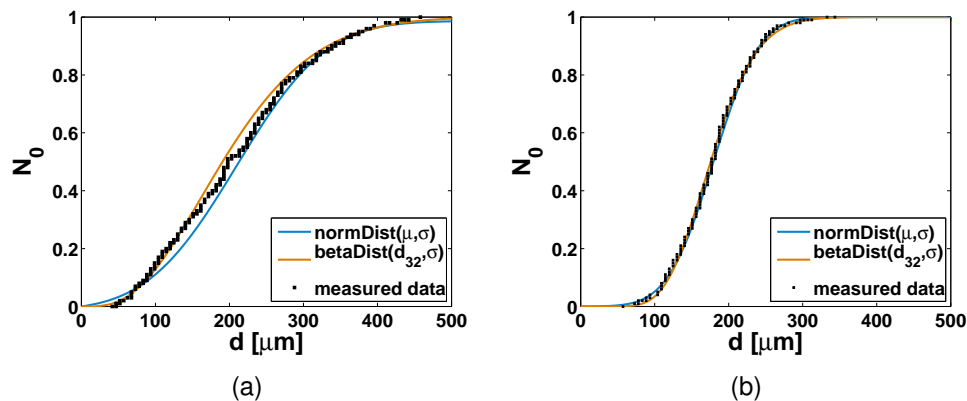


Figure 2: Comparison of the cumulative density distribution of measured drop sizes with the normal and beta distribution, constructed by means of standard deviation σ and mean value μ and standard deviation and Sauter diameter d_{32} , respectively, of the measured data. The setup is described in [5] for the pH 13 case. The measurements are taken after 60 minutes at constant stirrer speed of (a) 400 rpm and (b) 550 rpm. The maximum diameter in the definition of the beta distribution is $1200 \mu\text{m}$.

The choice of d_{max} indeed influences the quality of the approximation. The higher the upper bound the less it disturbs the distribution in the range of interest. However in this case that the distribution concentrates in the lower range a large d_{max} leads to very small values of $B(p, q)$ which is numerically problematic. Therefore the used algorithm suggests adapt the upper bound if $B(p, q)$ gets close to machine precision.

Having determined the parameters of the density function (7) one can compute the moments up to an arbitrary order using expression (8). Then the corresponding initial

weights and abscissae can be computed using the PD-algorithm described e.g. in [15, 17].

We will use starting and boundary values that change in time but which are independent of the spatial variables. An extension to multivariate, e.g. for inlet and outlet, or smoothly varying in space boundary values is straight forward.

3 The FASTEST 3D/MATLAB Interface

The coupling between the flow solver FASTEST 3D and the MATLAB control unit is realised in three steps:

- Initialisation of a new control setup out of MATLAB.
- Starting the MATLAB control unit out of FASTEST 3D based on the data at the current time step (e.g. Sauter diameter, velocity, turbulent energy).
- The control unit sets the input data (stirrer speed) for the next time step.

Thus, a single input/multiple output control system is designed which is illustrated in Figure 3.

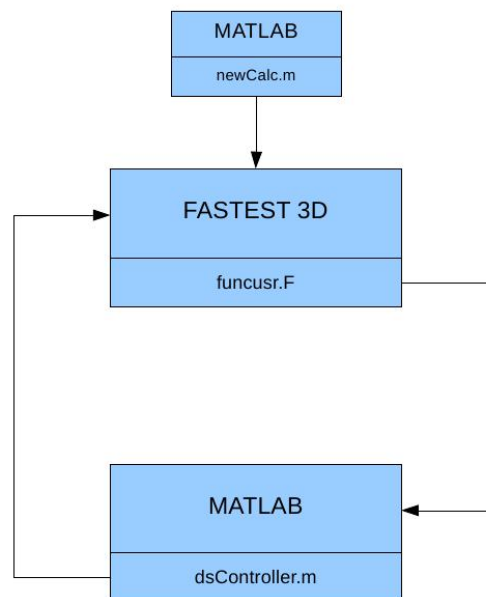


Figure 3: Design of the FASTEST 3D/MATLAB interface

3.1 Initialisation

A new calculation of FASTEST 3D can be started out of MATLAB using a script, let's say `newCalc.m`. At this point all initial values for a new flow computation are set. Since the mesh is blockstructured, the user can choose the number of the block to define the region which should be controlled. Also all input parameters, which are set in the id-file of FASTEST-3D like for instance the rotational speed of the stirrer ω^0 in first time

step, are defined at this stage. Additionally the initial values for the Sauter diameter d_{32}^0 and standard deviation σ^0 are saved in `.mat` format since MATLAB is closed afterwards. As a last step the user decides between an *open loop*-calculation and a *closed loop*-calculation. In the case of *open loop*-calculation the value for the rotational speed at every time step is set by a special subfunction. In the other case the value computed by means of control algorithms at every time step.

3.2 Flow Solver

During the computation the basic functionality of FasCon is the initiation of a MATLAB programme, i.e. `dsController.m`, to compute a future control input by means of actual data, which is then fed back to FASTEST 3D for the next time step. The data is provided by FasCon in ASCII format as well as the current time step. The control algorithm is specified and implemented within `dsController.m` and related subfunctions.

FasCon is a Fortran routine located in the source code function `funcusr.F` and executed before every time step or at a specified position in the multigrid cycle of the current time step. The parameters in FasCon are listed in Table 2 and have to be set in accordance to the control method defined in `dsController.m`. For simple PID and other low level controllers toggles are foreseen that allow the complete problem definition in FasCon. The call of MATLAB routines out of Fortran is realised by the help of the MATLAB Engine (function name `engOpen`), which operates by running in the background as an independent process. The following code extract shows, how the MATLAB function `dsController.m` is executed out of Fortran code:

```
integer*8 engOpen
integer status
logical control

control = .TRUE.

if (control) then
  ep = engOpen('matlab ')

  if (ep .eq. 0) then
    write(6,*) 'Can''t start MATLAB engine'
    stop
  endif

  if(engEvalString(ep,'cd ~/.dsController').ne.0)then
    write(6,*) 'engEvalString failed'
    stop
  endif

  status = engClose(ep)
endif
```

Parameter	Values	Description
blockNo*	1,2,...,#Blocks	Specifies the region for the output according to the blockstructure of the given grid
modControl	1,2,...,#time steps	Frequency of the Control unit (modControl=1 means the Control unit is executed at every time step)
dataVar*	$u, v, w, p, k, e, m_0, m_1, \dots$	Variable values for the output
timePoints	0,1,2,3,...	Additional datasets to the current time step
intermediate	0,1	Toggle for an intermediate output
VCycNo	1,2,..., #MG iterations	Specifies the inner iteration for data extraction

Table 2: Parameters for the output definition. The * indicates possible multiple definitions

3.3 Control Unit in MATLAB

After calling the control unit in MATLAB the time step and the FASTEST output are read from ASCII files by the MATLAB routine `dsController.m`. A special flag in the MATLAB code allows the usage of different control algorithms. The result of the control is a new rotational speed of the stirrer which is fed back to FASTEST 3D via an input file for the next time step.

4 Optimal Control of the Dispersion

The control is designed to consider only the standard deviation and the Sauter diameter as a measure for the average drop diameter instead of the whole PSD. These quantities are of main importance in practice and are investigated also in the experimental setup, c.f. [5]. The Sauter diameter and the standard deviation,

$$d_{32} = \frac{m_3}{m_2} \quad \text{and} \quad \sigma^2 = \frac{m_2}{m_0} - \frac{m_1^2}{m_0^2},$$

respectively, can be derived directly from the first four moments. The input for the control system is given by the stirrer speed ω .

Thus, a general form for the cost functional of the control reads

$$\mathcal{J}_{(\alpha,\beta)}(d_{32}, \sigma^2, \omega) = |||d_{32}(\omega) - d_{32}^*||| + \alpha||\sigma^2(\omega)|| + \beta|\omega|.$$

Here d_{32}^* denotes the aspired state and α, β are weight parameters. Minimizing \mathcal{J} , subject to the system dynamics given e.g. by equations (1a)-(1c) is equivalent to driving the system to an aspired state while minimizing the standard deviation of the PSD and the input. The norms used have to be specified with respect to the specific control problem. Since it is too costly to solve the control problem for the full model, one has to call on model reduction techniques. A possible approach is to drop the spatial dependencies of d_{32} and σ by using the average value and approximate the dynamics by a continuous linear state space model

$$\dot{x}(t) = Ax(t) + B\omega(t) \tag{11a}$$

$$\begin{bmatrix} d_{32}(t) \\ \sigma(t) \end{bmatrix} = Cx(t) + D\omega(t) \tag{11b}$$

- $D = 0.15m$ Diameter of the tank
- $d = 0.05m$ Diameter of the stirrer
- $H = 0.1m$ Height of the tank
- $h = 0.05m$ Height of the Stirrer

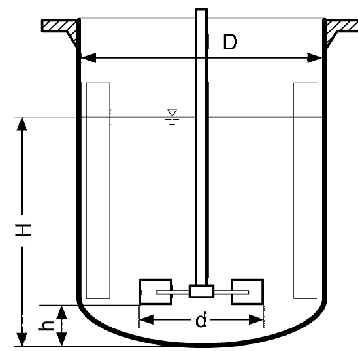


Figure 4: Geometrical parameters

with t ranging in the simulation interval. The problem specific matrices A, B, C, D can be obtained by identification techniques as described e.g. in [11]. Note that the state $x(t)$ is a purely artificial construct by the identification algorithm.

Given an identified model (11) common algorithm for optimal and robust feedback control apply, c.f. [21, 25].

5 Implementation of a Control Design

The presented control design aims at the control of the PSD via an identified model. Thereto several simulations with varying stirrer velocity as the input were run to generate a set data and to identify a linear state space model. Then controllers and regulators can be computed for the identified model. Depending on the quality of the model, the identified controller may then also perform in the full-scale simulation.

5.1 Setup

For the simulations of a liquid/liquid dispersion a Rushton turbine filled by 90% water and 10% toluene was chosen. The main parameters of the geometrical dimensions of the setup are summarised in Figure 4.

Furthermore, the stirrer system contains 6 blades and the the tank is equipped with 4 baffles. The computations were carried out for a half model of the stirrer, applying periodical boundary conditions at the symmetry plane and using a blockstructured hex-aedron finite volumes grid with about 1.6 million cells.

The physical properties of the involved fluids are depicted in Table 3

	ρ	ν	ϕ	σ
Water	$1000 \frac{kg}{m^3}$	$10^{-6} \frac{m^2}{s}$	0.9	$32 \cdot 10^{-3} \frac{N}{m}$
Toluene	$866 \frac{kg}{m^3}$	$0.63 \cdot 10^{-6} \frac{m^2}{s}$	0.1	

Table 3: Physical properties of water and toluene

The basic stirring regime was set to a stirrer speed of $\omega_0 := 25 \frac{rad}{s}$ which corresponds to approximately 239 revolutions per minute. The Reynolds number $Re \sim 10^5$ is calculated using the stirrer diameter, the velocity of the stirrer tip for $\omega_0 = 25 \frac{rad}{s} = 3.98$ rounds per second and the viscosity of the water.

We gave 1 second simulation time to accelerate the stirrer from the zero state to ω_0 and then started computing the Sauter diameter and the variance of the dispersion while varying the impeller speed ω smoothly by $\pm 20\%$.

The PBE is solved using the DQMOM, i.e. equations (5) are numerically integrated to obtain weights and abscissae where the moments can be extracted from. To save computational time only a segment of the tank, covering the space between two blades, was considered for the PBE, c.f. Figure 5.

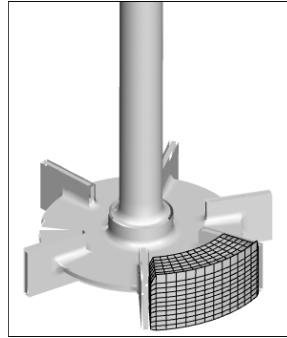


Figure 5: Illustration of the domain where the PBE is considered

To start the calculation one has to provide initial values for the weights w_α and abscissae ξ_α , $\alpha = 1, \dots, N$. Thereto we compute the first $2N$ moments m_0, \dots, m_{2N-1} of a Beta-distribution fitting a prescribed Sauter diameter $d_{32,0}$, standard deviation σ_0 and volume fraction ϕ , as described in section 2.5. The initial values for the w_α and ξ_α are then conceived from the moments by means of the PD-Algorithm, see e.g. [15].

The initial values mark the initial state of the simulated PSD can be set arbitrarily, in particular because the system is far away from the stationary state, which is expected after several minutes, depending on the models, c.f. [5]. Extrapolating experimental findings by [5] we set the initial values as shown in Table 4.

$d_{32,0} = 505\mu m$	Initial Sauter diameter
$\sigma_0 = 102\mu m$	Initial standard deviation
$\phi = 0.1$	Dispersed volume fraction

Table 4: Initial values for the simulation

According to [17] the main flow in the considered region is in radial direction. Thus we set uniform Dirichlet conditions at the boundary facing the center and zero gradient conditions elsewhere. The Dirichlet values for the weights and abscissae was set to be the initial values or the average values of the preceeding time step.

In this investigation the number of quadrature points was set to $N = 2$. Having computed w_1, w_2, ξ_1, ξ_2 for the current time step the actual moments, the Sauter diameter and deviation of the PSD are obtained by the algebraical relations

$$m_k = \sum_{\alpha=1}^2 w_\alpha \xi_\alpha^k, \quad d_{32} = \frac{m_3}{m_2} \quad \text{and} \quad \sigma^2 = \frac{m_2}{m_0} - \frac{m_1^2}{m_0^2},$$

respectively.

5.2 Data Generation and Identification of a State Space Model

Following the general approach of defining a set of test (input) functions and computing and recording their response, the FASTEST-3D/MATLAB coupling described in Section 3 can also be used to design a work cycle for data generation for identification.

Thereto in every time step the MATLAB routine `dsController.m` is set up to simply store the output in a `.mat`-file and return the actual value of the testfunction to the solver as the new stirrerspeed rather than running a control algorithm.

For our tests we chose a set of sine functions that are periodic on the interval $[0, 0.3]$ with varying wave number and a fixed amplitude of magnitude 5. To use the functions directly as the solver input they were shifted by ω_0 . Thus testfunction ω^k , $k = 0, 1, 2, 3, \dots$, reads

$$\omega^k = 5 \cdot \sin\left(\frac{k\pi t}{0.3}\right) + 25, \quad t \in [0, 0.1].$$

For the simulation ω^k is evaluated at the current discrete time level. The first 4 testfunctions are illustrated in Figure 6.

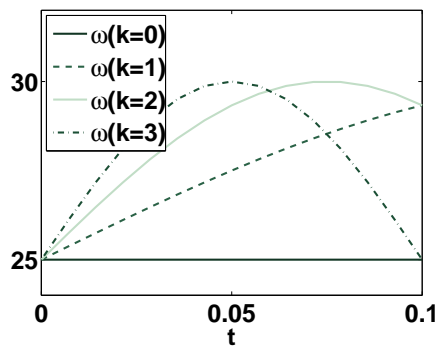


Figure 6: Illustration of the first 4 test functions

Running the simulation for the first 9 testfunctions ω^k , we obtained vectors $(\omega^k, d_{32}^k, \sigma^k)$, containing the values of ω^k , and their responses d_{32}^k and σ^k at all time steps the solve took to resolve the simulation interval.

Because of the clicking mesh technique, the time step size is restricted to discrete points, depending on the actual velocity and the number of cells the stirrer is moving, c.f. Section 2.1. As a result the values of the time steps in the described series differs from testfunction to testfunction. This issue was solved by linear interpolation of the data pairs to equidistant points at the interval $[0, t_{end}]$ in order to make the data suitable for discrete time identification.

The obtained data sets were imported to the MATLAB *System Identification Toolbox* [23] to describe a time-discrete linear time-invariant dynamical system of the form

$$\begin{aligned} x_{n+1} &= Ax_n + B\omega_n \\ \begin{bmatrix} d_{32,n} \\ \sigma_n \end{bmatrix} &= Cx_n + D\omega_n \end{aligned}$$

with the system output $= [d_{32}, \sigma]^T$ and the input data ω . The basic idea of identification is to fit a linear time invariant model to given input/output data, and use the model instead of the full scale control design depicted in Figure 3.

The MATLAB *Identification Toolbox* provides a vast variety of algorithms for identification. We used the `idss` algorithm to extract the matrices $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,1}$, $C \in \mathbb{R}^{2,n}$, $D \in \mathbb{R}^{2,1}$ which can be used for optimal control. Here n denotes the dimension of the artificial state x .

5.3 LQR State Feedback Control

Considering the identified time discrete system and an aspired sauter diameter sampled at discrete points, the optimisation problem may be formulated as: Minimize

$$\mathcal{J} = \sum_n (d_{32,n} - d_{32,n}^*)^2 + \alpha \sigma_n^2 + \beta \omega_n^2 \quad (12a)$$

subject to

$$x_{n+1} = Ax_n + B\omega_n \quad (12b)$$

$$\begin{bmatrix} d_{32,n} \\ \sigma_n \end{bmatrix} = Cx_n + D\omega_n. \quad (12c)$$

For this control system one can call on the theory for linear quadratic regulation, [25]. Thus, having computed by the MATLAB function `dlqry.m` the regulator K that minimizes

$$\mathcal{J}_0 = \sum_n \begin{bmatrix} d_{32,n} & \sigma_n \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} d_{32,n} \\ \sigma_n \end{bmatrix} + \beta \omega_n^2 \rightarrow \min,$$

the optimal control ω_n^* for (12) is given via

$$\omega_n^* = -K(x_n - \bar{x}_n) + \bar{\omega}_n.$$

Here x_n is the state of the closed-loop system

$$\begin{aligned} x_{n+1} &= (A - BK)x_n + B(K\bar{x}_n + \bar{\omega}_n) \\ \begin{bmatrix} d_{32,n} \\ \sigma_n \end{bmatrix} &= (C - DK)x_n + D(K\bar{x}_n + \bar{\omega}_n). \end{aligned}$$

and for the quantities \bar{x}_n and $\bar{\omega}_n$ one has to solve the equations

$$A\bar{x}_n + B\bar{\omega}_n = 0 \quad \text{and} \quad \begin{bmatrix} d_{32,n}^* \\ 0 \end{bmatrix} = C\bar{x}_n + D\bar{\omega}_n. \quad (13)$$

Note that in the given case of one input and two outputs, equations (13) have a solution only for specific choices of the reference outputs.

5.4 Discussion

As illustrated in Figure 7 the test functions showed no detectable response in the chosen simulation interval. For both the Sauter diameter and the standard deviation there are no tendencies to observe and the slight changes are indefinite and possibly caused by numerical errors.

A reason for this outcome may be fact that after only one second of stirring up the resting medium, the flow in the stirrer is still developing. Hence unsteady effects of the flow field superimpose the impact of the control. Thus the identified model holds to little information on the system and the computed LQR controller had no influence on the PSD.

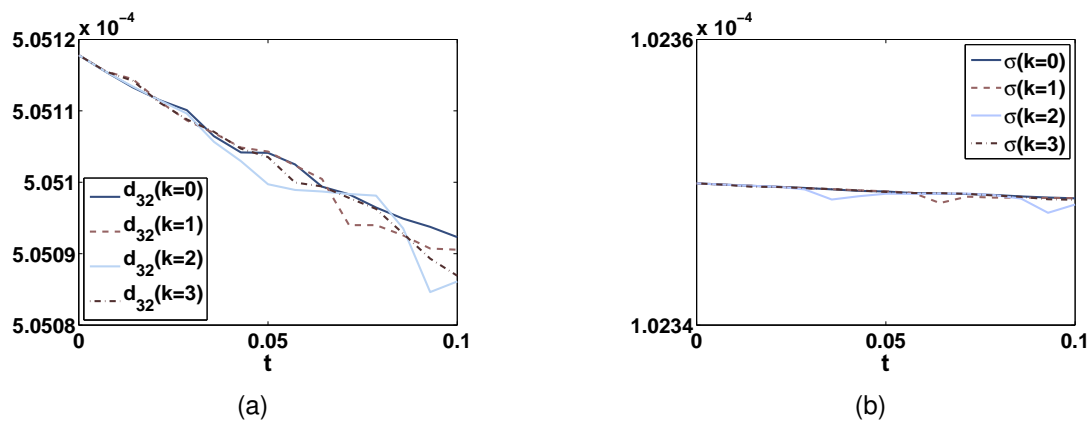


Figure 7: Illustration of the response of the first 4 test functions defined in Section 5.2

6 Conclusion and Outlook

The key result of this work is the efficient coupling of the flow solver FASTEST 3D with MATLAB. In the first place this makes the multiple control algorithms of the MATLAB toolboxes available for simulation of control designs including the flow solver.

In addition, the coupling can be used for the design of numerical experiments. As presented for the identification example one can use the functionality of MATLAB to define a series of simulations and use the interface to steer the solver and to manage the data. The interface is also suitable for postprocessing of the simulation results. For example the visualisation of the flow variables, using the open source tool PARAVIEW [16], was successfully implemented within this investigation.

The simulations of the proposed control design showed the need for further analysis of the problem. Also the algorithm implemented for the solution of the PBE proved to be numerically problematic because of the small values of the abscissae. A nondimensional formulation as discussed e.g. in [20] may put the things right.

Regarding the control a next step is the synthesis of suitable and robust controllers, both from measured data and from the underlying mathematical model.

Nomenclature

Symbol	Description	Symbol	Description
A, B, C, D	State space matrices	w	Weight
$c_{1,b}, c_{2,b}$	Model parameters	α, β	Weighting constants
$c_{1,c}, c_{2,c}$	Model parameters	Γ_t	Turbulent diffusion
d	Particle diameter	ε	Turbulent dissipation rate
d_{32}	Sauter diameter	ζ	Weighted abscissa
k	Turb. kinetic energy	μ	Dynamic viscosity
m	Number of daughter drops	ν	Kinematic viscosity
m_k	k-th moment	ξ	Abscissa
n	Number density function	ρ	Density
n_b	Approximation by β -distr.	σ	Standard derivation
N	#Nodes for quadrature	ϕ	Volume fraction
Re	Reynolds number	ω	Stirrer speed
s	Interfacial tension	CFD	Computational fluid dynamics
S	Source term of PBE	FVM	Finite volume method
t	Time	ILU	Incomplete lower-upper
T^*	Turbulent stress tensor	LQR	Linear-quadratic regulator
v	Velocity	PBE	Population balance equation
		PSD	Particle size distribution

References

- [1] V. Alopaeus, J. Koskinen & K.I. Keskinen, *Chem. Eng. Sci.* **54** (1999) 5887-5899.
- [2] A.E. Bouaswaig & S. Engell, *Ind. Eng. Chem. Res.* **49** (2010) 5911-5924.
- [3] C.A. Coulaloglou & L.L. Tavlarides, *Chem. Eng. Sci.* **32** (1977) 1289-1297.
- [4] FASTEST - User Manual, *FNB, TU Darmstadt* (2006).
- [5] A. Gäbler, M. Wegener, A.R. Paschedag & M. Kraume, *Chem. Eng. Sci.* **61** (2006) 3018-3024.
- [6] P.M. Gresho & S.T. Chan, *Int. J. Numer. Methods Fluids* **11** (1990) 621-659.
- [7] Z. Harth, Automated Numerical Shape Optimization of 3-dimensional Flow Geometry Configurations *PhD Thesis, TU Darmstadt* (2008).
- [8] L. Henning, D. Kuzmin, V. Mehrmann, M. Schmidt, A. Sokolov & S. Turek, Flow control on the basis of a FEATFLOW-MATLAB coupling. In R. King (ed.), *Active Flow Control* (2006).
- [9] H. Hulburt & S. Katz, *Chem. Eng. Sci.* **19** (1964) 555-574.
- [10] N.L. Johnson, S. Kotz & N. Balakrishnan, Continuous univariate distributions. Vol. 2. 2nd ed. *New York, NY: Wiley* (1995).
- [11] L. Ljung, System identification: theory for the user. *Prentice-Hall* (1987).
- [12] D.L. Marchisio & R.O. Fox, *J. Aerosol Sci.* **36** (2005) 43-73.
- [13] D.L. Marchisio, J.T. Pikturna, R.O. Fox, R.D. Vigil & A.A. Barresi, *AIChE Journal* **49** (2003) 1266-1276.
- [14] D.L. Marchisio, R.D. Vigil & R.O. Fox, *Chem. Eng. Sci.* **58** (2003) 3337-3351.
- [15] D.L. Marchisio, Precipitation in Turbulent Fluids *PhD thesis, Polytechnical University of Turin* (2001).
- [16] ParaView Version 3.6.1, www.paraview.org
- [17] S. Schlauch, Modeling of stirred liquid-liquid dispersions. *PhD Thesis, TU Berlin* (2007).
- [18] B. Selma, P. Proulx & R. Bannari, *Chem. Eng. Sci.* **65** (2010) 1925-1941.
- [19] R. Sieber, Numerische Simulation technischer Strömungen mit Fluid-Struktur-Kopplung *PhD Thesis, TU Darmstadt* (2002).
- [20] L.F.L.R. Silva, R.B. Damian & P.L.C. Lage, *Comput. Chem. Eng.* **32** (2008) 2933-2945.
- [21] S. Skogestad & I. Postlethwaite, Multivariable feedback control. Analysis and design. *Chichester: Wiley* (1997).
- [22] M. Srour, V.G. Gomes, I. Altarawneh & J. Romagnoli, *Chem. Eng. Sci.* **64** (2009) 2076-2087.
- [23] System Identification Toolbox 7 - User's Guide, *The MathWorks* (March 2010).
- [24] D.L. Wright, R. McGraw & D.E. Rosner, *J. Colloid Interface Sci.* **236** (2001) 242-251.
- [25] K. Zhou, J.C. Doyle & K. Glover, Robust and optimal control. *Prentice Hall* (1996).