Comparison between compartment and CFD models for simulating reactive crystallization processes

Andrea Querio,[†] Mohsen Shiea,[‡] Antonio Buffo,[†] and Daniele Luca Marchisio[*,†]

†*Dipartimento di Scienza Applicata e Tecnologia, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy*

‡*IFP Energies Nouvelles, Rond-point de l'échangeur de Solaize, BP 3, 69360 Solaize, France*

E-mail: daniele.marchisio@polito.it

Abstract

This work compares two different computational approaches aimed at describing the reactive crystallization or precipitation process in stirred tanks. The first approach is a full CFD model coupled with Population Balance Modeling, which is accelerated by the operator-splitting method and the hybrid MPI-OpenMP parallelization. Here emphasis is given to the hybrid MPI-OpenMP parallelization that improves parallel scalability, when the operator-splitting method is used to take relatively large time steps, despite of the large separation of time-scales in such processes. The second approach is a compartment model enhanced by an automatic tool for the generation of compartments based on some relevant features of the system. The two models are compared for a case study of particular interest: the reactive co-precipitation of Ni-Mn-Co hydroxide in a continuous stirred tank, main precursor to produce cathode active materials of lithium-ion batteries. The obtained results demonstrate the effectiveness

of the hybrid parallelization to improve the parallel scalability of the CFD model. In addition, it is shown that the compartment model can produce less accurate, but still relevant, predictions with relatively small computational cost.

1 Introduction

Several chemical engineering processes consist of phenomena that occur and are defined by characteristic time- and length-scales that differ several orders of magnitude. In this regard, of particular interest are reactive crystallization or precipitation processes, such as the production of active material precursor of Li-ion battery cathodes[1,2] or the recovery of critical raw materials.[3,4] These processes are often characterized by fast chemical reactions, which occur almost instantaneously, followed by fast nucleation, moderately fast molecular growth and aggregation, and slow breakage. These phenomena occur while the multiphase fluid dynamics of the system, as well as the residence time of the fluid and of the formed particles/crystals inside the reactor are characterized by larger time-scales. Additional issues arise because of the localized nature of the phenomena involved: most of the particles/crystals are nucleated in small portions of the reactor, whereas in others no particulate processes occur. Several examples can be found in the literature, namely nickel-manganese-cobalt hydroxides co-precipitation,[1,2] barium sulphate precipitation,[5–8] magnesium hydroxide precipitation,[3,4] titanium dioxide synthesis,[9–11] molybdenum sulfide precipitation,[12] but also organic compounds like adipic acid,[13] different type of drugs[14,15] or fuels[16] and many others.

In this work we focus on an industrially relevant process as case study for the implemented computational tools, that will be introduced later, namely the co-precipitation of nickel, manganese and cobalt hydroxide (NMC) in continuous stirred tanks. Reactive crystallization, precipitation and co-precipitation will be used in what follows interchangeably, as they all refer to a multiphase solid-liquid system, with solid crystalline particles evolving. This process has been extensively investigated experimentally.[17–22] The developments in the study of NMC are related to their use as precursor of cathode active material (pCAM)

for Li-ion batteries (LiB), which are growing exponentially in diverse applications, ranging from electric vehicles to stationary energy storage solutions, thanks to their long lifetime, safety and high energy density. Considering that LiBs play a key role in the transition to sustainable energy sources, it is crucial to develop computational tools capable of identifying the operating conditions that lead to the production of precursors, i.e. pCAM, with desired properties, that in turn result into batteries with excellent performances.

Nowadays, Computational Fluid Dynamics (CFD) is world-wide recognized as an accurate approach for process design, scale-up and optimization, allowing to reduce the extent of experimental campaigns, that are often time consuming and resource intensive. For reactive crystallization or precipitation processes, CFD is coupled with a Population Balance Model (PBM) to properly describe the evolution of solid crystals or particles due to nucleation, molecular growth, aggregation or breakage, getting important insights about crystallization kinetics to facilitate the optimization of operating conditions or the reactor design.[4,23–25] The model based on CFD and PBM presented in this study was developed and experimentally validated in previous works.[1,2] The model is therefore ready to be employed for the description of industrial scale equipment, usually large continuous stirred tanks characterized by long residence times. However, long residence times pose a challenge to the use of the model, since there is a significant jump of time and spatial scales involved by different physical phenomena occurring in the process. Micromixing and chemical reactions are fast: the former one describe the mixing at the Batchelor scale,[26] which is of the order of 1 ms, while the chemical reactions can be considered instantaneous. On the other hand, the dynamics of particle evolution is slow, in the order of hours,[22] especially particle growth, aggregation and breakage which are the main phenomenon influencing the final particle size distribution. These challenges increase the computational load, resulting in high computational costs that often require an unfeasible wall time to be completed. Additional burden comes from necessary features to be included in the simulation framework of reactive crystallization processes, such as chemical equilibria and activity coefficients calculations. For this purpose, it is cru-

cial to implement and employ strategies to reduce the simulation time, namely operator splitting, hybrid parallelization and compartment modeling, which have been investigated in this work, the first two as optional tools for the CFD-PBM approach, while the latter as a standalone method.

The first approach implemented is operator splitting, commonly employed to deal with reactive transport problems with large time-scale separation.[27–30] This method involves decoupling the different terms of the governing equations, namely slow physical transport and fast chemical reactions, in order to use different numerical scheme for their integration, resulting in faster and more robust numerical solution schemes.

Another interesting tool is hybrid parallelization, employing MPI (Message Passing Interface) tasks and OpenMP (Open Multi-Processing) threads[31,32] which can help in dealing efficiently with local processes characterized by large gradients in the computational domain,[33–35] where different phenomena may have locally a different impact on the computational load of the simulation. This particular approach will be compared with standard parallelization, the most used one in CFD, highlighting the increase of the computational efficiency and also the improvement of the load balance between the parallel processes.

Operator splitting and hybrid parallelization may speed up significantly a CFD simulation, but in some cases this is not enough, as in the example described in this work, in which a very large computational time (several weeks) is required to perform a single simulation. In these cases a different approach all together must be used, namely the compartment model (CM). It is based on a simple idea: the reactor and computational domain is divided in several completely mixed compartments that exchange fluid elements and particles/crystals between them. In this way, a cheaper albeit less accurate description of the hydrodynamics of the system is obtained, allowing a faster response given by all the additional models for the other significant quantities, such as the chemical species and the particles/crystals characteristics. Moreover, it is noteworthy to mention that other methods to create faster descriptions of the hydrodynamics can be found in the literature, such as the recurrence

Computational Fluid Dynamics (rCFD) developed by Lichtenegger and Piker.[36]

Previous works investigated the CM approach to describe precipitation and crystallization processes.[37,38] However, the novelty of our work relies on the compartmentalization of the reactor. In fact, in the aforementioned works, the compartmentalization of the reactors was done ad-hoc for specific reactor geometries and reactions, which means that the same procedure cannot be universally applied to other configurations or reactions. In other examples,[39–41] an automatic compartmentalization based on flow field and turbulence properties of the reactor is proposed. However, in this case, there is the risk of dividing the reactor into zones without taking into account other important phenomena, such as the mixing at the molecular scale, or micromixing, which takes into consideration the molecular segregation of the feed solutions. For this reason here we propose and implement an automatic tool to compartmentalize the reactor relying on the turbulence and mixing properties of the system, which are two factors that play a key role in the co-precipitation processes. The automatic tool mentioned and the CM solver are written in python. Both the CFD-PBM solver and the CM are publicly available and can be found on our group repository.[42]

In addition it is worth remarking that the computational tools here implemented and investigated can be used to describe systems other than the co-precipitation of Nickel, Manganese and Cobalt hydroxide, with minimal changes. A practical example is the work performed by Raponi et al.,[3,4] which simulate the precipitation of $Mg(OH)_2$ in a different type of micromixers, employing a full CFD-PBM approach very similar to the one used in this work.

In the next section we will provide a brief introduction to the model employed to describe the case study mentioned before, i.e., the co-precipitation of Ni-Mn-Co hydroxide. Then we will focus on the description of the computational tools used to optimize the computational costs and we will also focus on the comparison between the full CFD-PBM and the CM approaches. Eventually, in the last two sections we will discuss the results obtained and the conclusions that can be drawn.

2 Model description

Here the model is described for the co-precipitation of Ni-Mn-Co (NMC) hydroxide, however, it can be potentially applied to any reactive crystallization process. The co-precipitation is performed by contacting a mixture of metal sulphates and sodium hydroxide in an aqueous solution, which react together to form particles of $Ni_{1-x-y}Mn_xCo_y(OH)_2$. In this work, we consider the NMC811 chemistry corresponding to $x = 0.1$ and $y = 0.1$. This process is conducted under the presence of ammonia, which acts as a chelating agent to promote the production of dense spherical NMC hydroxide particles.[43] After the reactants get in contact, several chemical equilibrium reactions occur,[44] which generate a supersaturated solution, i.e., a solution with a concentration of NMC hydroxide higher than its solubility product. The formation of particles take place by the nucleation and growth, which are triggered by the generated supersaturation.

The model tracks the evolution in time and space of the chemical species, coupling them with a population balance model (PBM), which is used to describe the formation of particles and the evolution of the Particle Size Distribution (PSD) due to the nucleation, molecular growth, aggregation and breakage.[45] Moreover, considering that reactants are injected from different inlets, segregation at the molecular scale can occur. Therefore, a multi-environment micromixing model is implemented[46] to include the effect of this segregation. In Fig. 1 are shown the calculation steps performed by the solver to simulate the co-precipitation process. In the next subsections, we will provide a brief introduction to the model, a more detailed description of the model can be found in the Supplementary Material.

2.1 Micromixing model

As previously mentioned, due to the separate injection of the reactants and considering the very fast reactions, the segregation of the feed streams at the molecular scale can happen, which in turn causes a slowdown of the co-precipitation reaction. To account for this effect, a

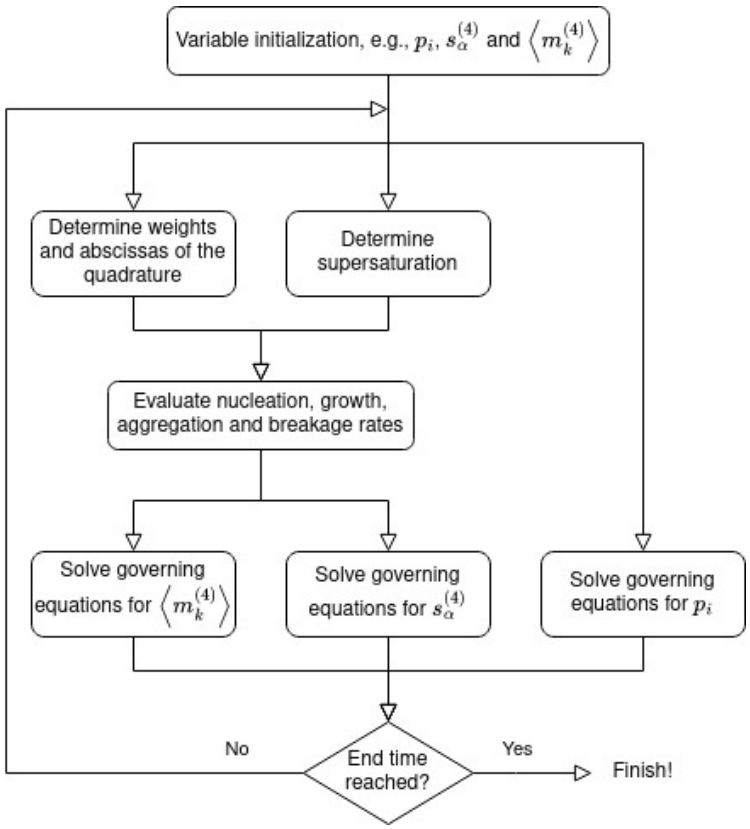


Figure 1: Flow diagram of the simulation steps performed by the model, where p_i, $s_\alpha^{(4)}$ and $\langle m_k^{(4)} \rangle$ are respectively described in Eq. (1), Eq. (2) and Eq. (3).

multi-environment micromixing model is employed. This model considers some pure environments representing feed solutions that co-exist in the entire domain with certain probabilities — or in other words, volume fractions. The pure environments mix together with a certain rate, i.e., micromixing rate, to create mixed environments where the reaction can take place between reactants. The model tracks the probability of the environments by solving transport equations that describe the convective and diffusive transport of the environments and their disappearance/appearance due to micromixing. Moreover, it is necessary to follow the evolution of species concentrations in the mixed environment, which are subject to vary as different environments mix together. Instead, the concentrations of pure environments are constant and equal to those of feed solutions The multi-environment micromixing model and the importance of mixing in reactive precipitation processes, have already been studied in previous works.[5,6,8,47–49]

This work assumes the simplest arrangement for a system of three feed streams. There are three pure environments corresponding to the three feed solutions, and one mixed environment formed by the mixing of the pure environments. The co-precipitation reaction is allowed to take place only in the mixed environment, called here as the reacting environment. The probability of the pure environments are governed by the following transport equations:[46]

$$\frac{\partial p_i}{\partial t} + \mathbf{u}_\mathrm{f} \cdot \frac{\partial p_i}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} \cdot \left(\Gamma_\mathrm{t} \frac{\partial p_i}{\partial \mathbf{x}} \right) - r_i \quad \text{for} \quad \mathrm{i} \in \{1, 2, 3\}, \tag{1}$$

where p_i is the volume fraction of the i-th environment, $\mathbf{u}_\mathrm{f}$ is the velocity of the fluid, Γ_t is the turbulent diffusivity and r_i is the micromixing rate, i.e., the flux from the i-th environment to the precipitation environment due to the micromixing, which depends on the probability of the i-th environment and the turbulent properties of the fluid, i.e., the turbulent kinetic energy k and the turbulent dissipation rate ε. Moreover, there is no need to solve a transport equation for the reacting environment since the sum of probabilities is equal to unity, so the probability of the reacting environment is $p_4 = 1 - \sum_{i=1}^{3} p_i$. Further details can be found in the Supplementary Material.

2.2 Chemical equilibria

As mentioned before, the driving force of the co-precipitation process is the supersaturation, which depends on the concentration of the species involved in the generation of the NMC hydroxide, i.e., metal and hydroxide ions, in the reacting environment. In order to determine the supersaturation, it is necessary to solve for all the chemical equilibria that occur. All the reactions involved in the process and their equilibrium constants can be found in the work of Van Bommel and Dahn.[44]

In order to solve the chemical equilibria, the transport equations for the concentrations

of nickel, manganese, cobalt, ammonia, sodium and sulfate are solved:

$$\frac{\partial s_{\alpha}^{(4)}}{\partial t} + \mathbf{u}_{\mathrm{f}} \cdot \frac{\partial s_{\alpha}^{(4)}}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} \cdot \left(D_{\mathrm{eff}} \frac{\partial s_{\alpha}^{(4)}}{\partial \mathbf{x}} \right) + \sum_{i=1}^{3} r_i c_{T,\alpha}^{(i)} - Q_{\alpha}^{(4)}, \tag{2}$$

where $s_{\alpha}^{(4)} = p_4 c_{T,\alpha}^{(4)}$ is the volume-weighted total concentrations of the species α, $c_{T,\alpha}^{(i)}$ is the total concentration of the species α in the i-th environment, D_{eff} is the effective diffusivity which takes into account both the molecular and turbulent diffusivities and $\sum_{i=1}^{3} r_i c_{T,\alpha}^{(i)}$ is the flux of species α from the feed environments to the reacting one. $Q_{\alpha}^{(4)}$ is the sink term, representing the consumption of nickel, manganese and cobalt due to the co-precipitation, and it is calculated from the volume of generated particles. It is noteworthy that the superscript $^{(4)}$ is used to indicate that the variable is evaluated in the reacting environment as the co-precipitation occurs only in this environment.

Based on the total concentrations, the equilibrium concentrations can be computed. These equilibrium concentrations are determined by employing the Newton-Raphson method to solve a system of non-linear equations.[50] This approach yields the concentrations of free metal and hydroxide ions necessary for determining the degree of supersaturation. This quantity is the driving force that determines the rates of nucleation and growth of crystals, which is defined as the ratio between the product of metal hydroxide activity coefficients and the solubility product of Ni-Mn-Co hydroxide.[51] In order to consider the deviation of the reacting medium from the ideal solution, the activities of metal hydroxides are taken into account in the calculation of the supersaturation, which are evaluated with the Bromley method.[52] Further details of the chemical equilibria are reported in the Supplementary material.

2.3 Population Balance Model

In this work, the only property used to characterize the co-precipitated particles is their size L and the evolution of the particle population can be statistically described by the PSD:

$n_(L)$.

The evolution of the PSD is described by the population balance equation (PBE),[45] which is solved by adopting the quadrature method of moments (QMOM).[6] With this approach, the PSD is approximated by a summation of N weighted Dirac delta functions each located at a quadrature abscissa (in this work $N = 2$, corresponding to tracking the evolution of four moments of the PSD). The weights and abscissas are obtained from low order moments. In this way, the problem of solving the PBE is reduced to tracking $2N$ moments that are defined as $m_k = \int_0^\infty L^k n(L) \mathrm{d}L$, where k is the order of the moment.

The transport equations of probabilities and concentrations, respectively Eq. (1) and Eq. (2), are coupled with the transport equations for the moments of the PSD, defined as follows:[7]

$$\frac{\partial \langle m_k^{(4)} \rangle}{\partial t} + \mathbf{u}_\mathrm{f} \cdot \frac{\partial \langle m_k^{(4)} \rangle}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} \cdot \left(\Gamma_\mathrm{t} \frac{\partial \langle m_k^{(4)} \rangle}{\partial \mathbf{x}} \right) + p_4 h_k^{(4)}, \tag{3}$$

where $\langle m_k^{(4)} \rangle = p_4 m_k^{(4)}$ is the volume-weighted moment of order k and $h_k^{(4)}$ is the source term related to the nucleation, growth, aggregation and breakage rates. The first two particulate processes depend mainly on the supersaturation level, while the aggregation and breakage rates depend on the physical and turbulence properties of the suspending fluid and on the particle sizes. The kinetic of these processes are described as function of some parameters that were identified by comparison with experimental data.[2] For further details on the nucleation, growth, aggregation and breakage rates the reader may refer to the Supplementary Material.

3 Computational tools

As previously mentioned, the goal of the developed model is to obtain information on the PSD resulting from the NMC co-precipitation, occurring in large continuous reactors characterized by very long residence times. This requires very long computational times, as the fast kinetics of the chemical reaction impose a stability limit on the maximum simulation time-step. As

a remedy, the operator-splitting method is adopted to be able to use larger time-steps for the CFD simulation. In addition, we implemented a hybrid MPI-OpenMP parallelization to improve the load balance between the MPI processes for the integration of the source terms in the operator-splitting method. Both the operator-splitting method and hybrid parallelization are implemented in OpenFOAM-8 by using the solver pisoFoam as the base for the co-precipitation model.

3.1 Operator-splitting approach

The operator-splitting is here implemented in its second-order version, i.e., the Strang Operator Splitting, which allows to increase the time-step and reduce simulation instabilities, with the result of accelerating the numerical solution of the computational model.[27,30,53] With this method, the governing equations are solved in different steps: first the physical transport of the probabilities, concentrations and moments due to the convection and diffusion is solved for a half time step, obtaining an intermediate solution that is used to evaluate the supersaturation and to perform the integration of the source terms over an entire time step. After updating the local variables, the physical transport is solved for the remaining half time step.

The integration of the source terms is performed by the CVODE module of the SUNDIALS package, which is coupled with the main solver.[54–56]

In Fig. 2, a schematic representation of the solution algorithm is depicted.

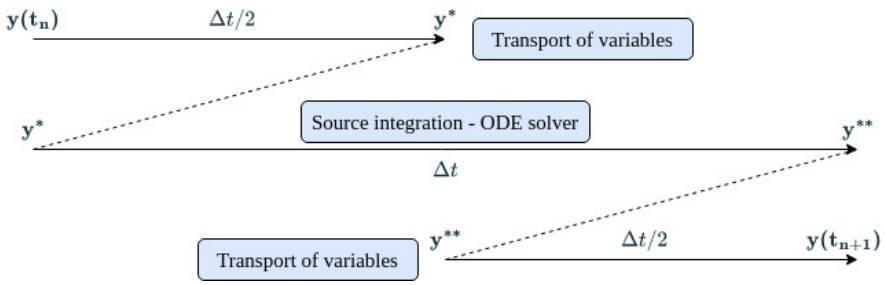


Figure 2: Schematic representation of the operator-splitting approach.

However, there is a drawback using this method, namely the load-balance issue. In fact,

in a decomposed domain parallelization, the operator-splitting approach usually leads to a load imbalance, since it is possible that some computational cores require more time to perform the integration of the source terms. This issue happens frequently since the target in classical domain decomposition is to minimize the number of boundaries between the sub-domains. However, in this case due to the large supersaturation gradients inside the reactor, the parallel processes associated with the sub-domains with high supersaturation values need a longer time to perform the integration of the source terms, slowing down the other parallel processes used for the simulation. This problem can also occur in other systems that are characterized by phenomena that occur locally. In order to limit this problem, a hybrid parallelization is implemented as described in the next section.

3.2 Hybrid parallelization

In this work, the hybrid parallelization consists in the two main parallelization strategies, i.e., distributed-memory and shared-memory, aiming at homogenizing the load-balance between cores, hence increasing the parallelization efficiency.[57–59] As mentioned before, the use of the operator-splitting approach may cause a great difference in the computational load between parallel processes, due to the different supersaturation levels in the partitions of the computational domain assigned to them. The hybrid parallelization scheme proposed here allows to make use of a large number of cores, that are necessary to perform the source integration in short times and, at the same time, decompose the spatial domain in less parts improving the load-balance of the simulation. MPI is a parallelization tool that employs distributed memory, while OpenMP employs shared memory. We refer to the literature for the detailed description of both approaches.[33–35,60] The hybrid parallelization approach makes use of both paradigms: the distributed-memory parallelization is used as in the standard domain decomposition for the CFD, while the shared-memory parallelization is employed to integrate simultaneously the source terms of computational cells in each region assigned to each MPI task. Figure 3 shows a schematic representation of the implemented hybrid

parallelization. The optimal number of MPI and OpenMP tasks is problem specific, and

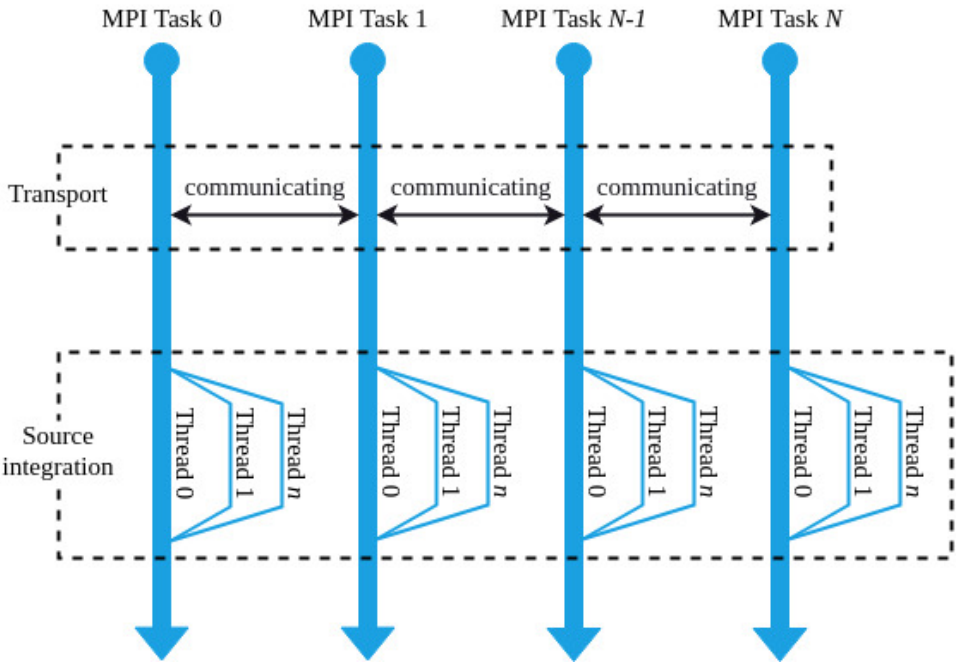


Figure 3: Schematic representation of the hybrid parallelization with N MPI tasks and n OpenMP threads.

must be found through run tests. The results of these tests will be shown in the following.

3.3 Compartment Model

To significantly reduce the computational load of the simulation, a more drastic approach inspired by domain decomposition, can be employed: the Compartment Model (CM). This approach consists in dividing the reactor into different completely mixed compartments which exchange mass with their neighbours. In fact, each compartment is a collection of cells characterized by some mean values for the variables tracked by the model, from the flow and turbulence properties, to species concentrations and moments of the PSD.

The idea behind the CM is not novel: a preliminary CFD simulation is performed to obtain the information about the flow field and possibly other relevant fields to create a network of compartments.[61,62] In this regard, it is interesting to mention another approach that allows to speed-up classical CFD simulations, which uses a different approach than

ours, but it tries to address a similar problem. Recently, Pirker, Lichtenegger and coworkers presented a novel technique called recurrence Computational Fluid Dynamics (rCFD) to simulate different types of industrial processes. Basically, rCFD is an alternative to the time-extrapolation of full CFD simulations based on recurrence statistics used to compare the flow field at different times. Both CM and rCFD approaches reduce the degrees-of-freedom, and therefore, decrease the computational cost of the simulation, both need previous CFD simulations (one to reproduce the recurrence statistics, the other to compartmentalize the reactor), both approaches can provide temporal evolution of variables of interest. The rCFD approach keeps the same spatial discretization used in the CFD simulation, while the CM combines several cells to create one compartment. Moreover, rCFD simulations can account for diffusive terms that are instead neglected in the CM. On the other hand, rCFD method suffers from the local violation of conservation laws, it has been never used to study the behavior of fast reactive scalars in a turbulent flow field, that then have a crucial importance on slower phenomena, such as in our case in the evolution in space and time of the particle population. Another important cons is that it needs a manual decision to choose the correct recurrence statistic, making the method not suited for being automatized.[36,63–65] For these reasons, we decided to invest our effort in developing and improving the Compartment model.

The CM approach implemented in this work is an in-house code developed in Python, in which predictions are an approximation of the results that can be obtained by the full CFD-PBM code, but it represents an interesting alternative due to its lower computational costs. In fact, the numerical solution of a large number of unknowns belonging to a system of partial differential equations (PDEs) is replaced by a less expensive system of ordinary differential equations (ODEs).

The governing equations solved by the CM are as follows:

$$\frac{\mathrm{d}m_{k,i}}{\mathrm{d}t} = \sum_j F_{j,i} m_{k,j} - \sum_j F_{i,j} m_{k,i} + S_{k,i}, \tag{4}$$

$$\frac{\mathrm{d}C_{\alpha,i}}{\mathrm{d}t} = \sum_j F_{j,i} C_{\alpha,j} - \sum_j F_{i,j} C_{\alpha,i} - S_{\alpha,i}, \tag{5}$$

where $m_{k,i}$ is the moment of order k in the i-th compartment, $F_{i,j}$ is the flux between the i-th compartment and the j-th one, i.e., the summation of the flux between the cells that are at the boundary between the two compartments, $C_{\alpha,i}$ is the concentration of the species α in the compartment i, while $S_{k,i} = h_k$ and and $S_{\alpha,i} = Q_\alpha$ are respectively the source term of the moment of order k and of the concentrations of the species α in the i-th compartment. The definition of Q_α and h_k is the same as that described in Sections 2.2 and 2.3 for the full CFD-PBM model.

The main difference between the CM and CFD-PBM is the lack of diffusion terms in the governing equations, while the convective contributions takes into account the fluxes between different compartments. Another big difference is the coarse spatial discretization adopted to reduce the computational costs of the simulation. Therefore, this implies that all phenomena which play a crucial role at the molecular scale would not be described accurately, such as the micromixing. Indeed, the governing equations of the environment probabilities are not solved in the CM, so basically the effect of mixing at the molecular scale on the co-precipitation is not considered directly. The micromixing is indirectly considered during the compartmentalization step as described later.

The CM was also tested with the governing equations for environment probabilities, however the solver showed robustness problems and the results were not physically meaningful, since the micromixing effects occur in very small regions of the reactor. Moreover, it is worth remarking that some strategies for speeding-up the CM simulations are applied. The operator-splitting approach is implemented to accelerate the integration of the ODEs, i.e., separating the integration of the transport and source terms. The transport terms are integrated with the backward Euler method while the source terms are integrated by the LSODA time integrator,[66] which can automatically switch between stiff (using Backward Differencing Scheme) and non-stiff (Adams-Moulton-Bashford scheme) methods. The latter

integration method is available within the *Scipy* python package. A further improvement of the CM consists in a domain decomposition parallelization, using the distributed-memory paradigm by assigning the solution of the system of ODEs to some MPI tasks. This strategy has been applied by using the *mpi4py* python package.

3.3.1 Reactor division

As previously mentioned, the approach used for the creation of compartments represents one of the elements of novelty of this work, since it is not purely based on geometrical considerations, but it is based on the specific features that characterize the co-precipitation process. In fact, the reactor division follows the spatial values of the turbulent dissipation rate and additionally the probability fields that characterize zones of the reactor with a relevant level of the supersaturation. In this way, the reactor is compartmentalized by the main properties that influence the particulate processes that determine the temporal evolution of the PSD, i.e., the supersaturation for the nucleation and molecular growth, and the turbulent dissipation rate for the aggregation and breakage.

To split the reactor into compartments according to the mentioned variable fields, we employ the agglomerative clustering: a method of cluster analysis which gathers objects (computational cells in our case) in clusters by minimizing the variance of target properties within each cluster. The agglomerative clustering method is a bottom-up approach that starts from single objects and merges them to create larger clusters as it climbs up the hierarchy until the desired number of clusters is achieved.[67] This clustering method is available in the scikit-learn package of python.[68] The advantages of this algorithm are that it can consider the connectivity between cells, essential to create compartment with contiguous cells, and moreover it can be extended to consider more than one criterion for clustering. The user inputs required by the algorithm are the desired number of regions to divide the reactor, the criteria on which the compartmentalization is based and the connectivity between the cells (this is a sparse matrix in which the element at the i-th row and j-th column is equal

to 1 if the i-th cell and j-th cell are adjacent). The linkage criterion used to perform the compartmentalization is the Ward's method.[69] At each step, the algorithm finds pairs of clusters (also considering the connectivity between cells) that leads to the minimum increase of the variance of the selected variables for the compartmentalization. The division of the reactor consists of three steps, which are summarized in Fig. 4:

1. The reactor is divided into five regions considering the value of the summation of the probability of metal sulphates and sodium hydroxide environments ($p_1 + p_2$). Accordingly, two lower and an upper thresholds are selected for this division. The cells with values below the lower threshold are assumed to be one of the pure feeds. These cells correspond to two regions inside the inlet tubes of metal sulphate and sodium hydroxide. Those with values higher than the upper threshold are assumed as completely micromixed. These cells correspond to the region far from both injection points. The rest of the cells are assumed to have some degree of segregation. They correspond to two (usually separate) regions close to the injection points of metal sulphate and sodium hydroxide.

2. After the first division, the zone far from the inlets is divided into N_{ext} compartments by using the turbulent dissipation rate as the criterion.

3. In the end, the two zones near the inlets are both divided into N_{int} different compartments by using, as criterion, both probabilities and turbulent dissipation rate. Moreover, the probability field is scaled by the following expression to have a better separation of regions with different level of micromixing.

$$p = \frac{1}{\sqrt[3]{1 - (p_1 + p_2)}} - \frac{1}{\sqrt[3]{p_1 + p_2}} \tag{6}$$

The user can decide the number of compartments by selecting N_{ext} and N_{int}. In this way, the reactor is divided in a total number of compartments corresponding to: $N = N_{ext} + 2N_{int}$.

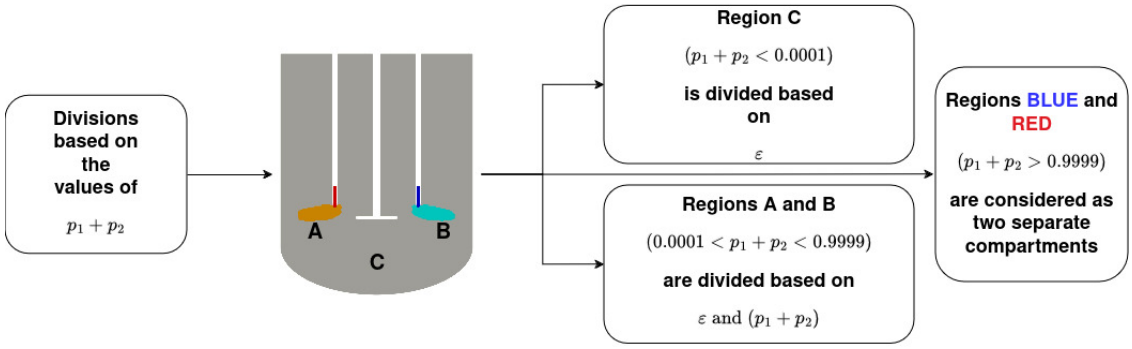


Figure 4: Flowchart of the reactor compartmentalization.

Since the number of variables solved for each compartment is equal to ten (four moments of the PSD and six chemical species concentrations), this means that the total number of ODEs solved by the CM will be equal to $10N$. This aspect has clearly an impact on the final computational time of the CM.

4 Results and discussion

The full CFD-PBM was used to simulate the co-precipitation of Ni-Mn-Co hydroxide in a 3L continuous stirred-tank reactor (CSTR), which is shown in Fig. 5. The CSTR has a cylindrical body of 150 mm diameter and height with a round bottom of 37.5 mm height. Moreover, it has four baffles of the same height as the cylindrical body with a width of 12 mm and a thickness of 2 mm. The reactants are injected from three different inlets of 3 mm diameter and 118 mm length, while the outlet is just a patch on the top wall of the reactor with diameter of 6 mm. The mixing of the solution is guaranteed by a Rushton turbine impeller of 50 mm diameter with six blades of 10 mm height and 2 mm thickness.

As mentioned in the introduction, the main aim of this work is not to show the effect of several operating conditions on the process, but mostly to demonstrate how to speed up the CFD-PBM simulations with hybrid parallelization, and proposing the CM as a less demanding alternative model. Consequently, just one set of operating conditions was ex-

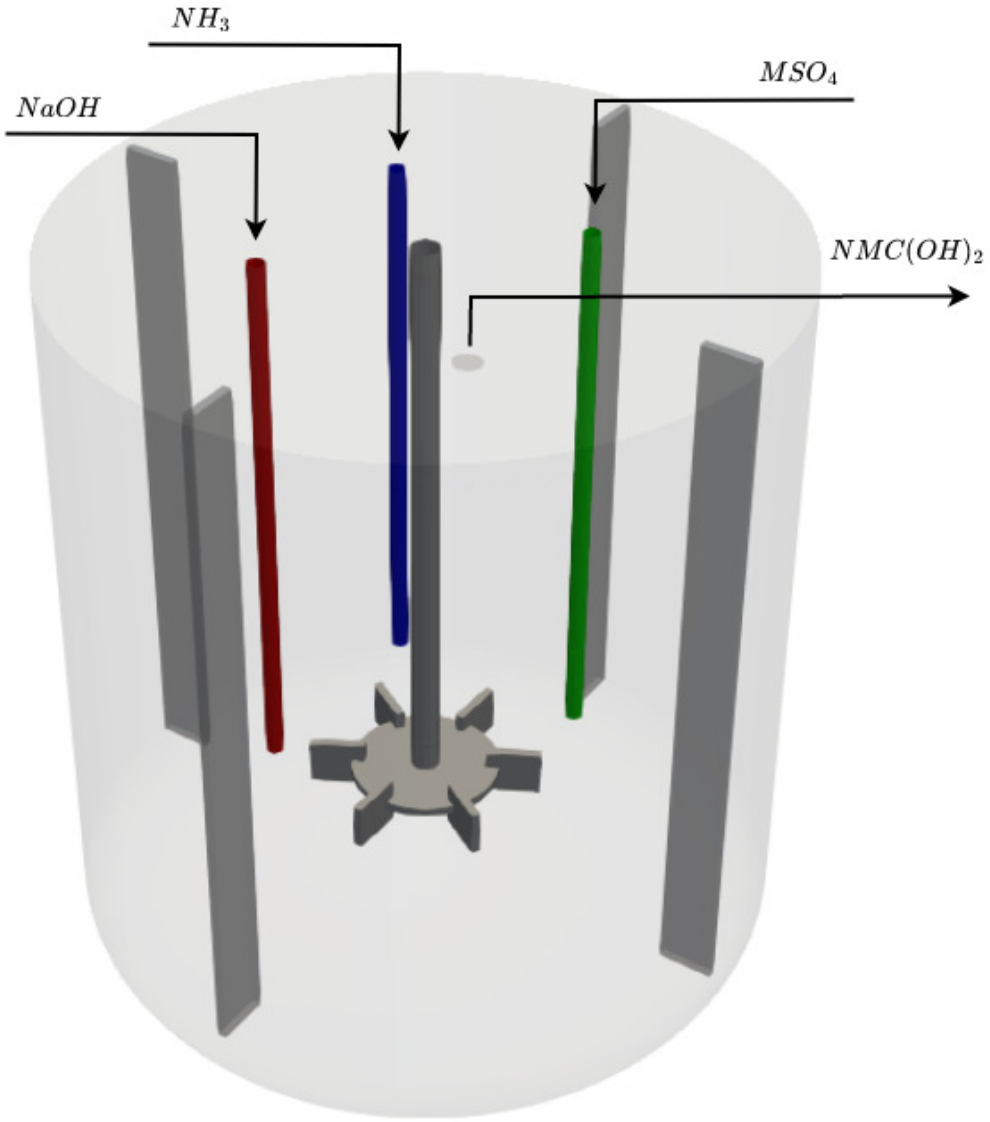


Figure 5: Sketch of the CSTR used for CFD-PBM simulations.

plored: the impeller rotation speed was set to 200 rpm, while the concentrations of the metal sulphates was 2 M. The ratio of metals in the metal sulphates feed was kept constant ([Ni]:[Mn]:[Co]=8:1:1). In addition, the ratio of inlet flow rates is $\dot{V}_{\mathrm{MSO_4}} : \dot{V}_{\mathrm{NaOH}} : \dot{V}_{\mathrm{NH_3}} = 5 : 4 : 1$ for a total inlet flow rate equal to 3 L/h. Ammonia is injected with a concentration five times higher than that of metal sulphates, while the sodium hydroxide concentration is two and half times the concentration of metals. The temperature is set equal to 25 °C.

4.1 Effect of hybrid parallelization

As already mentioned, the computational cost of the full CFD-PBM simulation is very high. Indeed the solver requires several weeks to reach the steady-state solution of the process. The first features implemented, the operator-splitting approach, gives the opportunity to increase the time-step of the simulation.

This method speeds up the simulation, however, the computational cost of the solver is still important. In order to reduce the computational time required by the model to reach the final solution, a scalability study was performed to better understan the most critical steps of the solver. Starting from a simulation that already reached convergence, we perform ten iterations increasing gradually the number of cores, then we take the averages of the computational time to compute the time spent for one time-step, which is shown in Fig. 6. For this study, the simulations were carried out using Gold Intel Xeon 6130 2.10 GHz CPUs and the only particle processes considered are nucleation and growth.

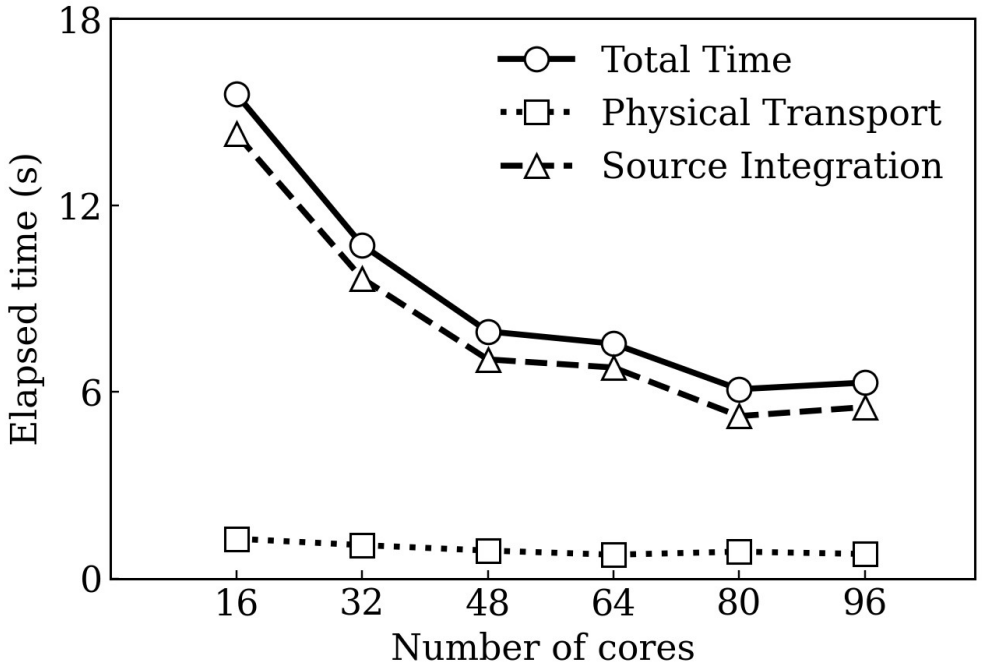
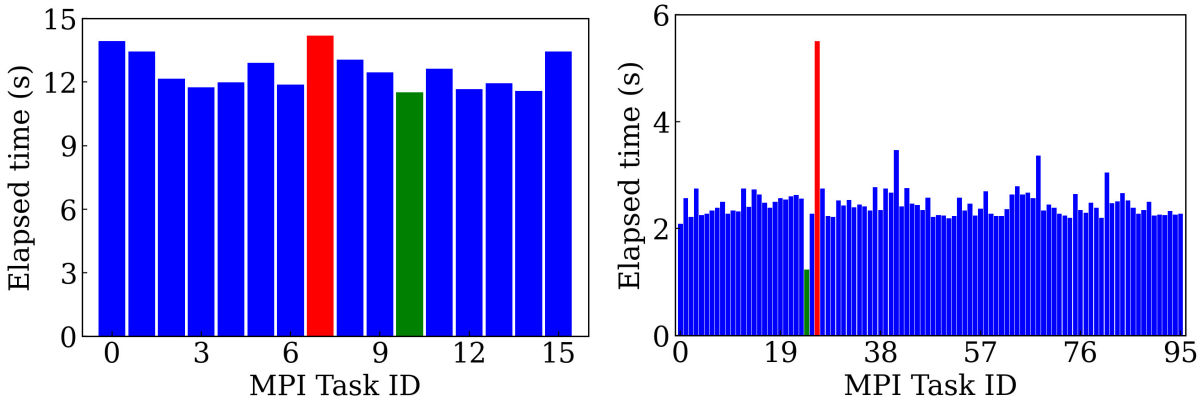


Figure 6: Different computational times to perform one time-step in function of the number of cores. In the plot the total time, the time to solve physical transport and the time to perform the source integration are shown.

As expected, the total time required to perform one time-step decreases by increasing the number of cores. The second interesting point is that the integration of the source terms is the most computationally expensive step. In fact, the transport needs around one second to be solved, while the remaining time is dedicated for the source integration.

Another important aspect to explore is the load-balance of computational cores. The procedure followed is the same as described before: we performed ten iterations, evaluated the time required by every processor to carry out the different steps of the model and make the average through ten time-steps. As we observed previously that the integration of the source terms takes much more time than the transport, we focus on the load balance of this step. In Fig. 7 the results obtained for the cases with 16 and 96 cores (or MPI tasks) are shown. It is evident that the load balance gets worse as the number of cores increases: the difference



(a) 16 cores **(b) 96 cores**

Figure 7: Load balance of the source integration step for different number of cores, (a) 16 and (b) 96 cores. Red and green bars represent the maximum and the minimum time required, respectively.

between the maximum time employed by a MPI task to perform the source integration and mean value between all cores changes from 13% (Fig. 7a) to 122% (Fig. 7b) as the number of MPI tasks increases from 16 to 96. This happens due to the high supersaturation gradient inside the reactor. In fact, the time required by the solver to perform the source integration is strictly related to the supersaturation, which increases in regions with high levels of the supersaturation. Therefore, the problem gets worst as more processors are used

for the simulation, i.e., as the reactor is decomposed in more regions, because the difference between the level of supersaturation in the regions increases. Moreover, considering that all MPI tasks should finish the source integration before they can proceed with the next calculation step, the majority of them remain in an idle state waiting for the one with the highest computational load. This can drastically reduce the parallelization efficiency.

In order to limit this problem, a hybrid MPI-OpenMP parallelization was implemented with the aim of increasing the parallelization efficiency as the number of cores increases. For this purpose, a selected number of computational cores (instead of only one core) are dedicated to each MPI tasks, which are used to launch OpenMP threads for the simultaneous source integration of cells located in the corresponding sub-domain of MPI tasks. Therefore, we can use a higher number of computational cores without increasing the number of sub-domains for the CFD parallel decomposition. The same study described before was performed by changing the number of cores, i.e., OpenMP threads, assigned to each MPI task and the results obtained are showed in Fig. 8.

As can be seen from Fig. 8a, the implementation of the hybrid parallelization reduces the computational time required by the solver to perform one time-step. Indeed, focusing on the case with 96 cores, the elapsed time decreases from 6.3 s with classic parallelization (black line) to 3.6 s with the hybrid parallelization employing 4 or 8 OpenMP threads for each MPI task (red and green line). The hybrid parallelization has a great effect on the time required for the source integration, showing always better results with respect to the classic parallelization, except when the number of the MPI tasks is too small. Indeed, it is possible to see in Fig. 8a that for the case with totally 16 number of cores, increasing the number of OpenMP threads (or in other words decreasing the number of MPI tasks) leads to longer computational times. It is due to the fact that when the number of MPI tasks is reduced too much, the time required for the transport becomes comparable to that required for the source integration, and therefore, the situation gets worse as the number of MPI tasks decreases further. This means that an optimal number of MPI tasks exists for the

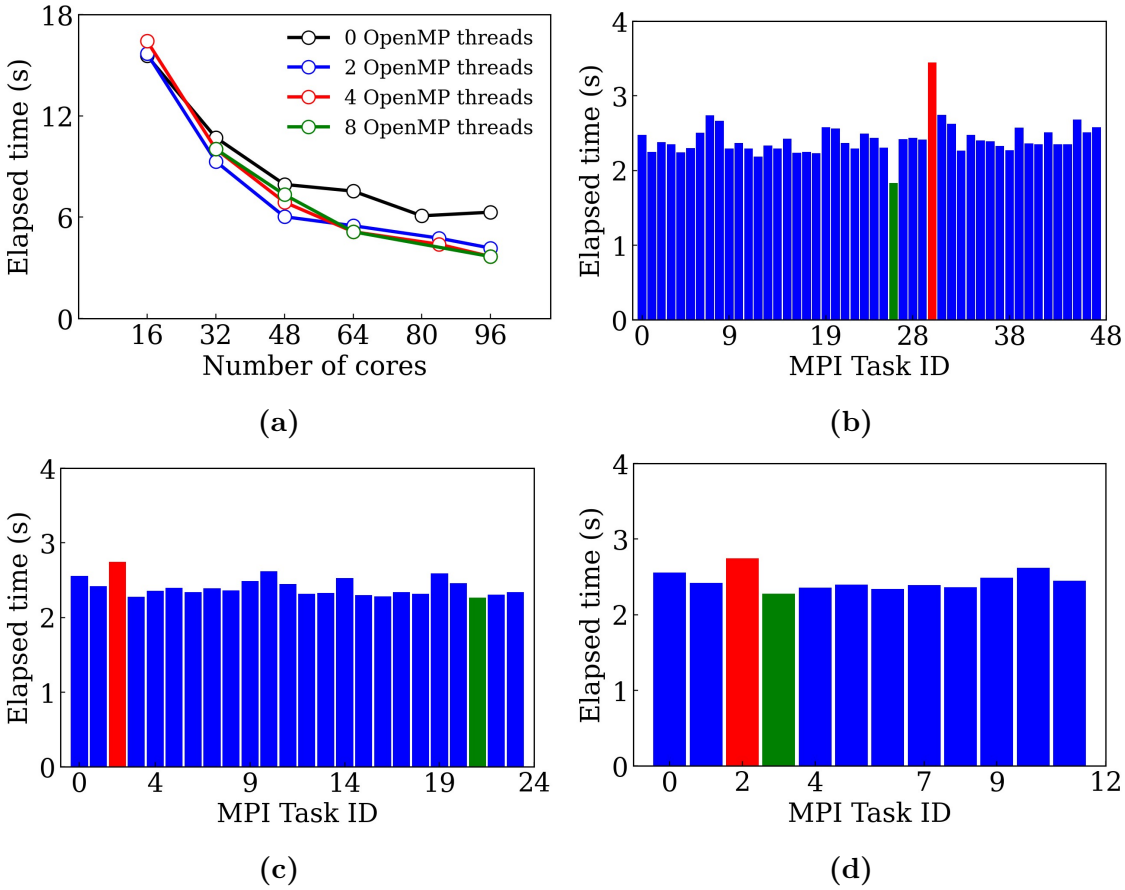


Figure 8: On the top-left (a), elapsed time to perform one time-step using different number of OpenMP threads. In the other plots (b, c and d), load balance of source integration step with 96 cores and different number of OpenMP threads for each MPI tasks, respectively 2, 4 and 8. Red and green bars represent the maximum and the minimum time required, respectively.

CFD parallelization below which the time required for the transport can be still reduced, and above which no advantage is offered due to increase in the load-balance and increased communication between a large number of MPI tasks. Also the choice of the number of OpenMP threads depends on the total number of cores available: if it is not possible to operate with high number of cores, the best option is to employ a configuration with a small number of OpenMP threads for each MPI task. This is also inferred from Fig. 8a, where it is shown that the configuration with two OpenMP threads (blue line) takes less time than the cases with four or eight OpenMP threads to perform one time-step when the total number of the used cores is relatively low. This can be associated to insufficient number of MPI tasks as

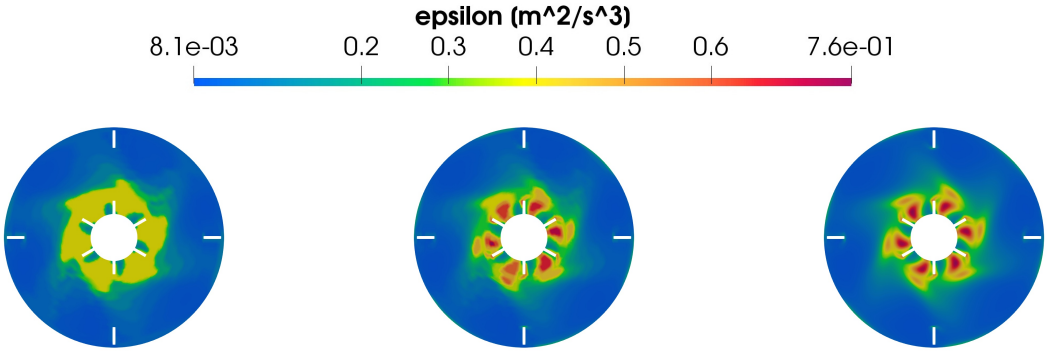
described before. As the total number of the cores increases, the computational time reduces by increasing the OpenMP threads because more cores are used for the source integration without increasing too much the number of MPI tasks that can cause load-balance issues. Therefore, it is advantageous to use a larger number of OpenMP threads as long as enough MPI tasks are dedicated for the CFD parallelization. Otherwise, it is better to limit the number of the OpenMP threads to have enough MPI tasks for the CFD. In addition, it is important to note that the number of OpenMP threads is limited to the number of physical cores that exist in one computational node and cannot increase beyond that.

Moreover, the efficiency of the parallelization improves as can be seen by comparing Fig. 7b with Figs. 8b to 8d. With the same total number of cores used, i.e., 96, the difference between the maximum and the mean value of the time spent by the MPI task for the source integration decreases from 122% to 42%, 14% and 12% by employing 2, 4 and 8 OpenMP threads, respectively, which confirms the great benefits of adopting the hybrid parallelization.

4.2 Comparison between CFD-PBM and CM

Now, let us compare the results obtained by the CM with the predictions by the full CFD-PBM. First, we focus on the turbulent dissipation rate. In Fig. 9 a comparison of the turbulent dissipation rate, ε, between CM with different numbers of compartments and a full CFD-PBM simulation is shown. The section is taken across the reactor, at the height where the impeller is located.

Considering that the compartmentalization was done on the basis of the turbulent dissipation rate, a good agreement, in terms of absolute values, between the full CFD-PBM and CM simulations was already taken for granted. However, it is noteworthy that, with a sufficient number of compartments, the CM can describe well also the spatial distribution of the turbulent dissipation rate in comparison to the results obtained by the CFD-PBM simulation. This can be observed by focusing in the zone near the impeller blades where the turbulent dissipation rate reaches the maximum value: here, the compartment model can



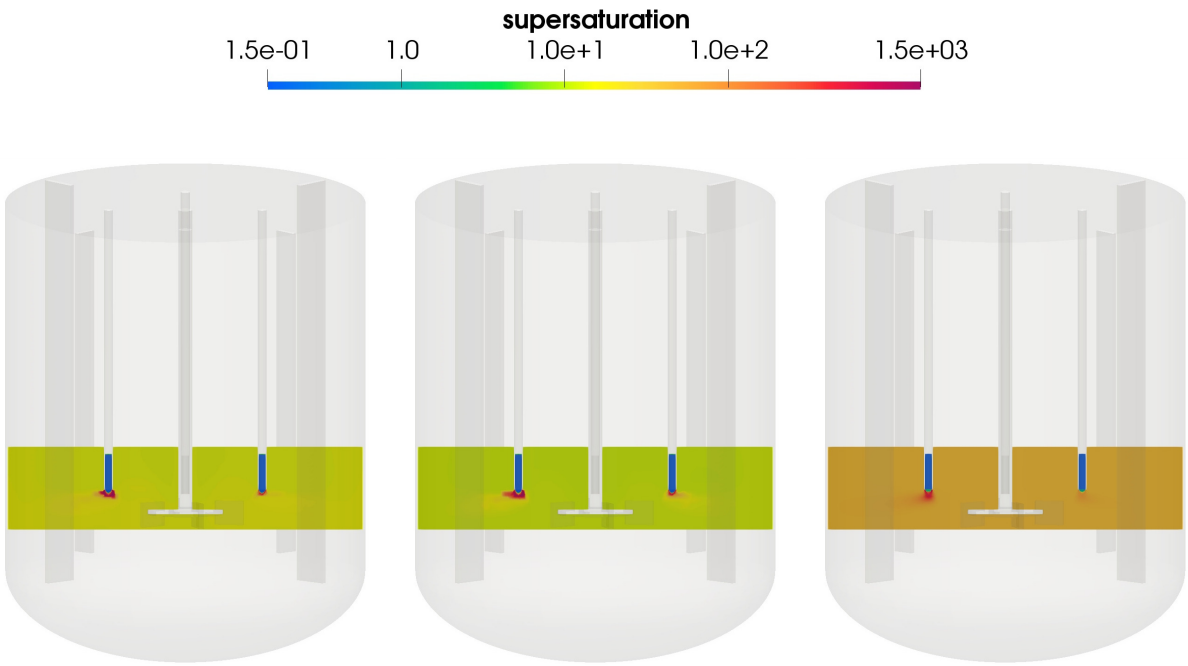
(a) 100 compartments **(b) 400 compartments** **(c) Full CFD**

Figure 9: Contour plots of the turbulent dissipation rate for different compartments, 100 (a) and 400 (b), and for the full CFD simulation (c). Transversal section of the reactor.

describe well both the profiles and the values of ε. Moreover, as the number of compartments increases, the turbulent dissipation rate is represented better in the compartment model as expected. This can be also confirmed by considering the average and maximum values of ε: the CFD-PBM predicts a mean and a maximum values that are respectively 0.0019 and 0.754 m^2/s^3. The corresponding values with 100 compartments are respectively 0.00188 and 0.389 m^2/s^3, which improve with increasing the number of compartments to 400, reaching 0.00191 and 0.763 m^2/s^3.

Now, considering that the CM would be used to solve the PBM, let us concentrate on the supersaturation which is, as mentioned previously, the driving force of the co-precipitation process. In Fig. 10 contour plots of the supersaturation predicted by the CM and CFD-PBM simulations in a vertical section of the reactor that passes through the inlets of metal sulphates and sodium hydroxide are reported.

Unlike the turbulent dissipation rate, the supersaturation profiles estimated by the CM are quite different with respect to that obtained by the full CFD-PBM simulation, mainly in terms of absolute value. As expected the supersaturation reaches the maximum value near the reactant inlets and then it is consumed rapidly due to the particle nucleation and growth. In this regard, we can notice a great difference between the two models, because the average value predicted by the full CFD-PBM is around 43, while the CM evaluates a



(a) 100 compartments **(b) 400 compartments** **(c) Full CFD-PBM**

Figure 10: Contour plot of the supersaturation for different compartments, 100 (a) and 400 (b), and for full CFD-PBM simulation (c). Detail of vertical section of the reactor.

mean value between 14 and 16, depending on the number of compartments. Obviously, this discrepancy has a large effect on the final PSD of the precipitated product, due to the crucial role played by the supersaturation in the particulate processes that determine the evolution of the particle size.

Despite this fact, it is important to underline the good agreement between models for the spatial profiles: both models predict the homogeneity of the supersaturation field far from the inlets but, more importantly, the CM can simulate correctly how the supersaturation is consumed during the co-precipitation process. Indeed, from the contour plots shown in Fig. 10, it is possible to note that also the simulation performed with the CM predicts the plume of supersaturation generated near the inlets that slowly disappears moving away from the feed tubes. The small differences that can be seen are due to the spatial discretization that for the CM is obviously less refined than in the full CFD-PBM.

The big difference found in terms of absolute values of supersaturation, i.e., maximum

and mean values, can instead be explained considering that, for CM simulations, the governing equations of the concentration of species and moments of the PSD lack of the diffusion terms and, moreover, the micromixing model is not solved (see Table 1 as reference). For the sake of completeness, a full CFD-PBM simulation without the micromixing model was performed. In this scenario we were able to see a smaller difference of supersaturation profiles and values, between the two models. Indeed, the average value predicted by the full CFD-PBM simulation is around 24, which is much closer to the values calculated by the CM. As mentioned before, the remaining difference is due to the lack of diffusion terms and, of course, to the fact that the CM is an approximation of the full CFD-PBM simulation which aims to provide a meaningful prediction in less time than the full simulation.

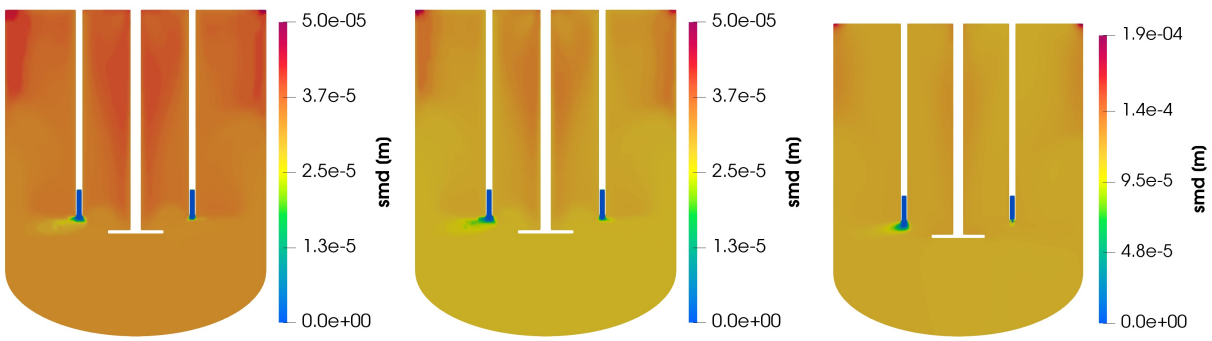
Table 1: Main features implemented in the two different approaches, the CFD-PBM and the CM models.

Features	Operator splitting	Chemical equilibria	Convective terms	Diffusion terms	Micromixing model
CFD-PBM	✓	✓	✓	✓	✓
CM	✓	✓	✓	✗	✗

As mentioned in the introduction, the main goal of precipitation processes is to obtain particles with specific sizes and morphologies, in order to optimize the properties of the final products. In this regard, one of the most relevant results that we can obtain from the developed models is the mean size of particles, e.g., the Sauter mean diameter (SMD), evaluated as SMD $= \frac{\langle m_3{}^{(4)} \rangle}{\langle m_2{}^{(4)} \rangle}$. In Fig. 11, the predicted contour plots of the SMD on a vertical section of the reactor are reported.

As expected, the error committed with the CM in the predictions of the supersaturation is reflected also on the SMD. In fact, the mean value of particle size varies between 35 μm and 30 μm, depending on the number of compartments, while the average size predicted by the full CFD-PBM simulation is around 115 μm.

Nevertheless, we can make some further discussion about the obtained results by con-



(a) 100 compartments **(b) 400 compartments** **(c) Full CFD-PBM**

Figure 11: Contour plot of the SMD for different compartments, 100 (a) and 400 (b), and for full CFD-PBM simulation (c). Section of the reactor.

ducting a qualitative comparison. Indeed, Fig. 11 shows that by increasing the number of compartments, the spatial profile of the SMD obtained by the CM tends to be more and more similar to the one obtained by the full CFD-PBM simulation. For all the cases, the scale used for the contour plots is a linear scale between 0 μm and the maximum value, allowing a better comparison between the CFD-PBM and CM predictions, focusing on the SMD profile more than on the numerical values that are influenced by the features lacking in the CM as mentioned previously. Let us focus first on the SMD profile obtained by the full CFD-PBM, Fig. 11c, where we can identify some crucial zones: first, it is possible to note the plume near the metal sulphates inlet, where the particles start to nucleate and grow due to the high supersaturation level, see Fig. 10c, and, secondly, we can observe two zones where the SMD of the particles reach higher values, which are around the impeller shaft and in proximity of the angle between the lateral and top walls of the reactor.

Focusing on the zones just mentioned and comparing the full CFD-PBM simulation with CM predictions, we can immediately observe that the plume related to the main growth of particles is accurately described by the CM, see Figs. 11a and 11b, with improved agreement as the number of compartments increases. Instead, as expected we can observe that with a lower number of compartments, Fig. 11a, the CM overestimates the size of the areas in which the SMD reaches higher values. This error decreases as the number of compartments increases. Indeed, with 400 compartments, Fig. 11b, we notice that the zones previously

mentioned are confined to a small area, exactly as predicted by the full CFD-PBM simulation. Moreover, although the mean value calculated from the CM simulation with 400 compartments is far away from the one obtained by the full CFD-PBM case, spatial profile comparison show an acceptable discrepancy and the crucial zones detected at the beginning are well described by the CM model. Lastly, it is noteworthy to consider that the spatial profiles of SMD and supersaturation can provide important insights about the reactive crystallization process, which can be used to optimize the operating conditions and the geometry of the reactor to achieve a final product with desired properties.

In Fig. 12 the temporal evolution of the SMD at the outlet of the reactor predicted by the CM with a different number of compartments is compared with the predictions obtained by the full CFD-PBM simulation. As described before, the asymptotic value gave by the CM differs from the prediction of the full CFD-PBM simulation, but the noteworthy result is the correct description of the profile evolution. Moreover, the delay of the particle growth observed in the full CFD-PBM prediction (dashed line) is due to the micromixing model, which causes the reactions and, therefore, the generation of the supersaturation to occur slower.

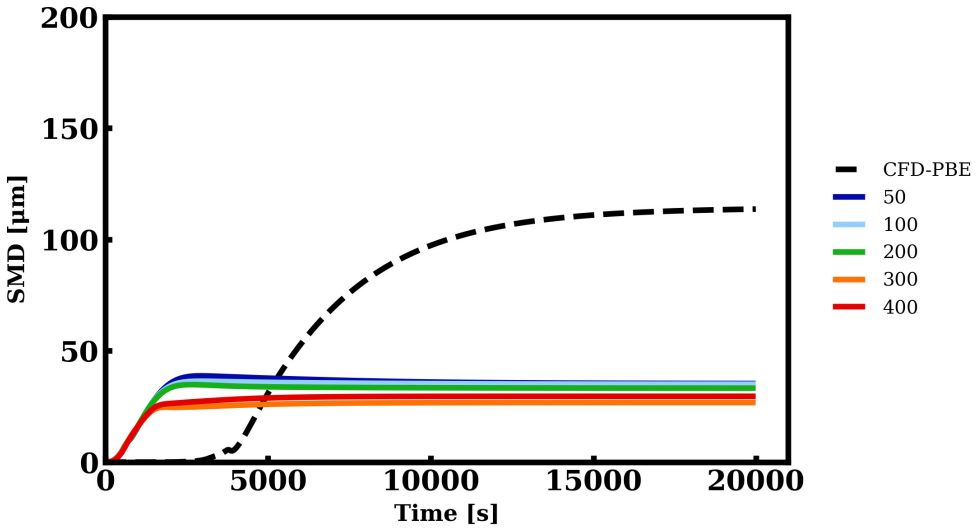


Figure 12: Time evolution of the SMD evaluated at the outlet of the reactor. Comparison between full CFD-PBM and CM predictions with different number of compartment.

In conclusion, the most important advantage of the CM is the drastic reduction of the computational cost, as reported in Table 2. In fact, a full CFD-PBM simulation takes around 2 months, while the computational time needed by the CM changes from 1 day if the reactor is compartmentalized in 100 parts to 5 days if the number of compartments is 400. In other words, with the CM, it is possible to reduce the computational time about 91-98%.

Table 2: Computational times required by the solver to complete a simulation. Comparison between CM with different compartments (comps) and full CFD-PBM approaches.

N° of comps	50	100	200	300	400	CFD-PBM
Time [h]	24	34	54	76	84	1296

5 Conclusions

In this work, we proposed two different computational tools to simulate reactive crystallization processes inside a continuous stirred tank reactor: one is based on a CFD-PBM approach in which fluid dynamics and the population balance equation are solved tightly coupled while, the other one, is based on a CM. The CM is constructed via an automatic script for the compartmentalization of the reactor, based on mixing and turbulence properties. This represents one the main novelty of this approach, when compared to other similar workflows. Moreover, the framework includes an operator-splitting approach and a hybrid parallelization technique in order to significantly reduce the computational costs, increase the parallelization efficiency and improve the robustness of the solver. We have analyzed both tools by simulating the co-precipitation process of nickel, manganese and cobalt hydroxide. However, these computational tools can be also applied to other reactive multiphase systems. Furthermore, the developed models can be adapted to simulate the precipitation of other metals.

The improvements achieved through the implementation of the operator-splitting ap-

proach and hybrid parallelization in the full CFD-PBM model shows clearly a reduction of the computational time required to perform the calculations. Furthermore, the parallelization efficiency has also been improved by uniforming the computational load through all the parallel processes.

The CM instead, despite the lack of the diffusion terms and micromixing model, is capable of producing results that are qualitatively in agreement with those of the full CFD-PBM. Indeed, the spatial profiles and the evolution of important variables, e.g., supersaturation and Sauter mean diameter, are described with an acceptable error by the compartment model, with the results that are increasingly similar to those of the CFD-PBM as the number of compartments increases. Therefore the CM can be seen as a less-demanding model that can provide meaningful qualitative predictions in a small amount of time, with the possibility to tune the desired accuracy through the choice of the number of compartments.

Supporting Information Available

- Detailed description of the micromixing model and equation employed
- Accurate description of the chemical equilibria and equations involved
- Detailed description of the population balance model

Author contributions

A. Querio: Methodology, Software, Formal analysis, Validation, Visualization, Writing - Original Draft; **M. Shiea**: Conceptualization, Methodology, Software, Formal analysis, Validation, Visualization, Writing - Review & Editing; **A. Buffo**: Conceptualization, Writing - Review & Editing, Supervision; **D. Marchisio**: Conceptualization, Writing - Review & Editing, Project administration, Funding acquisition.

Acknowledgement

Computational resources were provided by HPC@POLITO, a project of Academic Computing within the Department of Control and Computer Engineering at the Politecnico di Torino (http://www.hpc.polito.it).

We also acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support (https://www.hpc.cineca.it).

Data availability

The solvers developed and employed in this work can be found and downloaded from the GitHub repository of our research group MulMoPro (https://github.com/mulmopro/wet-synthesis-route).

References

(1) Shiea, M.; Querio, A.; Buffo, A.; Boccardo, G.; Marchisio, D. CFD-PBE modelling of continuous Ni-Mn-Co hydroxide co-precipitation for Li-ion batteries. *Chemical Engineering Research and Design* **2022**, *177*, 461–472.

(2) Para, M. L.; Alidoost, M.; Shiea, M.; Boccardo, G.; Buffo, A.; Barresi, A. A.; Marchisio, D. A modelling and experimental study on the co-precipitation of $Ni_{0.8}Mn_{0.1}Co_{0.1}(OH)_2$ as precursor for battery cathodes. *Chemical Engineering Science* **2022**, *254*, 117634.

(3) Raponi, A.; Romano, S.; Battaglia, G.; Buffo, A.; Vanni, M.; Cipollina, A.; Marchisio, D. Computational modeling of magnesium hydroxide precipitation and kinetics parameters identification. *Crystal Growth & Design* **2023**, *23*, 4748–4759.

(4) Raponi, A.; Achermann, R.; Romano, S.; Trespi, S.; Mazzotti, M.; Cipollina, A.; Buffo, A.; Vanni, M.; Marchisio, D. Population balance modelling of magnesium hydroxide precipitation: Full validation on different reactor configurations. *Chemical Engineering Journal* **2023**, *477*, 146540.

(5) Marchisio, D. L.; Fox, R. O.; Barresi, A. A.; Baldi, G. On the Comparison between Presumed and Full PDF Methods for Turbulent Precipitation. *Industrial & Engineering Chemistry Research* **2001**, *40*, 5132–5139.

(6) Marchisio, D.; Fox, R.; Barresi, A.; Garbero, M.; Baldi, G. On the simulation of turbulent precipitation in a tubular reactor via computational fluid dynamics (CFD). *Chemical Engineering Research and Design* **2001**, *79*, 998–1004.

(7) Marchisio, D. L.; Barresi, A. A.; Garbero, M. Nucleation, growth, and agglomeration in barium sulfate turbulent precipitation. *AIChE Journal* **2002**, *48*, 2039–2050.

(8) Barresi, A. A.; Marchisio, D.; Baldi, G. On the role of micro-and mesomixing in a continuous Couette-type precipitator. *Chemical engineering science* **1999**, *54*, 2339–2349.

(9) Marchisio, D. L.; Omegna, F.; Barresi, A. A. Production of TiO2 nanoparticles with controlled characteristics by means of a Vortex Reactor. *Chemical Engineering Journal* **2009**, *146*, 456–465.

(10) Marchisio, D. L.; Omegna, F.; Barresi, A. A.; Bowen, P. Effect of mixing and other operating parameters in sol- gel processes. *Industrial & engineering chemistry research* **2008**, *47*, 7202–7210.

(11) Hussain, M.; Ceccarelli, R.; Marchisio, D.; Fino, D.; Russo, N.; Geobaldo, F. Synthesis, characterization, and photocatalytic application of novel TiO2 nanoparticles. *Chemical Engineering Journal* **2010**, *157*, 45–51.

(12) Bensaid, S.; Deorsola, F. A.; Marchisio, D.; Russo, N.; Fino, D. Flow field simulation and mixing efficiency assessment of the multi-inlet vortex mixer for molybdenum sulfide nanoparticle precipitation. *Chemical Engineering Journal* **2014**, *238*, 66–77.

(13) Costa, C. B. B.; Maciel, M. R. W.; Maciel Filho, R. Considerations on the crystallization modeling: Population balance solution. *Computers & chemical engineering* **2007**, *31*, 206–218.

(14) de Azevedo Jacqueline, R.; Fabienne, E.; Jean-Jacques, L.; Inês, R. M. Antisolvent crystallization of a cardiotonic drug in ionic liquids: Effect of mixing on the crystal properties. *Journal of Crystal Growth* **2017**, *472*, 29–34.

(15) Melzig, S.; Finke, J.; Schilde, C.; Vierheller, A.; Dietzel, A.; Kwade, A. Fluid mechanics and process design of high-pressure antisolvent precipitation of fenofibrate nanoparticles using a customized microsystem. *Chemical Engineering Journal* **2019**, *371*, 554–564.

(16) Choi, Y.-J.; Chung, S.-T.; Oh, M.; Kim, H.-S. Investigation of crystallization in a jet Y-mixer by a hybrid computational fluid dynamics and process simulation approach. *Crystal growth & design* **2005**, *5*, 959–968.

(17) Kim, M.-H.; Shin, H.-S.; Shin, D.; Sun, Y.-K. Synthesis and electrochemical properties of Li [Ni0. 8Co0. 1Mn0. 1] O2 and Li [Ni0. 8Co0. 2] O2 via co-precipitation. *Journal of Power Sources* **2006**, *159*, 1328–1333.

(18) Zhang, S. Characterization of high tap density Li [Ni1/3Co1/3Mn1/3] O2 cathode material synthesized via hydroxide co-precipitation. *Electrochimica acta* **2007**, *52*, 7337–7342.

(19) Kim, J.-M.; Chang, S.-M.; Chang, J. H.; Kim, W.-S. Agglomeration of nickel/cobalt/manganese hydroxide crystals in Couette–Taylor crystallizer. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* **2011**, *384*, 31–39.

(20) Yang, Y.; Xu, S.; Xie, M.; He, Y.; Huang, G.; Yang, Y. Growth mechanisms for spherical mixed hydroxide agglomerates prepared by co-precipitation method: A case of $Ni_{1/3}Co_{1/3}Mn_{1/3}(OH)_2$. *Journal of Alloys and Compounds* **2015**, *619*, 846–853.

(21) Ngoepe, N.; Gutierrez, A.; Barai, P.; Chen, J.; Ngoepe, P. E.; Croy, J. R. The effects of process parameters on the properties of manganese-rich carbonate precursors: A study of co-precipitation synthesis using semi-batch reactors. *Chemical Engineering Science* **2021**, *241*, 116694.

(22) Para, M. L.; Querio, A.; Amici, J.; Versaci, D.; Barresi, A. A.; Bodoardo, S.; Marchisio, D. Electrochemical performance optimization of NMC811 through the structure design of its precursor. *Journal of Electroanalytical Chemistry* **2023**, 117630.

(23) Rane, C. V.; Ganguli, A. A.; Kalekudithi, E.; Patil, R. N.; Joshi, J. B.; Ramkrishna, D. CFD simulation and comparison of industrial crystallizers. *The Canadian Journal of Chemical Engineering* **2014**, *92*, 2138–2156.

(24) Ridder, B. J.; Majumder, A.; Nagy, Z. K. Population balance model-based multiobjective optimization of a multisegment multiaddition (MSMA) continuous plug-flow antisolvent crystallizer. *Industrial & Engineering Chemistry Research* **2014**, *53*, 4387–4397.

(25) Rasche, M. L.; Jiang, M.; Braatz, R. D. Mathematical modeling and optimal design of multi-stage slug-flow crystallization. *Computers & Chemical Engineering* **2016**, *95*, 240–248.

(26) Bałdyga, J.; Bourne, J. R. *Turbulent mixing and chemical reactions*; John Wiley & Sons, 1999.

(27) Cuoci, A.; Frassoldati, A.; Faravelli, T.; Ranzi, E. Numerical modeling of laminar flames with detailed kinetics based on the operator-splitting method. *Energy & fuels* **2013**, *27*, 7730–7753.

(28) Maffei, T.; Gentile, G.; Rebughini, S.; Bracconi, M.; Manelli, F.; Lipp, S.; Cuoci, A.; Maestri, M. A multiregion operator-splitting CFD approach for coupling microkinetic modeling with internal porous transport in heterogeneous catalytic reactors. *Chemical Engineering Journal* **2016**, *283*, 1392–1404.

(29) Bracconi, M.; Maestri, M.; Cuoci, A. In situ adaptive tabulation for the CFD simulation of heterogeneous reactors based on operator-splitting algorithm. *AIChE Journal* **2017**, *63*, 95–104.

(30) Buffo, A.; Vanni, M.; Marchisio, D. L. Simulation of a reacting gas–liquid bubbly flow with CFD and PBM: Validation with experiments. *Applied Mathematical Modelling* **2017**, *44*, 43–60.

(31) Smith, L.; Bull, M. Development of mixed mode MPI/OpenMP applications. *Scientific Programming* **2001**, *9*, 83–98.

(32) Rabenseifner, R.; Hager, G.; Jost, G.; Keller, R. Hybrid MPI and OpenMP parallel programming. 13th European PVM/MPI User's Group conference. 2006; p 11.

(33) Sampat, C.; Bettencourt, F.; Baranwal, Y.; Paraskevakos, I.; Chaturbedi, A.; Karkala, S.; Jha, S.; Ramachandran, R.; Ierapetritou, M. A parallel unidirectional coupled DEM-PBM model for the efficient simulation of computationally intensive particulate process systems. *Computers & Chemical Engineering* **2018**, *119*, 128–142.

(34) Bettencourt, F. E.; Chaturbedi, A.; Ramachandran, R. Parallelization methods for efficient simulation of high dimensional population balance models of granulation. *Computers & Chemical Engineering* **2017**, *107*, 158–170.

(35) Yakubov, S.; Cankurt, B.; Abdel-Maksoud, M.; Rung, T. Hybrid MPI/OpenMP parallelization of an Euler–Lagrange approach to cavitation modelling. *Computers & Fluids* **2013**, *80*, 365–371.

(36) Lichtenegger, T.; Pirker, S. Recurrence CFD–a novel approach to simulate multiphase flows with strongly separated time scales. *Chemical Engineering Science* **2016**, *153*, 394–410.

(37) Bermingham, S. K.; Kramer, H. J.; van Rosmalen, G. M. Towards on-scale crystalliser design using compartmental models. *Computers & chemical engineering* **1998**, *22*, S355–S362.

(38) Zauner, R.; Jones, A. G. Scale-up of continuous and semibatch precipitation processes. *Industrial & engineering chemistry research* **2000**, *39*, 2392–2403.

(39) Zhao, W.; Buffo, A.; Alopaeus, V.; Han, B.; Louhi-Kultanen, M. Application of the compartmental model to the gas–liquid precipitation of CO2-Ca (OH) 2 aqueous system in a stirred tank. *AIChE Journal* **2017**, *63*, 378–386.

(40) Tajsoleiman, T.; Spann, R.; Bach, C.; Gernaey, K. V.; Huusom, J. K.; Krühne, U. A CFD based automatic method for compartment model development. *Computers & Chemical Engineering* **2019**, *123*, 236–245.

(41) Darand, J.; Jafarian, A.; Tizbin, S. A hybrid compartmental-CFD model to investigate forced circulation crystallizer performance. *Desalination* **2022**, *533*, 115743.

(42) Shiea, M.; Querio, A.; Buffo, A.; Boccardo, G.; Marchisio, D. Simulation of Ni-Mn-Co hydroxide co-precipitation by the CFD-PBE approach. `https://github.com/mulmopro/wet-synthesis-route/tree/main/cfd\_pbe\_fluent\_udf`, 2021.

(43) Lee, M.-H.; Kang, Y.-J.; Myung, S.-T.; Sun, Y.-K. Synthetic optimization of $Li[Ni_{1/3}Co_{1/3}Mn_{1/3}]O_2$ via co-precipitation. *Electrochimica Acta* **2004**, *50*, 939–948.

(44) Van Bommel, A.; Dahn, J. Analysis of the growth mechanism of coprecipitated spherical and dense nickel, manganese, and cobalt-containing hydroxides in the presence of aqueous ammonia. *Chemistry of Materials* **2009**, *21*, 1500–1503.

(45) Marchisio, D. L.; Pikturna, J. T.; Fox, R. O.; Vigil, R. D.; Barresi, A. A. Quadrature method of moments for population-balance equations. *AIChE Journal* **2003**, *49*, 1266–1276.

(46) Fox, R. O. On the relationship between Lagrangian micromixing models and computational fluid dynamics. *Chemical Engineering and Processing: Process Intensification* **1998**, *37*, 521–535.

(47) Gavi, E.; Marchisio, D.; Barresi, A. On the importance of mixing for the production of nanoparticles. *Journal of dispersion science and technology* **2008**, *29*, 548–554.

(48) Sierra-Pallares, J.; Marchisio, D. L.; Alonso, E.; Parra-Santos, M. T.; Castro, F.; Cocero, M. J. Quantification of mixing efficiency in turbulent supercritical water hydrothermal reactors. *Chemical engineering science* **2011**,

(49) Soos, M.; Marchisio, D. L.; Sefcik, J. Assessment of gel formation in colloidal dispersions during mixing in turbulent jets. *AIChE Journal* **2013**, *59*, 4567–4581.

(50) Isaacson, E.; Keller, H. B. *Analysis of numerical methods*; Wiley, 1966.

(51) Mersmann, A. Physical and chemical properties of crystalline systems. *Crystallization technology handbook. 2nd ed. New York: Marcel Dekker, Inc* **2001**, 1–44.

(52) Bromley, L. A. Thermodynamic properties of strong electrolytes in aqueous solutions. *AIChE journal* **1973**, *19*, 313–320.

(53) Strang, G. On the construction and comparison of difference schemes. *SIAM journal on numerical analysis* **1968**, *5*, 506–517.

(54) Hindmarsh, A. C.; Serban, R.; Balos, C. J.; Gardner, D. J.; Reynolds, D. R.; Woodward, C. S. User Documentation for CVODE. 2023; v6.7.0.

(55) Hindmarsh, A. C.; Brown, P. N.; Grant, K. E.; Lee, S. L.; Serban, R.; Shumaker, D. E.; Woodward, C. S. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)* **2005**, *31*, 363–396.

(56) Gardner, D. J.; Reynolds, D. R.; Woodward, C. S.; Balos, C. J. Enabling new flexibility in the SUNDIALS suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)* **2022**,

(57) Gropp, W. D.; Kaushik, D. K.; Keyes, D. E.; Smith, B. F. High-performance parallel implicit CFD. *Parallel Computing* **2001**, *27*, 337–362.

(58) Liu, H.; Tafti, D. K.; Li, T. Hybrid parallelism in MFIX CFD-DEM using OpenMP. *Powder technology* **2014**, *259*, 22–29.

(59) Giovannini, M.; Marconcini, M.; Arnone, A.; Dominguez, A. A hybrid parallelization strategy of a cfd code for turbomachinery applications. 11 th European Conference on Turbomachinery Fluid dynamics & Thermodynamics. 2015.

(60) Jin, H.; Jespersen, D.; Mehrotra, P.; Biswas, R.; Huang, L.; Chapman, B. High performance computing using MPI and OpenMP on multi-core parallel systems. *Parallel Computing* **2011**, *37*, 562–575.

(61) Rigopoulos, S.; Jones, A. A hybrid CFD—reaction engineering framework for multiphase reactor modelling: basic concept and application to bubble column reactors. *Chemical Engineering Science* **2003**, *58*, 3077–3089.

(62) Bezzo, F.; Macchietto, S. A general methodology for hybrid multizonal/CFD models: Part II. Automatic zoning. *Computers & chemical engineering* **2004**, *28*, 513–525.

(63) Pirker, S.; Lichtenegger, T. Efficient time-extrapolation of single-and multiphase simulations by transport based recurrence CFD (rCFD). *Chemical Engineering Science* **2018**, *188*, 65–83.

(64) Pirker, S.; Lichtenegger, T. Process control of through-flow reactor operation by real-time recurrence CFD (rCFD) simulations–Proof of concept. *Chemical Engineering Science* **2019**, *198*, 241–252.

(65) Pirker, S.; Puttinger, S.; Rössler, R.; Lichtenegger, T. Steel Alloy Homogenization During Rheinsahl–Heraeus Vacuum Treatment: Conventional Computational Fluid Dynamics, Recurrence Computational Fluid Dynamics, and Plant Observations. *steel research international* **2020**, *91*, 2000214.

(66) Hindmarsh, A. C. In *Scientific computing*; Stepleman, R. S., Ed.; North-Holland, 1983; pp 55–64.

(67) Blashfield, R. K.; Aldenderfer, M. S. The literature on cluster analysis. *Multivariate behavioral research* **1978**, *13*, 271–295.

(68) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blon-

del, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; others Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* **2011**, *12*, 2825–2830.

(69) Ward Jr, J. H. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association* **1963**, *58*, 236–244.