MODELING AND SIMULATION OF ANTISOLVENT CRYSTALLIZATION: MIXING AND CONTROL

WOO, XING YI

NATIONAL UNIVERSITY OF SINGAPORE

&

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

2007

MODELING AND SIMULATION OF ANTISOLVENT CRYSTALLIZATION: MIXING AND CONTROL

WOO, XING YI

B. Eng (Hons.), National University of Singapore

A THESIS SUBMITTED

FOR THE DEGREE OF PHILOSOPHY IN ENGINEERING

DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

&

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

2007

Acknowledgements

Firstly, my deepest gratitude goes to my advisors, Professor Richard D. Braatz and Associate Professor Reginald B. H. Tan. They have not only taught me knowledge in the area of crystallization, modeling, and control, but have also imparted me scientific research and problem solving skills, as well as a positive attitude in carrying out challenging research in the pursuit of good science. I deeply appreciate the encouragement they have given throughout these years, especially those kind advices that will stay important throughout my future career.

I would like to thank my PhD committee, Professor Shamsuzzaman Farooq, Professor Jonathan J. L. Higdon, and Dr. Paul J. A. Kenis for their suggestions and advice for my PhD research. The NUS-UIUC joint PhD program has given me an unique graduate school experience, this would not have been possible without the research scholarship from A*STAR, and efforts of the faculty and staff from both Departments of Chemical and Biomolecular Engineering, especially Professor Koon Gee Neoh, Associate Professor Reginald B. H. Tan, Professor Charles F. Zukoski, Professor Edmund G. Seebauer, Dr. Ashgar A. Mirarefi, and Mdm Cheok Bee Khim. I had the opportunity to gain valuable teaching experiences with Professor Richard D. Braatz, Professor Shamsuzzaman Farooq, Dr. Marina Milectic, Dr. Rajarathnam Dharmarajan and Mdm Jamie Siew. I would also like to thank Professor Richard C. Alkire for sharing his computing resources and for inviting me to the wonderful parties at his house, and Dr. Brian K. Johnson for sharing his PhD work. Much of my research work would not be completed without the computational resources from the National

Center for Supercomputing Applications (NCSA) in Illinois and the Institute of Chemical and Engineering Sciences (ICES) in Singapore, as well as the technical support provided by them.

I deeply treasure the help and friendship of my fellow group mates from UIUC and ICES. Ann, for guiding me through the first steps in research, Chin Lee, for the nights and weekends we spent in the lab, Xiaohai, for helping me with all sorts of computer problems, Effendi, for giving me ideas in solving numerical problems, Mitsuko, for making many suggestions for my research and manuscripts, Mo, for showing me how to use the NCSA machines and the crazy parties, Charlotte, for the lunches, dinners, beer, long chats and friendship, Shuyi, for listening to me and giving me a chance to share my ideas, as well as Rudi, Eric, Kim Seng, Zaiqun, Nicholas, Li May, Juan, Guangwen, and Sendhil. There are also many more new friends whom I have got to known in my short stay in Urbana-Champaign and ICES, and everyone of you have added happy moments to my graduate school memories.

To my family and friends whom I've known for years, I know I have neglected all of you while trying to put as much time as possible into work and research. I am thankful for all the understanding, support and love that you have given me. To my dearest Matthew, I could not imagine how this journey would be like without you. Though we are not sure what the future holds, but I am excited to pursue our many dreams, and to overcome any difficulties together.

Lastly, I would like to thank God, for the many gifts He has given me. I continue to pray for the strength to do His will and to serve others through my work.

Contents

Chapter 3 Theory

Chapter 4 Simulation of Antisolvent Crystallization in an Agitated Tank

Chapter 5 Simulation of Competitive Reactions in Confined Impinging Jet Reactors

Chapter 6 Modeling of Impinging Jet Crystallization

Chapter 7 Concentration Control of Antisolvent Crystallization with Laser Backscattering Measurement

Chapter 8 Precise Tailoring of the Crystal Size Distribution by Optimal Control of Impinging Jet Crystallizers

Summary

In the pharmaceutical industry, both company internal and regulatory authorities impose stringent requirements on the product quality, which includes crystal size distribution, of active pharmaceutical ingredients (APIs) obtained from crystallization processes. In addition, the development of the crystallization process for a given API includes the design of control strategies to ensure the crystal product meets the demands of the drug administration method and the bioavailability, as well as the required physical attributes for the efficiency of downstream processes (e.g., filtration and drying).

The design of crystallization processes becomes more complicated if mixing has an effect on the final crystal product quality (e.g., crystal size distribution and polymorphic form). Such mixing effects are more apparent in antisolvent and reactive crystallizations, which involve the blending of different fluids, and in large-scale crystallizers, where homogeneity cannot be easily achieved. Hence, it is necessary to develop tools to understand the interactions between hydrodynamics and the kinetics of crystallization in order to develop appropriate design methodologies.

The first part of the dissertation presents the development of an integrated algorithm, which couples macromixing and micromixing models with the population balance equation. It is applied to simulate the antisolvent crystallization in a stirred vessel and impinging jet crystallizers. The dependency of the crystal size distribution on the mixing speed, addition mode, and scale for a stirred vessel, and the effects of jet

velocity on the crystal size distribution and polymorphic form for an impinging jet crystallizer, were numerically investigated.

For the crystallization of paracetamol in a stirred tank considering primary nucleation and growth, larger crystals were observed for higher agitation speeds. In addition, smaller crystals with a narrower size distribution were observed for the revered addition of saturated solution into antisolvent, and similar crystal size distributions were observed for scaling up with constant tip speed and constant power per volume. The simulation of impinging jet crystallization using lovastatin as a model system predicted the formation of larger crystals with lower jet velocities. For the crystallization of L-histidine polymorphs, the ratio of polymorphs was observed to be affected by the jet velocities.

The goal of such computational tools is to enable the numerical determination of the crystal size distribution and polymorphic form for a wide range of operating conditions for a given set of crystallizer designs and control schemes. Subsequently, the mixer, vessel internal design, and operating conditions which result in the desired crystal size distribution and polymorph form could be determined. This systematic design approach would be especially useful for scale-up, where the product quality must be maintained at the industrial scale. In addition, the use of numerical simulations to design crystallization processes would significantly reduce the amount of API required for experiments to arrive at a robust design.

Besides mixing, the quality and consistency of the crystal product can be improved by applying various control strategies. The second part of the dissertation focuses on the

theoretical development and analysis of control strategies applied to antisolvent crystallization in semibatch stirred tanks and impinging jet crystallizers. For batch crystallization, the advantage of concentration control over the specification of antisolvent addition rate is illustrated by the insensitivity of concentration control to a number of process disturbances and parameter uncertainties. However, for disturbances and uncertainties that cause excessive nucleation with concentration control, it is shown that crystal count measurements (e.e. laser backscattering) can be used to detect the onset of nucleation, so that the deviations in the product quality can be subsequently reduced by adjusting the supersaturation profile.

For impinging jet crystallizers, it is shown that, by coupling it with an aging vessel of controlled growth, specific crystal size distributions can be tailored. The first approach is to use the crystals with the narrowest distribution from the impinging jet and seed them into an aging vessel according to a optimal seeding profile. The second approach is to operate the impinging jet according to a optimal jet velocity profile while the crystals are sent directly into an aging vessel. The numerical evaluation of both approaches concludes that a wide variety of crystal size distributions can be targeted. This potentially allows the crystal size distribution of an API to be first designed based on its drug administration requirements, followed by design of the crystallization process using the proposed control strategy to give the desired crystal size distribution. This also avoids the need for milling which poses a series of problems.

The overall contribution of the dissertation is the development of various simulation tools to help process engineers address the issues of mixing and control in antisolvent

crystallization processes. These simulation tools can be used to develop a better understanding of the crystallization process of different systems, and when coupled with carefully designed experiments, the design of crystallization process for different systems can be executed in a systematic and scientific manner.

List of Tables

List of Figures

Nomenclature

A_c	Crystal surface area [m^2]
b	Nucleation rate exponent
B	Nucleation rate [#/m^3 s]
c	Concentration [kg/m^3 or kg/kg]
c^*	Solubility or saturation concentration [kg/m^3 or kg/kg]
Δc	Supersaturation [kg/m^3 or kg/kg]
c_i	Interfacial concentration [kg/m^3]
$D,\ D_m$	Diffusion coefficient or laminar diffusivity [m^2/s]
d_p	Particle size [m]
D_t	Turbulent diffusivity [m^2/s]
f	Number density function [#/m]
F	Target number density function [#/m]
f_{jet}	Production rate of crystals from impinging jets [#/μm s]
f_r	Derivative of number density function [#/m^2]
f_w	Mass density function [kg/m]
g	Growth rate exponent
$\vec{g}$	Gravitational acceleration [m/s^2]
G	Growth rate [m/s]
i	Exponent for solute integration
k	Turbulent kinetic energy [m^2/s^2] in turbulence and micromixing equations Boltzmann's constant in nucleation rate expression
K	Tradeoff ratio between growth and nucleation rates [(μm/s)/(#/m^3-s)]

k_1, k_2	Reaction rate constant [m^3/mol s]
k_b	Nucleation rate prefactor [#/m^3 s $(kg/kg)^b$]
k_a	Area shape factor
k_d	Mass transfer coefficient [m/s]
k_g	Growth rate prefactor [m/s $(kg/kg)^g$]
k_i	Integration rate constant [m^{3i-1}/kg^{i-1} s]
k_v	Volume shape factor
L	Crystal size [μm]
m	Mass [kg]
$\dot{m}$	Mass flow rate [kg/s]
N_A	Avogadro's number
p	Pressure [Pa] in momentum conservation equation Volume fraction of environment in micromixing model
r	Crystal size [m]
r_0	Nuclei size [m]
Δr	Discretized bin size for crystal size [m]
Re	Reynolds number
$\langle \mathbf{s} \rangle$	Weight concentration of scalar
S	Relative supersaturation
S_j	Parameter sensitivity
Sc	Schmidt number
Sc_t	Turbulent Schmidt number
Sh	Sherwood number
t	Time [s]
T	Temperature [^oC]
t_I	Induction time [s]

t_M	Micromixing time [s]
t_s	Sampling time [s]
v	Molar volume in nucleation rate expression [m^3/mol] Jet velocity of impinging jets [m/s]
$\vec{v}$	Velocity vector [m/s]
V	Volume fraction of antisolvent
w	Antisolvent mass per cent [%] Weight of crystals from impinging jets [kg]
$\mathbf{x}$	Spatial position vector [m]
X	Reaction conversion
X_A	Fraction of polymorph A

Symbols

α	Scalar
β	Geometric shape factor
ε	Turbulent kinetic energy dissipation rate [m^2/s^3]
ε_ξ	Scalar dissipation rate [1/s]
ϕ	Volume fraction of solids in effective viscosity expression
ϕ_k	Scalar
$\langle\phi\rangle$	Mean composition of scalar in environment
γ	Interfacial tension [N/m]
λ_k	Kolmogoroff length scale
μ	Viscosity [kg/m s] Effective viscosity of suspension [kg/m s] in effective viscosity expression
μ_n	n^{th} moment of number density function
μ_s	Viscosity of suspending medium [kg/m s]
μ_t	Viscosity [kg/m s]

θ	Constant in minmod limiter
θ_j	Parameter value
ρ	Density [kg/m^3]
ρ_c	Crystal density [kg/m^3]
$\bar{\bar{\tau}}$	Stress tensor [kg/m s^2]
ν	Kinematic viscosity [m^2/s]
$\langle \xi \rangle$	Mixture fraction
$\langle \xi'^2 \rangle$	Mixture fraction variance

Subscripts

i	Crystal dimension in population balance equation Jet velocity of impinging jet Instance for dropping seed crystals
j	Discretized bin for crystal size for population balance equation Polymorphic form in crystallization kinetic expressions of polymorphs
n	Environment in micromixing model Order of moment

Chapter 1
Introduction

1.1 Background and Motivation

A common product specification for crystallization processes is large, uniform crystals to ensure the efficiency of subsequent downstream processes like filtration and drying. However, other than crystal growth, kinetic processes and rates like nucleation (primary and secondary), attrition and aggregation can result in wide particle distributions, as well as multi-modal distributions. As the kinetic processes and rates can vary significantly for different crystallization systems, the development of crystallization processes in the pharmaceutical industries rely primarily on numerous laboratory experiments, conducted on a trial-and-error or factorial design basis, to identify an optimal operating recipe. However, as mixing and particle-particle interactions change with scale (Green, 2002), the operating recipe is no longer optimal on scale up. This either calls for a redesign of the process, which again requires a large number of experiments, or operating the process using the non-optimal recipe.

Over the years, considerable research effort has been put into developing crystallization processes design onto a more scientific and engineering basis. This includes development of inline sensors for the measurement of accurate and reliable process data (Barrett et al., 2005; Birch et al., 2005), advancement in modeling and simulation software (Braatz et al., 2002), application of optimization and control strategies to give consistent and desired particle size distributions and polymorphic

forms (Braatz, 2002; Fujiwara et al., 2005; Larsen et al., 2006), and accurate estimation of crystallization kinetics (Rawlings et al., 1993). More detailed review of these research developments will be covered in subsequent chapters.

While many of these engineering strategies have been applied to actual pharmaceutical crystallization processes (Togkalidou et al., 2004; Zhou et al., 2006), the systematic and reliable design of crystallizers in-silico has remained an open problem. In addition, the objectives for crystallization control are usually focused on reducing fines and producing large crystals of unimodal distribution, with additional milling is used to reduce the crystals to the size to give the desired dissolution rate and bioavailability. However, there are numerous operational problems related to milling. While it is possible to design the crystal product to meet specified bioavailability requirements, the control of the crystallization process to tailor a specified crystal size obtained from product design distribution has been an open problem.

1.2 Goal and Objectives

Based on past research efforts, the main research goal in the area of crystallization is to develop systematic and scientific design approaches for crystallization processes. With a combination of simulations and experiments, an in-depth understanding can be established for any crystallization systems, and subsequently, an optimal crystallizer design and control strategy can be developed to produce crystals of the desired crystal size distribution. This would ultimately improve the efficiency of pharmaceutical crystallization process development.

In working towards this research goal, several objectives have been established for this dissertation. The focus will be in the theoretical and computational studies in the area of mixing and control, while the insights gained from theory can provide a basis for experimental design and data collection.

1. Develop state-of-the-art simulation algorithms for modeling the full crystal size distributions in crystallizers taking into account different scales of fluid mixing.
2. Conduct numerical experiments using the simulation algorithms to investigate the effects of various operating parameters in different crystallizer configurations.
3. Conduct robustness analysis for different control strategies.
4. Develop new control strategies to target crystals of any specified distribution.

Antisolvent crystallization systems are the application focus here, due to potential high sensitivity to mixing, and their control is not well-studied, compared to cooling crystallization.

1.3 Organization of Dissertation

According to the objectives, the dissertation will be divided into two areas, mixing (Chapters 3 to 6) and control (Chapters 7 and 8). The subsequent chapters are organized as follows.

Chapter 2 Literature Review

This chapter reviews the experimental studies for antisolvent crystallization systems, the modeling of crystallization processes for the application of

mixing studies, model identification and control, and the development of the impinging jet crystallizer.

Chapter 3 Theory

The chapter covers the macromixing (computational fluid dynamics, CFD) and micromixing models as well as the numerical solution for population balance equations. The nucleation and growth kinetic models and the expressions for effective viscosity in suspensions are also presented. All the models are coupled together and solved within the CFD solver for the modeling of crystallizers.

Chapter 4 Simulation of Antisolvent Crystallization in an Agitated Tank

This chapter illustrates the application of the coupled CFD-micromixing-population balance model to simulate the antisolvent crystallization process in a semibatch stirred tank. The effects of agitation speed, addition mode and scale up on the final crystal size distribution are numerically investigated.

Chapter 5 Simulation of Competitive Reactions in Confined Impinging Jet Reactors

This chapter presents the simulation of reactions in confined impinging jet reactors using the CFD-micromixing model described in Chapter 3, and the simulated results are compared with experimental data reported in literature. Sensitivity analysis is used to improve the simulation parameters to give better predictions.

Chapter 6 Modeling of Impinging Jet Crystallization

This chapter shows the application of the CFD-micromixing-population balance model to simulate antisolvent crystallization in confined impinging jets using the improved parameters in Chapter 5. The effect of jet velocity on crystal size distribution and polymorphic forms are presented.

Chapter 7 Concentration Control of Antisolvent Crystallization with Laser Backscattering Measurement

This chapter compares antisolvent addition rate control and concentration control for antisolvent crystallization in terms of sensitivity to disturbances.

Chapter 8 Precise Tailoring of the Crystal Size Distribution by Optimal Control of Impinging Jet Crystallizers

This chapter explores different control strategies to precisely produce crystals of a target distribution by combining the impinging jets and the stirred tank crystallizer with concentration control.

Chapter 9 Conclusions and Future Directions

This final chapter summarizes the key scientific accomplishments achieved in this dissertation and the possible future research directions based on the current developments and findings.

Appendices

The examples of the CFD-micromixing-population balance algorithm (Chapters 4 and 6), which are implemented as user-defined functions, are included in the appendices. These codes are to be linked with the corresponding Fluent files of the examples.

1.4 References

Barrett, P., Smith, B., Worlitschek, J., Bracken, V., O'Sullivan, B. and O'Grady, D. (2005). A review of the use of process analytical technology for the understanding and optimization of production batch crystallization processes. *Organic Process Research & Development* **9**(3): 348-355.

Birch, M., Fussell, S. J., Higginson, P. D., McDowall, N. and Marziano, I. (2005). Towards a PAT-based strategy for crystallization development. *Organic Process Research & Development* **9**(3): 360-364.

Braatz, R. D. (2002). Advanced control of crystallization processes. *Annual Reviews in Control* **26**(1): 87-99.

Braatz, R. D., Fujiwara, M., Ma, D. L., Togkalidou, T. and Tafti, D. K. (2002). Simulation and new sensor technologies for industrial crystallization: A review. *International Journal of Modern Physics B* **16**(1-2): 346-353.

Fujiwara, M., Nagy, Z. K., Chew, J. W. and Braatz, R. D. (2005). First-principles and direct design approaches for the control of pharmaceutical crystallization. *Journal of Process Control* **15**(5): 493-504.

Green, D. (2002). Crystallizer Mixing: Understanding and Modeling Crystallizer Mixing and Suspension Flow. *Handbook of Industrial Crystallization*. Boston, Butterworth-Heinemann**:** 181-200.

Larsen, P. A., Patience, D. B. and Rawlings, J. B. (2006). Industrial crystallization process control. *IEEE Control Systems Magazine* **26**(4): 70-80.

Rawlings, J. B., Miller, S. M. and Witkowski, W. R. (1993). Model identification and control of solution crystallization processes - a review. *Industrial & Engineering Chemistry Research* **32**(7): 1275-1296.

Togkalidou, T., Tung, H. H., Sun, Y., Andrews, A. T. and Braatz, R. D. (2004). Parameter estimation and optimization of a loosely bound aggregating pharmaceutical crystallization using in situ infrared and laser backscattering measurements. *Industrial & Engineering Chemistry Research* **43**(19): 6168-6181.

Zhou, G. X., Fujiwara, M., Woo, X. Y., Rusli, E., Tung, H. H., Starbuck, C., Davidson, O., Ge, Z. H. and Braatz, R. D. (2006). Direct design of pharmaceutical antisolvent crystallization through concentration control. *Crystal Growth & Design* **6**(4): 892-898.

Chapter 2
Literature Review

2.1 Introduction

This chapter reviews the work performed by both academic and industrial researchers in the area of antisolvent crystallization and modeling of crystallization processes. In particular, the advancement of simulation methods for modeling different mixing scales in crystallizers and for developing different control systems is described.

2.2 Antisolvent Crystallization

Antisolvent crystallization is used widely in the pharmaceutical industry. This enables the crystallization of thermally sensitive pharmaceuticals without introducing large temperature changes in the process (Mullin, 2001; Wey and Karpinski, 2002). Current state-of-the-art crystallization technology such as impinging jet crystallizers (see Figure 2.1) utilizes high intensity mixing of the antisolvent and the solution to produce crystals smaller than 25 μm with improved bioavailability and increased dissolution rates (Lindrud et al., 2001; Mahajan and Kirwan, 1996; Midler et al., 1994), which, at the same time, reduces the undesirable effects of milling (am Ende and Brenek, 2004; Leung et al., 1998). Various experimental studies of antisolvent crystallization in an agitated semibatch vessel (see Figure 2.2) indicate that the crystal size distribution (CSD) depends strongly on the operating conditions, such as agitation rate, mode of addition (direct or reverse), addition rate, solvent composition, and size of the crystallizer

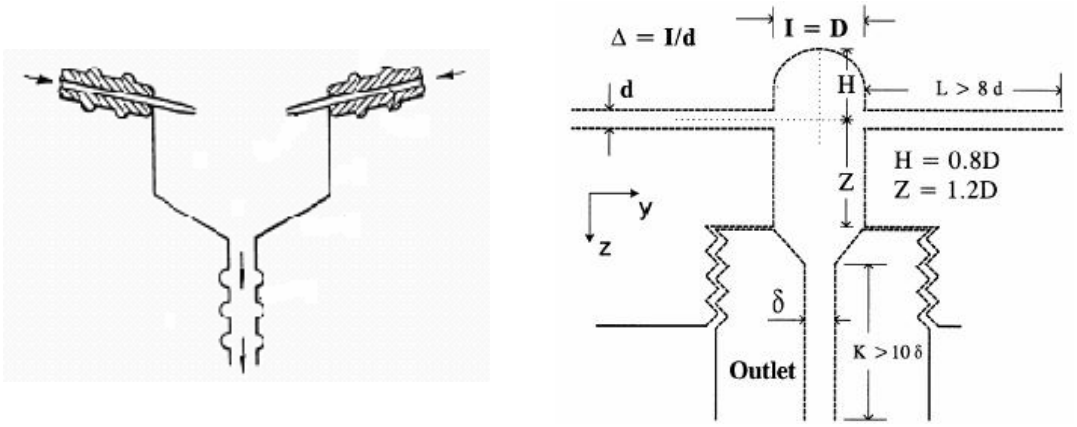

Figure 2.1 Schematics of impinging jet crystallizers. Left: free impinging jets (Midler et al., 1994). Right: confined impinging jets (Johnson and Prud'homme, 2003a).

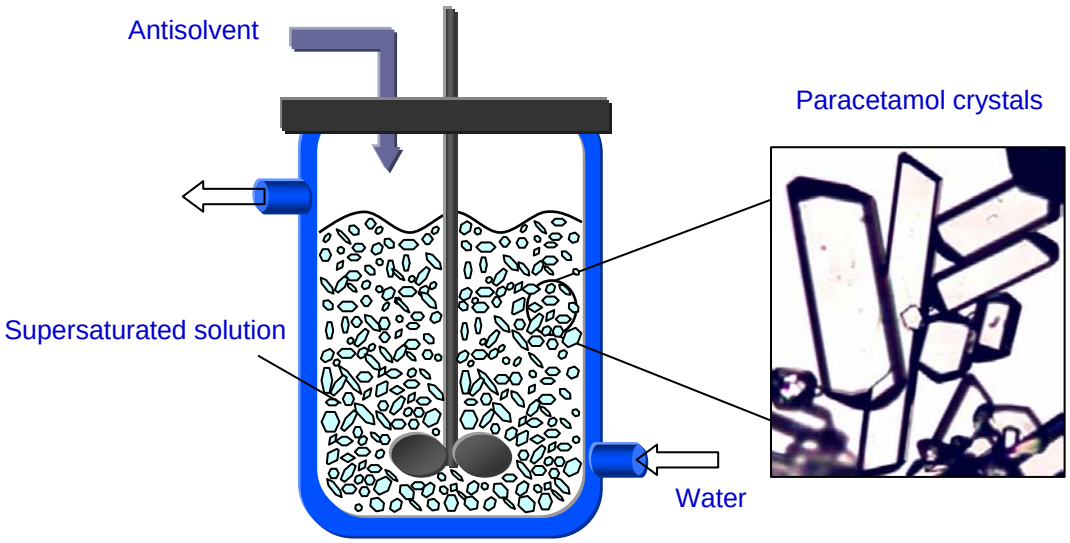

Figure 2.2 Schematic of semibatch antisolvent crystallization process in a stirred tank.

(Borissova et al., 2004; Budz et al., 1986; Charmolue and Rousseau, 1991; Doki et al., 2002; Granberg et al., 1999; Holmback and Rasmuson, 1999; Kaneko et al., 2002; Midler et al., 1994; Mullin et al., 1989; Mydlarz and Jones, 1991; Nyvlt and Zacek, 1986; Plasari et al., 1997; Shin and Kim, 2002; Takiyama et al., 1998). The polymorphic or pseudopolymorphic form can also depend on the operating conditions (Kim et al., 2003; Kitamura, 2002; Kitamura and Nakamura, 2002; Kitamura and Sugimoto, 2003; Okamoto et al., 2004; Schroer and Ng, 2003).

Most variations in the operating conditions have a direct influence on the mixing of the antisolvent and the solution, which affects the localized supersaturation and, thus, the crystal product. Because the dependence of nucleation and growth rates on supersaturation is highly system specific, determining the optimal process conditions that produce the desirable crystal product can require numerous bench-scale laboratory experiments, which might not be optimal after the scale-up of the crystallizer, as the mixing effects and spatial distribution of supersaturation can be vastly different (Green, 2002; Paul et al., 2004). In addition, control strategies developed on the basis of the assumption of perfect mixing may not result in the intended crystal product when implemented at the industrial scale (Ma et al., 2002c). A pressing issue for the pharmaceutical industry is the regulatory requirement of consistency in the various chemical and physical properties of the crystals, including the CSD (Paul et al., 2005). Such concerns motivate the development of a computational model to simulate the antisolvent crystallization process to quantify the effects of mixing on the product crystal characteristics such as the CSD, which determines the bioavailability of the drug and efficiency of downstream processes (e.g., filtration and drying) (Fujiwara et al., 2005).

2.3 Modeling of Crystallizers: Mixing

The modeling of well-mixed crystallizers involves the computation of the population balance equation (PBE) together with the material balance equations for each species in solution. Numerous numerical techniques that compute the full CSD have been used to model well-mixed batch, semibatch, or continuous crystallizers (Gerstlauer et al., 2002; Gunawan et al., 2004; Haseltine et al., 2005; Hill and Ng, 1997; Hounslow, 1990; Hounslow et al., 1988; Hu et al., 2004; Kumar and Ramkrishna, 1997; Ma et al., 2002b; Motz et al., 2002; Puel et al., 2003a, 2003b; Quintana-Hernandez et al., 2004; Wulkow et al., 2001). To account for nonideal mixing, the PBE has to be coupled with the transport equations of mass, momentum, and energy (Hulburt and Katz, 1964). One approach is to couple turbulent computational fluid dynamics (CFD) codes with the solution of the PBE, and most of the literature studies focus on reactive crystallization systems (Jaworski and Nienow, 2003; Van Leeuwen et al., 1996; Wei and Garside, 1997; Wei et al., 2001). A recent paper by Choi et al. (Choi et al., 2005) models the antisolvent crystallization process in a jet Y-mixer using a hybrid CFD-PBE approach, but neglects the micromixing effects. Compartmental modeling, where the crystallizer is divided into a number of well-mixed compartments connected by interchanging flows, is a less computational intensive approach (Kramer et al., 1996). One strategy is to compartmentalize the crystallizer into regions that are, to some degree, homogeneous in properties of interest (e.g., suspension density, energy dissipation, supersaturation), as determined by CFD simulations (Kougoulos et al., 2005). However, compartmental modeling oversimplifies the flow field and, most importantly, it loses the spatial resolution of the supersaturation and turbulent energy dissipation distribution in the crystallizer.

Subsequently, the effects of micromixing have been included in coupled CFD-PBE computations to model turbulent precipitators (Baldyga and Orciuch, 1997, 2001; Falk and Schaer, 2001; Marchisio et al., 2001a; Marchisio et al., 2001b; Marchisio et al., 2001c; Piton et al., 2000; Wang and Fox, 2003, 2004) (here, the term "precipitation" is reserved to refer to reactive crystallization), in which a variety of methods were used to approximate the probability density function (PDF) (Pope, 1985, 2000), which is a statistical description of the fluctuating scalars (e.g., species concentrations) at a subgrid scale. The solution of the PBE was obtained by the method of moments, which only computes the average and aggregate properties of the crystalline phase. Recently, a supercritical antisolvent crystallization process was modeled using this strategy (Henczka et al., 2005).

An alternative method used to include micromixing effects in precipitation models utilizes a multi-zonal approach in a Lagrangian framework (Baldyga and Bourne, 1999), in which the precipitator is divided into a few segregated zones (e.g., feed/reactant zone, mixed/reaction zone, contact zone, bulk zone). The volume change of the zones and the material exchange between the zones are determined by the meso- and micromixing rates (Phillips et al., 1999). The reduction in the computational expense, by eliminating the direct linkage to CFD computations, enabled the simulation of the PBE equation for the full CSD (Baldyga et al., 1995; Schwarzer and Peukert, 2004; Zauner and Jones, 2000a, 2000b, 2002). In some instances, additional approximations included the confinement of nucleation and crystal growth to certain zones. A variation of this approach by Kresta et al. (Kresta et al., 2005) used a multiscale Eulerian-Lagrangian framework to couple the zones in the

bulk fluid, governed by long time and length scales, with the discretized volumes of the feed plume, governed by short time and length scales.

Compartmental modeling, coupled to the solution of the PBE for the full particle size and shape distribution, also has been applied to cooling crystallization and polymerization processes (Alexopoulos et al., 2002). Recent publications show that it is possible to simulate the CSD, while taking into account the spatial distribution of the solid particles of different sizes (Ma et al., 2002a; Ma et al., 2002c; Sha and Palosaari, 2002), which is important when the crystalline phase is much denser than the solution. This was an advancement over the earlier works in modeling crystallizers that assumed that the solid particles follow the liquid streamlines, which avoided the use of multiphase models. In contrast, the coupling of CFD, PBE, and multiphase models has been an ongoing effort in the modeling of bubble size distribution as a result of coalescence and breakup in gas-liquid processes (e.g. bioreactor) (Dhanasekharan et al., 2005; Venneker et al., 2002).

2.4 Modeling of Crystallizers: Identification and Control

The advancement in simulation methods for crystallization processes, along with new in-situ measurement technologies for particle size distribution and solution concentration, has allowed more efficient model identification of crystallization processes, as well as a more in-depth evaluation of different control strategies (Braatz et al., 2002; Larsen et al., 2006). In identifying crystallization kinetics, the use of optimal model-based experimental design and parameter estimation allows the nucleation and growth kinetic parameters to be estimated systematically while minimizing the number of experiments required, as well as minimizing the

uncertainty of the parameter estimates (Chung et al., 2000; Gunawan et al., 2002; Togkalidou et al., 2004). Besides identifying model parameters for the subsequent development of model-based control strategies (Ma and Braatz, 2003), the iterative procedure of constructing a model also increases process understanding, and thus enabling process development to be carried out using systematic engineering approaches.

Following model identification, the model-based open-loop control of the crystallization process requires the optimization of process conditions (e.g. seed distribution and loading, temperature or antisolvent addition profile) based on the desired performance objective (e.g. minimize fines formation, maximize crystal size), subject to physical constraints of the process (e.g. maximum cooling rate or antisolvent flow rate, maximum batch time) (Chung et al., 1999; Kaneko et al., 2002; Matthews and Rawlings, 1998; Nagy and Braatz, 2003a, 2004; Sarkar et al., 2006; Worlitschek and Mazzotti, 2004). With improvements in computational power and optimization algorithms, the application of advanced control techniques on crystallization processes, with the potential to improve process performance, can be explored (Nagy and Braatz, 2003a). In addition, the robustness and sensitivity of the control trajectory, due to parameter and control implementation inaccuracies and practical process disturbances, can be quickly computed and analyzed to justify the need for further improvements before implementation on the actual process (Ma et al., 1999; Nagy and Braatz, 2003b).

With additional aggregation and breakage processes, the complexity of model development, identification and validation increases and it can be considered as a time

consuming task for process engineers developing the crystallization process. Alternatively, advances in in-situ measurement technology and automation of crystallization processes have enabled the direct design of crystallization batch recipes to follow a supersaturation profile, based on the solubility and metastable limit curves, by feedback control (Fujiwara et al., 2002; Fujiwara et al., 2005; Gron et al., 2003; Yu et al., 2006; Zhou et al., 2006). Despite the fact that the supersaturation profile is a sub-optimal one, the shorter development time, and the low sensitivities to practical disturbance and variation in kinetics of the closed-loop feedback control approach, are important advantages.

2.5 Impinging Jet Crystallizers

To date, most of the modeling and control studies of crystallization processes are focused on stirred-tank crystallizers. Impinging jet crystallization, which was developed more than a decade ago, is recognized as one of the most reliable approaches to produce small crystals with a narrow size distribution (Midler et al., 1994). The basic principle in this design is to utilize high intensity micromixing of fluids to achieve a homogeneous composition of high supersaturation before the onset of nucleation. (D'Aquino, 2004)This technology is now being used by various pharmaceutical companies in their commercial drug production (am Ende et al., 2003; D'Aquino, 2004; Dauer et al., 1996; Lindrud et al., 2001). The direct production of small uniform crystals of high surface area that meet the bioavailability and dissolution requirements can eliminate the need for milling, which causes problems like noise and dust issues, yield losses, long production times, polymorphic transformation, and amorphization (am Ende and Brenek, 2004; Bauer-Brandl, 1996a, 1996b; Leung et al., 1998). A narrow particle size distribution is especially important

for inhalation drugs, in which the specific size range would depend on the region of the human respiratory tract where the drug is targeted (Nagao et al., 2005; Shekunov and York, 2000; Taylor, 2002). Continuous crystallization using T-mixers and Y-mixers to produce small, uniform particles also have been experimentally and numerically studied (Choi et al., 2005; Haselhuhn and Kind, 2003; Schwarzer and Peukert, 2004; Schwarzer et al., 2006; Stahl et al., 2001).

Numerous experimental studies have been carried out by academic as well as industrial researchers to gain a deeper understanding of the impinging jet crystallization process so as to increase efficiency during process development and optimization. For antisolvent crystallization of lovastatin, Mahajan and Kirwan (1996) measured the dependency of the crystal size distribution (CSD) on the jet velocity and level of supersaturation in the impinging jet crystallizer operated in non-submerged mode. Hacherl et al. (2003) investigated the effects of jet velocity on the crystal size distribution and the proportion of hydrates formed for the reactive crystallization of calcium oxalate. Johnson and Prud'homme (2003b) experimentally investigated the dependency of the crystal size distribution on the jet velocity and the inlet concentrations for a confined and submerged impinging jet crystallizer in which amphiphilic diblock copolymers were added to inhibit crystal growth. Recently, Marchisio et al. (2006) applied the confined impinging jet for reactive precipitation, and reported on the dependence of particle size distribution with jet velocity and inlet reactant concentrations.

The mixing in impinging jets has been characterized by an overall micromixing time by Mahajan and Kirwan and Johnson and Prud'homme using competitive reactions

(Johnson and Prud'homme, 2003a; Mahajan and Kirwan, 1996). The micromixing times, correlated to the jet velocity, the fluid properties, and some geometrical aspects of the impinging jet crystallizer, can be used to establish scale-up criteria for impinging jets. The three-dimensional turbulent flow field in the impinging jet chamber also can be visualized experimentally (for example, by using laser-induced fluorescence, laser doppler anemometry, or particle image velocimetry) or simulated using computational fluid dynamics (CFD) (Stan and Johnson, 2001; Teixeira et al., 2005; Unger and Muzzio, 1999; Zhao and Brodkey, 1998). The modeling of reactions in impinging jets has been widely studied as well (Kusch et al., 1989; Liu and Fox, 2006; Santos et al., 2005; Sohrabi and Marvast, 2000). The extension of experimental visualization and simulations that include the crystal nucleation and growth would offer a deeper understanding of impinging jet crystallization, which could speed up the design of impinging jets and reduce the time required to identify the operating conditions that produces the desired crystal size distribution. Consequently, the time required to bring a new drug to the market could be reduced.

2.6 Experimental Validations

The simulation of mixing effects using CFD and micromixing models have been frequently validated using competitive parallel reactions for different configurations (Akiti and Armenante, 2004; Liu and Fox, 2006; Tsai et al., 2002; Vicum et al., 2004). This typically involves comparing the conversion of the slower reaction at the outlet of a continuous process or at the end of a batch process. Alternatively, the velocity and concentration fields can be validated using particle image velocimetry (PIV) and planar laser-induced fluorescence (PLIF) (Feng et al., 2005).

As mentioned earlier, most of the past works on coupling CFD-micromixing simulations with the population balance equation used the method of moments to predict the mean and aggregate properties of the crystal product. Some authors have reported experimental validation of mean particle sizes or number density at the outlet of a continuous process or at the end of a batch process (Marchisio et al., 2001a; Shekunov et al., 2001; Zauner and Jones, 2000a). Nonetheless, a full spatial and temporal validation of crystal size distribution has yet to be reported.

2.7 Conclusions

In the past ten years, a large quantity of publications in the crystallization literature was focused on modeling, estimation of kinetics and control, as well as the development of in-situ measurement technology. Evidently, a major effort of both academic and industrial researchers, along with FDA's support, is geared towards developing crystallization process design as a scientific and engineering discipline, and reducing the number of trial-and-error experiments required to arrive at a optimal and robust design.

2.8 References

Akiti, O. and Armenante, P. M. (2004). Experimentally-validated micromixing-based CFD model for fed-batch stirred-tank reactors. *AIChE Journal* **50**(3): 566-577.

Alexopoulos, A. H., Maggioris, D. and Kiparissides, C. (2002). CFD analysis of turbulence non-homogeneity in mixing vessels - A two-compartment model. *Chemical Engineering Science* **57**(10): 1735-1752.

am Ende, D. J. and Brenek, S. J. (2004). Strategies to control particle size during crystallization processes. *American Pharmaceutical Review* **7**(3): 98-104.

am Ende, D. J., Crawford, T. C. and Weston, N. P. (2003). Reactive crystallization method to improve particle size. Pfizer, Inc. and Pfizer Products, Inc. *Patent, U. S.* 6,558,435.

Baldyga, J. and Bourne, J. R. (1999). *Turbulent Mixing and Chemical Reactions*. Chichester, U. K., John Wiley & Sons Ltd.

Baldyga, J. and Orciuch, W. (1997). Closure problem for precipitation. *Chemical Engineering Research & Design* **75**(A2): 160-170.

Baldyga, J. and Orciuch, W. (2001). Some hydrodynamic aspects of precipitation. *Powder Technology* **121**(1): 9-19.

Baldyga, J., Podgorska, W. and Pohorecki, R. (1995). Mixing-precipitation model with application to double feed semibatch precipitation. *Chemical Engineering Science* **50**(8): 1281-1300.

Bauer-Brandl, A. (1996a). Erratum to "Polymorphic transitions of cimetidine during manufacture of solid dosage forms" [Int. J. Pharm. 140 (1996) 195]. *International Journal of Pharmaceutics* **145**(1-2): 253.

Bauer-Brandl, A. (1996b). Polymorphic transitions of cimetidine during manufacture of solid dosage forms. *International Journal of Pharmaceutics* **140**(2): 195.

Borissova, A., Dashova, Z., Lai, X. and Roberts, K. J. (2004). Examination of the semi-batch crystallization of benzophenone from saturated methanol solution via aqueous antisolvent drowning-out as monitored in-process using ATR FTIR spectroscopy. *Crystal Growth & Design* **4**(5): 1053-1060.

Braatz, R. D., Fujiwara, M., Ma, D. L., Togkalidou, T. and Tafti, D. K. (2002). Simulation and new sensor technologies for industrial crystallization: A review. *International Journal of Modern Physics B* **16**(1-2): 346-353.

Budz, J., Karpinski, P. H., Mydlarz, J. and Nyvlt, J. (1986). Salting-out precipitation of cocarboxylase hydrochloride from aqueous-solution by addition of acetone. *Industrial & Engineering Chemistry Product Research and Development* **25**(4): 657-664.

Charmolue, H. and Rousseau, R. W. (1991). L-Serine obtained by methanol addition in batch crystallization. *AIChE Journal* **37**(8): 1121-1128.

Choi, Y. J., Chung, S. T., Oh, M. and Kim, H. S. (2005). Investigation of crystallization in a jet Y-mixer by a hybrid computational fluid dynamics and process simulation approach. *Crystal Growth & Design* **5**(3): 959-968.

Chung, S. H., Ma, D. L. and Braatz, R. D. (1999). Optimal seeding in batch crystallization. *Canadian Journal of Chemical Engineering* **77**(3): 590-596.

Chung, S. H., Ma, D. L. and Braatz, R. D. (2000). Optimal model-based experimental design in batch crystallization. *Chemometrics and Intelligent Laboratory Systems* **50**(1): 83-90.

D'Aquino, R. (2004). Masterminding mixing technology. *CEP Magazine* **100**(8): 7-14.

Dauer, R., Mokrauer, J. E. and McKeel, W. J. (1996). Dual jet crystallizer apparatus. Merck & Co., Inc. *Patent, U. S.* 5,578,279.

Dhanasekharan, K. M., Sanyal, J., Jain, A. and Haidari, A. (2005). A generalized approach to model oxygen transfer in bioreactors using population balances and computational fluid dynamics. *Chemical Engineering Science* **60**(1): 213-218.

Doki, N., Kubota, N., Yokota, M., Kimura, S. and Sasaki, S. (2002). Production of sodium chloride crystals of uni-modal size distribution by batch dilution crystallization. *Journal of Chemical Engineering of Japan* **35**(11): 1099-1104.

Falk, L. and Schaer, E. (2001). A PDF modelling of precipitation reactors. *Chemical Engineering Science* **56**(7): 2445-2457.

Feng, H., Olsen, M. G., Liu, Y., Fox, R. O. and Hill, J. C. (2005). Investigation of turbulent mixing in a confined planar-jet reactor. *AIChE Journal* **51**(10): 2649-2664.

Fujiwara, M., Chow, P. S., Ma, D. L. and Braatz, R. D. (2002). Paracetamol crystallization using laser backscattering and ATR-FTIR spectroscopy: Metastability, agglomeration, and control. *Crystal Growth & Design* **2**(5): 363-370.

Fujiwara, M., Nagy, Z. K., Chew, J. W. and Braatz, R. D. (2005). First-principles and direct design approaches for the control of pharmaceutical crystallization. *Journal of Process Control* **15**(5): 493-504.

Gerstlauer, A., Motz, S., Mitrovic, A. and Gilles, E. D. (2002). Development, analysis and validation of population models for continuous and batch crystallizers. *Chemical Engineering Science* **57**(20): 4311-4327.

Granberg, R. A., Bloch, D. G. and Rasmuson, A. C. (1999). Crystallization of paracetamol in acetone-water mixtures. *Journal of Crystal Growth* **199**: 1287-1293.

Green, D. (2002). Crystallizer Mixing: Understanding and Modeling Crystallizer Mixing and Suspension Flow. *Handbook of Industrial Crystallization*. Boston, Butterworth-Heinemann**:** 181-200.

Gron, H., Borissova, A. and Roberts, K. J. (2003). In-process ATR-FTIR spectroscopy for closed-loop supersaturation control of a batch crystallizer producing monosodium glutamate crystals of defined size. *Industrial & Engineering Chemistry Research* **42**(1): 198-206.

Gunawan, R., Fusman, I. and Braatz, R. D. (2004). High resolution algorithms for multidimensional population balance equations. *AIChE Journal* **50**(11): 2738-2749.

Gunawan, R., Ma, D. L., Fujiwara, M. and Braatz, R. D. (2002). Identification of kinetic parameters in multidimensional crystallization processes. *International Journal of Modern Physics B* **16**(1-2): 367-374.

Hacherl, J. M., Paul, E. L. and Buettner, H. M. (2003). Investigation of impinging-jet crystallization with a calcium oxalate model system. *AIChE Journal* **49**(9): 2352-2362.

Haselhuhn, F. and Kind, M. (2003). Pseudo-polymorphic behavior of precipitated calcium oxalate. *Chemical Engineering & Technology* **26**(3): 347-353.

Haseltine, E. L., Patience, D. B. and Rawlings, J. B. (2005). On the stochastic simulation of particulate systems. *Chemical Engineering Science* **60**(10): 2627-2641.

Henczka, M., Baldyga, J. and Shekunov, B. Y. (2005). Particle formation by turbulent mixing with supercritical antisolvent. *Chemical Engineering Science* **60**(8-9): 2193-2201.

Hill, P. J. and Ng, K. M. (1997). Simulation of solids processes accounting for particle-size distribution. *AIChE Journal* **43**(3): 715-726.

Holmback, X. and Rasmuson, A. C. (1999). Size and morphology of benzoic acid crystals produced by drowning-out crystallisation. *Journal of Crystal Growth* **199**: 780-788.

Hounslow, M. J. (1990). A discretized population balance for continuous systems at steady-state. *AIChE Journal* **36**(1): 106-116.

Hounslow, M. J., Ryall, R. L. and Marshall, V. R. (1988). A discretized population balance for nucleation, growth, and aggregation. *AIChE Journal* **34**(11): 1821-1832.

Hu, Q., Rohani, S., Wang, D. X. and Jutan, A. (2004). Nonlinear kinetic parameter estimation for batch cooling seeded crystallization. *AIChE Journal* **50**(8): 1786-1794.

Hulburt, H. M. and Katz, S. (1964). Some problems in particle technology: A statistical mechanical formulation. *Chemical Engineering Science* **19**(8): 555.

Jaworski, Z. and Nienow, A. W. (2003). CFD modelling of continuous precipitation of barium sulphate in a stirred tank. *Chemical Engineering Journal* **91**(2-3): 167-174.

Johnson, B. K. and Prud'homme, R. K. (2003a). Chemical processing and micromixing in confined impinging jets. *AIChE Journal* **49**(9): 2264-2282.

Johnson, B. K. and Prud'homme, R. K. (2003b). Flash NanoPrecipitation of organic actives and block copolymers using a confined impinging jets mixer. *Australian Journal of Chemistry* **56**(10): 1021-1024.

Kaneko, S., Yamagami, Y., Tochihara, H. and Hirasawa, I. (2002). Effect of supersaturation on crystal size and number of crystals produced in antisolvent crystallization. *Journal of Chemical Engineering of Japan* **35**(11): 1219-1223.

Kim, Y., Haam, S., Shul, Y. G., Kim, W. S., Jung, J. K., Eun, H. C. and Koo, K. K. (2003). Pseudopolymorphic crystallization of L-ornithine-L-aspartate by drowning out. *Industrial & Engineering Chemistry Research* **42**(4): 883-889.

Kitamura, M. (2002). Controlling factor of polymorphism in crystallization process. *Journal of Crystal Growth* **237**: 2205-2214.

Kitamura, M. and Nakamura, K. (2002). Dependence of polymorphic transformation on anti-solvent composition and crystallization behavior of thiazole-derivative pharmaceutical. *Journal of Chemical Engineering of Japan* **35**(11): 1116-1122.

Kitamura, M. and Sugimoto, M. (2003). Anti-solvent crystallization and transformation of thiazole-derivative polymorphs - I: effect of addition rate and initial concentrations. *Journal of Crystal Growth* **257**(1-2): 177-184.

Kougoulos, E., Jones, A. G. and Wood-Kaczmar, M. (2005). CFD modelling of mixing and heat transfer in batch cooling crystallizers - Aiding the development of a hybrid predictive compartmental model. *Chemical Engineering Research & Design* **83**(A1): 30-39.

Kramer, H. J. M., Dijkstra, J. W., Neumann, A. M., Meadhra, R. O. and van Rosmalen, G. M. (1996). Modelling of industrial crystallizers, a compartmental

approach using a dynamic flow-sheeting tool. *Journal of Crystal Growth* **166**(1-4): 1084-1088.

Kresta, S., Anthieren, G. and Parsiegla, K. (2005). Model reduction for prediction of silver halide precipitation. *Chemical Engineering Science* **60**(8-9): 2135-2153.

Kumar, S. and Ramkrishna, D. (1997). On the solution of population balance equations by discretization - III. Nucleation, growth and aggregation of particles. *Chemical Engineering Science* **52**(24): 4659-4679.

Kusch, H. A., Ottino, J. M. and Shannon, D. M. (1989). Analysis of impingement mixing-reaction data - use of a lamellar model to generate fluid mixing information. *Industrial & Engineering Chemistry Research* **28**(3): 302-315.

Larsen, P. A., Patience, D. B. and Rawlings, J. B. (2006). Industrial crystallization process control. *IEEE Control Systems Magazine* **26**(4): 70-80.

Leung, S. S., Padden, B. E., Munson, E. J. and Grant, D. J. W. (1998). Solid-state characterization of two polymorphs of aspartame hemihydrate. *Journal of Pharmaceutical Sciences* **87**(4): 501-507.

Lindrud, M. D., Kim, S. and Wei, C. (2001). Sonic impinging jet crystallization apparatus and process. Bristol-Myers Squibb Company. *Patent, U. S.* 6,302,958.

Liu, Y. and Fox, R. O. (2006). CFD predictions for chemical processing in a confined impinging-jets reactor. *AIChE Journal* **52**(2): 731-744.

Ma, D. L. and Braatz, R. D. (2003). Robust identification and control of batch processes. *Computers & Chemical Engineering* **27**(8-9): 1175-1184.

Ma, D. L., Braatz, R. D. and Tafti, D. K. (2002a). Compartmental modeling of multidimensional crystallization. *International Journal of Modern Physics B* **16**(1-2): 383-390.

Ma, D. L., Chung, S. H. and Braatz, R. D. (1999). Worst-case performance analysis of optimal batch control trajectories. *AIChE Journal* **45**(7): 1469-1476.

Ma, D. L., Tafti, D. K. and Braatz, R. D. (2002b). High-resolution simulation of multidimensional crystal growth. *Industrial & Engineering Chemistry Research* **41**(25): 6217-6223.

Ma, D. L., Tafti, D. K. and Braatz, R. D. (2002c). Optimal control and simulation of multidimensional crystallization processes. *Computers & Chemical Engineering* **26**(7-8): 1103-1116.

Mahajan, A. J. and Kirwan, D. J. (1996). Micromixing effects in a two-impinging-jets precipitator. *AIChE Journal* **42**(7): 1801-1814.

Marchisio, D. L., Barresi, A. A. and Fox, R. O. (2001a). Simulation of turbulent precipitation in a semi-batch Taylor-Couette reactor using CFD. *AIChE Journal* **47**(3): 664-676.

Marchisio, D. L., Fox, R. O., Barresi, A. A. and Baldi, G. (2001b). On the comparison between presumed and full PDF methods for turbulent precipitation. *Industrial & Engineering Chemistry Research* **40**(23): 5132-5139.

Marchisio, D. L., Fox, R. O., Barresi, A. A., Garbero, M. and Baldi, G. (2001c). On the simulation of turbulent precipitation in a tubular reactor via computational fluid dynamics (CFD). *Chemical Engineering Research & Design* **79**(A8): 998-1004.

Marchisio, D. L., Rivautella, L. and Barresi, A. A. (2006). Design and scale-up of chemical reactors for nanoparticle precipitation. *AIChE Journal* **52**(5): 1877-1887.

Matthews, H. B. and Rawlings, J. B. (1998). Batch crystallization of a photochemical: Modeling, control, and filtration. *AIChE Journal* **44**(5): 1119-1127.

Midler, M., Paul, E. L., Whittington, E. F., Futran, M., Liu, P. D., Hsu, J. and Pan, S.-H. (1994). Crystallization method to improve crystal structure and size. Merck & Co., Inc. *Patent, U. S.* 5,314,506.

Motz, S., Mitrovic, A. and Gilles, E. D. (2002). Comparison of numerical methods for the simulation of dispersed phase systems. *Chemical Engineering Science* **57**(20): 4329-4344.

Mullin, J. W. (2001). *Crystallization*. Oxford, U. K., Elsevier Butterworth-Heinemann.

Mullin, J. W., Teodossiev, N. and Sohnel, O. (1989). Potassium-sulfate precipitation from aqueous-solution by salting-out with acetone. *Chemical Engineering and Processing* **26**(2): 93-99.

Mydlarz, J. and Jones, A. G. (1991). Crystallization and agglomeration kinetics during the batch drowning-out precipitation of potash alum with aqueous acetone. *Powder Technology* **65**(1-3): 187-194.

Nagao, L. M., Lyapustina, S., Munos, M. K. and Capizzi, M. D. (2005). Aspects of particle science and regulation in pharmaceutical inhalation drug products. *Crystal Growth & Design* **5**(6): 2261-2267.

Nagy, Z. K. and Braatz, R. D. (2003a). Robust nonlinear model predictive control of batch processes. *AIChE Journal* **49**(7): 1776-1786.

Nagy, Z. K. and Braatz, R. D. (2003b). Worst-case and distributional robustness analysis of finite-time control trajectories for nonlinear distributed parameter systems. *IEEE Transactions on Control Systems Technology* **11**(5): 694-704.

Nagy, Z. K. and Braatz, R. D. (2004). Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. *Journal of Process Control* **14**(4): 411-422.

Nyvlt, J. and Zacek, S. (1986). Effect of the rate of stirring on crystal size in precipitating or salting-out systems. *Collection of Czechoslovak Chemical Communications* **51**(8): 1609-1617.

Okamoto, M., Hamano, M. and Ooshima, H. (2004). Active utilization of solvent-mediated transformation for exclusive production of metastable polymorph crystals of AE1-923. *Journal of Chemical Engineering of Japan* **37**(1): 95-101.

Paul, E. L., Midler, M. and Sun, Y. (2004). Mixing in the Fine Chemicals and Pharmaceutical Industries. *Handbook of Industrial Mixing*. Hoboken, N. J., John Wiley & Sons, Inc.**:** 1027-1070.

Paul, E. L., Tung, H. H. and Midler, M. (2005). Organic crystallization processes. *Powder Technology* **150**(2): 133-143.

Phillips, R., Rohani, S. and Baldyga, J. (1999). Micromixing in a single-feed semi-batch precipitation process. *AIChE Journal* **45**(1): 82-92.

Piton, D., Fox, R. O. and Marcant, B. (2000). Simulation of fine particle formation by precipitation using computational fluid dynamics. *Canadian Journal of Chemical Engineering* **78**(5): 983-993.

Plasari, E., Grisoni, P. and Villermaux, J. (1997). Influence of process parameters on the precipitation of organic nanoparticles by drowning-out. *Chemical Engineering Research & Design* **75**(A2): 237-244.

Pope, S. B. (1985). PDF methods for turbulent reactive flows. *Progress in Energy and Combustion Science* **11**(2): 119-192.

Pope, S. B. (2000). *Turbulent Flows*. Cambridge, U. K., Cambridge University Press.

Puel, F., Fevotte, G. and Klein, J. P. (2003a). Simulation and analysis of industrial crystallization processes through multidimensional population balance equations. Part 1: A resolution algorithm based on the method of classes. *Chemical Engineering Science* **58**(16): 3715-3727.

Puel, F., Fevotte, G. and Klein, J. P. (2003b). Simulation and analysis of industrial crystallization processes through multidimensional population balance equations.

Part 2: A study of semi-batch crystallization. *Chemical Engineering Science* **58**(16): 3729-3740.

Quintana-Hernandez, P., Bolanos-Reynoso, E., Miranda-Castro, B. and Salcedo-Estrada, L. (2004). Mathematical modeling and kinetic parameter estimation in batch crystallization. *AIChE Journal* **50**(7): 1407-1417.

Santos, R. J., Teixeira, A. M. and Lopes, J. C. B. (2005). Study of mixing and chemical reaction in RIM. *Chemical Engineering Science* **60**(8-9): 2381-2398.

Sarkar, D., Rohani, S. and Jutan, A. (2006). Multi-objective optimization of seeded batch crystallization processes. *Chemical Engineering Science* **61**(16): 5282-5295.

Schroer, J. W. and Ng, K. M. (2003). Process paths of kinetically controlled crystallization: enantionters and polymorphs. *Industrial & Engineering Chemistry Research* **42**(10): 2230-2244.

Schwarzer, H. C. and Peukert, W. (2004). Combined experimental/numerical study on the precipitation of nanoparticles. *AIChE Journal* **50**(12): 3234-3247.

Schwarzer, H. C., Schwertfirm, F., Manhart, M., Schmid, H. J. and Peukert, W. (2006). Predictive simulation of nanoparticle precipitation based on the population balance equation. *Chemical Engineering Science* **61**(1): 167-181.

Sha, Z. L. and Palosaari, S. (2002). Modeling and simulation of crystal size distribution in imperfectly mixed suspension crystallization. *Journal of Chemical Engineering of Japan* **35**(11): 1188-1195.

Shekunov, B. Y., Baldyga, J. and York, P. (2001). Particle formation by mixing with supercritical antisolvent at high Reynolds numbers. *Chemical Engineering Science* **56**(7): 2421-2433.

Shekunov, B. Y. and York, P. (2000). Crystallization processes in pharmaceutical technology and drug delivery design. *Journal of Crystal Growth* **211**(1-4): 122-136.

Shin, D. M. and Kim, W. S. (2002). Drowning-out crystallization of L-ornithine-aspartate in turbulent agitated reactor. *Journal of Chemical Engineering of Japan* **35**(11): 1083-1090.

Sohrabi, M. and Marvast, M. A. (2000). Application of a continuous two impinging streams reactor in solid-liquid enzyme reactions. *Industrial & Engineering Chemistry Research* **39**(6): 1903-1910.

Stahl, M., Alund, B. L. and Rasmuson, A. C. (2001). Reaction crystallization kinetics of benzoic acid. *AIChE Journal* **47**(7): 1544-1560.

Stan, G. and Johnson, D. A. (2001). Experimental and numerical analysis of turbulent opposed impinging jets. *AIAA Journal* **39**(10): 1901-1908.

Takiyama, H., Otsuhata, T. and Matsuoka, M. (1998). Morphology of NaCl crystals in drowning-out precipitation operation. *Chemical Engineering Research & Design* **76**(A7): 809-814.

Taylor, K. (2002). Pulmonary Drug Delivery. *Pharmaceutics*. Spain, Churchill Livingstone**:** 473-488.

Teixeira, A. M., Santos, R. J., Rui, M., Costa, P. F. N. and Lopes, J. C. B. (2005). Hydrodynamics of the mixing head in RIM: LDA flow-field characterization. *AIChE Journal* **51**(6): 1608-1619.

Togkalidou, T., Tung, H. H., Sun, Y., Andrews, A. T. and Braatz, R. D. (2004). Parameter estimation and optimization of a loosely bound aggregating pharmaceutical crystallization using in situ infrared and laser backscattering measurements. *Industrial & Engineering Chemistry Research* **43**(19): 6168-6181.

Tsai, K., Gillis, P. A., Sen, S. and Fox, R. O. (2002). A finite-mode PDF model for turbulent reacting flows. *Journal Of Fluids Engineering-Transactions Of The ASME* **124**(1): 102-107.

Unger, D. R. and Muzzio, F. J. (1999). Laser-induced fluorescence technique for the quantification of mixing in impinging jets. *AIChE Journal* **45**(12): 2477-2486.

Van Leeuwen, M. L. J., Bruinsma, O. S. L. and Van Rosmalen, G. M. (1996). Influence of mixing on the product quality in precipitation. *Chemical Engineering Science* **51**(11): 2595-2600.

Venneker, B. C. H., Derksen, J. J. and Van den Akker, H. E. A. (2002). Population balance modeling of aerated stirred vessels based on CFD. *AIChE Journal* **48**(4): 673-685.

Vicum, L., Ottiger, S., Mazzotti, M., Makowski, L. and Baldyga, J. (2004). Multi-scale modeling of a reactive mixing process in a semibatch stirred tank. *Chemical Engineering Science* **59**(8-9): 1767-1781.

Wang, L. G. and Fox, R. O. (2003). Application of in situ adaptive tabulation to CFD simulation of nano-particle formation by reactive precipitation. *Chemical Engineering Science* **58**(19): 4387-4401.

Wang, L. G. and Fox, R. O. (2004). Comparison of micromixing models for CFD simulation of nanoparticle formation. *AIChE Journal* **50**(9): 2217-2232.

Wei, H. and Garside, J. (1997). Application of CFD modelling to precipitation systems. *Chemical Engineering Research & Design* **75**(A2): 219-227.

Wei, H. Y., Zhou, W. and Garside, J. (2001). Computational fluid dynamics modeling of the precipitation process in a semibatch crystallizer. *Industrial & Engineering Chemistry Research* **40**(23): 5255-5261.

Wey, J. S. and Karpinski, P. H. (2002). Batch Crystallization. *Handbook of Industrial Crystallization*. Boston, Butterworth-Heinemann**:** 231-248.

Worlitschek, J. and Mazzotti, M. (2004). Model-based optimization of particle size distribution in batch-cooling crystallization of paracetamol. *Crystal Growth & Design* **4**(5): 891-903.

Wulkow, M., Gerstlauer, A. and Nieken, U. (2001). Modeling and simulation of crystallization processes using parsival. *Chemical Engineering Science* **56**(7): 2575-2588.

Yu, Z. Q., Chow, P. S. and Tan, R. B. H. (2006). Seeding and constant-supersaturation control by ATR-FTIR in anti-solvent crystallization. *Organic Process Research & Development* **10**(4): 717-722.

Zauner, R. and Jones, A. G. (2000a). Mixing effects on product particle characteristics from semi-batch crystal precipitation. *Chemical Engineering Research & Design* **78**(A6): 894-902.

Zauner, R. and Jones, A. G. (2000b). Scale-up of continuous and semibatch precipitation processes. *Industrial & Engineering Chemistry Research* **39**(7): 2392-2403.

Zauner, R. and Jones, A. G. (2002). On the influence of mixing on crystal precipitation processes - application of the segregated feed model. *Chemical Engineering Science* **57**(5): 821-831.

Zhao, Y. and Brodkey, R. S. (1998). Averaged and time-resolved, full-field (three-dimensional), measurements of unsteady opposed jets. *Canadian Journal of Chemical Engineering* **76**(3): 536-545.

Zhou, G. X., Fujiwara, M., Woo, X. Y., Rusli, E., Tung, H. H., Starbuck, C., Davidson, O., Ge, Z. H. and Braatz, R. D. (2006). Direct design of pharmaceutical antisolvent crystallization through concentration control. *Crystal Growth & Design* **6**(4): 892-898.

Chapter 3
Theory

3.1 Introduction

Based on the past research efforts highlighted in the previous chapter, it is apparent that the next step in modeling crystallizers requires a higher resolution of the flow field in order to establish a better understanding of the interactions between hydrodynamics and crystal nucleation and growth, and the impact on the crystal size distribution (CSD). In this chapter, an approach to couple the computational fluid dynamics (CFD) computations to the solution of the population balance equation (PBE) to simulate the full CSD, and the solution of the probability density function (PDF) that describes the local fluctuations in the turbulent flow field, is presented. The development of this algorithm also is motivated by emerging sensor technologies, which allow the in-situ measurement of the solution concentration and the full CSD as a function of time (Braatz et al., 2002) and space (Bachalo, 1994; Bardow et al., 2003; Gladden, 2003; Stanley et al., 2005). Thus a complete validation of the coupled simulation algorithm is feasible, which would provide support for any theoretical understandings or conclusions made based on the simulation results.

Although direct numerical simulations (DNS) can resolve all flow structures of the turbulent flow, and avoids the prediction of the PDF, the application of DNS in highly turbulent flow in a full-scale reactor with chemical reactions and crystallization is still computationally intractable (Fox, 2003; Moody and Collins, 2003). Instead, the full CSD is solved numerically using the high-resolution, finite-volume, semidiscrete

central scheme (Kurganov and Tadmor, 2000) and the micromixing is modeled by a multi-environment presumed-PDF model (Fox, 2003). This approach can be integrated within commercially available CFD codes, in which the additional models are simulated within the CFD solver.

The following sections describe the numerical methods used to compute (1) the PBE for the evolution of CSD and (2) the PDF of the local turbulent fluctuations, which are directly integrated into a CFD code. In addition, the CFD equations, and the expressions for the crystallization kinetics and the effective viscosity, are presented.

3.2 Computational Fluid Dynamics

In this work, the macromixing was modeled by the Reynolds-averaged Navier-Stokes (RANS) and transport equations and the *k*-ε turbulence model (Fluent 6.1 User's Guide, 2003; Pope, 2000; Shih et al., 1995), computed using a commercial CFD software (Fluent 6.1.22 and 6.2.16, Fluent Inc.). The equations are presented as follows; the reader is referred to the software documentation for the nomenclature and for more details of the equations (Fluent 6.1 UDF Manual, 2003; Fluent 6.1 User's Guide, 2003).

$$\text{Continuity equation:} \quad \frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \vec{v}) = 0 \tag{3.1}$$

$$\text{Momentum conservation equation:} \quad \frac{\partial}{\partial t}(\rho \vec{v}) + \nabla \bullet (\rho \vec{v} \vec{v}) = -\nabla p + \nabla \bullet (\bar{\bar{\tau}}) + \rho \vec{g} \tag{3.2}$$

Standard $k-\varepsilon$ equations:

$$\begin{aligned}
&\frac{\partial}{\partial t}(\rho k)+\nabla\bullet(\rho k\vec{v})=\nabla\bullet\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\nabla\bullet k\right]+G_k-\rho\varepsilon+S_k\\
&\frac{\partial}{\partial t}(\rho\varepsilon)+\nabla\bullet(\rho\varepsilon\vec{v})=\nabla\bullet\left[\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\bullet\varepsilon\right]+C_{1\varepsilon}\frac{\varepsilon}{k}G_k-C_{2\varepsilon}\rho\frac{\varepsilon^2}{k}+S_\varepsilon\\
&\text{where}\quad \mu_t=\rho C_\mu\frac{k^2}{\varepsilon}
\end{aligned}\tag{3.3}$$

Realizable $k-\varepsilon$ equations:

$$\begin{aligned}
&\frac{\partial}{\partial t}(\rho k)+\nabla\bullet(\rho k\vec{v})=\nabla\bullet\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\nabla\bullet k\right]+G_k-\rho\varepsilon+S_k\\
&\frac{\partial}{\partial t}(\rho\varepsilon)+\nabla\bullet(\rho\varepsilon\vec{v})=\nabla\bullet\left[\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\bullet\varepsilon\right]+\rho C_1 S\varepsilon-\rho C_2\frac{\varepsilon^2}{k+\sqrt{\nu\varepsilon}}+S_\varepsilon
\end{aligned}\tag{3.4}$$

Scalar transport equation:

$$\begin{aligned}
&\frac{\partial}{\partial}(\rho\phi_k)+\nabla\bullet(\rho\vec{v}\phi_k-\rho(D_m+D_t)\nabla\bullet\phi_k)=S_{\phi_k}\\
&D_t=\frac{\mu_t}{\rho\mathrm{Sc}_t}
\end{aligned}\tag{3.5}$$

3.3 High-Resolution, Finite-Volume, Semidiscrete Central Schemes

High-resolution finite-volume methods have been investigated primarily in the applied mathematics and computational physics literature (Leveque, 2002). These methods provide high accuracy for simulating hyperbolic conservation laws while reducing numerical diffusion and eliminating nonphysical oscillations that can occur with classical methods. Being in the class of finite volume methods, such methods are conservative, which ensures the accurate tracking of discontinuities and preserves the total mass within the computational domain subject to the applied boundary conditions. Another advantage is that these numerical schemes can be easily extended to solve multidimensional and variable-coefficient conservation laws.

High-resolution central schemes for nonlinear conservation laws, starting from the Nessyahu and Tadmor (NT) scheme, have the advantages of retaining the simplicity of the Riemann-solver-free approach, while achieving at least second-order accuracy (Nessyahu and Tadmor, 1990). Kurganov and Tadmor (2000) and Kurganov et al. (2001) extended the NT scheme to reduce numerical viscosity (nonphysical smoothing of the numerical solution) arising from discrete approximations of the advection term. This Kurganov and Tadmor (KT) high-resolution finite-volume central scheme accumulates less dissipation for a fixed Δy as compared to the NT scheme, and can be used efficiently with small time steps since the numerical viscosity is independent of $(1/\Delta t)$. The limiting case, $\Delta t \rightarrow 0$, results in the second-order semidiscrete version. In addition, the KT method satisfies the scalar total-variation-diminishing (TVD) property with minmod reconstruction, which implies that the nonphysical oscillations that occur with many second-order accurate numerical methods cannot occur with this method. The KT semidiscrete scheme is particularly effective when combined with high-order ODE solvers for the time evolution.

Consider the nonlinear conservation law,

$$\frac{\partial}{\partial t}u(y,t)+\frac{\partial}{\partial y}q(u(y,t))=0 \tag{3.6}$$

The KT semidiscrete central scheme is classified as a finite-volume method, since it involves keeping track of the integral of *u* over each grid cell. The use of cell averages

$$u_j(t)=\frac{1}{\Delta y}\int_{y_{j-1/2}}^{y_{j+1/2}}u(y,t)dy \tag{3.7}$$

to represent computed values, where $\Delta y = y_{j+1/2} - y_{j-1/2}$, ensures that the numerical method is conservative. The second-order semidiscrete scheme admits the conservative form:

$$\frac{d}{dt}u_j(t) = -\frac{H_{j+1/2}(t) - H_{j-1/2}(t)}{\Delta y} \tag{3.8}$$

with the numerical flux

$$H_{j+1/2}(t) := \frac{q(u^+_{j+1/2}(t)) + q(u^-_{j+1/2}(t))}{2} - \frac{a_{j+1/2}}{2}[u^+_{j+1/2}(t) - u^-_{j+1/2}(t)] \tag{3.9}$$

and the intermediate values given by

$$\begin{aligned} u^+_{j+1/2} &:= u_{j+1}(t) - \frac{\Delta y}{2}(u_y)_{j+1}(t) \\ u^-_{j+1/2} &:= u_j(t) + \frac{\Delta y}{2}(u_y)_j(t) \end{aligned} \tag{3.10}$$

while the local propagation of speeds, for the scalar case, is

$$a_{j+1/2}(t) := \max_{u \in [u^-_{j+1/2}(t),\, u^+_{j+1/2}(t)]} \left| q'\left(u^-_{j+1/2}(t)\right) \right| \tag{3.11}$$

The derivatives are approximated with the minmod limiter:

$$(u_y)^n_j := \operatorname{minmod}\left(\theta \frac{u^n_j - u^n_{j-1}}{\Delta y}, \frac{u^n_{j+1} - u^n_{j-1}}{2\Delta y}, \theta \frac{u^n_{j+1} - u^n_j}{\Delta y} \right) \qquad 1 \le \theta \le 2 \tag{3.12}$$

which is defined as

$$\operatorname{minmod}(\alpha_1, \alpha_2, ...) = \begin{cases} \min\limits_i \{\alpha_i\} & \text{if } \alpha_i > 0 \ \forall i \\ \max\limits_i \{\alpha_i\} & \text{if } \alpha_i < 0 \ \forall i \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

Selecting the value of $\theta = 1$ results in nonphysical smoothing of the numerical solution. A value of $\theta = 2$ results in minimal nonphysical smoothing, but can introduce some nonphysical oscillation. The value $\theta = 1.5$ is commonly selected to

trade off minimizing the amount of nonphysical dissipation/smoothing with minimizing nonphysical oscillation. More details on such limiters can be found in the references in this section.

3.4 Coupling the Population Balance Equation to CFD

A spatially inhomogeneous crystallization process can be described by the population balance equation (PBE) (Hulburt and Katz, 1964; Randolph and Larson, 1988):

$$\begin{aligned}&\frac{\partial f}{\partial t}+\sum_i \frac{\partial[G_i(r_i,c,T)\,f]}{\partial r_i}+\nabla\bullet(\vec{v}f-D_t\nabla\bullet f)\\&=B(f,c,T)\prod_i\delta(r_i-r_{i0})+h(f,c,T)\end{aligned} \tag{3.14}$$

where the particle number density function (f) is a function of external coordinates (x_i), internal coordinates (r_i), and time (t). The rates of growth (G_i) and nucleation (B) are functions of the vector of solution concentrations (c) and the temperature (T), and for size-dependent growth, G_i also varies with r_i. The Dirac delta function is δ. The term h represents the creation and destruction of crystals due to aggregation, agglomeration, and breakage. As the solution concentrations and temperature vary with spatial position and time, equation (3.14) must be solved together with the bulk transport equations for mass, energy, momentum, and turbulence to obtain f (x, r, t), c(x, t), T(x, t), the velocity field v(x, t), and the local turbulent diffusivity $D_t(x,t)$. This enables the determination of the effects of the localized solution environment on the nucleation and growth rates, as well as on the CSD. With v and D_t being obtained by solving the momentum and turbulence conservation equations of the liquid phase, respectively, equation (3.14) assumes that the particles follow the streamlines in the flow field (Hulburt and Katz, 1964). This is a good approximation for organic pharmaceutical crystals where the density is close to the liquid phase, and for primary

nucleation in a crystallizer for short times. The approximation becomes less accurate as the crystals increase in size.

High-resolution finite-volume methods can be utilized to solve the PBE, due to the similarity of its mathematical structure to that of hyperbolic conservation equations. Recently, the capability of using such methods to numerically solve multidimensional PBEs that simulate the evolution of crystal size and shape distribution had been demonstrated (Gunawan et al., 2004; Ma et al., 2002a; Ma et al., 2002b, 2002c). The following will briefly present the application of the high-resolution finite-volume semidiscrete central scheme to the PBE. The main advantage of using the high-resolution central scheme to discretize the growth term is that its second-order accuracy allows the use of a larger Δr, while retaining the same numerical accuracy obtained from first-order methods (e.g., upwind method). This is important because the number of transport equations that can be solved in the CFD algorithm is limited. Moreover, the method does not produce spurious oscillations in the solution, which are common in second-order methods such as Lax-Wendroff. Another advantage of using the high-resolution central scheme is that the numerical dissipation depends on Δr, but not $1/\Delta t$. This is essential due to the fact that, in most cases, very small time steps, much smaller than that limited by the Courant-Friedrichs-Lewy (CFL) condition, is required to resolve the turbulent flow and concentration field in the CFD computation. Hence, this method avoids any additional numerical dissipation due to using small Δt. Although the approach taken here is applicable to the general equation (3.14), the subsequent equations will focus on the case of primary nucleation and size-dependent growth along one internal principal axis, as this will simplify the

presentation and these assumptions are consistent with the crystallization systems simulated in this thesis.

Focusing first only on the first two terms of equation (3.14), the following semidiscrete PBEs are obtained after integrating over *r* over each cell and canceling terms:

$$\frac{d}{dt} f_j(t) = \begin{cases} -\frac{1}{\Delta r}\left\{ G_{j+1/2}\left[f_j(t) + \frac{\Delta r}{2}(f_r)_j(t) \right] \right. & G > 0\ (\Delta c > 0) \\ \left. -G_{j-1/2}\left[f_{j-1}(t) + \frac{\Delta r}{2}(f_r)_{j-1}(t) \right] \right\}, & \text{(crystal growth)} \\ -\frac{1}{\Delta r}\left\{ G_{j+1/2}\left[f_{j+1}(t) - \frac{\Delta r}{2}(f_r)_{j+1}(t) \right] \right. & G < 0\ (\Delta c < 0) \\ \left. -G_{j-1/2}\left[f_j(t) - \frac{\Delta r}{2}(f_r)_j(t) \right] \right\}, & \text{(crystal dissolution)} \end{cases} \tag{3.15}$$

where f_j is the cell-averaged population density, based on equation (3.7) and the derivatives, $(f_r)_j$, are approximated by the minmod limiter (equations (3.12) and (3.13)). Note that the growth rates are evaluated at the end points of each grid cell. The nucleation term is included in the cell corresponding to the nuclei size by averaging the nucleation rate (the number of nuclei per unit time per unit volume) over the cell width, $B/\Delta r$. The computation of the average population density for the first grid cell, f_1, requires the values of f_0 and f_{-1}, which are fictitious points with population densities of zero at all times. At the other end, the computation of f_N in the last grid cell assumes that $f_{N+1} = f_{N+2} = f_N$ at all times, which is known as the absorbing boundary condition (Leveque, 2002).

To couple the semidiscrete PBE with the CFD algorithm, equation (3.15) is rewritten on a mass basis so that the set of crystals within each grid cell is treated as a separate

species. Thus, when this equation is coupled with the transport equations of other species present in the system (solute, solvent, and antisolvent), also written on a mass basis, the overall mass balance of the system is also satisfied. The cell-averaged crystal mass can be evaluated as

$$f_{w,j} = \rho_c k_v \int_{r_{j-1/2}}^{r_{j+1/2}} r^3 f_j dr = \frac{\rho_c k_v f_j}{4}\left((r_{j+1/2})^4 - (r_{j-1/2})^4\right) \tag{3.16}$$

and the transport equation for crystal mass between size $r_{j-1/2}$ and $r_{j+1/2}$ is

$$\begin{aligned}
&\frac{d}{dt} f_{w,j} + \nabla\bullet(\vec{v} f_{w,j} - D_t \nabla\bullet f_{w,j}) \\
&= \begin{cases}
\dfrac{\rho_c k_v}{4\Delta r}\left[(r_{j+1/2})^4 - (r_{j-1/2})^4\right] & \\
\times\left\{-G_{j+1/2}\left[f_j + \dfrac{\Delta r}{2}(f_r)_j\right] + G_{j-1/2}\left[f_{j-1} + \dfrac{\Delta r}{2}(f_r)_{j-1}\right] + \underbrace{J}_{j=0}\right\}, & \Delta c > 0 \\
\dfrac{\rho_c k_v}{4\Delta r}\left[(r_{j+1/2})^4 - (r_{j-1/2})^4\right] & \\
\times\left\{-G_{j+1/2}\left[f_{j+1} - \dfrac{\Delta r}{2}(f_r)_{j+1}\right] + G_{j-1/2}\left[f_j - \dfrac{\Delta r}{2}(f_r)_j\right]\right\}, & \Delta c < 0
\end{cases}
\end{aligned} \tag{3.17}$$

Equation (3.17) can be directly incorporated into the CFD code as a transport equation when micromixing effects are not important, such that the right hand side is treated as an additional source term. A corresponding source term is added to the solute transport equation to account for its depletion due to nucleation and crystal growth, or its increment due to crystal dissolution, which is a negative sum of equation (3.17) for $j = 1, \ldots, N$.

3.5 Multienvironment Presumed-PDF Model

A multienvironment CFD micromixing model, also known as the presumed finite-mode PDF method, is used to model the micromixing effects (Fox, 2003). In this

approach, each computational cell in the CFD grid is divided into N_e different probability modes or environments, which correspond to a discretization of the presumed composition PDF into a finite set of delta (δ) functions:

$$f_\phi(\boldsymbol{\psi};\mathbf{x},t)=\sum_{n=1}^{N_e}p_n(\mathbf{x},t)\prod_{\alpha=1}^{N_s}\delta\left[\psi_\alpha-\left\langle\phi_\alpha\right\rangle_n(\mathbf{x},t)\right] \tag{3.18}$$

where f_ϕ is the joint PDF of all scalars, N_s is the total number of scalars (species), p_n is the probability of mode n or volume fraction of environment n, and $\left\langle\phi_\alpha\right\rangle_n$ is the mean composition of scalar α corresponding to mode n. The weighted concentration is defined as

$$\left\langle\mathbf{s}\right\rangle_n\equiv p_n\left\langle\phi\right\rangle_n \tag{3.19}$$

The transport of probability and species in inhomogeneous flows is modeled by:

$$\frac{\partial\mathbf{p}}{\partial t}+\nabla\bullet(\vec{v}\mathbf{p}-D_t\nabla\bullet\mathbf{p})=\mathbf{G}(\mathbf{p})+\mathbf{G}_\mathbf{s}(\mathbf{p}) \tag{3.20}$$

$$\begin{aligned}&\frac{\partial\left\langle\mathbf{s}\right\rangle_n}{\partial t}+\nabla\bullet(\vec{v}\left\langle\mathbf{s}\right\rangle_n-D_t\nabla\bullet\left\langle\mathbf{s}\right\rangle_n)\\&=\mathbf{M}^n(\mathbf{p},\left\langle\mathbf{s}\right\rangle_1,\ldots,\left\langle\mathbf{s}\right\rangle_{N_e})+\mathbf{M}_\mathbf{s}^n(\mathbf{p},\left\langle\mathbf{s}\right\rangle_1,\ldots,\left\langle\mathbf{s}\right\rangle_{N_e})+p_n\mathbf{S}(\left\langle\phi\right\rangle_n)\end{aligned} \tag{3.21}$$

where $\mathbf{G}$ and $\mathbf{M}^n$ are the rates of change of $\mathbf{p}=\begin{bmatrix}p_1 & p_2 & \ldots & p_N\end{bmatrix}$ and $\left\langle\mathbf{s}\right\rangle_n$ due to micromixing, respectively, $\mathbf{G}_\mathbf{s}$ and $\mathbf{M}_\mathbf{s}^n$ are additional micromixing terms to eliminate the spurious dissipation rate in the mixture-fraction-variance transport equation (for details see in Fox, 2003), and $\mathbf{S}$ is the chemical source term. The conservation of probability requires that

$$\sum_{n=1}^{N}p_n=1 \tag{3.22}$$

and

$$\sum_{n=1}^{N_e}G_n(\mathbf{p})=0 \tag{3.23}$$

The mean compositions of the scalars are given by

$$\langle \boldsymbol{\phi} \rangle = \sum_{n=1}^{N_e} p_n \langle \phi \rangle_n = \sum_{n=1}^{N_e} \langle \mathbf{s} \rangle_n \tag{3.24}$$

and, since the means remain unchanged by micromixing, the following must be satisfied:

$$\sum_{n=1}^{N_e} \mathbf{M}^n(\mathbf{p}, \langle \mathbf{s} \rangle_1, \ldots, \langle \mathbf{s} \rangle_{N_e}) = 0 \tag{3.25}$$

The simulations in this paper utilize a three-environment model, as shown in Figure 3.1. This approach had been used to model precipitation, in which the method of moments to model the average properties of the crystalline phase (Marchisio et al., 2001a; Marchisio et al., 2001b; Marchisio et al., 2001c). The authors suggested that three environments are sufficient in capturing the micromixing effects in non-premixed flows with satisfactory accuracy. The extension to a larger number of environments is possible (Fox, 2003; Piton et al., 2000; Wang and Fox, 2004), but at a larger computational expense, since one set of semidiscrete PBE has to be solved in each mixed environment. An advantage of using the multienvironment PDF model instead of other micromixing models is that it can be easily incorporated into existing CFD codes, in which the transport Equations (3.20) and (3.21) can be computed directly by the CFD solver. Since the compositions in environments 1 and 2 are known on the basis of on the feed and initial conditions, equation (3.21) will be applied to all species in environment 3 only, which includes the solute, the solvent, the antisolvent, and the crystal mass in each grid cell of the semidiscrete PBE.

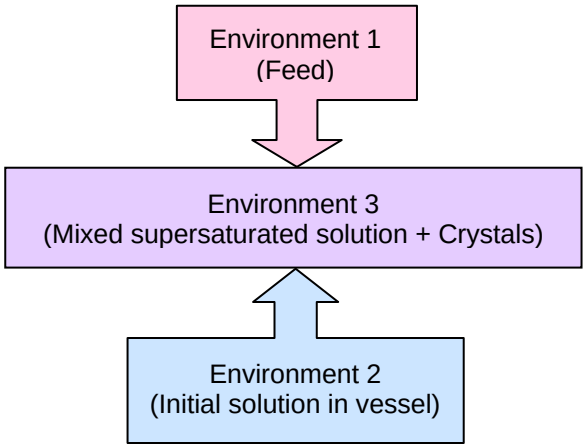

Figure 3.1 Three-environment micromixing model.

Furthermore, equation (3.21) is used to evaluate the mixture fraction in environment 3 (which represents the relative fractions of fluids from environments 1 and 2), $\langle\xi\rangle_3$, while the mixture fractions in environments 1 and 2 are $\langle\xi\rangle_1 = 1$ and $\langle\xi\rangle_2 = 0$, respectively. With $N_e = 3$, it is possible to simulate the mean, the variance, and the skewness of the mixture fraction correctly.

The micromixing terms are given by (Fox, 2003):

Table 3.1 Micromixing terms for equations three-environment presumed-PDF micromixing model (Fox, 2003).

Model variables	$\mathbf{G},\ \mathbf{M}^n$	$\mathbf{G_s},\ \mathbf{M_s}^n$
p_1	$-\gamma p_1(1-p_1)$	$\gamma_s p_3$
p_2	$-\gamma p_2(1-p_2)$	$\gamma_s p_3$
p_3	$\gamma\left[p_1(1-p_1)+p_2(1-p_2)\right]$	$-2\gamma_s p_3$
$\langle\mathbf{s}\rangle_3$	$\gamma\left[p_1(1-p_1)\langle\phi\rangle_1+p_2(1-p_2)\langle\phi\rangle_2\right]$	$-\gamma_s p_3\left(\langle\phi\rangle_1+\langle\phi\rangle_2\right)$

$$\gamma = \frac{\varepsilon_\xi}{p_1(1-p_1)(1-\langle\xi\rangle_3)^2 + p_2(1-p_2)\langle\xi\rangle_3^2}$$

$$\gamma_s = \frac{2D_t}{(1-\langle\xi\rangle_3)^2+\langle\xi\rangle_3^2}\frac{\partial\langle\xi\rangle_3}{\partial x_i}\frac{\partial\langle\xi\rangle_3}{\partial x_i}$$

$$\langle\xi''^2\rangle = p_1(1-p_1) - 2p_1p_3\langle\xi\rangle_3 + p_3(1-p_3)\langle\xi\rangle_3^2$$

where $\langle\phi\rangle_n$ is evaluated using equation (3.24). The value of p_3 can also be determined from equation (3.22), but its value can be erroneous when $(p_1 + p_2)$ is close to 1 due to numerical errors. For a fully-developed scalar spectrum, the scalar dissipation rate, ε_ξ, is related to the turbulent frequency, ε/k, by

$$\varepsilon_\xi = C_\phi\langle\xi''^2\rangle\frac{\varepsilon}{k} \tag{3.26}$$

where $C_\phi = 2$ (Wang and Fox, 2004), and ε and *k* are the turbulent dissipation rate and kinetic energy, respectively. The chemical source terms in equation (3.21) for the solute and crystals are substituted with the right-hand side of equation (3.17) along with the appropriate nucleation and growth kinetics that are not limited by micromixing (Mahajan and Kirwan, 1994). For the case of unseeded crystallization, the micromixing terms for the crystals are zero.

3.6 Nucleation and Growth Kinetics

The kinetic rate expressions of primary nucleation and crystal growth is described in this section. The values of the kinetics parameters should be obtained for well-micromixed and well-macromixed conditions, that is, not limited by any mixing rates. The primary nucleation rate, *B*, can be expressed as (Granberg et al., 1999; Sohnel and Mullin, 1988):

$$\frac{1}{t_I} \propto B = A\exp\left\{-\frac{\beta\gamma^3\nu^2}{k^3T^3\left[\ln\left(c/c^*\right)\right]^2}\right\} \tag{3.27}$$

where t_I is the induction time, *A* is a constant, *v* is the molar volume, *k* is the Boltzmann's constant, and *c** is the saturated solute concentration. The geometric shape factor is given by

$$\beta = \frac{4}{27}\frac{k_a^{\,3}}{k_v^{\,2}} \tag{3.28}$$

and the interfacial tension is expressed as (Mersmann, 1990)

$$\gamma = 0.414kT(\rho_c N_A)^{2/3}\ln(\rho_c / c^*) \tag{3.29}$$

where k_v and k_a are the volume and area shape factors, ρ_c is the density of the crystal, and N_A is the Avogadro's number.

For crystal growth kinetics, Karpinski (1985) discussed the importance of using the two-step crystal growth model that describes the crystal growth as the diffusion of solute molecules to the crystal surface followed by the arrangement of solute molecules into the crystal lattice. It was shown that the two-step growth model has a better accuracy than the overall kinetic growth equation. The diffusion layer model proposed by Nyvlt et al. (1985) described crystal growth into the following steps (see Figure 3.2):

1. Transfer of solute from the bulk solution to the diffusion layer.
2. Diffusion of the solute through the diffusion layer, whose thickness depends on the hydrodynamic conditions in the solution.
3. Incorporation of the solute molecules into the crystal lattice.

The diffusion rate across the diffusion layer can be written as

$$\frac{dm_c}{dt} = k_d A_c (c - c_i) \tag{3.30}$$

where (dm_c / dt) is the solute flux across the area, A_c, and k_d is the mass transfer coefficient. The concentration of species in the bulk phase of the supersaturated solution is *c*, and c_i is the interfacial concentration between the Volmer boundary layer and the diffusion layer. The rate of incorporation of solute into the crystal lattice can be written as

$$\frac{dm_c}{dt} = k_i A_c (c_i - c^*)^i \tag{3.31}$$

where k_i is the integration rate constant and *i* is an exponent whose value, between 1 and 2, depends on the surface integration mechanism. Equations (3.30) and (3.31) can be combined to eliminate c_i.

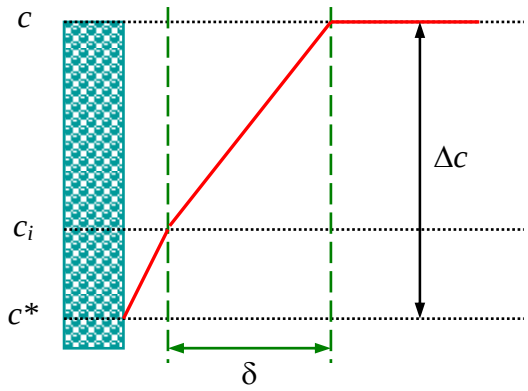

Figure 3.2 Diffusion layer growth model (Nyvlt et al., 1985).

For $i = 1$,

$$\frac{dm_c}{dt} = \left(\frac{1}{k_d} + \frac{1}{k_i} \right)^{-1} A_c (\Delta c) \tag{3.32}$$

where $\Delta c = c - c^*$ is the supersaturation. For $i = 2$,

$$\frac{dm_c}{dt} = A_c \frac{1}{2k_i / {k_d}^2} \left(1 + \frac{2k_i}{k_d} \Delta c - \sqrt{\frac{4k_i}{k_d} \Delta c + 1} \right) \tag{3.33}$$

For the case of undersaturation (negative Δc), the dissolution of crystals is a result of by mass transfer (Mullin, 2001):

$$\frac{dm_c}{dt} = k_d A_c \Delta c \tag{3.34}$$

Equations (3.33) and (3.34) can be expressed as a linear growth rate G by the relation (Myerson and Ginde, 2002)

$$\frac{1}{A_c} \frac{dm_c}{dt} = 3 \frac{k_v}{k_a} \rho_c \frac{dr}{dt} = 3 \frac{k_v}{k_a} \rho_c G \tag{3.35}$$

where k_v and k_a are the volume and area shape factors, respectively. A similar expression to equation (3.33) had been proposed by Mersmann et al. (1992), and it has been widely used (Gahn and Mersmann, 1999; Gerstlauer et al., 2002; Motz et al., 2002; Westhoff et al., 2002).

To determine the solid-liquid mass transfer coefficient, the Frossling equation can be applied for conditions of forced convection in steady flows (Mullin, 2001; Ohara and Reid, 1973). For particle-liquid mass transfer in agitated systems, various authors have applied Kolmogoroff theory of local isotropic turbulence in the mass transfer coefficient correlations (Calderbank and Moo-Young, 1961; Hughmark, 1969; Levich, 1962; Levins and Glastonbury, 1972b; Middleman, 1965). For microparticles, whose largest size dimension is smaller than the Kolmogoroff's length scale, it had

been proposed and validated that the turbulent contribution to mass transfer is by the boundary layer development resulting from the exchange of microparticles among decaying eddies (Armenante and Kirwan, 1989), which can be characterized by

$$\begin{aligned} Sh &= 2 + 0.52 Re^{0.52} Sc^{1/3} \\ d_p &\le \lambda_k = (\nu^3 / \varepsilon)^{0.25} \end{aligned} \tag{3.36}$$

where $Sh = k_d d_p / D$, $Re = \varepsilon^{1/3} d_p^{4/3} / \nu$ and $Sc = \nu / D$. The particle size is d_p, ν is the kinematic viscosity, and D is the laminar diffusivity, which is approximated as 10^{-9} m/s^2 (Deen, 1998). For macroparticles, it was suggested that the mass transfer occurs predominantly by a slip velocity mechanism with some contribution from unsteady state transfer (Levins and Glastonbury, 1972b):

$$\begin{aligned} Sh &= 2 + 0.5 Re^{0.62} Sc^{1/3} \\ d_p &> \lambda_k \end{aligned} \tag{3.37}$$

Both of these correlations were developed for an agitated vessel using the overall power input per unit mass to represent ε. However, the localized energy dissipation rate will be used in this work (Levins and Glastonbury, 1972a; Mori et al., 2002). The two-step growth rate expression is size-dependent as the mass transfer coefficient is a function of particle size.

For the simulations presented in the later chapters of the thesis, the kinetic parameters of the different systems simulated were obtained from published literature. The details on how the parameters are estimated are given in the cited references in the subsequent chapters. However, the uncertainties of the parameters were often not reported. The use of the published kinetics in the mixing simulations assumes that kinetic parameters are hydrodynamic-independent, which cannot be fully justified as the kinetics estimated would be convoluted with the transport limitations in the

crystallizer. Thus, suggestions will be made on how experiments could be designed to estimate mixing-independent crystallization kinetics in the following chapters.

3.7 Effective Viscosity of Suspensions

It is possible to incorporate the semidiscretized PBE to compute the CSD with the multiphase Eulerian model available in CFD codes. A direct approach would be to specify the crystals in each grid cell ($r_{j-1/2}$ to $r_{j+1/2}$) as a discrete solid phase with an average size in the same range. However, the Eulerian model requires much more intensive computations since a separate set of conservation equations has to be solved for each discrete phase. In this work, a simpler approach was used by treating the suspension as a pseudo-homogeneous phase with a spatial variation in the effective viscosity based on the localized suspension density. Numerous effective viscosity expressions for suspensions, valid for different ranges of solids concentration, can be found in the rheology literature, with most work extending from the Einstein's equation, which is valid for uniform spheres at low solids concentration (<10%) (Barnes et al., 1989; Einstein, 1906, 1911):

$$\mu = \mu_s(1 + 2.5\phi) \tag{3.38}$$

where μ is the viscosity of the suspension, μ_s is the viscosity of the suspending medium, and ϕ is the volume fraction of the solid phase. Many of the expressions for effective viscosity reduce to the Einstein equation in the limit of dilute suspension, which applies to organic crystals with low solubility. A significant amount of effort has also been put in to address the effects of particle size distribution and shape (Chang and Powell, 2002; Farris, 1968; Mooney, 1951; Parkinson et al., 1970; Phan-Thien and Pham, 2000; Sudduth, 1993). Except for rods and disks with large aspect ratios that can dramatically increase the effective viscosity even at low solids

concentration, the increase in viscosity and its sensitivities to the particle shape, particle size distribution, and particle-particle interaction are insignificant at low solids concentration ($\ll 10\%$ volume fraction) (Barnes et al., 1989).

3.8 Conclusions

The models and numerical approaches described in the above sections are integrated into the CFD solver using user-defined functions (Fluent 6.1 UDF Manual, 2003). The coupled algorithm is applied to simulate different crystallizer geometries and crystallization systems presented in the subsequent chapters.

3.9 References

Armenante, P. M. and Kirwan, D. J. (1989). Mass-transfer to microparticles in agitated systems. *Chemical Engineering Science* **44**(12): 2781-2796.

Bachalo, W. D. (1994). Experimental methods in multiphase flows. *International Journal of Multiphase Flow* **20**: 261-295.

Bardow, A., Marquardt, W., Goke, V., Koss, H. J. and Lucas, K. (2003). Model-based measurement of diffusion using Raman spectroscopy. *AIChE Journal* **49**(2): 323-334.

Barnes, H. A., Hutton, J. F. and Walters, K. (1989). An Introduction to Rheology. Amsterdam, Elsevier Science Publishers B. V.**:** 115-139.

Braatz, R. D., Fujiwara, M., Ma, D. L., Togkalidou, T. and Tafti, D. K. (2002). Simulation and new sensor technologies for industrial crystallization: A review. *International Journal of Modern Physics B* **16**(1-2): 346-353.

Calderbank, P. H. and Moo-Young, M. B. (1961). The continuous phase heat and mass-transfer properties of dispersions. *Chemical Engineering Science* **16**(1-2): 39.

Chang, C. Y. and Powell, R. L. (2002). Hydrodynamic transport properties of concentrated suspensions. *AIChE Journal* **48**(11): 2475-2480.

Deen, W. M. (1998). *Analysis of Transport Phenomena*. New York, Oxford University Press.

Einstein, A. (1906). A new determination of molecular dimensions. *Annalen der Physik* **19**: 289-306.

Einstein, A. (1911). Corrections: a new determination of molecular dimensions. *Annalen der Physik* **34**: 591-592.

Farris, R. J. (1968). Prediction of the viscosity of multimodal suspensions from unimodal viscosity data. *Journal of Rheology* **12**(2): 281-301.

Fluent 6.1 UDF Manual. (2003). Lebanon, New Hampshire, Fluent Inc.

Fluent 6.1 User's Guide. (2003). Lebanon, New Hampshire, Fluent Inc.

Fox, R. O. (2003). *Computational Models for Turbulent Reacting Flows*. Cambridge, U. K., Cambridge University Press.

Gahn, C. and Mersmann, A. (1999). Brittle fracture in crystallization processes. Part B. Growth of fragments and scale-up of suspension crystallizers. *Chemical Engineering Science* **54**(9): 1283-1292.

Gerstlauer, A., Motz, S., Mitrovic, A. and Gilles, E. D. (2002). Development, analysis and validation of population models for continuous and batch crystallizers. *Chemical Engineering Science* **57**(20): 4311-4327.

Gladden, L. F. (2003). Recent advances in MRI studies of chemical reactors: ultrafast imaging of multiphase flows. *Topics in Catalysis* **24**(1-4): 19-28.

Granberg, R. A., Bloch, D. G. and Rasmuson, A. C. (1999). Crystallization of paracetamol in acetone-water mixtures. *Journal of Crystal Growth* **199**: 1287-1293.

Gunawan, R., Fusman, I. and Braatz, R. D. (2004). High resolution algorithms for multidimensional population balance equations. *AIChE Journal* **50**(11): 2738-2749.

Hughmark, G. A. (1969). Mass transfer for suspended solid particles in agitated liquids. *Chemical Engineering Science* **24**(2): 291-297.

Hulburt, H. M. and Katz, S. (1964). Some problems in particle technology: A statistical mechanical formulation. *Chemical Engineering Science* **19**(8): 555.

Karpinski, P. H. (1985). Importance of the 2-step crystal-growth model. *Chemical Engineering Science* **40**(4): 641-646.

Kurganov, A., Noelle, S. and Petrova, G. (2001). Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations. *SIAM Journal on Scientific Computing* **23**(3): 707-740.

Kurganov, A. and Tadmor, E. (2000). New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *Journal of Computational Physics* **160**(1): 241-282.

Leveque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge, U. K., Cambridge University Press.

Levich, V. G. (1962). *Physiochemical Hydrodynamics*. Englewood Cliffs, N. J., Prentice-Hall, Inc.

Levins, D. M. and Glastonbury, J. R. (1972a). Application of Kolmogorofff's theory to particle--liquid mass transfer in agitated vessels. *Chemical Engineering Science* **27**(3): 537-543.

Levins, D. M. and Glastonbury, J. R. (1972b). Particle-liquid hydrodynamics and mass transfer in a stirred vessel. Part II - mass transfer. *Transactions of the Institution of Chemical Engineers* **50**: 132-146.

Ma, D. L., Braatz, R. D. and Tafti, D. K. (2002a). Compartmental modeling of multidimensional crystallization. *International Journal of Modern Physics B* **16**(1-2): 383-390.

Ma, D. L., Tafti, D. K. and Braatz, R. D. (2002b). High-resolution simulation of multidimensional crystal growth. *Industrial & Engineering Chemistry Research* **41**(25): 6217-6223.

Ma, D. L., Tafti, D. K. and Braatz, R. D. (2002c). Optimal control and simulation of multidimensional crystallization processes. *Computers & Chemical Engineering* **26**(7-8): 1103-1116.

Mahajan, A. J. and Kirwan, D. J. (1994). Nucleation and growth-kinetics of biochemicals measured at high supersaturations. *Journal of Crystal Growth* **144**(3-4): 281-290.

Marchisio, D. L., Barresi, A. A. and Fox, R. O. (2001a). Simulation of turbulent precipitation in a semi-batch Taylor-Couette reactor using CFD. *AIChE Journal* **47**(3): 664-676.

Marchisio, D. L., Fox, R. O., Barresi, A. A. and Baldi, G. (2001b). On the comparison between presumed and full PDF methods for turbulent precipitation. *Industrial & Engineering Chemistry Research* **40**(23): 5132-5139.

Marchisio, D. L., Fox, R. O., Barresi, A. A., Garbero, M. and Baldi, G. (2001c). On the simulation of turbulent precipitation in a tubular reactor via computational fluid dynamics (CFD). *Chemical Engineering Research & Design* **79**(A8): 998-1004.

Mersmann, A. (1990). Calculation of interfacial-tensions. *Journal of Crystal Growth* **102**(4): 841-847.

Mersmann, A., Angerhofer, M., Gutwald, T., Sangl, R. and Wang, S. (1992). General prediction of median crystal sizes. *Separations Technology* **2**(2): 85-97.

Middleman, S. (1965). Mass transfer from particles in agitated systems: application of the kolmogoroff theory. *AIChE Journal* **11**(4): 750-752, 760-761.

Moody, E. G. and Collins, L. R. (2003). Effect of mixing on the nucleation and growth of titania particles. *Aerosol Science and Technology* **37**(5): 403-424.

Mooney, M. (1951). The viscosity of a concentrated suspension of spherical particles. *Journal of Colloid Science* **6**(2): 162-170.

Mori, Y., Sha, Z. L., Louhi-Kultanen, M. and Kallas, J. (2002). CFD study of local crystal growth rate in a continuous suspension crystallizer. *Journal of Chemical Engineering of Japan* **35**(11): 1178-1187.

Motz, S., Mitrovic, A. and Gilles, E. D. (2002). Comparison of numerical methods for the simulation of dispersed phase systems. *Chemical Engineering Science* **57**(20): 4329-4344.

Mullin, J. W. (2001). *Crystallization*. Oxford, U. K., Elsevier Butterworth-Heinemann.

Myerson, A. S. and Ginde, R. (2002). Crystals, Crystal Growth, and Nucleation. *Handbook of Industrial Crystallization*. Boston, Butterworth-Heinemann**:** 33-66.

Nessyahu, H. and Tadmor, E. (1990). Non-oscillatory central differencing for hyperbolic conservation-laws. *Journal of Computational Physics* **87**(2): 408-463.

Nyvlt, J., Sohnel, O., Matuchova, M. and Broul, M. (1985). *The Kinetics of Industrial Crystallization*. New York, Elsevier.

Ohara, M. and Reid, R. C. (1973). *Modeling Crystal Growth Rates from Solution*. Englewood Cliffs, N. J., Prentice-Hall, Inc.

Parkinson, C., Matsumoto, S. and Sherman, P. (1970). The influence of particle-size distribution on the apparent viscosity of non-newtonian dispersed systems. *Journal of Colloid and Interface Science* **33**(1): 150-160.

Phan-Thien, N. and Pham, D. C. (2000). Differential multiphase models for polydispersed spheroidal inclusions: thermal conductivity and effective viscosity. *International Journal of Engineering Science* **38**(1): 73-88.

Piton, D., Fox, R. O. and Marcant, B. (2000). Simulation of fine particle formation by precipitation using computational fluid dynamics. *Canadian Journal of Chemical Engineering* **78**(5): 983-993.

Pope, S. B. (2000). *Turbulent Flows*. Cambridge, U. K., Cambridge University Press.

Randolph, A. D. and Larson, M. A. (1988). *Theory of Particulate Processes*. San Diego, California, Academic Press, Inc.

Shih, T. H., Zhu, J. and Lumley, J. L. (1995). A new reynolds stress algebraic equation. *Computer Methods in Applied Mechanics and Engineering* **125**(1-4): 287-302.

Sohnel, O. and Mullin, J. W. (1988). Interpretation of crystallization induction periods. *Journal of Colloid and Interface Science* **123**(1): 43-50.

Stanley, S. J., Mann, R. and Primrose, K. (2005). Interrogation of a precipitation reaction by electrical resistance tomography (ERT). *AIChE Journal* **51**(2): 607-614.

Sudduth, R. D. (1993). A generalized-model to predict the viscosity of solutions with suspended particles. 3. Effects of particle interaction and particle-size distribution. *Journal of Applied Polymer Science* **50**(1): 123-147.

Wang, L. G. and Fox, R. O. (2004). Comparison of micromixing models for CFD simulation of nanoparticle formation. *AIChE Journal* **50**(9): 2217-2232.

Westhoff, G. M., Butler, B. K., Kramer, H. J. M. and Jansens, P. J. (2002). Growth behaviour of crystals formed by primary nucleation on different crystalliser scales. *Journal of Crystal Growth* **237**: 2136-2141.

Chapter 4
Simulation of Antisolvent Crystallization in an Agitated Tank

4.1 Introduction

In the previous chapter, an approach to couple the computational fluid dynamics (CFD) computations to the solution of the population balance equation (PBE) to simulate the full crystal size distribution (CSD), and the solution of the probability density function (PDF) that describes the local fluctuations in the turbulent flow field, was presented. This approach can be integrated within commercially available CFD codes, in which the additional models are simulated within the CFD solver.

In this chapter, the coupled algorithm is applied to simulate the antisolvent crystallization of paracetamol from an acetone-water mixture (Granberg et al., 1999) in a semibatch stirred vessel. The rise in liquid level is captured by a dynamic mesh, which is commonly used for aeroelastic and free surface simulations (Duvigneau and Visonneau, 2004; Gao et al., 2002). The presence of solids is modeled by treating the slurry as a pseudo-homogeneous fluid with a spatial distribution of effective viscosity that depends on the local solids fraction (Barnes et al., 1989). The effects of agitation rate, addition mode, and scale-up on the transient CSD were numerically investigated.

4.2 Crystallization Kinetics of Paracetamol

The solubility, c^*, of paracetamol in an acetone-water mixture at 16^{o}C is given by (Granberg and Rasmuson, 2000):

$$
\begin{aligned}
&c^*(\text{kg of solute/kg of solvents}) \\
&= -5.01902\times10^{-12}w^6 + 1.69767\times10^{-9}w^5 - 2.46765\times10^{-7}w^4 + \\
&\quad 2.19262\times10^{-5}w^3 - 1.27018\times10^{-3}w^2 + 3.42614\times10^{-2}w + 7.96086\times10^{-2},
\end{aligned} \tag{4.1}
$$

or

$$
\begin{aligned}
&c^*(\text{kg of solute/m}^3) \\
&= -7.56719\times10^{-9}w^6 + 2.52296\times10^{-6}w^5 - 3.32604\times10^{-4}w^4 \\
&\quad + 2.33867\times10^{-2}w^3 - 1.01740w^2 + 2.33555\times10^{1}w + 6.08849\times10^{1},
\end{aligned} \tag{4.2}
$$

where w is the antisolvent mass percent on a solute-free basis. The kinetic rates of primary nucleation and growth of paracetamol in an acetone-water mixture at 16°C were approximated from the experimental data given by Granberg et al. (1999). In this work, we assume that the experiments in this publication were performed under well-mixed conditions. The primary nucleation rate, B, is expressed as

$$
B(\text{no. of nuclei/s-m}^3) = 8.56080\times10^{8}\exp\left\{-1.22850\times10^{-3}\frac{\left[\ln\left(\frac{\rho_c(\text{kg/m}^3)}{c^*(\text{kg/m}^3)}\right)\right]^3}{\left[\ln\left(\frac{c(\text{kg/kg})}{c^*(\text{kg/kg})}\right)\right]^2}\right\} \tag{4.3}
$$

where ρ_c is the density of the crystal and c is the supersaturated solute concentration.

Crystal growth was modeled using the diffusion layer model (Nyvlt et al., 1985) described in the previous chapter.

$$
G = \frac{k_a}{6\rho_c k_v}\frac{1}{k_i/{k_d}^2}\left(1+\frac{2k_i}{k_d}\Delta c - \sqrt{\frac{4k_i}{k_d}\Delta c + 1}\right) \tag{4.4}
$$

$$
\Delta c = c - c^*
$$

Assuming that the crystal growth in Granberg et al. (1999) is measured without mass transfer limitations, the integration rate constant is

$$
k_i(\text{kg/(m}^2\text{-s-(kg/m}^3)^2)) = 1.95\times10^{-7}w - 7.35\times10^{-6} \quad (w \geq 30\%) \tag{4.5}
$$

Average values of the shape factors, $k_v = 0.605$ and $k_a = 4.63$, are assumed in the computations. For the case of undersaturation (negative Δc), the dissolution of crystals is a result of by mass transfer (Mullin, 2001):

$$G = \frac{k_a}{3\rho_c k_v} k_d \Delta c \tag{4.6}$$

The correlations of the mass transfer coefficient are given in the previous chapter.

For this crystallization system, only the kinetic parameters for primary nucleation and growth have been reported in the literature (Granberg et al., 1999). Hence, the goal of the simulations is to gain insights into how hydrodynamics affect crystal nucleation and growth. Agglomeration, breakage, and secondary nucleation for this system can be minimized by controlled seeding or by adjusting some of the operating parameters such as the agitation speed and antisolvent addition rate (Fujiwara et al., 2005; Yu et al., 2006; Yu et al., 2005). If more complex crystallization kinetic parameters were made available, then a wider range of operating conditions could be simulated. For effective viscosity, the Einstein equation remains valid for this system as the suspension is sufficiently dilute (< 4%) and no needles or thin plates are formed (Barnes et al., 1989).

4.3 Validation of High-Resolution Central Scheme

To assess the accuracy of the high-resolution central scheme in solving the PBE written on a mass basis, it was applied to the well-mixed case of antisolvent crystallization of paracetamol in an acetone-water mixture. Equation (4.7), without the external coordinate terms, was solved for nucleation and size-independent growth using the ODE solver in Matlab 6.1.

$$\frac{d}{dt} f_{w,j} + \sum_{i}^{3} \left\{ \frac{\partial [v_i f_{w,j}]}{\partial x_i} - \frac{\partial}{\partial x_i} \left[D_t \frac{\partial f_{w,j}}{\partial x_i} \right] \right\}$$

$$= \begin{cases} \dfrac{\rho_c k_v}{4\Delta r} \left[(r_{j+1/2})^4 - (r_{j-1/2})^4 \right] & \\ \times \left\{ -G_{j+1/2} \left[f_j + \dfrac{\Delta r}{2} (f_r)_j \right] + G_{j-1/2} \left[f_{j-1} + \dfrac{\Delta r}{2} (f_r)_{j-1} \right] + \underbrace{B}_{j=0} \right\}, & \Delta c > 0 \\ \dfrac{\rho_c k_v}{4\Delta r} \left[(r_{j+1/2})^4 - (r_{j-1/2})^4 \right] & \\ \times \left\{ -G_{j+1/2} \left[f_{j+1} - \dfrac{\Delta r}{2} (f_r)_{j+1} \right] + G_{j-1/2} \left[f_j - \dfrac{\Delta r}{2} (f_r)_j \right] \right\}, & \Delta c < 0 \end{cases} \tag{4.7}$$

The nucleation rate expression was Equation (4.3), while Granberg et al. (1999) fitted the growth rate expression:

$$G\,(\text{m/s}) = k_g (\Delta c\,(\text{kg of solute/kg of solvents}))^g \tag{4.8}$$

to experimental data, with the kinetic constants

$$\begin{aligned} & k_g\,(\text{m/s-(kg of solute/kg of solvents)}^{\text{g}}) \\ & = -1.60\times10^{-10} w^3 + 5.59\times10^{-8} w^2 - 2.10\times10^{-6} w + 6.14\times10^{-5} \end{aligned} \tag{4.9}$$

and

$$g = -1.11\times10^{-4} w^2 + 1.02\times10^{-2} w + 1.43 \tag{4.10}$$

In this well-mixed validation run, a 1 L vessel was initially half-filled with saturated solution of 65% by mass water, and antisolvent added at a constant rate over 1 h to fill the vessel (i.e., direct addition). As shown in Figure 4.1, the zeroth through seventh order moments of the CSD computed from equation (4.7) for various Δr are compared to those obtained from the method of moments (Randolph and Larson, 1988). For $\Delta r = 1$ μm, the moments agree well and, as Δr increases, the overestimation of the higher order moments increases due to numerical diffusion, which depends on Δr.

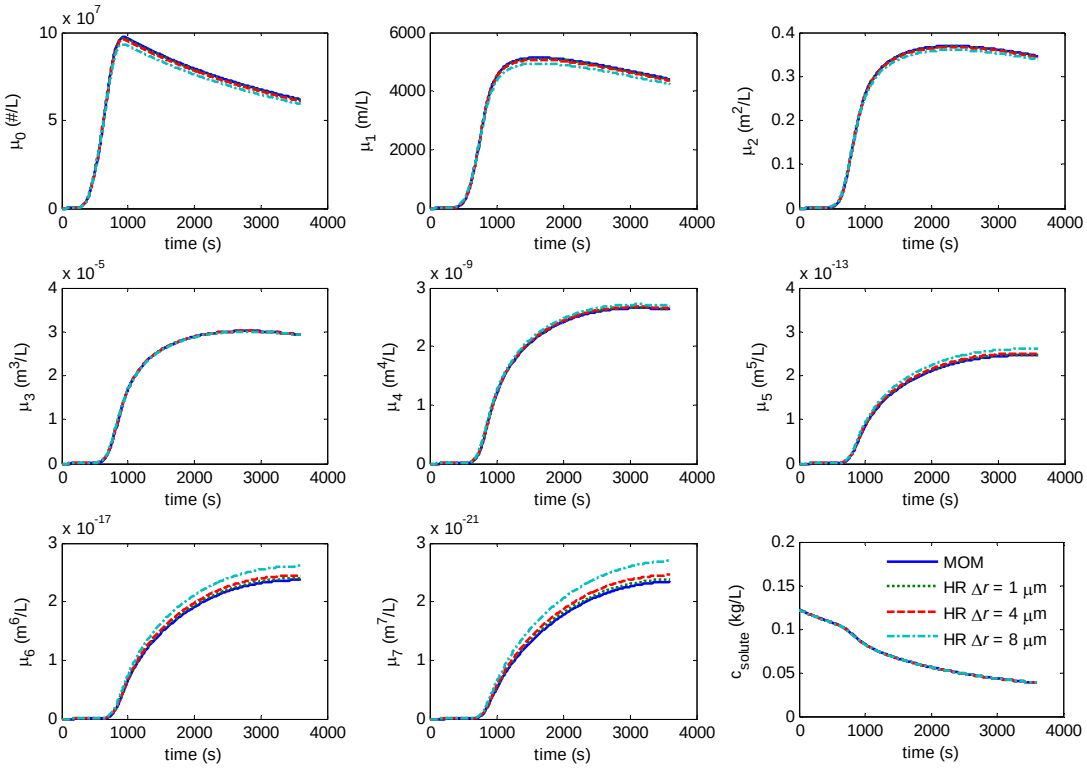

Figure 4.1 Zeroth through seventh order moments and solute concentration from the method of moments (MOM) and the high-resolution central scheme (HR), Equation (3.17), for various Δr.

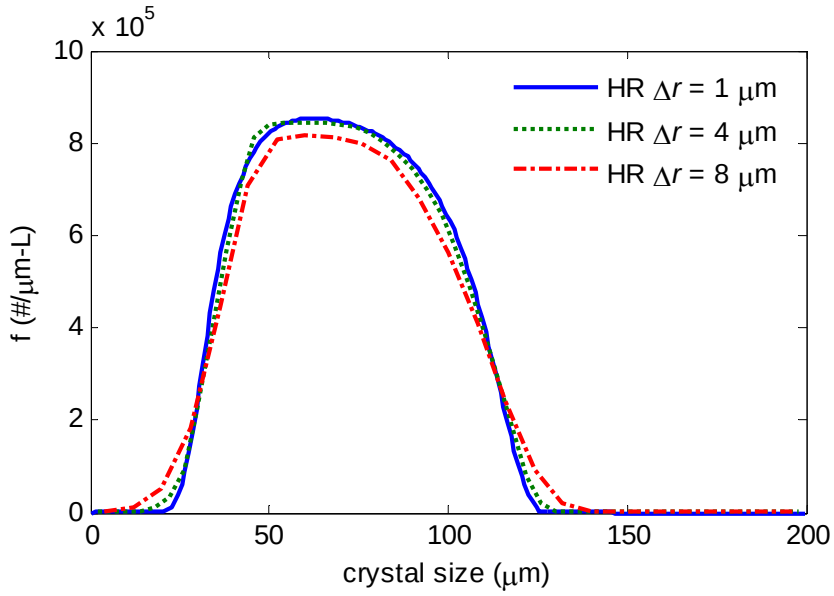

Figure 4.2 CSD from high-resolution central scheme (HR), Equation (3.17), for various Δr.

Figure 4.2 shows the final CSD at the end of the batch. Small numerical diffusion is observed, with the absence of nonphysical oscillations.

4.4 Implementation of CFD-PBE-Micromixing Algorithm

This section describes the implementation the coupled CFD-PBE-PDF algorithm for simulating the antisolvent crystallization in a semibatch agitated tank. Fluent 6.1.22 (Fluent Inc., Lebanon) was employed as the CFD solver, which solves the Reynolds-averaged Navier-Stokes (RANS) and conservation equations (see previous chapter). The 2D axisymmetric mesh for the 1 L cylindrical vessel was generated using Gambit 2.2 (Fluent Inc., Lebanon). The use of 2D simulation can be justified by the use of a well-baffled stirred tank where the mixing is dominated by axial flow. The turbulence in the vessel was modeled by the realizable *k*-ε model (see previous chapter) with standard wall functions, while the impeller (Lightnin A200), placed a quarter of the vessel height from the bottom, was modeled by fixed velocity data[i] (Marshall and Bakker, 2004) for simplicity.

A steady-state flow field was first obtained for the fluid in the half-filled vessel, where $p_2 = 1$, before the introduction of the feed. The addition of the feed was modeled by a mass and p_1 source located at the feed point, corresponding to the feed flow rate, along with a momentum source to capture the downward injection. The feed rate was determined by the constant flow rate required to fill the other half of the vessel in a 1 h batch time. The rise in liquid level was modeled using a dynamic mesh by the constant height dynamic layering method (Fluent 6.1 User's Guide, 2003), in

[i] Fixed velocity data was obtained from http://www.fluentusers.com/mixsim/faq/faq5.htm and was applied as momentum and turbulence sources in the row of cells below the impeller region.

which computational cells were added adjacent to the moving boundary. The rate at which the boundary of the liquid surface moved was computed on the basis of the addition rate of the antisolvent.

The additional equations described in the previous section were included into the CFD algorithm through user-defined functions, and the additional transport equations were solved as user-defined scalars (Fluent 6.1 UDF Manual, 2003) . All CFD simulations were carried out on a Dell PowerEdge Linux cluster with an Intel Xeon 3.2 GHz processor[ii]. Here, antisolvent crystallization by the direct addition mode, similar to the system described in the well-mixed case, was modeled. The PBE was discretized into 40 grid cells with $\Delta r = 8\,\mu\text{m}$. Each simulation, for a batch time of 1 h, took approximately 5 days to complete on a single CPU. All simulation runs for this paper can be run in parallel on a modest-sized Linux cluster (with each simulation run sent to a different processor). Further, the time for each simulation run can be reduced by a factor of ~10 on such a cluster by using the parallelization capability in Fluent.

4.5 Effects of Agitation Rate

This section illustrates the capability of the coupled CFD-PBE-PDF approach through the investigation of the effects of agitation rate on the CSD. Figure 4.3 shows that the segregation of the feed and the initial solution, on the basis of the three-environment micromixing model, only occurs for a very short time (< 10 s), and the reduction of segregation is faster at a higher agitation rate. The initial mixing has small regions of mixed solution (environment 3) with equal proportions of the solution and the antisolvent ($\langle \xi \rangle_3 = 0.5$), as shown in Figure 4.4, with localized regions of high

[ii] http://www.ncsa.uiuc.edu/UserInfo/Resources/Hardware/XeonCluster/

supersaturation and rapid nucleation and growth rates (see Figures 4.6 to 4.9). Subsequently, the mixed environment extends throughout the vessel, with its major proportion being the initial solution in the vessel ($\langle \xi \rangle_3 \to 0$). The amount of antisolvent in environment 3 increases throughout the batch ($\langle \xi \rangle_3 \to 0.5$). This mixing sequence, as shown in Figure 4.5, results in a drop in antisolvent composition, supersaturation, and thus nucleation and growth rates, in the mixed environment, which then slowly increases with time. This is followed by the supersaturation going through a maximum (at t = 440 s for 500 rpm) and decreasing due to consumption of solute for crystal growth after a substantial amount of nuclei has formed.

Figure 4.10 shows the evolution of CSD during the first few minutes and throughout the entire batch at 500 rpm. Due to the initial micromixing effects, small amounts of crystals are formed, with lower agitation speeds resulting in more crystals due to slower micromixing (plot not shown). The nucleation and growth rates at the inlet drop quickly during the first few seconds (see Figures 4.8 and 4.9), after which the crystals grow with increasing growth rate (see top plot in Figure 4.10). Such crystals formed during the initial contact of the solution and the antisolvent have been observed in some experiments (Mullin et al., 1989) and do not represent the detection of a metastable limit for the overall solution. It is not surprising to observe higher growth rates at the impeller region due to higher turbulence (Figure 4.9). Higher growth rates are observed for higher agitation rates (see Figure 4.5), due to the reduction of mass transfer limitations on crystal growth. Consequently, the faster desupersaturation at higher agitation results in lower overall nucleation rates (Figure 4.5). This explains the final CSD (at the end of 1 h) for different agitation rates, in

which fewer and slightly larger crystals are obtained with higher agitation rates (see Figure 4.11).

The dependence of the final CSD on agitation rate is consistent with the experimental results of Kim et al. (2000), where the crystal size of titanium (IV) oxide formed by a supercritical reaction increased with mid-range stirring rate. Torbacke and Rasmuson (2004) also report an increase in product mean size with stirring rate for semibatch reaction crystallization of benzoic acid. However, no influence of stirring speed on the mean size of nanoparticles formed was observed in the experimental study of drowning out of ethylcellulose (Plasari et al., 1997). For the crystallization of griseofulvin by compressed carbon dioxide as an antisolvent (De Gioannis et al., 2004), and the salting out of $KAl(SO_4)_2$ (Nyvlt and Zacek, 1986), it was experimentally observed that the mean crystal size decreased with increasing stirring rate. Such opposing observations can be attributed to the different kinetics of each individual system.

A recent publication on the paracetamol-acetone-water system reports that the mean particle (crystals and agglomerates) size increases and subsequently decreases with mixing speed (Yu et al., 2005). However, no direct comparison with the crystal size distribution can be made due to the high degree of agglomeration at the lower mixing speeds and higher antisolvent addition rates. The authors of that paper loaned us the crystal samples from their experiments, and the larger, less agglomerated crystals were sieved out and measured under an optical microscope. It can be seen from Figure 4.12 that larger crystals were obtained at a higher agitation rate, consistent with the simulation results shown in Figure 4.11.

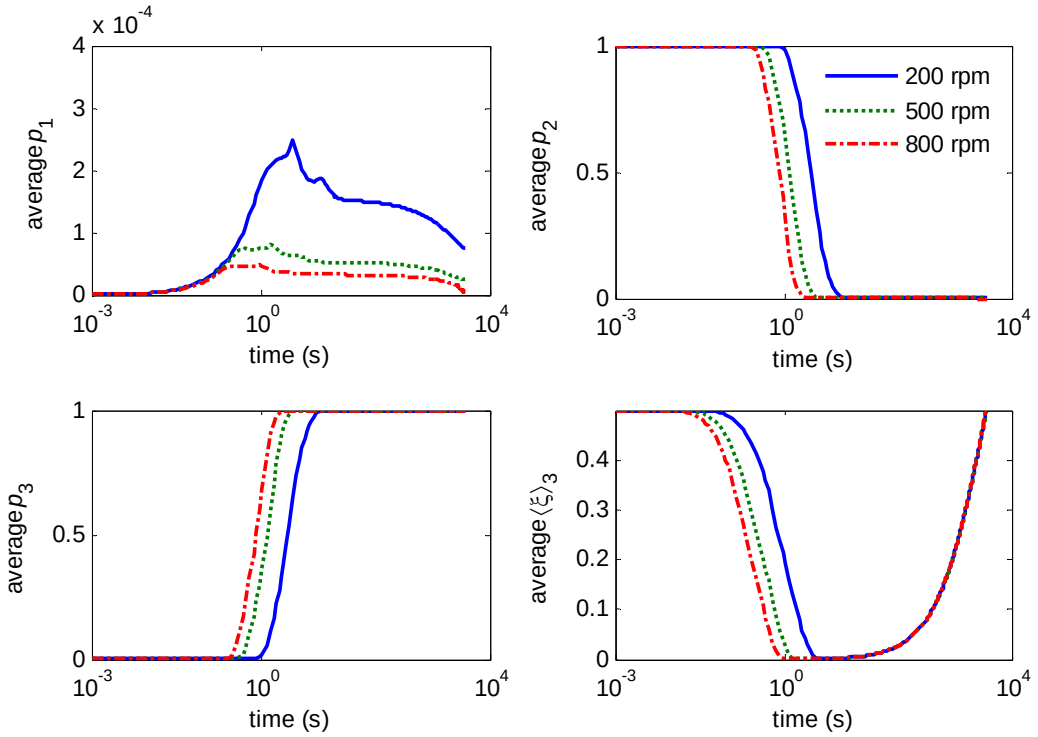

Figure 4.3 Volume-averaged p_1, p_2, p_3 and $\langle \xi \rangle_3$ for various agitation rates.

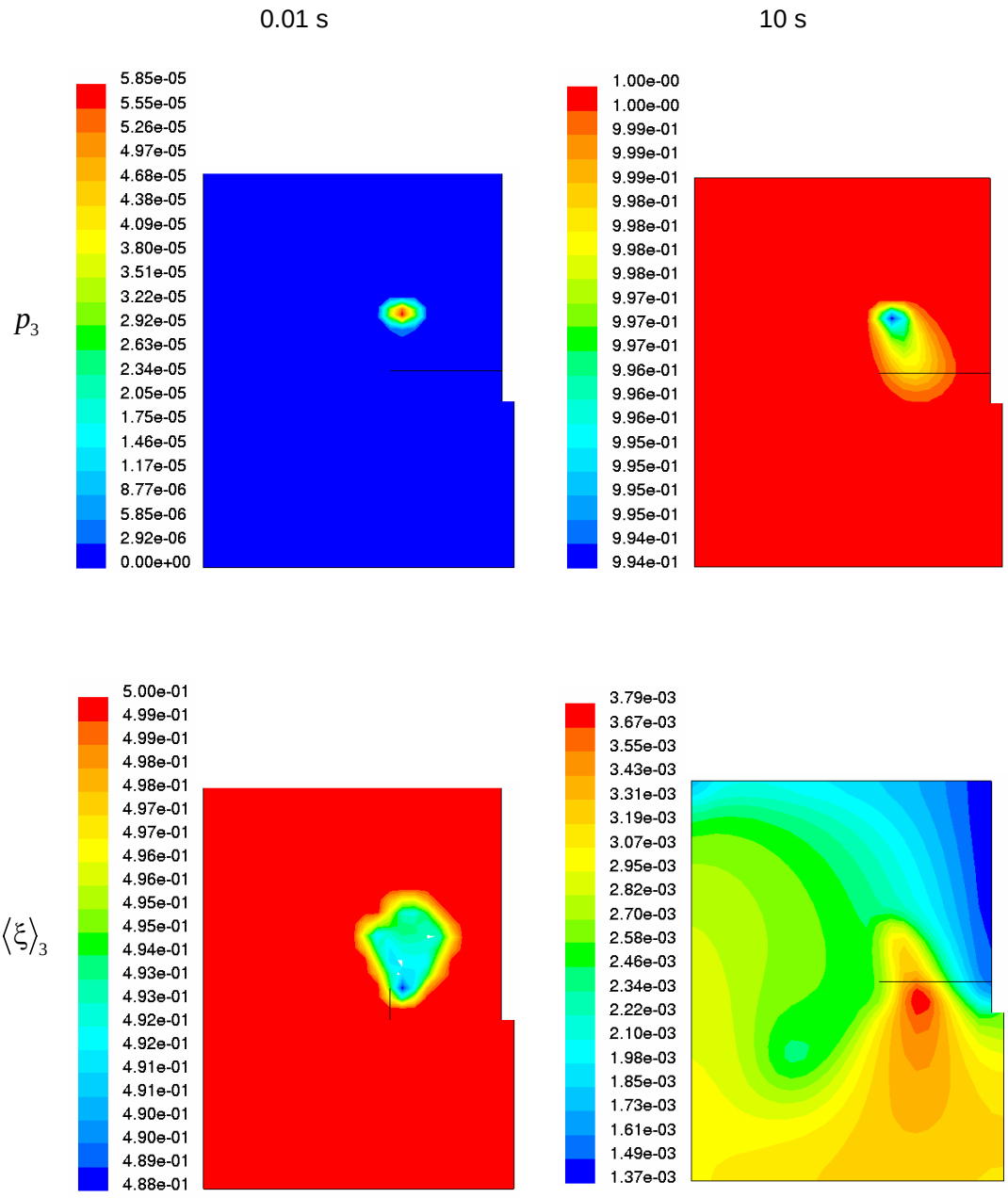

Figure 4.4 Spatial distributions of p_3(mixed), and $\langle\xi\rangle_3$ at 500 rpm during initial mixing.

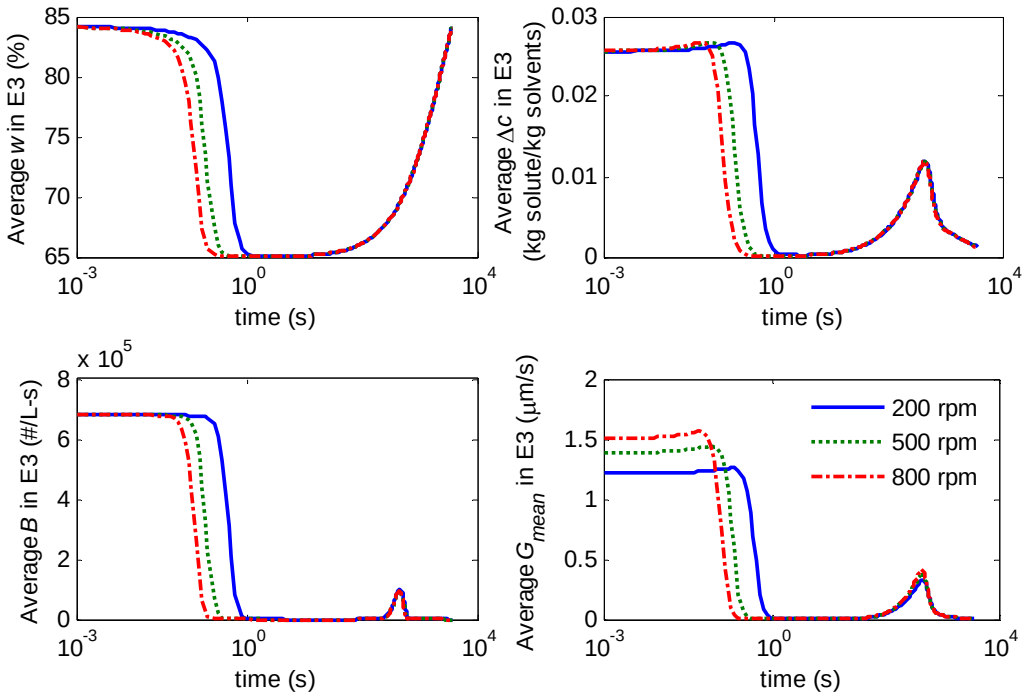

Figure 4.5 Volume-averaged antisolvent mass% (w), supersaturation (Δc), nucleation rate (B), and mean growth rate of crystals of all sizes (G_{mean}), in the mixed Environment 3 (E3) for various agitation rates.

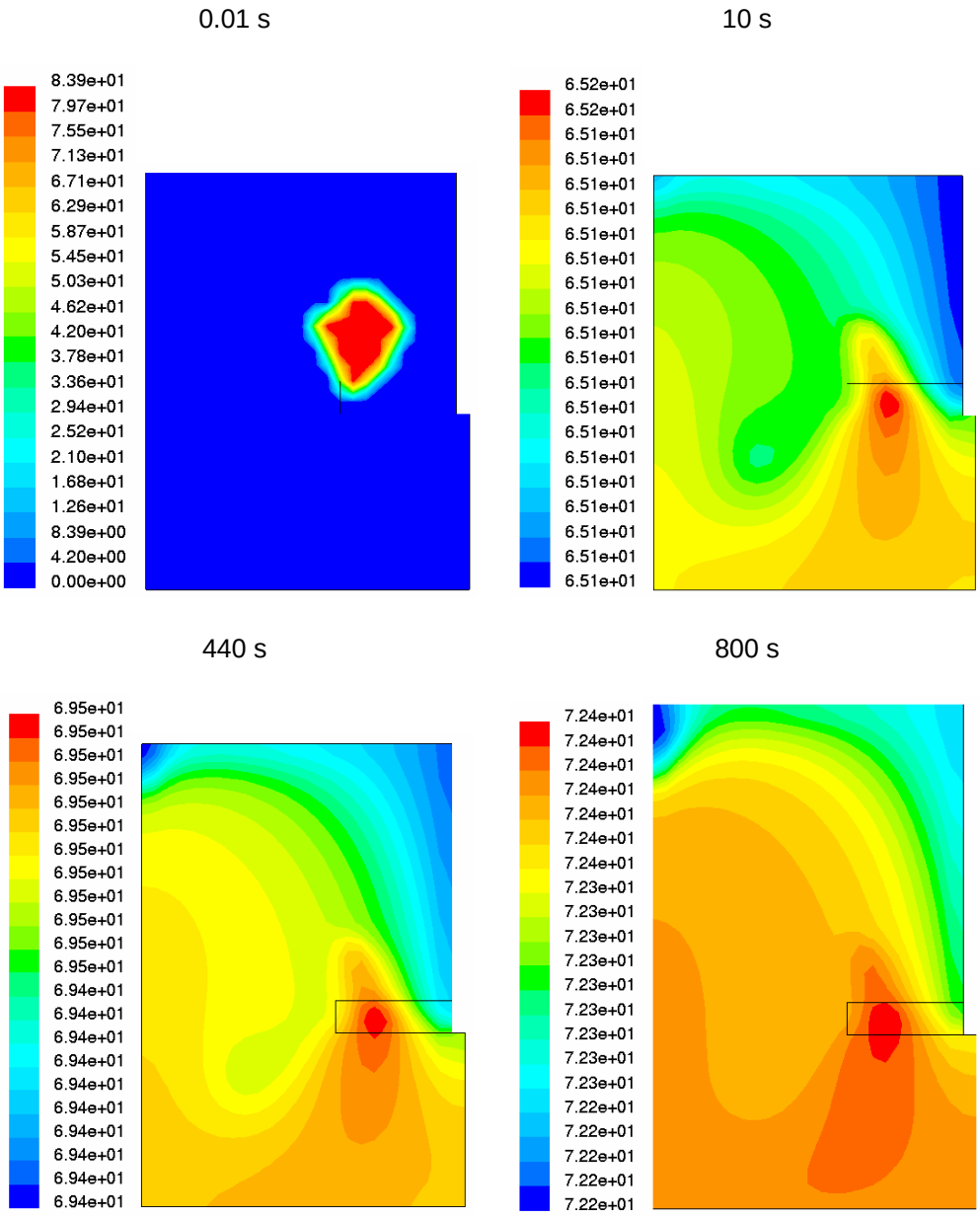

Figure 4.6 Spatial distribution of *w* (antisolvent mass%) in Environment 3 at 500 rpm for various times.

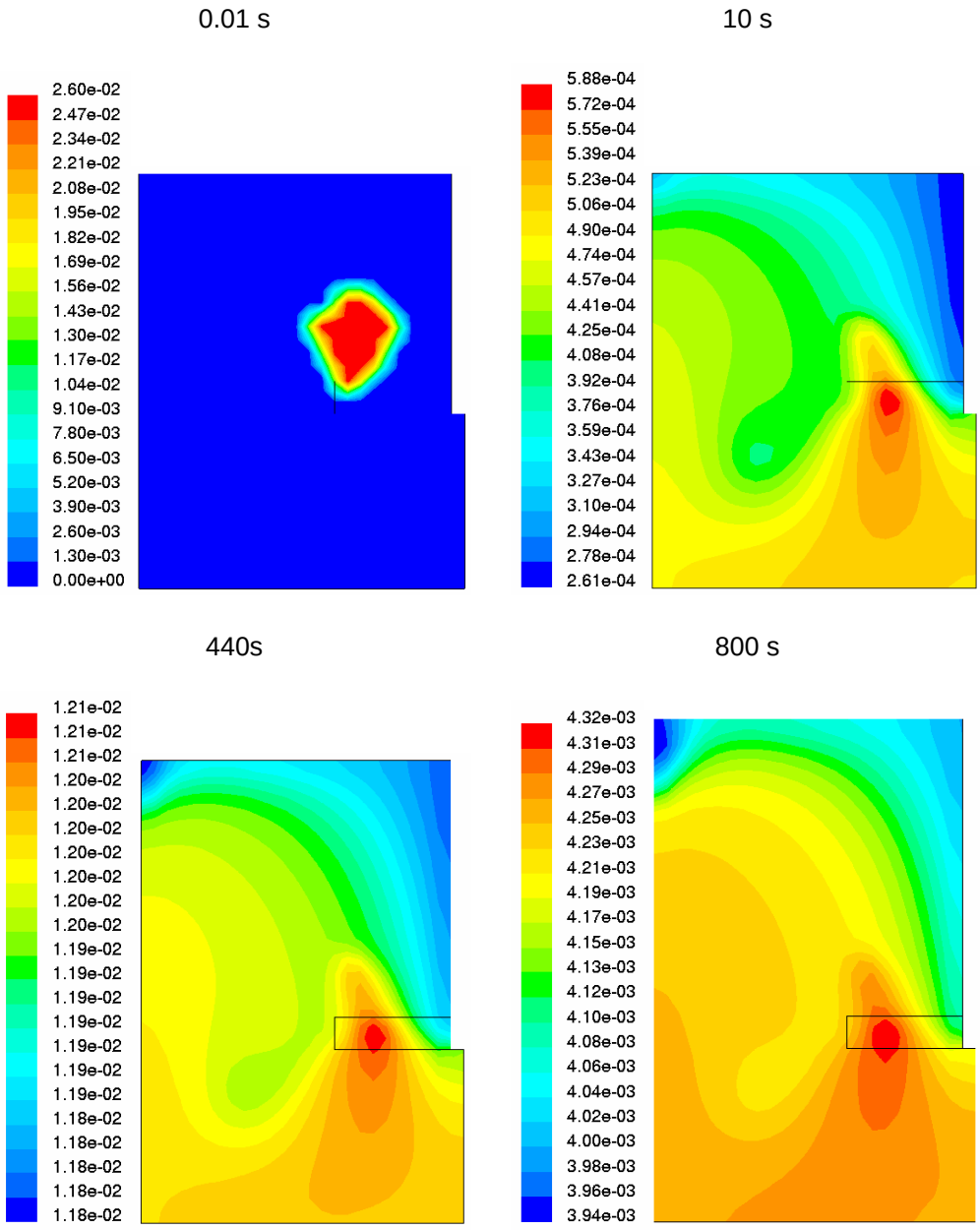

Figure 4.7 Spatial distributions of supersaturation Δc (kg solute/kg solvents) in Environment 3 at 500 rpm for various times.

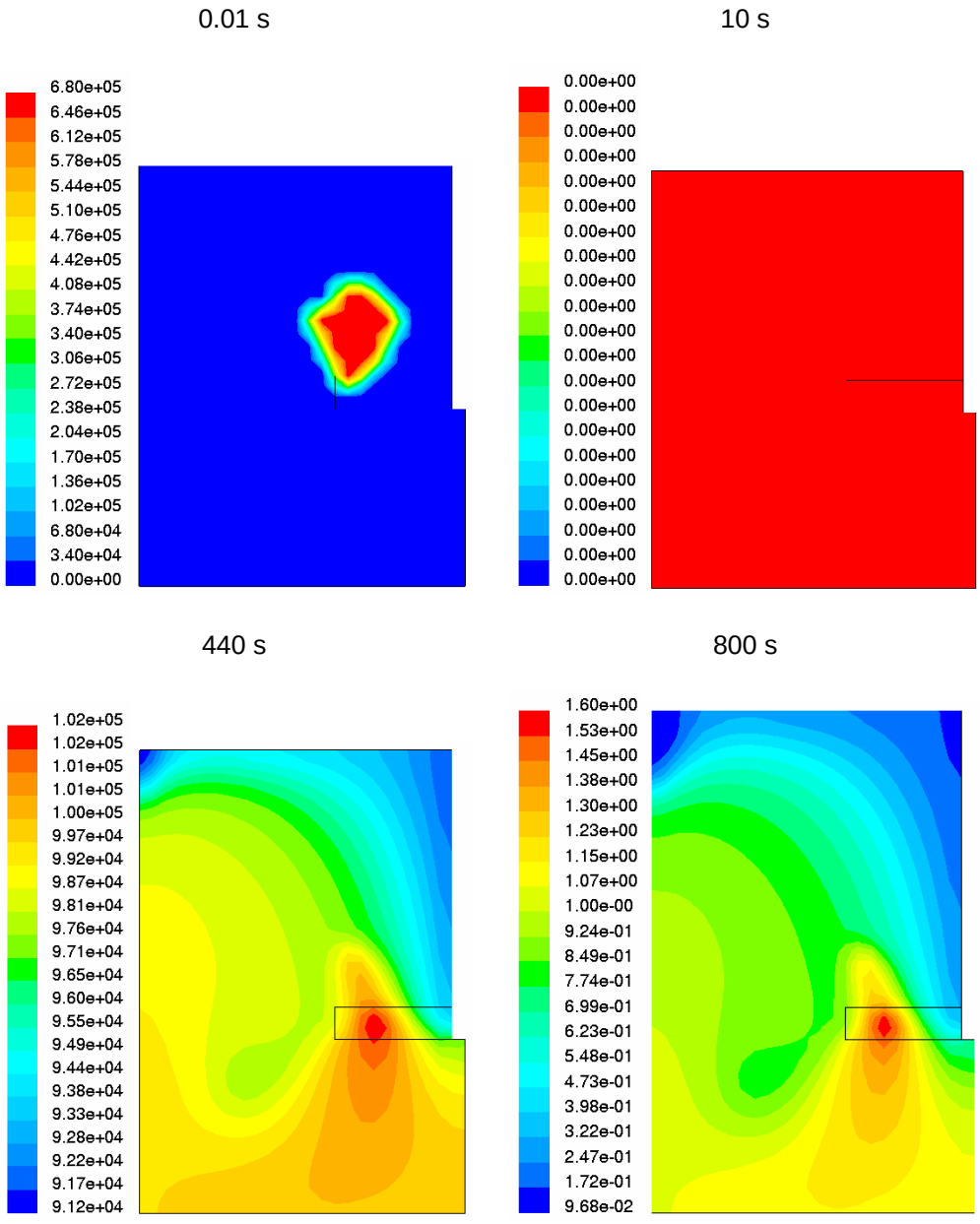

Figure 4.8 Spatial distributions of the nucleation rate *B* (#/L-s) in Environment 3 at 500 rpm for various times.

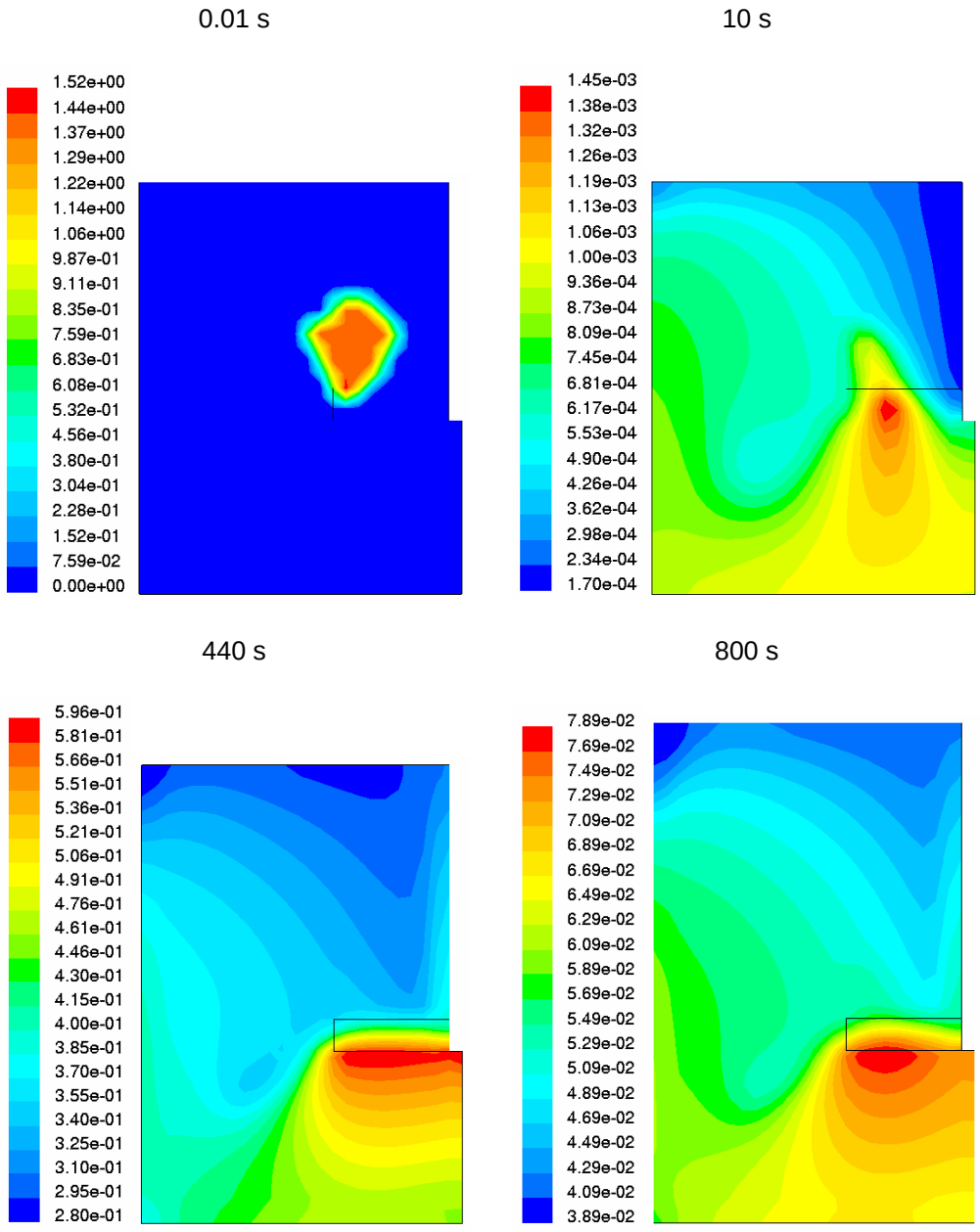

Figure 4.9 Spatial distributions of the mean growth rate G_{mean} (μm/s) in Environment 3 at 500 rpm for various times.

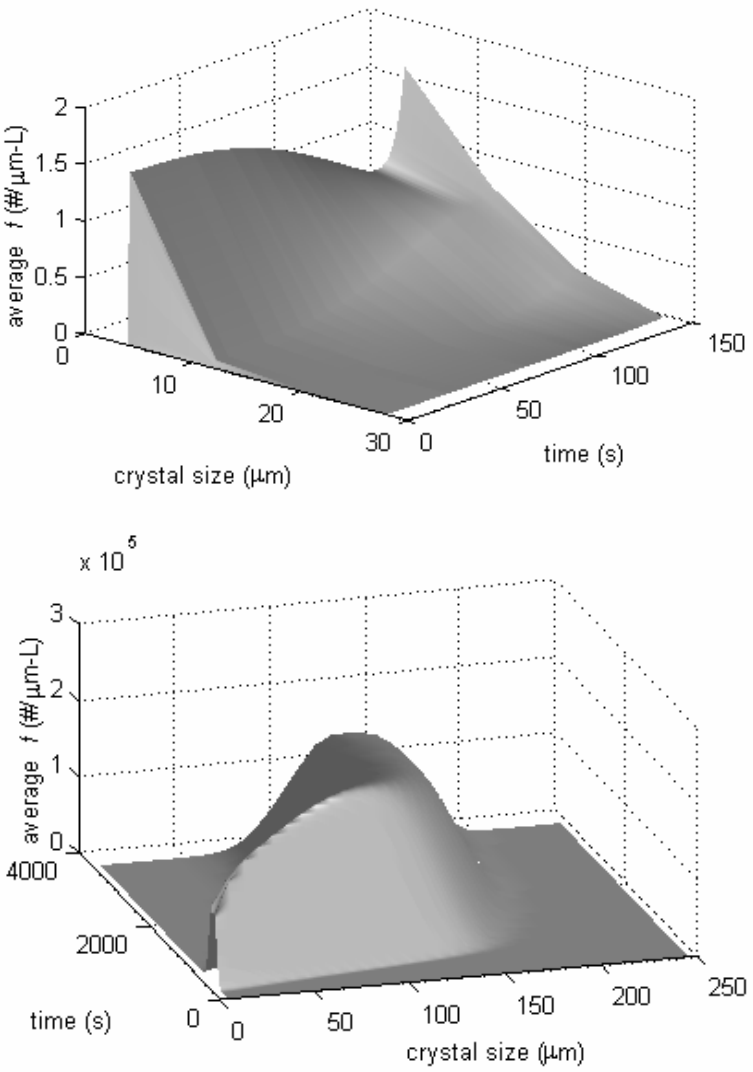

Figure 4.10 Evolution of the volume-averaged CSD at 500 rpm.

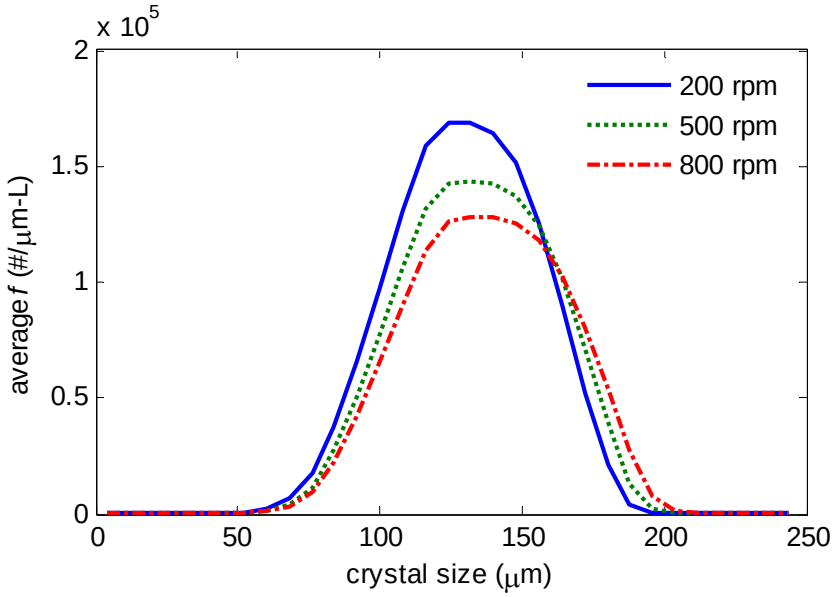

Figure 4.11 Final volume-averaged CSD for various agitation rates (direct addition).

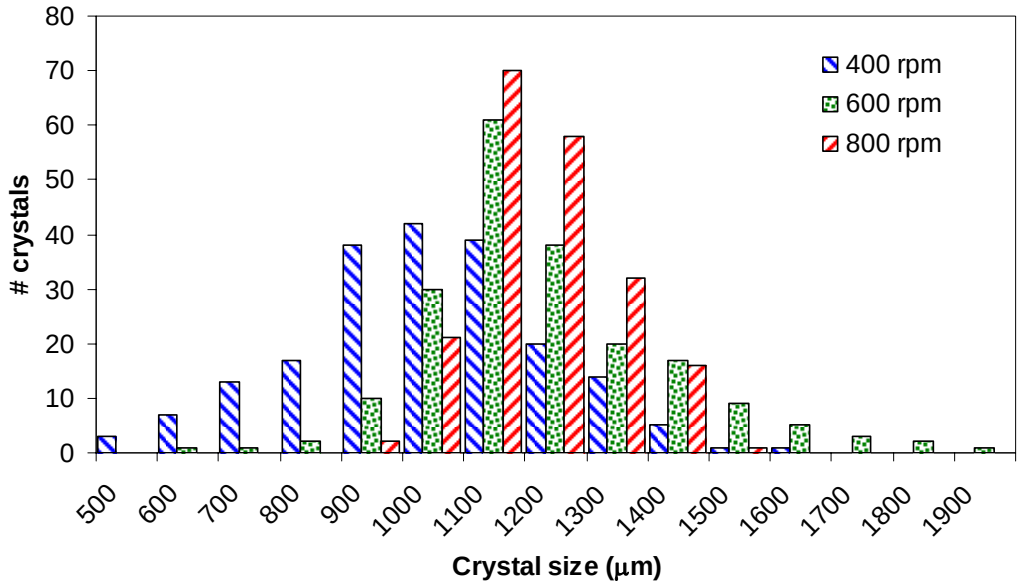

Figure 4.12 Crystal size distribution of paracetamol crystals obtained from Yu et al. (Yu et al., 2005) for an antisolvent addition rate of 2 g/min for various agitation rates. The larger and less agglomerated crystals were obtained by sieving (600 μm sieve) and the length of the longest axis of the single crystals (200 crystals total) were measured under an optical microscope (Olympus BX51).

4.6 Effects of Addition Mode

Reverse addition (addition of saturated solution to antisolvent) is used in the pharmaceutical industry to crystallize small particles (Midler et al., 1994). This addition mode was used in the crystallization studies by other researchers as well Kim et al. (Kim et al., 2003; Plasari et al., 1997; Shin and Kim, 2002). To the authors' knowledge, the comparison between the effects of both addition modes has yet to be extensively studied. Here, the reverse addition mode was modeled with the same volumes of saturated solution and antisolvent as the direct addition case. This simply involved switching the concentrations in environments 1 and 2.

The time profiles of the average antisolvent composition, supersaturation, and nucleation and growth rates in environment 3 during the initial contact of the feed solution and the antisolvent are the same as the direct addition case (see Figure 4.13). Subsequently, during reverse addition the excessive dilution of the saturated solution by the antisolvent results in undersaturation and dissolution of the crystals formed at the initial contact (see Figure 4.14). For most of the crystallization the supersaturation at longer times during reverse addition is not as high as in direct addition, while the peak nucleation rate is significantly higher (see Figure 4.13). This is a result of the dependence of the nucleation and surface integration rates on solvent composition. In the paracetamol-acetone-water system, the nucleation and surface integration rates increase with increasing antisolvent composition. During reverse addition, a high antisolvent composition is achieved, which results in the formation of a larger number of nuclei and reduced growth on individual crystals (see Figure 4.13). Consequently, the final CSD for the reverse addition mode, shown in Figure 4.15, has more crystals of significantly smaller size.

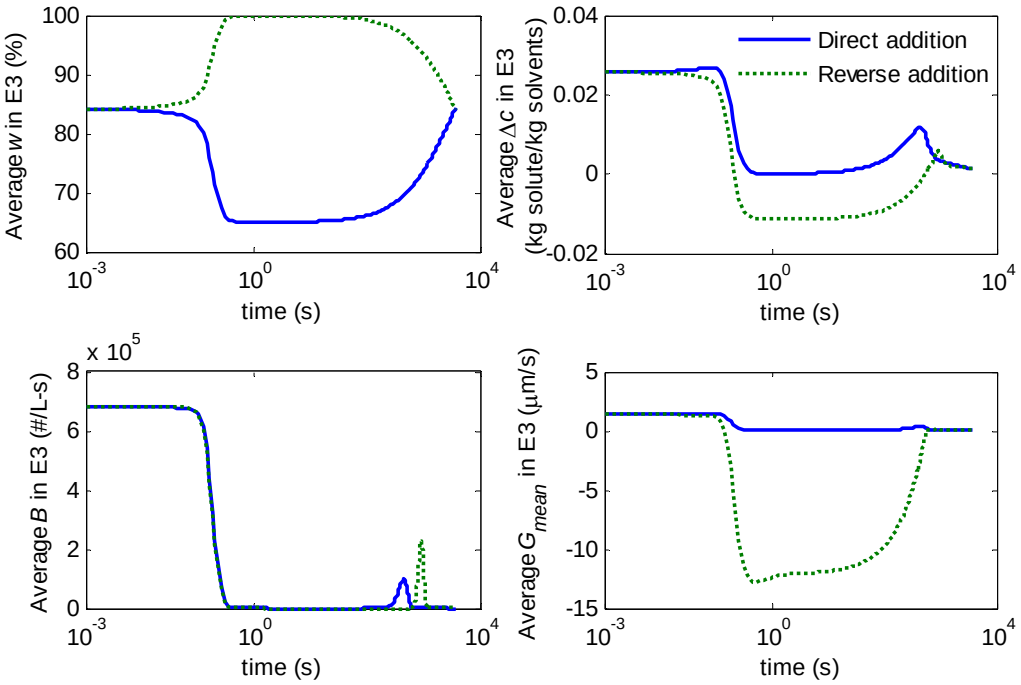

Figure 4.13 Volume-averaged antisolvent mass% (w), supersaturation (Δc), nucleation rate (B). and mean growth rate (G_{mean}) in Environment 3 (E3) at 500 rpm for direct and reverse addition modes.

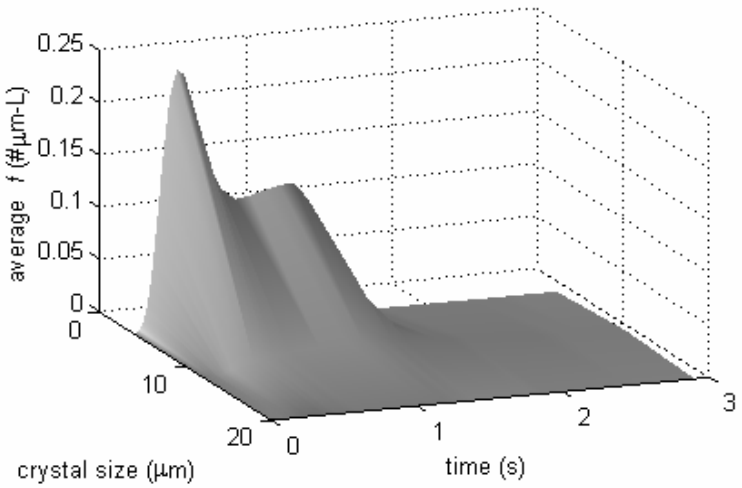

Figure 4.14 Evolution of volume-averaged CSD at 500 rpm for reverse addition.

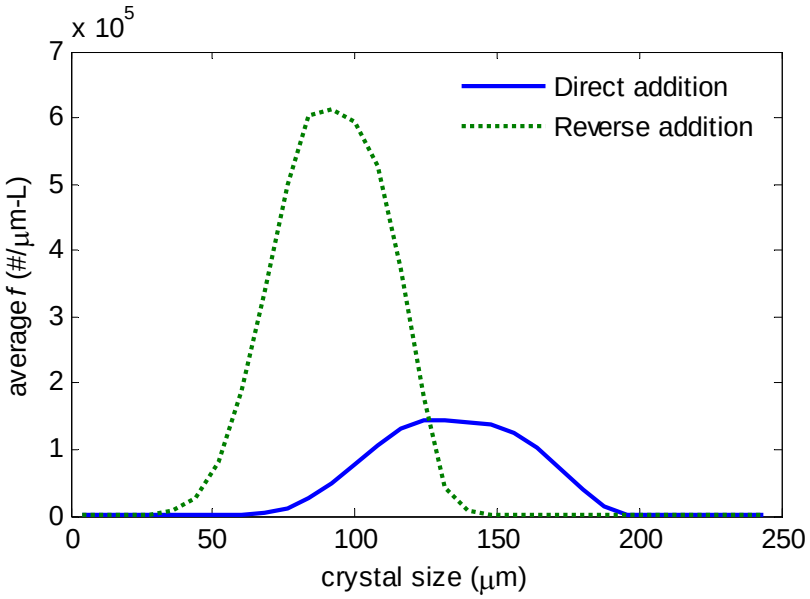

Figure 4.15 Final volume-averaged CSD at 500 rpm for direct and reverse addition modes.

4.7 Effects of Scale-up

The scale-up of crystallizers has long been a challenge for the pharmaceutical industry. The coupled algorithm enables the investigation of the effects of scale-up. Here, the 1 L vessel was scaled up to 125 L while its geometric similarity was maintained. The impeller speed, based on 500 rpm for the small scale, was scaled up according to two common scale-up rules: (i) constant tip speed (100 rpm) and (ii) constant power per unit volume (171 rpm) (Green, 2002; Kresta and Brodkey, 2004). The simulations for the scale-up studies for direct addition were performed for 20 min of batch time, after which the change in CSD was minimal, due to slow growth for the rest of the batch.

As shown in Figures 4.16 and 4.17, the dispersion of the feed solution is less effective on a larger scale for both scale-up criteria. By comparison of the spatial plots for the 1 L scale with both scale-up cases for the 125 L-scale (not shown) at the time of highest supersaturation (t = 440 s), somewhat higher inhomogeneities in the antisolvent composition, the supersaturation, and the nucleation rate can be observed for the large vessel. The difference in spatial variation of the growth rate is a consequence of the spatial distribution of the turbulent energy dissipation not being preserved after scale-up. The CSD obtained at the end of 20 min is shown in Figure 4.18, with the scale-up based on constant power per unit volume giving a better match with the CSD of the small scale. Nevertheless, no significant change in the crystal size distribution was observed on scale-up, which is consistent with the experimental findings reported in literature (Torbacke and Rasmuson, 2004), where the reactor size was observed to have no influence on the product mean size. More drastic differences in the CSD are expected after scale-up for crystallization systems where secondary nucleation,

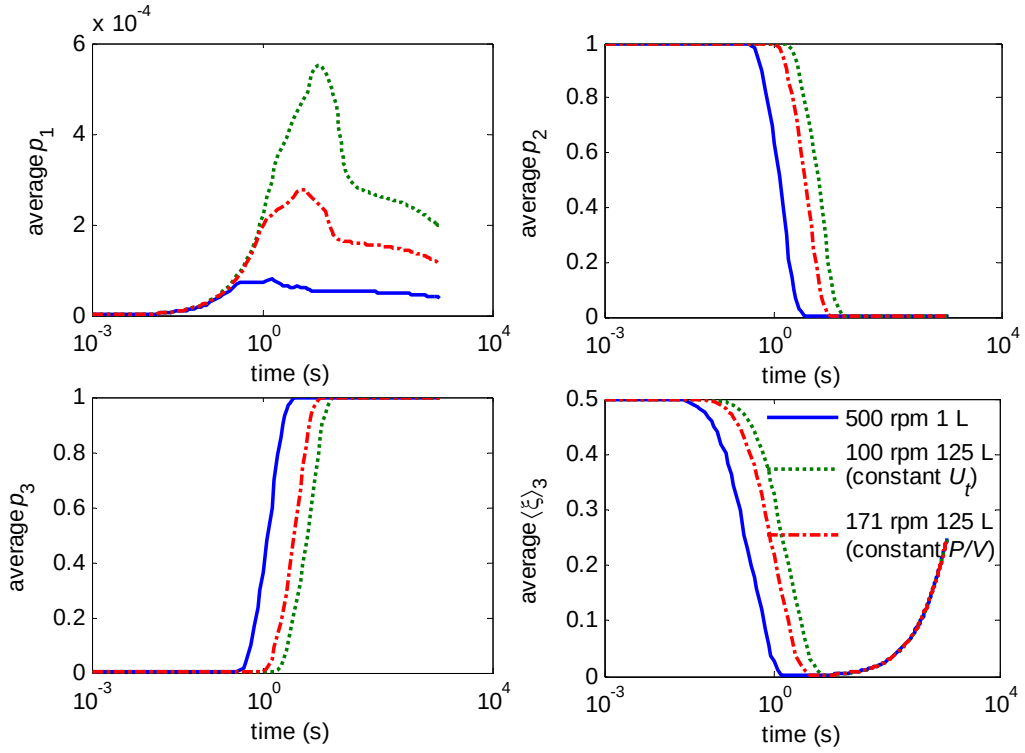

Figure 4.16 Volume-averaged p_1 (feed), p_2 (initial solution), p_3 (mixed), and $\langle \xi \rangle_3$ for scale-up based on constant tip speed (U_t) and constant power per unit volume (P/V).

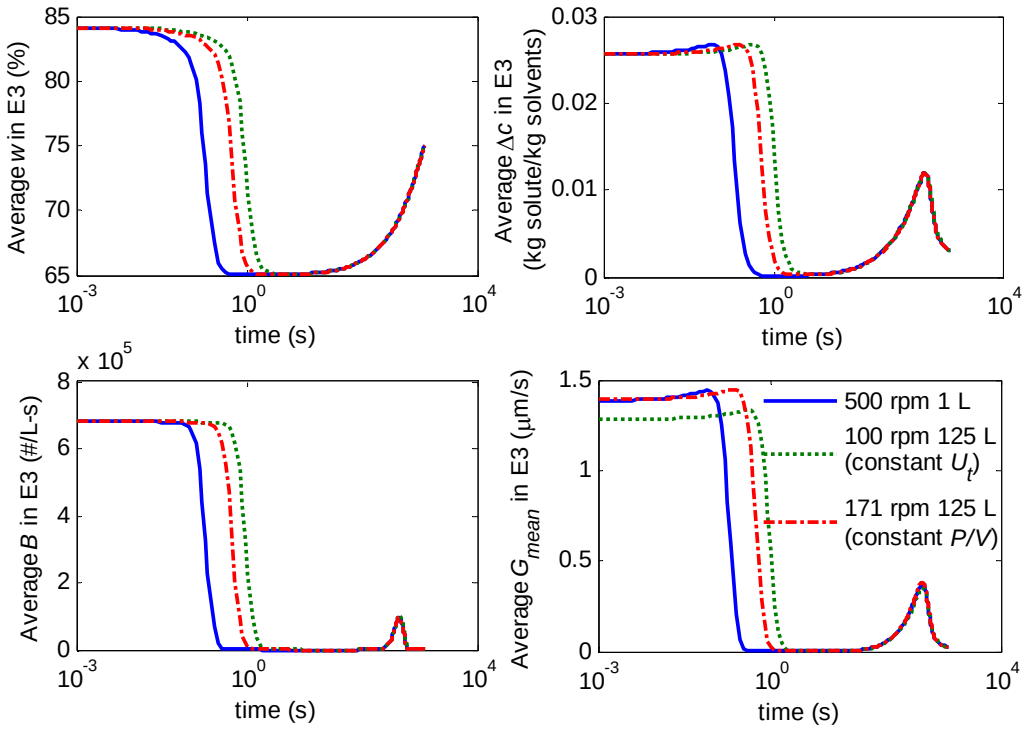

Figure 4.17 Volume-averaged antisolvent mass% (w), supersaturation (Δc), nucleation rate (B), and mean growth rate (G_{mean}) in the mixed Environment 3 (E3) for scale-up based on constant tip speed (U_t) and constant power per unit volume (P/V).

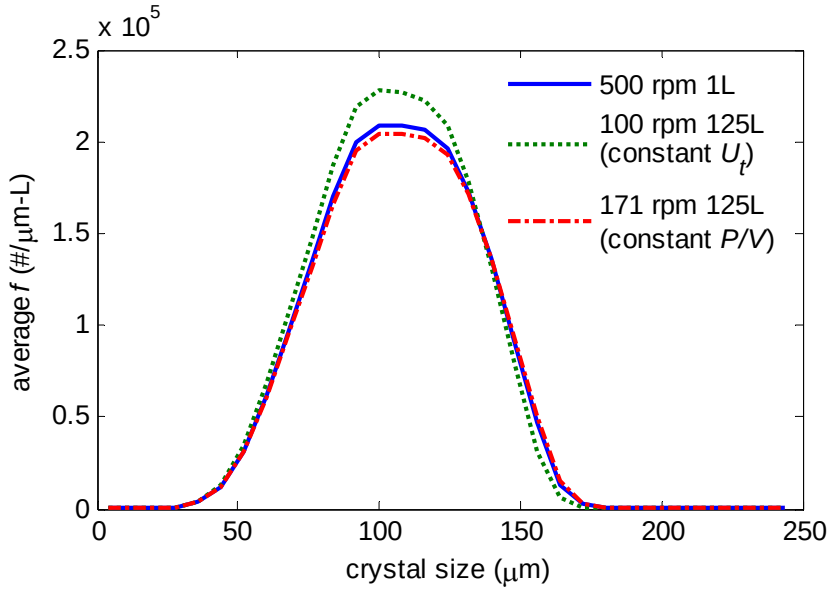

Figure 4.18 Volume-averaged CSD at 20 minutes after scale-up.

aggregation, and breakage are important. This suggest that scale-up rules based on hydrodynamic parameters which secondary nucleation, aggregation, and breakage are sensitive to can be developed to address crystallization scale-up issues.

4.8 Conclusions and Future Directions

The PBE, discretized along the internal coordinate using a high-resolution central scheme, and the multi-environment presumed-PDF model, which captures the micromixing effects, were integrated into a commercial CFD solver to simulate the effects of mixing on the full CSD in antisolvent crystallization. In a simulation study, larger and fewer crystals were produced when the agitation rate was increased during direct addition, and the reverse addition mode produced smaller crystals. Two rules for scaling up tip speed were compared, with constant power per unit volume resulting in a CSD closer to that of the bench-scale crystallizer.

The examples were chosen to show the effects of different operating conditions and scale on the crystal size distribution for a model system and can be extended to other systems which crystallization kinetics are known. While gaining insights into how different operating parameters can affect the crystal product quality, the design and scale-up of crystallizers to meet a desired product specification can be carried out in a more systematic way, which is a challenging problem in industrial crystallization (Braun et al., 2004). This can reduce the number of laboratory experiments required, especially when a pharmaceutical drug is only available in small quantities in the early drug development stage, and shorten the time required to develop the manufacturing process. In this chapter, we have shown how agitation rate and

addition mode can be adjusted to produce a specified crystal size distribution and what scale-up rules can be used to maintain the desired crystal size distribution during scale-up.

At this point, qualitative comparisons were made with experimental data reported in the published literature, which gave some confidence that the current model can predict the trends observed in experiments qualitatively. Predicting experimental data quantitatively would require the input of accurate parameters into the computational model. Hence, a step forward from this paper is to design experimental systems to accurately estimate the parameters in the nucleation and growth expressions and additional agglomeration, breakage, and secondary nucleation kinetics, as well as the turbulence and micromixing models.

This coupled algorithm can be further coupled with multiphase models to achieve a better accuracy for the particle flow field. This approach applies with minor modification to precipitation and cooling crystallization and, with increased computational requirements, can be extended to secondary nucleation, aggregation, and breakage processes. The secondary nucleation, aggregation and breakage kinetics can be included using additional terms in the population balance equation. The handling of such terms with the high-resolution methods has been reported for the well-mixed case (Gunawan et al., 2004, In preparation), which is also applicable when coupled with macromixing and micromixing models. The additional computational time can be reduced with parallel computations and the availability of faster processors (Moore's Law). The development of this integrated model would provide a better understanding of the effects of mixing on crystallization, thus offering a more

scientific basis for the robust and optimal design and scale-up of crystallizers. Also, it creates the opportunity to estimate crystallizer-independent crystallization kinetics, in contrast to the current literature approaches which are really estimating, whether acknowledged or not, kinetic parameters averaged over the crystallizer and, hence, are averaged over the hydrodynamics. The identification of crystallization kinetics that is independent of hydrodynamics is required for the systematic design of crystallizers of different configurations.

4.9 References

Barnes, H. A., Hutton, J. F. and Walters, K. (1989). An Introduction to Rheology. Amsterdam, Elsevier Science Publishers B. V.**:** 115-139.

Braun, B., Groen, H. and Tschernjaew, J. (2004). Production-scale particle design of a pharmaceutical intermediate. *Crystal Growth & Design* **4**(5): 915-920.

De Gioannis, B., Jestin, P. and Subra, P. (2004). Morphology and growth control of griseofulvin recrystallized by compressed carbon dioxide as antisolvent. *Journal of Crystal Growth* **262**(1-4): 519-526.

Duvigneau, R. and Visonneau, M. (2004). Hydrodynamic design using a derivative-free method. *Structural and Multidisciplinary Optimization* **28**(2-3): 195-205.

Fluent 6.1 UDF Manual. (2003). Lebanon, New Hampshire, Fluent Inc.

Fluent 6.1 User's Guide. (2003). Lebanon, New Hampshire, Fluent Inc.

Fujiwara, M., Nagy, Z. K., Chew, J. W. and Braatz, R. D. (2005). First-principles and direct design approaches for the control of pharmaceutical crystallization. *Journal of Process Control* **15**(5): 493-504.

Gao, X. W., Chen, P. C. and Tang, L. (2002). Deforming mesh for computational aeroelasticity using a nonlinear elastic boundary element method. *AIAA Journal* **40**(8): 1512-1517.

Granberg, R. A., Bloch, D. G. and Rasmuson, A. C. (1999). Crystallization of paracetamol in acetone-water mixtures. *Journal of Crystal Growth* **199**: 1287-1293.

Granberg, R. A. and Rasmuson, A. C. (2000). Solubility of paracetamol in binary and ternary mixtures of water plus acetone plus toluene. *Journal of Chemical and Engineering Data* **45**(3): 478-483.

Green, D. (2002). Crystallizer Mixing: Understanding and Modeling Crystallizer Mixing and Suspension Flow. *Handbook of Industrial Crystallization*. Boston, Butterworth-Heinemann**:** 181-200.

Gunawan, R., Fusman, I. and Braatz, R. D. (2004). High resolution algorithms for multidimensional population balance equations. *AIChE Journal* **50**(11): 2738-2749.

Gunawan, R., Fusman, I. and Braatz, R. D. (In preparation). ParticleSolver: High Resolution Methods for Population Balance Equations.

Kim, T. H., Lim, D. Y., Yu, B. S., Lee, J. H. and Goto, M. (2000). Effect of stirring and heating rate on the formation of TiO2 powders using supercritical fluid. *Industrial & Engineering Chemistry Research* **39**(12): 4702-4706.

Kim, Y., Haam, S., Shul, Y. G., Kim, W. S., Jung, J. K., Eun, H. C. and Koo, K. K. (2003). Pseudopolymorphic crystallization of L-ornithine-L-aspartate by drowning out. *Industrial & Engineering Chemistry Research* **42**(4): 883-889.

Kresta, S. M. and Brodkey, R. S. (2004). Turbulence in Mixing Applications. *Handbook of Industrial Mixing*. Hoboken, N. J., John Wiley & Sons, Inc.**:** 19-88.

Marshall, E. M. and Bakker, A. (2004). Computational Fluid Mixing. *Handbook of Industrial Mixing*. Hoboken, N. J., John Wiley & Sons, Inc.**:** 257-344.

Midler, M., Paul, E. L., Whittington, E. F., Futran, M., Liu, P. D., Hsu, J. and Pan, S.-H. (1994). Crystallization method to improve crystal structure and size. Merck & Co., Inc. *Patent, U. S.* 5,314,506.

Mullin, J. W. (2001). *Crystallization*. Oxford, U. K., Elsevier Butterworth-Heinemann.

Mullin, J. W., Teodossiev, N. and Sohnel, O. (1989). Potassium-sulfate precipitation from aqueous-solution by salting-out with acetone. *Chemical Engineering and Processing* **26**(2): 93-99.

Nyvlt, J., Sohnel, O., Matuchova, M. and Broul, M. (1985). *The Kinetics of Industrial Crystallization*. New York, Elsevier.

Nyvlt, J. and Zacek, S. (1986). Effect of the rate of stirring on crystal size in precipitating or salting-out systems. *Collection of Czechoslovak Chemical Communications* **51**(8): 1609-1617.

Plasari, E., Grisoni, P. and Villermaux, J. (1997). Influence of process parameters on the precipitation of organic nanoparticles by drowning-out. *Chemical Engineering Research & Design* **75**(A2): 237-244.

Randolph, A. D. and Larson, M. A. (1988). *Theory of Particulate Processes*. San Diego, California, Academic Press, Inc.

Shin, D. M. and Kim, W. S. (2002). Drowning-out crystallization of L-ornithine-aspartate in turbulent agitated reactor. *Journal of Chemical Engineering of Japan* **35**(11): 1083-1090.

Torbacke, M. and Rasmuson, A. C. (2004). Mesomixing in semi-batch reaction crystallization and influence of reactor size. *AIChE Journal* **50**(12): 3107-3119.

Yu, Z. Q., Chow, P. S. and Tan, R. B. H. (2006). Application of attenuated total reflectance-Fourier transform infrared (ATR-FTIR) technique in the monitoring and control of anti-solvent crystallization. *Industrial & Engineering Chemistry Research* **45**(1): 438-444.

Yu, Z. Q., Tan, R. B. H. and Chow, P. S. (2005). Effects of operating conditions on agglomeration and habit of paracetamol crystals in anti-solvent crystallization. *Journal of Crystal Growth* **279**(3-4): 477-488.

Chapter 5
Simulation of Competitive Reactions in Confined Impinging Jet Reactors

5.1 Introduction

Other than the conventional stirred tanks, the use of impinging jet crystallizers for fast crystallization processes is gaining popularity due to its consistency to produce crystals with narrow and uni-modal distributions (see review in Chapter 2). A better understanding of the impinging jet crystallizers through modeling and simulation can reduce the time for developing the impinging jet process for different crystallization systems.

This chapter covers the modeling and simulation of the macromixing and micromixing in confined impinging jets. A coupled CFD-Micromixing-Population Balance algorithm (Woo et al., 2006) is used to model the crystallization process in an impinging jet crystallizer. The geometry of the confined impinging jet is adapted from Johnson and Prud'homme (2003). In the first part, the CFD-Micromixing algorithm is combined with the kinetics of competitive reactions to model the experimental work reported in Johnson and Prud'homme (2003), and the simulation results are compared with the reported experimental data.

A sensitivity analysis of the parameters (Varma et al., 1999) in the turbulence and micromixing models is performed to determine changes in the parameters to increase the consistency between simulation results and experimental observations. While this

approach of sensitivity analysis and parameter estimation is widely used in other modeling applications (Braatz et al., 2006; Kontoravdi et al., 2005; Verenich et al., 2003), in turbulent fluid dynamics modeling, the default parameter values in the closure models for the Reynolds-averaged Navier-Stokes (RANS) equations are usually used. *Such default empirical parameters are not necessarily appropriate when applying the RANS equations to impinging jet crystallization*, however, because the default parameter values in CFD software were a compromise choice to give good agreement with experimental data for a limited set of flows (e.g., pipe flows, stirred tanks) and flow conditions (Pope, 2000). Although many CFD studies have compared different closure models without checking whether the parameters were optimized for the specific flow modeled (Armenante et al., 1997; Guardo et al., 2005; Hjertager et al., 2002; Jaworski and Zakrzewska, 2002; Mulvany et al., 2004; Ranade et al., 2001), it has been found that the prediction of turbulent flows using the RANS approach can be very sensitive to some of the parameters in the closure models (Bischof et al., 2004; Colin et al., 2005; Green et al., 1996; Sanders and Lamers, 1992; Turgeon et al., 2004), and for some flows, it may be necessary to define the parameters based on local flow conditions to get agreement between simulations and experiments (Launder and Sharma, 1974; Launder and Spalding, 1974; Liu and Fox, 2006; Vaidyanathan et al., 2003).

5.2 CFD-Micromixing Modeling of Competitive Reactions

The modeling of fast reactions in turbulent flows has been an ongoing interest for many academic researchers and many modeling approaches have been reported (Baldyga and Bourne, 1999; Fox, 2003; Woo et al., 2006). In this work, the macromixing was modeled by the Reynolds-averaged Navier-Stokes (RANS) and

scalar transport equations and the *k*-ε turbulence model (Fluent 6.2 User's Guide, 2005; Pope, 2000), computed using a commercial CFD software (Fluent 6.2.16, Fluent Inc.). The equations, are summarized below; the reader is referred to the software documentation for the nomenclature and more details on the equations (Fluent 6.2 UDF Manual, 2005; Fluent 6.2 User's Guide, 2005).

$$\text{Continuity equation:} \quad \frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \vec{v}) = 0 \tag{5.1}$$

Momentum conservation equation:

$$\frac{\partial}{\partial t}(\rho \vec{v}) + \nabla \bullet (\rho \vec{v} \vec{v}) = -\nabla p + \nabla \bullet (\bar{\bar{\tau}}) + \rho \vec{g} \tag{5.2}$$

k-ε equations:

$$\begin{aligned} &\frac{\partial}{\partial t}(\rho k) + \nabla \bullet (\rho k \vec{v}) = \nabla \bullet \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \nabla \bullet k \right] + G_k - \rho \varepsilon, \\ &\frac{\partial}{\partial t}(\rho \varepsilon) + \nabla \bullet (\rho \varepsilon \vec{v}) = \nabla \bullet \left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon} \right) \nabla \bullet \varepsilon \right] + C_{1\varepsilon} \frac{\varepsilon}{k} G_k - C_{2\varepsilon} \rho \frac{\varepsilon^2}{k} \\ &\text{where} \quad \mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \end{aligned} \tag{5.3}$$

Scalar transport equation:

$$\frac{\partial}{\partial}(\rho \phi_k) + \nabla \bullet (\rho \vec{v} \phi_k - \rho (D_m + D_t) \nabla \bullet \phi_k) = S_{\phi_k} \tag{5.4}$$

$$\text{where} \quad D_t = \frac{\mu_t}{\rho \mathrm{Sc}_t}$$

The micromixing was modeled by a three-environment presumed-PDF model, which approximates the fluctuations of the species concentrations on the subgrid scale (Fox, 2003). One inlet stream is in environment 1, the other inlet stream is in environment 2, and environment 3 is a mixture of the two fluids. The equations (also presented in Chapter 3), which are coupled with the CFD algorithm (Woo et al., 2006), are

$$\frac{\partial \mathbf{p}}{\partial t} + \nabla \bullet (\vec{v} \mathbf{p} - D_t \nabla \bullet \mathbf{p}) = \mathbf{G} + \mathbf{G_s} \tag{5.5}$$

$$\frac{\partial \langle \mathbf{s} \rangle_n}{\partial t} + \nabla \bullet (\vec{v} \langle \mathbf{s} \rangle_n - D_t \nabla \bullet \langle \mathbf{s} \rangle_n) = \mathbf{M}^n + \mathbf{M}_\mathbf{s}^n + p_n \mathbf{S}^n \tag{5.6}$$

$$\langle \mathbf{s} \rangle_n \equiv p_n \langle \phi \rangle_n \tag{5.7}$$

where **p** is the volume fraction vector and, $\langle \phi \rangle_n$ is the mean composition vector in environment *n*, and $\langle \mathbf{s} \rangle_n$ is the weighted composition vector. The transport equation for the mixture fraction, $\langle \xi \rangle_3$, is also represented by equation (5.6). The micromixing terms, $\mathbf{M}^n$ and $\mathbf{M}_\mathbf{s}^n$, in the right-hand-side of the equations are given in Table 5.1 and $\mathbf{S}^n$ is the source term (e.g., reaction, crystallization) for the scalars, which is a function of $\langle \phi \rangle_n$.

Table 5.1 Micromixing terms for equations (5.5) and (5.6) (Fox, 2003).

Model variables	$\mathbf{G}$, $\mathbf{M}^n$	$\mathbf{G}_\mathbf{s}$, $\mathbf{M}_\mathbf{s}^n$
p_1	$-\gamma p_1(1-p_1)$	$\gamma_s p_3$
p_2	$-\gamma p_2(1-p_2)$	$\gamma_s p_3$
p_3	$\gamma\left[p_1(1-p_1)+p_2(1-p_2)\right]$	$-2\gamma_s p_3$
$\langle \mathbf{s} \rangle_3$	$\gamma\left[p_1(1-p_1)\langle\phi\rangle_1 + p_2(1-p_2)\langle\phi\rangle_2\right]$	$-\gamma_s p_3\left(\langle\phi\rangle_1 + \langle\phi\rangle_2\right)$
	$\gamma = \dfrac{\varepsilon_\xi}{p_1(1-p_1)(1-\langle\xi\rangle_3)^2 + p_2(1-p_2)\langle\xi\rangle_3^2}$	
	$\gamma_s = \dfrac{2D_t}{(1-\langle\xi\rangle_3)^2 + \langle\xi\rangle_3^2} \dfrac{\partial\langle\xi\rangle_3}{\partial x_i} \dfrac{\partial\langle\xi\rangle_3}{\partial x_i}$	
	$\langle \xi'^2 \rangle = p_1(1-p_1) - 2p_1p_3\langle\xi\rangle_3 + p_3(1-p_3)\langle\xi\rangle_3^2$	

The scalar dissipation rate (ε_ξ), described in detail by Pope (1985), is expressed as:

$$\varepsilon_\xi = \frac{1}{2} C_\phi \langle \xi'^2 \rangle \frac{\varepsilon}{k} \tag{5.8}$$

The default values of the parameters are given in Table 5.2.

Table 5.2 Default values of the parameters in the CFD-micromixing model.

Parameter #	1	2	3	4	5	6	7
	$C_{1\varepsilon}$	$C_{2\varepsilon}$	C_{μ}	σ_k	σ_{ε}	S_{ct}	C_{ϕ}
Default values	1.44	1.92	0.09	1.0	1.3	0.7	2.0

Competitive-parallel and competitive-consecutive reactions, in which one reaction is faster and the other reaction is slower than the micromixing rate, are used to characterize the extent of mixing through the product distribution. Experimental data obtained from these mixing-reaction experiments can be used to check the predictability and accuracy of macromixing and micromixing models.

In the work of Johnson and Prud'homme (Johnson and Prud'homme, 2003), the following set of competitive-parallel reactions developed by Baldyga et al. (Baldyga et al., 1998) was used to characterize the mixing in a confined impinging jet:

$$\text{Reaction 1: } \mathrm{NaOH} + \mathrm{HCl} \xrightarrow{k_1} \mathrm{H_2O} + \mathrm{NaCl}$$

$$\text{Reaction 2: } \mathrm{CH_3C(OCH_3)_2CH_3} + \mathrm{H_2O} + \mathrm{H^+} \xrightarrow{k_2} \mathrm{CH_3COCH_3} + 2\mathrm{CH_3OH} + \mathrm{H^+}$$

The details of the reaction kinetics are described in Baldyga et al. (Baldyga et al., 1998), and summarized as follows:

$$\begin{gathered}\text{Rate of Reaction } 1 = k_1[\mathrm{HCl}][\mathrm{NaOH}] \\ k_1\,(\mathrm{m^3/mol\text{-}s}) = 1.4\times10^8\end{gathered} \tag{5.9}$$

$$\begin{gathered}\text{Rate of Reaction } 2 = k_2[\mathrm{CH_3C(OCH_3)_2CH_3}][\mathrm{H^+}] \\ k_2\,(\mathrm{m^3/mol\text{-}s}) = 7.32\times10^7 e^{-5556/T} 10^{(0.05434+7.07\times10^{-5}[\mathrm{NaCl}])}\end{gathered} \tag{5.10}$$

Equations (5.9) and (5.10) were inserted into the term $\mathbf{S}^n$, and equations (5.5) and (5.6) were solved as user-defined scalar transport equations by the CFD solver. This CFD-micromixing model was applied to model the confined impinging jet, labeled as mixing head 500A-Y2X (Johnson and Prud'homme, 2003). The use of CFD and

micromixing models to predict the conversion of the slow reaction in a confined impinging jet also has been recently reported (Liu and Fox, 2006).

Due to the symmetry of the geometry, only half the confined impinging jet needs to be modeled and the numerical grid, generated using Gambit 2.2.30 (Fluent Inc.), is shown in Figure 5.1. The *k*-ε model was combined with enhanced wall treatment (Fluent 6.2 User's Guide, 2005) due to the small volume of the mixing chamber and the confinement of the impinging region by the chamber walls. The effect of spatial temperature (*T*) variation on the reaction kinetics was neglected as it was observed in the experimental studies that the temperature change was sufficiently small to result in any significant changes in the reaction conversion. To improve numerical stability and convergence, constant density and viscosity were used, which was calculated based on reacting both inlet streams with no conversion of the slow reaction. This assumption would not cause significant changes in the flow field, since there is less than 10% variation in the density and viscosity for different conversions of the two reactions.

For inlet velocities with laminar flow in the inlet pipes (jet Reynolds Number < 2000), the inlet pipe region was specified as a laminar zone (Fluent 6.2 User's Guide, 2005). Also, as shown in the numerical grid (Figure 5.1), the entire length of the outlet pipe was not simulated because the turbulent micromixing model would not be valid for the laminar flow cases. Moreover, nearly all of the mixing occurs in the chamber due to high turbulence at the impingement region. Thus, the species concentrations obtained at the end of the chamber are representative of the concentrations exiting the outlet pipe.

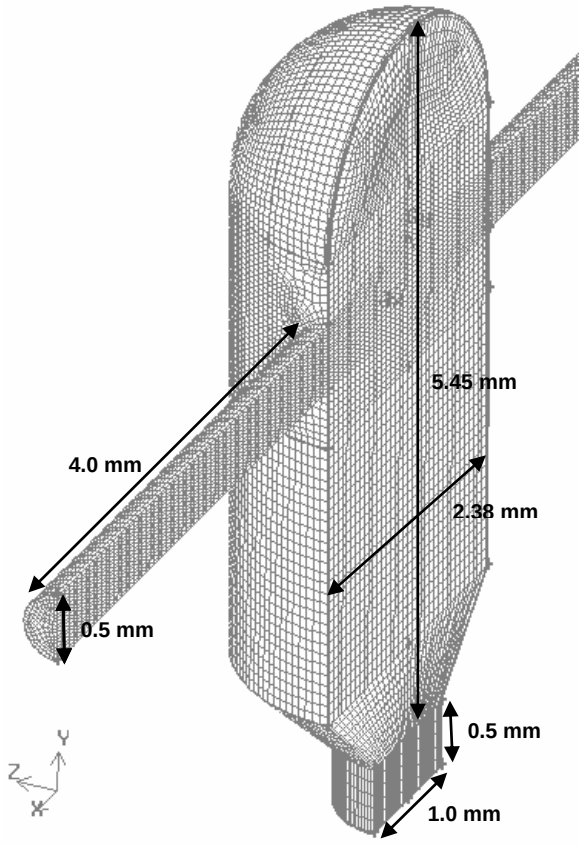

Figure 5.1 Numerical grid of confined impinging jet with geometrical details.

For a reaction time of the slower reaction (τ_{rxn}) of 4.8 ms, simulation results for a range of inlet velocities (v) and the conversions (X) of the slow reaction were compared with the experimental results (Johnson and Prud'homme, 2003). The compositions of the species at the inlets are given in Table 5.3. For this case, the average density and viscosity were 974.71 kg/m^3 and 2.03519×10^{-3} Pa-s respectively. First the steady-state flow field was calculated, followed by the time-dependent computation of the transport equations (5.5) and (5.6). A steady state solution was obtained within six residence times. All computations were carried in parallel, across four to eight CPUs, on Xeon 3.06 GHz and 3.2 GHz Linux clusters.

Table 5.3 Boundary conditions and concentrations of the reagents (corresponding to τ_{rxn} = 4.8 ms) at the inlets.

	Inlet 1	Inlet 2
	$p_1 = 1$ and $\langle\xi\rangle_1 = 1$	$p_2 = 1$ and $\langle\xi\rangle_2 = 0$
HCl	600 mM	-
NaOH	-	630 mM
$CH_3C(OCH_3)_2CH_3$	-	600 mM
NaCl	90 mM	90 mM

5.3 Parameter Sensitivity Analysis of Mixing Models

Using the default values of the parameters in the macromixing and micromixing models (see Table 5.2), it can be seen from Figure 5.2 (Simulation A) that the prediction of the actual conversion is approximately a factor of 4 less than the experimental data (~ –75% deviation). However, the overall trend was well predicted, where the conversion increases by a factor of 2 with a factor of 4 decrease in inlet velocities. The results gives some confidence that the k-ε model and the three-environment presumed-PDF model are able to capture the flow behavior in the

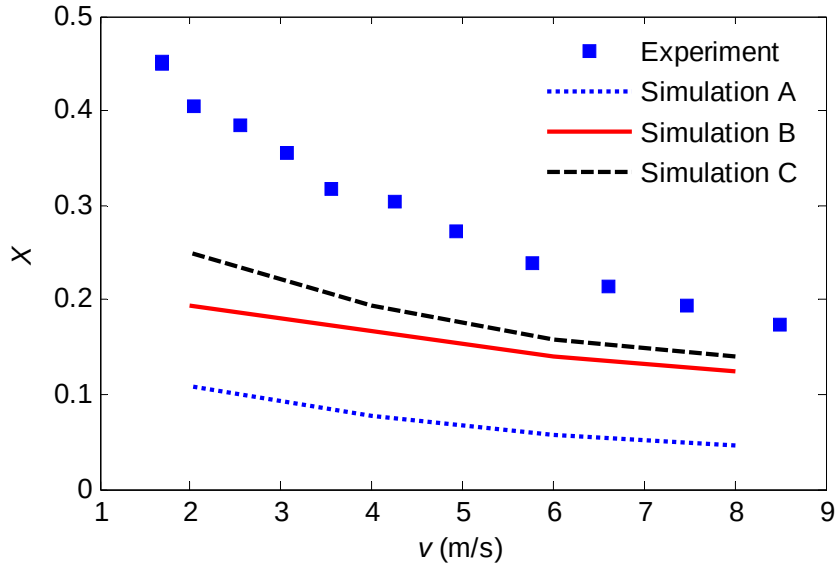

Figure 5.2 Conversion (X) of the slow reaction obtained from experiment and simulations. Simulation A: default parameter values. Simulation B: 30% perturbation of all parameters to increase conversion. Simulation C: 20% perturbation for $C_{1\varepsilon}$, $C_{2\varepsilon}$, C_{μ}, σ_k, and σ_{ε}, and 100% perturbation for S_{ct} and C_{ϕ} to increase conversion.

confined impinging jet, while indicating a need to adjust the parameter values in the models to provide a more accurate prediction.

In parameter sensitivity analysis, sensitivities (S_j) are computed to quantitatively describe the change in process behavior in response to perturbations in the parameters of the process models (Gunawan et al., 2003; Kamrunnahar et al., 2004). These sensitivities indicate how to adjust the parameters so that the simulation results more closely agree with experiments. The sensitivity of the conversion X of the slow reaction with respect to the j^{th} parameter θ_j is

$$S_j = \frac{\partial X}{\partial \theta_j} \tag{5.11}$$

The normalized sensitivity, $\hat{S}_j$, which quantifies the relative change in the conversion with respect to a relative change in a parameter, is

$$\hat{S}_j = \frac{\theta_j}{X}\frac{\partial X}{\partial \theta_j} \tag{5.12}$$

This normalized sensitivity is useful for comparisons for parameters of different units or widely varying magnitudes. The two-point central finite difference expression was used to estimate the sensitivities:

$$S_j \approx \frac{\Delta X}{\Delta \theta_j} = \frac{X(\theta_j + \Delta\theta_j) - X(\theta_j - \Delta\theta_j)}{2\Delta\theta_j} \tag{5.13}$$

The CFD-micromixing model was simulated twice for each parameter by perturbing each parameter ±10%. The sensitivities for the inlet velocity of 8 m/s are shown in Figure 5.3.

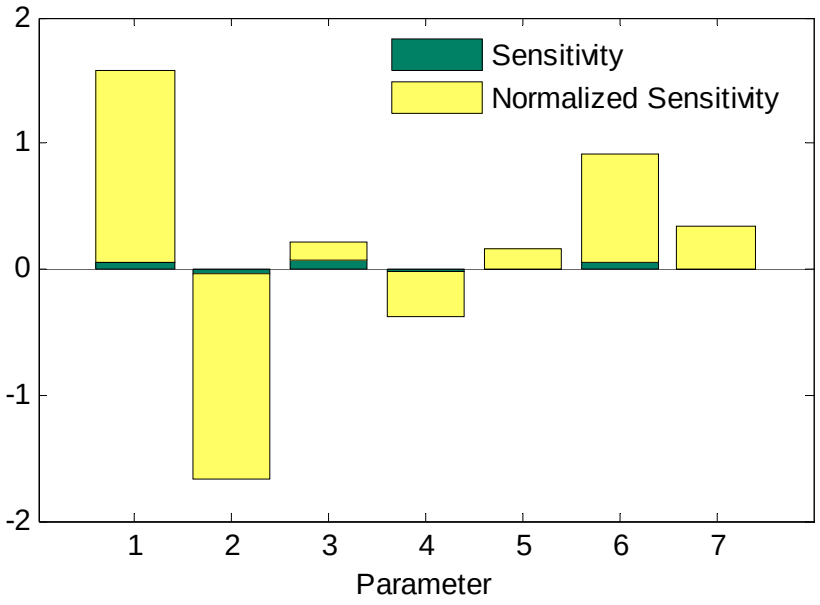

Figure 5.3 Sensitivities and normalized sensitivities of parameters in CFD-micromixing model at v = 8 m/s according to Table 3. Sensitivities were computed with 10% perturbation.

As observed in Figure 5.3, the simulated conversion is most sensitive to $C_{1\varepsilon}$ and $C_{2\varepsilon}$. To increase the conversion X towards the experimental values (Figure 5.2), the values of all the parameters, except for $C_{2\varepsilon}$ and σ_k, must be perturbed in the positive direction. With all parameters perturbed in the direction to increase the conversion of the slow reaction, the deviation of the simulated output from the experimental results was significantly reduced (see Figure 5.2).The set of parameters corresponding to Simulation C (Figure 5.2) predicted the experimental data within a factor of 2 (maximum deviation < −40%), which was a considerable improvement from the default parameters, and was used to model the impinging jet crystallizer in the following chapter. This set should result in more reliable predictions for the spatial concentration fields and the crystal size distribution.

A more complete validation of the CFD-micromixing model and estimated model parameters would compare simulations and experimental measurements of the full three-dimensional concentration and velocity field. Unfortunately, such experimental data sets are not available in the literature. To obtain such experimental data, more sophisticated measurement techniques such as laser-induced fluorescence, laser doppler anemometry, or particle image velocimetry would need to be incorporated within an experimental impinging jet crystallizer.

5.4 Conclusions

In this paper, the capability of the CFD-micromixing model in modeling the flow and mixing in confined impinging jets was first validated with competitive reactions data. Parameter sensitivity analysis was used to elucidate how the parameters can be adjusted to give a better agreement with experiments. An improved set of parameters

were obtained for the subsequent modeling of antisolvent crystallization in impinging jets.

This step of the study could be improved by measuring the three-dimensional flow field and concentrations (using particle imaging and fluorescence and Raman spectroscopy techniques), and estimate the CFD parameters to fit the full solution from the simulation model. Since such data are not available, these parameters were fixed in CFD-micromixing model used in the subsequent studies that included crystallization.

5.5 References

Armenante, P. M., Luo, C. G., Chou, C. C., Fort, I. and Medek, J. (1997). Velocity profiles in a closed, unbaffled vessel: comparison between experimental LDV data and numerical CFD predictions. *Chemical Engineering Science* **52**(20): 3483-3492.

Baldyga, J. and Bourne, J. R. (1999). *Turbulent Mixing and Chemical Reactions*. Chichester, U. K., John Wiley & Sons Ltd.

Baldyga, J., Bourne, J. R. and Walker, B. (1998). Non-isothermal micromixing in turbulent liquids: Theory and experiment. *Canadian Journal of Chemical Engineering* **76**(3): 641-649.

Bischof, C. H., Bucker, H. M. and Rasch, A. (2004). Sensitivity analysis of turbulence models using automatic differentiation. *SIAM Journal on Scientific Computing* **26**(2): 510-522.

Braatz, R. D., Alkire, R. C., Seebauer, E., Rusli, E., Gunawan, R., Drews, T. O., Li, X. and He, Y. (2006). Perspectives on the design and control of multiscale systems. *Journal of Process Control* **16**(3): 193-204.

Colin, E., Etienne, S., Pelletier, D. and Borggaard, J. (2005). Application of a sensitivity equation method to turbulent flows with heat transfer. *International Journal of Thermal Sciences* **44**(11): 1024-1038.

Fluent 6.2 UDF Manual. (2005). Lebanon, NH, Fluent Inc.

Fluent 6.2 User's Guide. (2005). Lebanon, NH, Fluent Inc.

Fox, R. O. (2003). *Computational Models for Turbulent Reacting Flows*. Cambridge, U. K., Cambridge University Press.

Green, L. L., Newman, P. A. and Haigler, K. J. (1996). Sensitivity derivatives for advanced CFD algorithm and viscous modeling parameters via automatic differentiation. *Journal of Computational Physics* **125**(2): 313-324.

Guardo, A., Coussirat, M., Larrayoz, M. A., Recasens, F. and Egusquiza, E. (2005). Influence of the turbulence model in CFD modeling of wall-to-fluid heat transfer in packed beds. *Chemical Engineering Science* **60**(6): 1733-1742.

Gunawan, R., Jung, M. Y. L., Braatz, R. D. and Seebauer, E. G. (2003). Parameter sensitivity analysis applied to modeling transient enhanced diffusion and activation of boron in silicon. *Journal of the Electrochemical Society* **150**(12): G758-G765.

Hjertager, L. K., Hjertager, B. H. and Solberg, T. (2002). CFD modelling of fast chemical reactions in turbulent liquid flows. *Computers & Chemical Engineering* **26**(4-5): 507-515.

Jaworski, Z. and Zakrzewska, B. (2002). Modelling of the turbulent wall jet generated by a pitched blade turbine impeller - The effect of turbulence model. *Chemical Engineering Research & Design* **80**(A8): 846-854.

Johnson, B. K. and Prud'homme, R. K. (2003). Chemical processing and micromixing in confined impinging jets. *AIChE Journal* **49**(9): 2264-2282.

Kamrunnahar, M., Braatz, R. D. and Alkire, R. C. (2004). Parameter sensitivity analysis of pit initiation at single sulfide inclusions in stainless steel. *Journal of the Electrochemical Society* **151**(2): B90-B97.

Kontoravdi, C., Asprey, S. P., Pistikopoulos, E. N. and Mantalaris, A. (2005). Application of global sensitivity analysis to determine goals for design of experiments: An example study on antibody-producing cell cultures. *Biotechnology Progress* **21**(4): 1128-1135.

Launder, B. E. and Sharma, B. I. (1974). Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters in Heat and Mass Transfer* **1**(2): 131.

Launder, B. E. and Spalding, D. B. (1974). The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering* **3**: 269-289.

Liu, Y. and Fox, R. O. (2006). CFD predictions for chemical processing in a confined impinging-jets reactor. *AIChE Journal* **52**(2): 731-744.

Mulvany, N., Tu, J. Y., Chen, L. and Anderson, B. (2004). Assessment of two-equation turbulence modelling for high Reynolds number hydrofoil flows. *International Journal for Numerical Methods in Fluids* **45**(3): 275-299.

Pope, S. B. (1985). PDF methods for turbulent reactive flows. *Progress in Energy and Combustion Science* **11**(2): 119-192.

Pope, S. B. (2000). *Turbulent Flows*. Cambridge, U. K., Cambridge University Press.

Ranade, V. V., Perrard, M., Le Sauze, N. and Bertrand, J. (2001). Trailing vortices of Rushton turbine: PIV measurements and CFD simulations with snapshot approach. *Chemical Engineering Research & Design* **79**(A1): 3-12.

Sanders, J. P. H. and Lamers, A. P. G. G. (1992). Scalar transport in a turbulent jet. *International Communications in Heat and Mass Transfer* **19**(6): 851-858.

Turgeon, E., Pelletier, D. and Borggaard, J. (2004). A general continuous sensitivity equation formulation for the kappa-epsilon model of turbulence. *International Journal of Computational Fluid Dynamics* **18**(1): 29-46.

Vaidyanathan, R., Senocak, I., Wu, J. Y. and Shyy, W. (2003). Sensitivity evaluation of a transport-based turbulent cavitation model. *Journal of Fluids Engineering-Transactions of the ASME* **125**(3): 447-458.

Varma, A., Morbidelli, M. and Wu, H. (1999). *Parametric Sensitivity in Chemical Systems*. Cambridge, Cambridge University Press.

Verenich, S., Laari, A. and Kallas, J. (2003). Parameter estimation and sensitivity analysis of lumped kinetic models for wet oxidation of concentrated wastewaters. *Industrial & Engineering Chemistry Research* **42**(21): 5091-5098.

Woo, X. Y., Tan, R. B. H., Chow, P. S. and Braatz, R. D. (2006). Simulation of mixing effects in antisolvent crystallization using a coupled CFD-PDF-PBE approach. *Crystal Growth & Design* **6**(6): 1291-1303.

Chapter 6

Modeling of Impinging Jet Crystallization

6.1 Introduction

In this chapter, a coupled CFD-Micromixing-Population Balance algorithm (Woo et al., 2006) is used to model the crystallization process in an impinging jet crystallizer. The geometry of the confined impinging jet is adapted from Johnson and Prud'homme (2003). The parameters for the CFD-Micromixing model are based on the improved values in Chapter 5.

The first part of the paper covers the simulation of the full crystal size distribution for the antisolvent crystallization of lovastatin in the confined impinging jet crystallizer, with the effects of macromixing and micromixing included. The effects of jet velocity are numerically investigated and compared with experimental data in the published literature. In the second part, the antisolvent crystallization of two polymorphic forms of L-histidine in the confined impinging jet crystallizer is presented. To the authors' knowledge, this is the most thorough simulation study on impinging jet crystallizers reported to date.

6.2 Coupling Population Balance with CFD-Micromixing Model

To compute the crystal size distribution in the impinging jet crystallizer, the transport equations in the CFD-micromixing model are coupled to a spatially-varying population balance equation. Since this is discussed in detail by Woo et al. (2006) and

in Chapter 3, the approach is only summarized here. The spatially-varying population balance equation, discretized along the crystal growth axis using a high-resolution finite-volume method (Kurganov and Tadmor, 2000; Leveque, 2002), is solved as a set of scalar transport equations in the CFD solver (see Appendix B). With f_j denoting the cell-averaged population density between size $r_{j-1/2}$ and $r_{j+1/2}$, the cell-averaged crystal mass, $f_{w,j}$, is written as

$$f_{w,j} = \rho_c k_v \int_{r_{j-1/2}}^{r_{j+1/2}} r^3 f_j dr = \frac{\rho_c k_v f_j}{4}\left((r_{j+1/2})^4 - (r_{j-1/2})^4\right) \tag{6.1}$$

where ρ_c is the crystal density and k_v is the crystal volume shape factor. For the well-micromixed case, the scalar transport equation for $f_{w,j}$ with size-dependent growth is

$$\frac{d}{dt} f_{w,j} + \nabla\bullet(\vec{v} f_{w,j} - D_t \nabla\bullet f_{w,j}) = \begin{cases} \dfrac{\rho_c k_v}{4\Delta r}\left[(r_{j+1/2})^4 - (r_{j-1/2})^4\right]\times \\ \left\{-G_{j+1/2}\left[f_j + \dfrac{\Delta r}{2}(f_r)_j\right] + G_{j-1/2}\left[f_{j-1} + \dfrac{\Delta r}{2}(f_r)_{j-1}\right] + B(j=0)\right\} & \text{for } S>1 \\ \dfrac{\rho_c k_v}{4\Delta r}\left[(r_{j+1/2})^4 - (r_{j-1/2})^4\right]\times \\ \left\{-G_{j+1/2}\left[f_{j+1} - \dfrac{\Delta r}{2}(f_r)_{j+1}\right] + G_{j-1/2}\left[f_j - \dfrac{\Delta r}{2}(f_r)_j\right]\right\} & \text{for } S<1 \end{cases} \tag{6.2}$$

where $(f_r)_j$ is the derivative approximated by the minmod limiter (Kurganov and Tadmor, 2000), $S = c/c^*$ is the relative supersaturation, c and c^* are the solution and saturated concentrations respectively, G is the growth or dissolution rate, and B is the nucleation rate.

To include the effects of micromixing, equation (6.2) is written in the form of equation (5.6), with the right-hand-side of equation (6.2) inserted into the chemical source term in equation (5.6). The inlet stream with saturated solution is in environment 1, the inlet stream with antisolvent is in environment 2, and environment

3 is a mixture of supersaturated solution and crystals. The assumption of particles following the fluid streamlines is valid in impinging jets as the particles are very small in these systems.

6.3 Crystallization Kinetics of Lovastatin

The crystallization of lovastatin from a methanol-water mixture, where water is the antisolvent, was used as a model system due to its fast kinetics. The solubility data were provided by Merck & Co., Inc. (Rahway, NJ):

$$c^* \text{ at } 23^\circ\text{C (g/kg of solvents)} = \begin{cases} -541.00V^3 + 534.30V^2 - 207.45V + 33.089 & \text{for } V \le 0.4 \\ -1.6196V + 1.6200 & \text{for } V > 0.4 \end{cases} \tag{6.3}$$

where V is the volume fraction of antisolvent in the solvents. The primary nucleation and growth kinetics are given as (Mahajan and Kirwan, 1994):

$$B = B_{\text{homogeneous}} + B_{\text{heterogeneous}}$$

$$B_{\text{homogeneous}} \text{ at } 23^\circ\text{C (\#/s-m}^3) = 6.97\times10^{14} \exp\left(\frac{-15.8}{[\ln S]^2}\right) \tag{6.4}$$

$$B_{\text{heterogeneous}} \text{ at } 23^\circ\text{C (\#/s-m}^3) = 2.18\times10^{8} \exp\left(\frac{-0.994}{[\ln S]^2}\right)$$

$$G \text{ at } 23^\circ\text{C (m/s)} = 8.33\times10^{-30}(2.46\times10^{3} \ln S)^{6.7} \tag{6.5}$$

Crystal growth for this system is surface integration limited, and the growth rate is size-independent (Mahajan and Kirwan, 1994). Secondary nucleation can be neglected due to the small solids density in impinging jets.

The simulation was carried out with methanol solution saturated with lovastatin in one inlet and water in the other inlet, with an average density of 891.6 kg/m^3 and viscosity of 8.0336×10^{-4} Pa-s. The density of lovastatin is 1273 kg/m^3 and the volume shape factor is 0.000625. The population balance equation was discretized into 30 bins, with

Δr = 3 μm, for the longest growth axis. The use of effective viscosity, as described in Woo et al. (2006) and Chapter 3, was omitted because the solid fraction was too small to significantly affect the effective viscosity.

6.4 Mixing in Confined Impinging Jets

Figures 6.1 and 6.2 show the volume fraction of the mixed environment (p_3) and its mixture fraction ($\langle \xi \rangle_3$), the fraction of fluid in environment 3 that came from environment 1, for low and high inlet velocities. At low inlet velocity, partial segregation of the inlet fluids is still present at the outlet of the mixing chamber ($p_3 < 1$) due to the low turbulence intensity. On the other hand, the mixed environment contains nearly equal quantities of fluid from each inlet, $\langle \xi \rangle_3 \to 0.5$, across the radius of the outlet pipe. At high inlet velocity, the segregation extends from the inlet pipes to the impingement plane at the center, after which the inlet fluids become de-segregated due to high turbulence intensity. Although there is only the mixed environment ($p_3 \approx 1$) at the outlet for high inlet velocity, the concentrations of antisolvent and solvent within the mixed environment vary across the radius of the output pipe, that is, $\langle \xi \rangle_3 \neq 0.5$ except at the center of the outlet pipe. Different spatial inhomogeneities are observed at high and low inlet velocities and this can have different effects on the product distribution for reactions or crystallization, depending on the kinetics of the system.

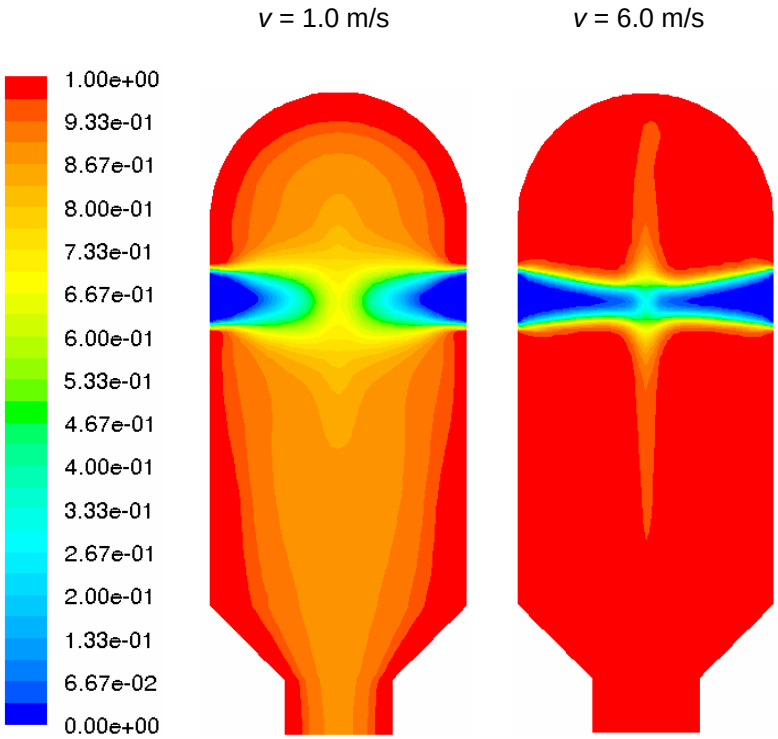

Figure 6.1 Volume fraction of the mixed environment (p_3) along the symmetry plane of the mixing chamber of the confined impinging jet. (Left inlet: Inlet 1, Right inlet: Inlet 2)

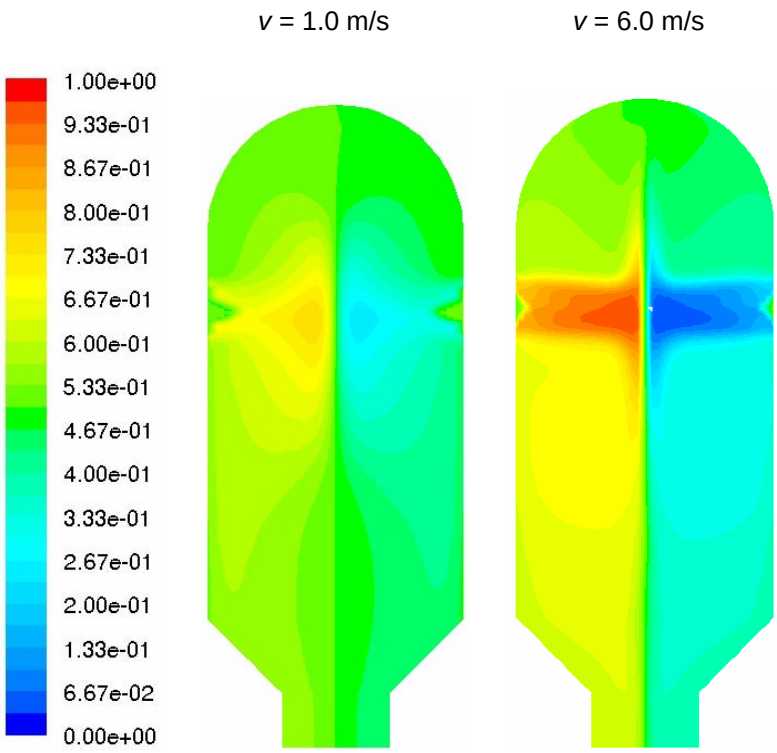

Figure 6.2 Mixture fraction of the mixed environment ($\langle \xi \rangle_3$) along the symmetry plane of the mixing chamber of the confined impinging jet.

6.5 Crystallization Dynamics

The effects of mixing on the crystallization dynamics in the confined impinging jet are further explored in this section. The steady-state spatial distributions are plotted for the mixed environment, where crystallization occurs. The spatial plot for the antisolvent volume fraction (see Figure 6.3) is analogous to the mixture fraction plot, since the mixture fraction describes the proportion of fluids from each inlet being mixed. It is important to keep in mind that even though there is higher homogeneity of antisolvent within the mixed environment at low inlet velocity due to longer time for mixing, these conditions result in much larger segregation of the inlet fluids throughout the crystallizer than at high inlet velocity (Figure 6.1).

High supersaturation is achieved in the regions of high antisolvent composition (Figure 6.3) where the solubility of lovastatin is low (Figure 6.4). The spatial inhomogeneities of the supersaturation for different inlet velocities are coincident with the spatial distribution of nucleation and growth rates (see Figures 6.5 and 6.6), due to their dependency on the supersaturation. Based on the supersaturation values in Figure 6.4, it can be concluded that nucleation in the impinging jet crystallizer was dominated by the primary homogeneous mechanism. The spatially averaged nucleation and growth rates are given in Table 6.1.

Table 6.1 Spatially averaged nucleation and growth rates for different jet velocities, v.

	v = 1.0 m/s	v = 6.0 m/s
Nucleation rate (#/s-m^3)	9.01×10^{13}	7.51×10^{13}
Growth rate (μm/s)	4.63×10^{2}	3.84×10^{2}

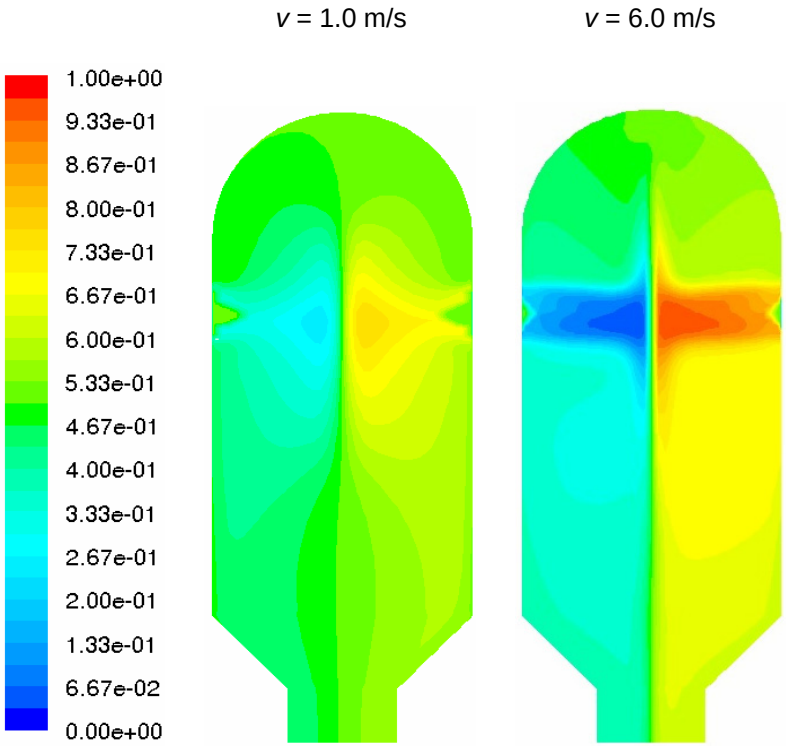

Figure 6.3 Volume fraction of antisolvent (water) in the mixed environment along the symmetry plane of the mixing chamber of the confined impinging jet. Left inlet: lovastatin saturated in methanol; right inlet: water.

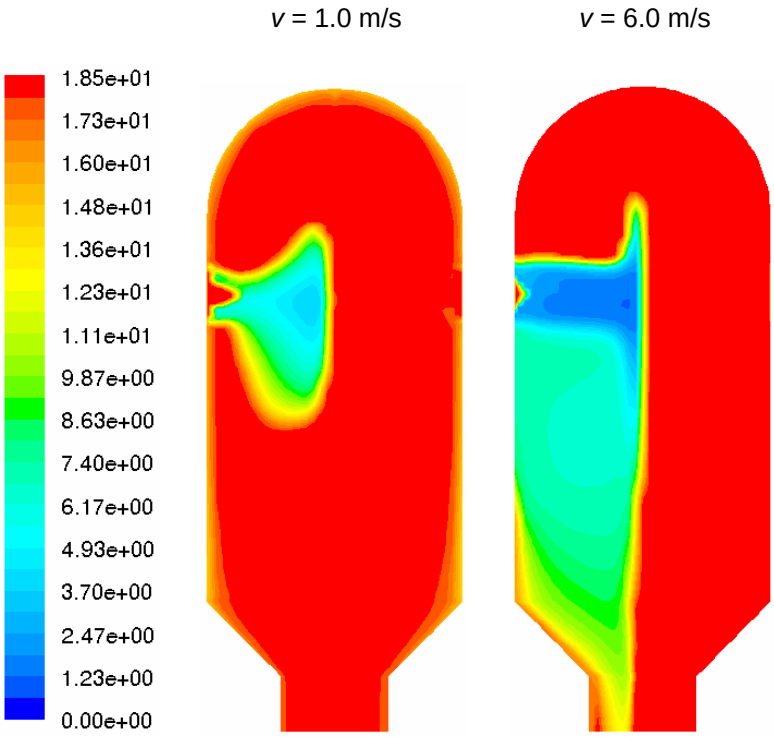

Figure 6.4 Supersaturation (c/c^*) of lovastatin in the mixed environment along the symmetry plane of the mixing chamber of the confined impinging jet. Left inlet: lovastatin saturated in methanol; right inlet: water.

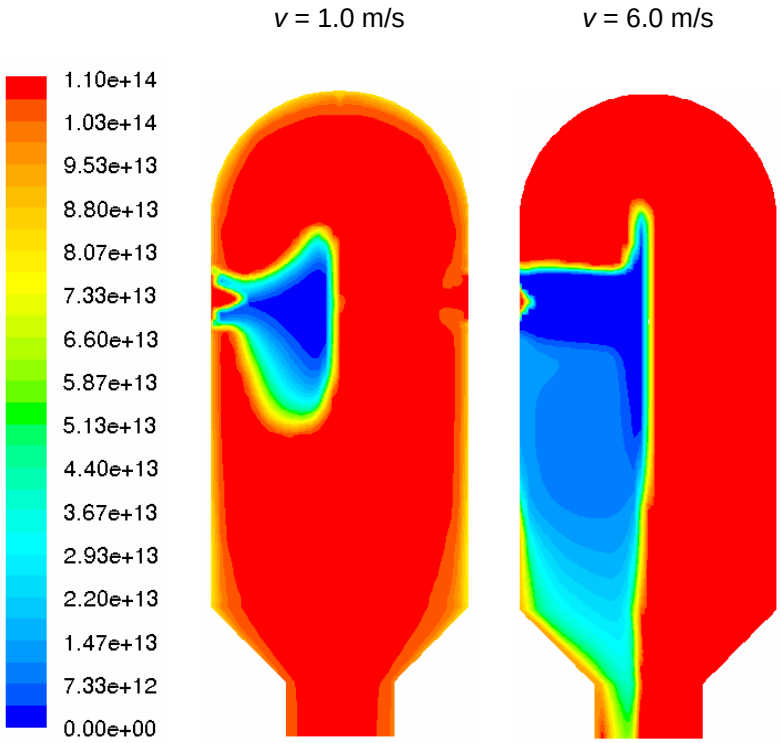

Figure 6.5 Nucleation rates (#./s-m^3) of lovastatin in the mixed environment along the symmetry plane of the mixing chamber of the confined impinging jet. Left inlet: lovastatin saturated in methanol; right inlet: water.

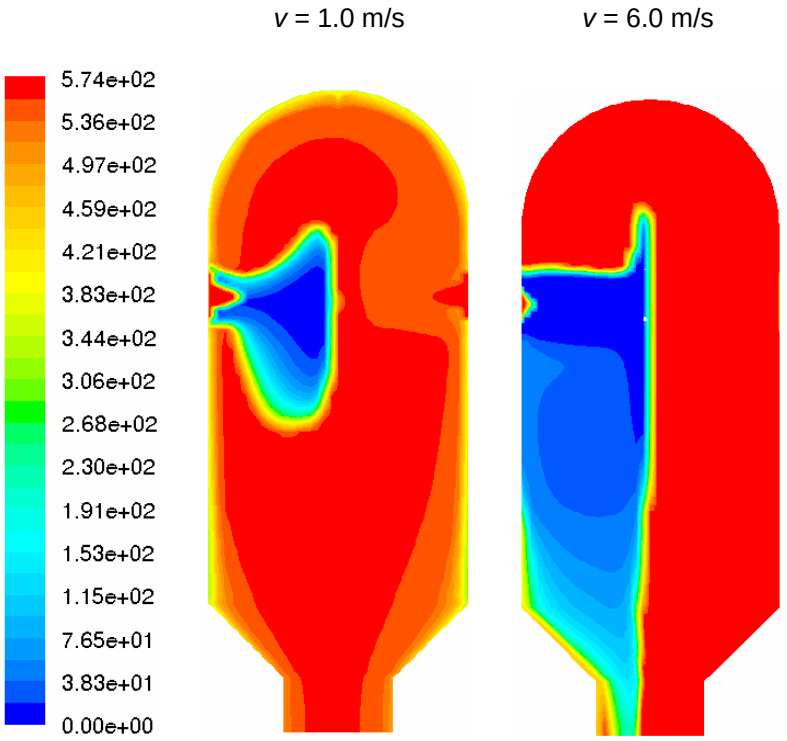

Figure 6.6 Growth rates (μm/s) of lovastatin in the mixed environment along the symmetry plane of the mixing chamber of the confined impinging jet. Left inlet: lovastatin saturated in methanol; right inlet: water.

6.6 Effect of Jet Velocity on Crystal Size Distribution

Figure 6.7 shows the crystal size distribution at the outlet of the impinging jet for different inlet velocities. The crystal size distribution is much broader with much higher crystal numbers and mass produced with relatively small reduction in the inlet velocity. It is primarily the longer residence time at lower inlet velocity that results in higher overall nucleation and growth of crystals, since the spatial averages of the nucleation and growth rates over the crystallizer are of similar magnitudes for different inlet velocities (see Figures 6.5 and 6.6, and Table 6.1). The longer residence time for the lower inlet velocities allow more time for crystals to grow, hence broadening the crystal size distribution.

The spatial distributions of the environment-weighted, cell-averaged-number density in the first and tenth bins are shown in Figure 6.8. These results indicate that the spatial distributions of small and larger crystals are much more symmetric about the central plane perpendicular to the inlet pipes for lower inlet velocities. In particular, at high inlet velocities many more crystals are located on the “antisolvent” side of the impinging jet crystallizer.

The simulation results agree with the experimental observations reported in Mahajan and Kirwan (1996) for the same lovastatin system in the following aspects (see Figure 6.9 for closer comparison):

- The shape of the crystal size distribution computed from the simulations is the same as that obtained from the experiments (Mahajan and Kirwan, 1996), despite the impinging jet used in their experiments is a non-submerged and unconfined one. However, the number population density of the smallest

crystals cannot be compared because it is not possible to measure the smallest particles accurately with the particle counter used in their experiments. In addition, the number population density for the smallest crystals from the simulations may be somewhat under-predicted due to numerical diffusion (Kurganov and Tadmor, 2000; Leveque, 2002).

- The experimental observations of less crystals and narrower size distribution for higher inlet velocities are seen in the simulations (see Figure 6.7). Following the comparisons based on micromixing time (t_M) and induction time (t_I) in Mahajan and Kirwan (1996), Figure 6.9 indicates that, for $t_M/t_I > 1$, the simulated crystal size distribution for the confined impinging jet is very similar to that of the CSD obtained from experiments for the free impinging jet. For $t_M/t_I < 1$, the crystals from the confined impinging jet are larger than the crystals from the free impinging jet. Due to mixing with the supersaturated solution already present in the mixing chamber, the crystals in the confined impinging jet have more time to grow to larger crystals.
- At sufficiently high inlet velocities, or low t_M/t_I, the crystal size distribution remains unchanged in the experiments. In the simulations, much higher inlet velocities were not simulated due to numerical instabilities in the CFD solver. Nonetheless, the crystal size distributions get closer together with decreasing t_M/t_I (see Figure 6.7) and we can expect the CSD to converge into a single distribution with further reduction in t_M/t_I. The value of t_M/t_I for the crystal size distributions to converge would be expected to be much lower for the confined impinging jet due to the difference in mixing discussed in the previous point.

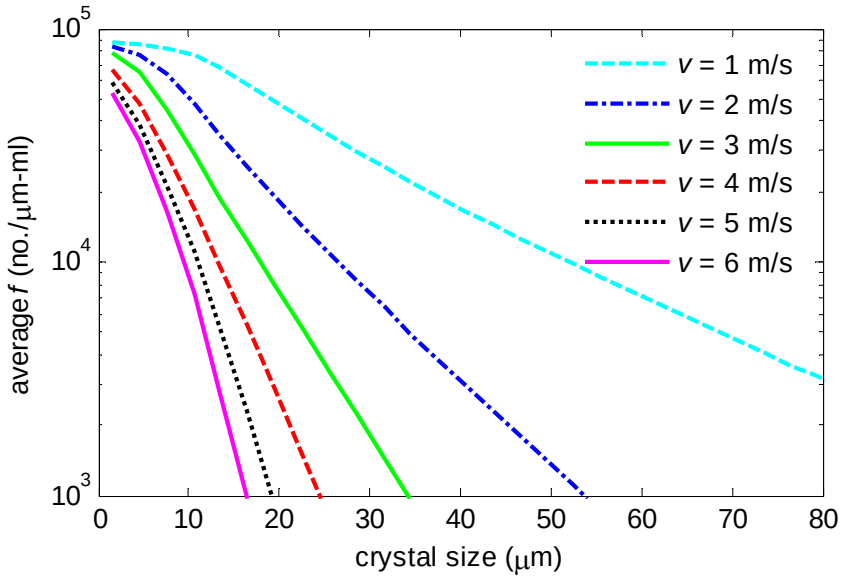

Figure 6.7 Crystal size (longest dimension) distributions of lovastatin obtained from the confined impinging jet crystallizer for different inlet velocities.

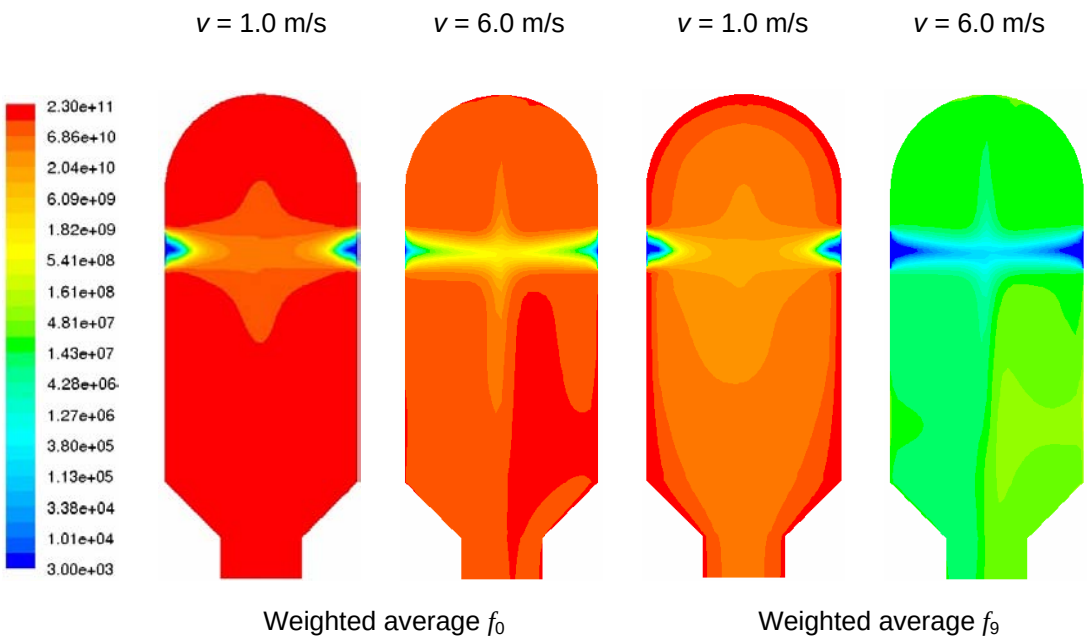

Figure 6.8 Environment-weighted, cell-averaged, number population density (#/μm-m^3) of lovastatin crystals in the first and tenth bins of the population balance equation along the symmetry plane of the mixing chamber of the confined impinging jet. Left inlet: lovastatin saturated in methanol, Right inlet: water.

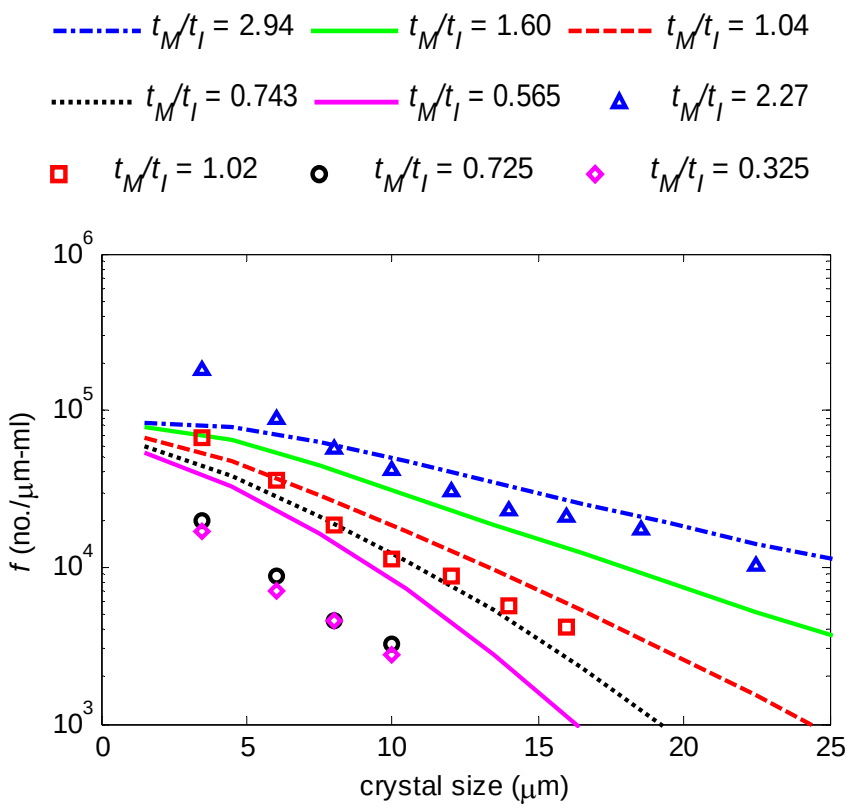

Figure 6.9 Crystal size distributions of lovastatin obtained from simulations using the confined impinging jets (lines) and experiments using the free impinging jets (markers) for different ratios of mixing time to induction time (t_M/t_I). t_M for confined impinging jets was calculated using equation (32) in reference (Johnson and Prud'homme, 2003) and t_I was obtained from reference (Mahajan and Kirwan, 1996).

The above consistencies between simulation and experiments indicate that the CFD-micromixing-population balance model is promising approach to simulate the crystal size distribution for antisolvent crystallization in impinging jet crystallizers. A next step forward would be to develop experimental systems with suitable sensors to measure the three-dimensional field of solute and solvent concentrations and crystal size distribution, and compare these fields with those predicted from the model.

From the industrial point of view, Figure 6.7 are already accurate enough to guide the selection of operations of a confined impinging jet towards a desired crystal size distribution, which depends on the method of drug administration and its dissolution requirements. Similar plots as Figure 6.7 could be computed for different three-dimensional geometries for the confined impinging jet (for example, geometries with different amounts of space above the confined jets, or with different diameters for the inlet and outlet pipes), and used to guide the design of the geometry to achieve a particular desired CSD.

6.7 Polymorphic Crystallization of L-Histidine

The regulatory and drug delivery reasons that necessitate the production of one specific polymorphic form of an active pharmaceutical ingredient are well recognized (Chemburkar et al., 2000). The development of a robust and efficient crystallization process to produce crystals of a specific polymorphic form has been considered challenging and, as a minimum, requires a detailed understanding of both the thermodynamics and crystallization kinetics of the polymorphs for a given system (Kitamura, 2004; Muller et al., 2006). The choice of solvent or solvent composition can affect the formation and the subsequent transformation of polymorphs

(Khoshkhoo and Anwar, 1993; Shekunov and York, 2000; Wang et al., 2005). For antisolvent crystallization, the crystallization and subsequent transformation of polymorphs can be affected by initial concentration, addition rate of antisolvent, seeding, and temperature (Kitamura, 2004; Kitamura and Hironaka, 2006; Kitamura and Nakamura, 2002; Kitamura and Sugimoto, 2003; Toth et al., 2005).

To the authors' knowledge, there are currently no published works on polymorphic crystallization in impinging jets. However, the formation of multiple hydrates of calcium oxalate in an impinging jet and Y-mixer has been reported by Hacherl et al. (Hacherl et al., 2003) and Haselhuhn and Kind (Haselhuhn and Kind, 2003), respectively. In this section, the polymorphic crystallization of L-histidine (stable form A and metastable form B) in a water-ethanol mixture in a confined impinging jet, where ethanol is the antisolvent, was simulated with the CFD-micromixing-population balance model. The solubility data and growth kinetics are (Kitamura et al., 1994):

$$\begin{aligned} c_A^* \text{ at } 20^\circ\text{C (mol/L of solvents)} &= -1.33V^3 + 1.86V^2 - 1.02V + 0.256 \quad (V \le 0.6) \\ c_B^* \text{ at } 20^\circ\text{C (mol/L of solvents)} &= -1.21V^3 + 1.78V^2 - 1.05V + 0.274 \quad (V \le 0.6) \end{aligned} \tag{6.6}$$

$$\begin{aligned} G_j \text{ at } 20^\circ\text{C (m/s)} &= k_{g,j}(S_j - 1)^{g_j} \qquad \text{for } V \le 0.4 \\ k_{g,A} &= 3.68\times10^{-5}V^2 - 2.90\times10^{-5}V + 5.77\times10^{-6} \\ k_{g,B} &= -1.37\times10^{-6}V^2 - 2.74\times10^{-7}V + 5.37\times10^{-7} \\ g_A &= -47.0V^2 + 16.5V + 3.40 \\ g_B &= -11.4V^2 - 6.91V + 1.40 \end{aligned} \tag{6.7}$$

where V is the volume fraction of ethanol and $S_j = c/c_j^*$. The size independence of these growth kinetics is consistent with the reported observation that the crystal

growth for this system is limited by the polynuclear growth mechanism (Roelands et al., 2006). The nucleation kinetics are (Roelands et al., 2006):

$$B_j \text{ at } 20^{o}\text{C (\#/s-m}^3) = B_{j,\text{ heterogeneous}} = 10^{20} \exp\left[\frac{-1.19\psi_j}{(\ln S_j)^2}\left(\ln\frac{1}{0.108c_{j*}}\right)^3\right] \qquad (6.8)$$

$$\psi_A = 0.3, \quad \psi_B = 0.255$$

Only the primary heterogeneous nucleation kinetic expression was applied as the metastable limit for the homogeneous nucleation was not exceeded in the simulations, and secondary nucleation was neglected due to the very small particle volume in impinging jets. The solvent-mediated transformation kinetics are excluded as the solution was supersaturated for both polymorphs throughout the crystallizer.

The simulation had saturated water solution in one inlet and saturated solution of water and ethanol in a 3:2 volume ratio in the other inlet (the inlet saturated concentration was calculated from the solubility curve of polymorph B). The average density of 956.56 kg/m^3 and viscosity of 1.0355×10^{-3} Pa-s were used to compute the flow field. The density of L-histidine is 1440 kg/m^3 and the volume shape factor is $2/3$. The population balance equation was discretized into 20 bins for both polymorphs, with Δr = 0.0002 μm. As in the previous section, the use of effective viscosity was omitted because the solid fraction was too small to significantly affect the effective viscosity.

6.8 Crystallization Dynamics and Crystal Size Distribution of Polymorphs

The supersaturation and nucleation rates and growth rates of form A and B in the mixed environment for the inlet velocity of 6 m/s are shown in Figures 6.10 to 6.12.

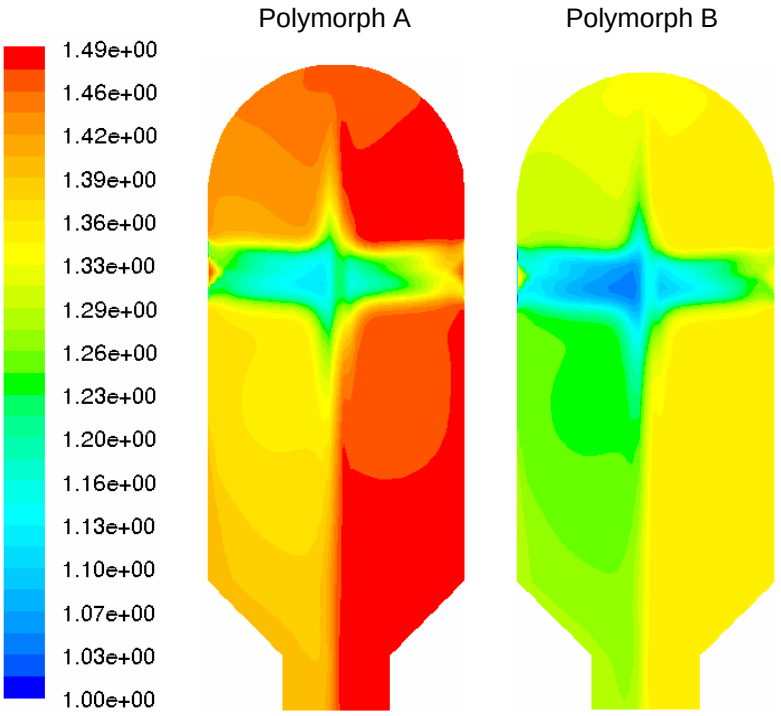

Figure 6.10 Supersaturation (c/c^*) of L-histidine in the mixed environment along the symmetry plane of the mixing chamber of the confined impinging jet for the inlet velocity of 6 m/s. Left inlet: L-histidine saturated in water; right inlet: L-histidine saturated in water and ethanol in a 3:2 volume ratio.

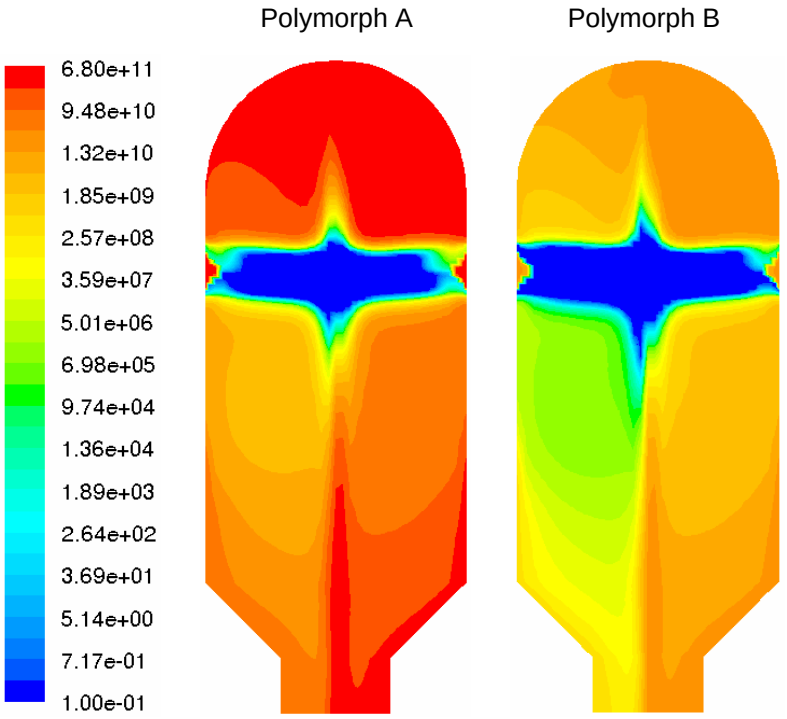

Figure 6.11 Nucleation rates (#/s-m^3) of L-histidine in the mixed environment along the symmetry plane of the mixing chamber of the confined impinging jet for the inlet velocity of 6 m/s. Left inlet: L-histidine saturated in water; right inlet: L-histidine saturated in water and ethanol in a 3:2 volume ratio.

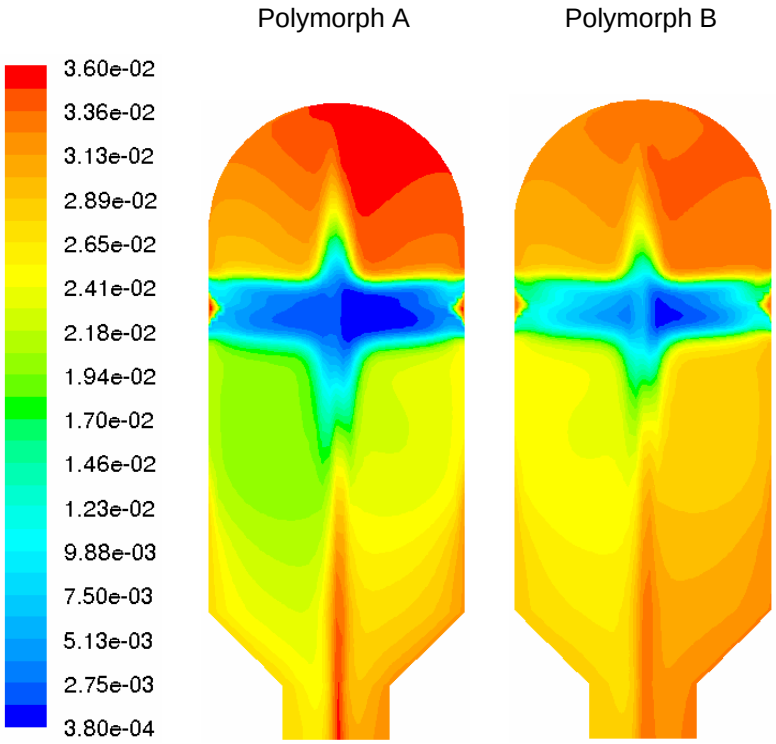

Figure 6.12 Growth rates (μm/s) of L-histidine in the mixed environment along the symmetry plane of the mixing chamber of the confined impinging jet for the inlet velocity of 6 m/s. Left inlet: L-histidine saturated in water; right inlet: L-histidine saturated in water and ethanol in a 3:2 volume ratio.

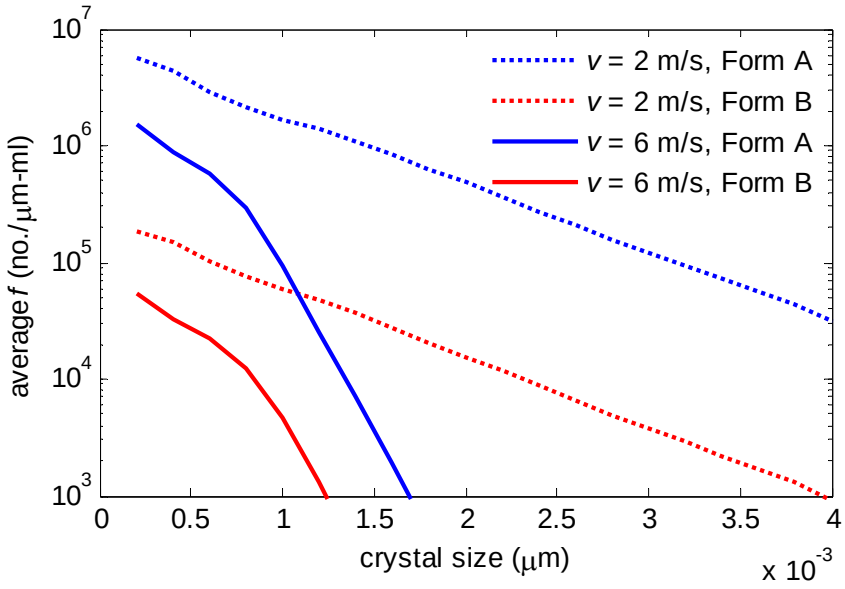

Figure 6.13 Crystal size distributions of polymorphs A and B of L-histidine obtained from the confined impinging jet crystallizer for different inlet velocities.

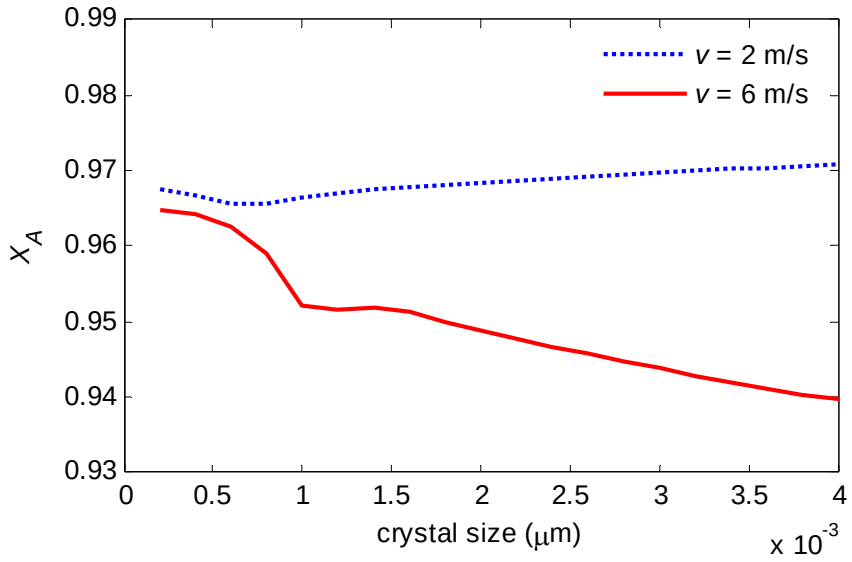

Figure 6.14 Fraction of polymorph A ($X_A = f_A/(f_A + f_B)$) obtained from the confined impinging jet crystallizer for different inlet velocities.

Overall, the supersaturation is lower and the nucleation rates are slightly higher for polymorph B, but the growth rates are comparable for both polymorphs. The crystal size distribution at the outlet is shown in Figure 6.13. The higher nucleation rates for polymorph B is consistent with the larger number of form B crystals produced. The shape of the crystal size distributions for each polymorph is similar due to the similar growth rates of the polymorphs, under these process conditions. Here, the Ostwald's rule of stages is not entirely satisfied since both stable and metastable polymorphs are nucleated simultaneously, which is also known as concomitant polymorphs (Bernstein et al., 1999). This is consistent with the experimental data reported in literature (Kitamura et al., 1994; Roelands et al., 2006).

As illustrated in Figure 6.13, the variation of the crystal size distribution with inlet velocity is consistent with the observations for the lovastatin system in the previous section. The fraction of polymorph A, on the basis on number density, can vary with crystal size and is dependent on the inlet velocity (see Figure 6.14). For the supersaturation and solvent composition values similar to the spatial averages in the impinging jet, the polymorph ratio in Figure 6.14 is consistent with the experimental observations reported for rapid cooling in an agitated vessel (Kitamura et al., 1994; Roelands et al., 2006). Figures 6.13 and 6.14 indicate that it is possible to tailor the crystal size distribution and polymorph ratio, by adjusting the inlet velocity, for achieving the desired crystal size distribution, as well as designing a more efficient solvent-mediated transformation process in the subsequent step (Kitamura, 2004).

6.9 Conclusions and a Look into the Future

This chapter illustrates the application of the CFD-Micromixing-Population Balance model to simulate antisolvent crystallization in confined impinging jets. For crystallization of lovastatin in a confined impinging jet, the simulation results from the CFD-micromixing-population balance model was compared with published experimental results and a good agreement between model predictions and experimental data was established. However, a full validation for the crystal size distribution in the entire three-dimensional field in the impinging jet remains to be done. This would ensure confidence when using this coupled model for other crystallization systems. The polymorphic crystallization of L-histidine was also simulated to demonstrate the capability of the CFD-micromixing-population balance model to simulate the crystallization of polymorphic systems.

The simulations related the crystal size distribution and ratio of polymorphs to changes in the inlet jet velocity. The simulation method enables an investigation into the variation of other operating variables such as the inlet concentrations and design variables such as the 3D geometry of the impinging jet to determine their effects on the crystal size distribution and the ratio of polymorphs. Note that in the initial stages of process development, the pharmaceutical compound is usually available in very small amounts. This makes it difficult to carry out numerous trial-and-error experiments to determine the operating conditions that give a desired crystal size distribution. Simulation tools could provide a technological advancement to industrial crystallization and process development in the pharmaceutical industry, by making it possible to design impinging jet crystallizers in-silico to give desired crystal size

distributions. The development of the crystallization process would be done in the following steps:

1. For a given antisolvent-solvent-solute crystallization system, hydrodynamics-independent crystallization kinetics would be first determined by using small amounts of the pharmaceutical compound, for example, with the use of microfluidic devices. Many designs for microfluidic devices have been developed to mix fluids in very short times, so that the system is not transport-limited (Squires and Quake, 2005). For particular systems or conditions where transport limitations exist, the parameter estimation procedure can be carried based on the macromixing-micromixing-PBE model.
2. For a given impinging jet geometry, the parameters for the CFD-micromixing model are estimated using competitive reactions data, which are inexpensive and easy to obtain.
3. With the estimated parameters and crystallization kinetics, the CFD-micromixing-population balance model is used to simulate the crystal size distribution for a wide range of operating conditions (for example inlet velocities and inlet concentrations). The simulation results then can be used to identify the set of operating conditions that produces the crystal size distribution that meets the bioavailability requirements.

Once the CFD parameters have been determined by comparisons of simulation results with experimental solution concentration and velocity fields for several impinging jet geometries, then Step 2 could be skipped in future applications. Instead, the operating conditions and geometry of the impinging jet could be optimized to give a desired crystal size distribution. This design strategy would allow an impinging jet crystallizer

and its operations to be designed in a systematic engineering manner. In addition, it would reduce the use of pharmaceutical compound to a minimum, and with the availability of faster computers, this process design would be done in shorter times.

6.10 References

Bernstein, J., Davey, R. J. and Henck, J. O. (1999). Concomitant polymorphs. *Angewandte Chemie-International Edition* **38**(23): 3441-3461.

Chemburkar, S. R., Bauer, J., Deming, K., Spiwek, H., Patel, K., Morris, J., Henry, R., Spanton, S., Dziki, W., Porter, W., Quick, J., Bauer, P., Donaubauer, J., Narayanan, B. A., Soldani, M., Riley, D. and McFarland, K. (2000). Dealing with the impact of ritonavir polymorphs on the late stages of bulk drug process development. *Organic Process Research & Development* **4**(5): 413-417.

Hacherl, J. M., Paul, E. L. and Buettner, H. M. (2003). Investigation of impinging-jet crystallization with a calcium oxalate model system. *AIChE Journal* **49**(9): 2352-2362.

Haselhuhn, F. and Kind, M. (2003). Pseudo-polymorphic behavior of precipitated calcium oxalate. *Chemical Engineering & Technology* **26**(3): 347-353.

Johnson, B. K. and Prud'homme, R. K. (2003). Chemical processing and micromixing in confined impinging jets. *AIChE Journal* **49**(9): 2264-2282.

Khoshkhoo, S. and Anwar, J. (1993). Crystallization of polymorphs - the effect of solvent. *Journal of Physics D-Applied Physics* **26**(8B): B90-B93.

Kitamura, M. (2004). Controlling factors and mechanism of polymorphic crystallization. *Crystal Growth & Design* **4**(6): 1153-1159.

Kitamura, M., Furukawa, H. and Asaeda, M. (1994). Solvent effect of ethanol on crystallization and growth-process of L-histidine polymorphs. *Journal of Crystal Growth* **141**(1-2): 193-199.

Kitamura, M. and Hironaka, S. (2006). Effect of temperature on antisolvent crystallization and transformation behaviors of thiazole-derivative polymorphs. *Crystal Growth & Design*.

Kitamura, M. and Nakamura, K. (2002). Dependence of polymorphic transformation on anti-solvent composition and crystallization behavior of thiazole-derivative pharmaceutical. *Journal of Chemical Engineering of Japan* **35**(11): 1116-1122.

Kitamura, M. and Sugimoto, M. (2003). Anti-solvent crystallization and transformation of thiazole-derivative polymorphs - I: effect of addition rate and initial concentrations. *Journal of Crystal Growth* **257**(1-2): 177-184.

Kurganov, A. and Tadmor, E. (2000). New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *Journal of Computational Physics* **160**(1): 241-282.

Leveque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge, U. K., Cambridge University Press.

Mahajan, A. J. and Kirwan, D. J. (1994). Nucleation and growth kinetics of biochemicals measured at high supersaturations. *Journal of Crystal Growth* **144**(3-4): 281-290.

Mahajan, A. J. and Kirwan, D. J. (1996). Micromixing effects in a two-impinging-jets precipitator. *AIChE Journal* **42**(7): 1801-1814.

Muller, M., Meier, U., Wieckhusen, D., Beck, R., Pfeffer-Hennig, S. and Schneeberger, R. (2006). Process development strategy to ascertain reproducible API polymorph manufacture. *Crystal Growth & Design* **6**(4): 946-954.

Roelands, C. P. M., Jiang, S., Kitamura, M., ter Horst, J. H., Kramer, H. J. M. and Jansens, P. J. (2006). Antisolvent crystallization of the polymorphs of L-histidine as a function of supersaturation ratio and of solvent composition. *Crystal Growth & Design* **6**(4): 955-963.

Shekunov, B. Y. and York, P. (2000). Crystallization processes in pharmaceutical technology and drug delivery design. *Journal of Crystal Growth* **211**(1-4): 122-136.

Squires, T. M. and Quake, S. R. (2005). Microfluidics: Fluid physics at the nanoliter scale. *Reviews of Modern Physics* **77**(3): 977-1026.

Toth, J., Kardos-Fodor, A. and Halasz-Peterfi, S. (2005). The formation of fine particles by salting-out precipitation. *Chemical Engineering and Processing* **44**(2): 193-200.

Wang, J., Loose, C., Baxter, J., Cai, D. W., Wang, Y. L., Tom, J. and Lepore, J. (2005). Growth promotion by H2O in organic solvent-selective isolation of a target polymorph. *Journal of Crystal Growth* **283**(3-4): 469-478.

Woo, X. Y., Tan, R. B. H., Chow, P. S. and Braatz, R. D. (2006). Simulation of mixing effects in antisolvent crystallization using a coupled CFD-PDF-PBE approach. *Crystal Growth & Design* **6**(6): 1291-1303.

Chapter 7
Concentration Control of Antisolvent Crystallization with Laser Backscattering Measurement

7.1 Introduction

The previous chapters described the use of simulation methods to develop a deeper understanding of mixing effects in antisolvent crystallization, to provide a capability for addressing a key aspect of process development—how to scale up a crystallization process from the bench-scale to the production scale. The main objective of this and the subsequent chapter is to address another key aspect of process development—how to control the batch antisolvent crystallization process to produce crystals of a desired quality. The chapters provide a sensitivity and disturbance analysis of various control strategies for antisolvent crystallization, with this chapter focusing on concentration control, with and without inclusion of laser backscattering measurement to increase robustness with respect to disturbances that result in excessive nucleation events. The next chapter explores new strategies for operating and controlling impinging jet crystallizers, to produce a target crystal size distribution.

7.2 Control of Crystallization Processes

The control of industrial crystallization processes have received increased research attention in the recent years (Braatz, 2002; Fujiwara et al., 2005; Larsen et al., 2006). This is motivated by the critical need to consistently meet the specifications on purity, crystal size and shape distributions, and polymorphic form in the crystallization of pharmaceuticals (Paul et al., 2005), as well as the advancement of simulation and

sensor technologies (Braatz et al., 2002; Yu et al., 2004). Along with the Food and Drug Administration's (FDA) initiative to introduce process analytical technologies (PAT) in the pharmaceutical industries, the development of control strategies to improve the performance of the manufacturing process, as well as the quality of the product, becomes increasingly important (Barrett et al., 2005; Birch et al., 2005; Yu et al., 2004).

One method for operating pharmaceutical crystallization process is to follow a predetermined temperature or antisolvent composition (or addition rate) profile (Figure 7.1a). A first-principles approach to determination of the optimal profile, as well as seed characteristics and loading, involves solving an optimization defined by a performance objective and the process constraints (Chung et al., 1999; Ma et al., 2002; Sarkar et al., 2006; Togkalidou et al., 2004; Ward et al., 2006; Worlitschek and Mazzotti, 2004). A requirement of this approach is the need to simulate the crystallization process with accurate nucleation and growth kinetics, which only can be determined in a series of experiments (Chung et al., 2000; Gunawan et al., 2002; Ma and Braatz, 2003; Togkalidou et al., 2004). An important practical consideration is that the kinetic and thermodynamic parameters can vary from batch to batch due to varying amounts of impurities. Hence, the robustness of the optimal profile depends on the sensitivity of the performance objective on the parametric variations (Ma et al., 1999).

An alternative approach that does not require accurate kinetics and numerous trial-and-error experimentation is to adjust the cooling or antisolvent addition rate to

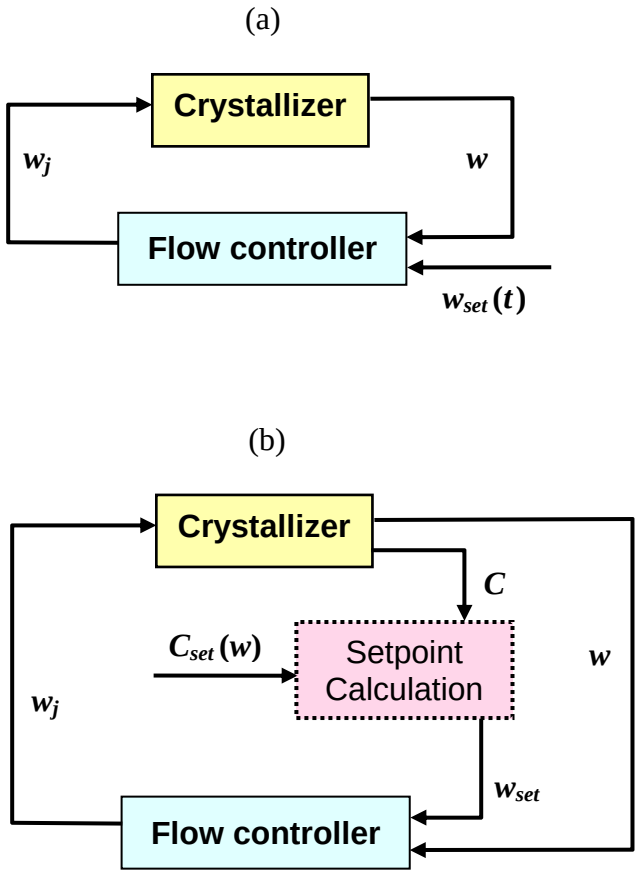

Figure 7.1 Schematic block diagrams for **(a)** antisolvent composition (w) versus time (t) approach, and **(b)** concentration (C) versus antisolvent composition approach.

follow a concentration (or supersaturation) profile within the metastable zone using feedback control based on in-process concentration measurement (see Figure 7.1b) (Fujiwara et al., 2002; Gron et al., 2003; Yu et al., 2006). This approach has been implemented successfully in the pharmaceutical industry and is commonly known as concentration control, supersaturation control, or direct design (Liotta and Sabesan, 2004; Nonoyama et al., 2006; Zhou et al., 2006). The main advantage of this approach is its insensitivity to most process disturbances. In the following sections, the sensitivities of concentration control in antisolvent crystallization are investigated for a wide range of process disturbances, which extends past work on cooling crystallization (Fujiwara et al., 2005).

7.3 Concentration Control

Concentration control in crystallization processes involves the feedback control of antisolvent flow rate to maintain a preset supersaturation profile based on concentration measurements. The equations below consider the case of following a concentration setpoint based on constant supersaturation setpoint profile within the metastable limit (it was shown in earlier chapters that the solution concentration in stirred vessels are sufficiently homogeneous). The solute concentration at time k (c_k) is

$$c_k = \frac{m_{solute,k}}{m_{solvent} + m_{antisolvent,k}} \tag{7.1}$$

and the setpoint concentration at the new time step $k+1$ is

$$c_{setpoint} = c_{k+1} = c^*(w_{k+1}) + \Delta c \tag{7.2}$$

where m is mass, w is antisolvent mass % on solute-free basis, and c^* is the saturated solution concentration as a function of w. Taking into account the dilution effect,

$$w_{k+1} = 100\left(\frac{m_{antisolvent,k} + t_s \dot{m}_{antisolvent,k+1}}{m_{solvent} + m_{antisolvent,k} + t_s \dot{m}_{antisolvent,k+1}} \right) \tag{7.3}$$

and that $m_{solute,k+1} \approx m_{solute,k}$ to very high accuracy when the sampling time is small,

$$c_{setpoint} = c_{k+1} = \frac{m_{solute,k}}{m_{solvent} + m_{antisolvent,k} + t_s \dot{m}_{antisolvent,k+1}} \tag{7.4}$$

where $\dot{m}$ is the mass flow rate, and t_s is the sampling time. Combining the above equations gives the following equation to be solved to obtain w_{k+1}:

$$c^*(w_{k+1}) + \Delta c + \frac{m_{solute,k}}{m_{solvent}} \left(\frac{w_{k+1}}{100} - 1 \right) = 0 \tag{7.5}$$

with $\dot{m}_{antisolvent}$ subsequently determined from equation (7.3). Equation (7.5) can also be applied with any supersaturation profiles as a function of antisolvent composition.

The above equations were implemented in simulations of the seeded antisolvent crystallization of paracetamol in an acetone-water mixture (see Figure 7.2, and see Section 4.2 and 4.3 for details on the simulation model) (Granberg et al., 1999; Granberg and Rasmuson, 2000). It can be seen that equation (7.5) follows the supersaturation setpoint profile very closely. The experimental implementation has been carried out for a proprietary pharmaceutical compound reported by Zhou et al. (2006). In the following sections, the sensitivity and robustness of the concentration control approach in antisolvent crystallization are investigated and compared with the control system where the supersaturation setpoint is specified as an explicit function of time. Similar studies had been performed for cooling crystallization (Fujiwara et al., 2005).

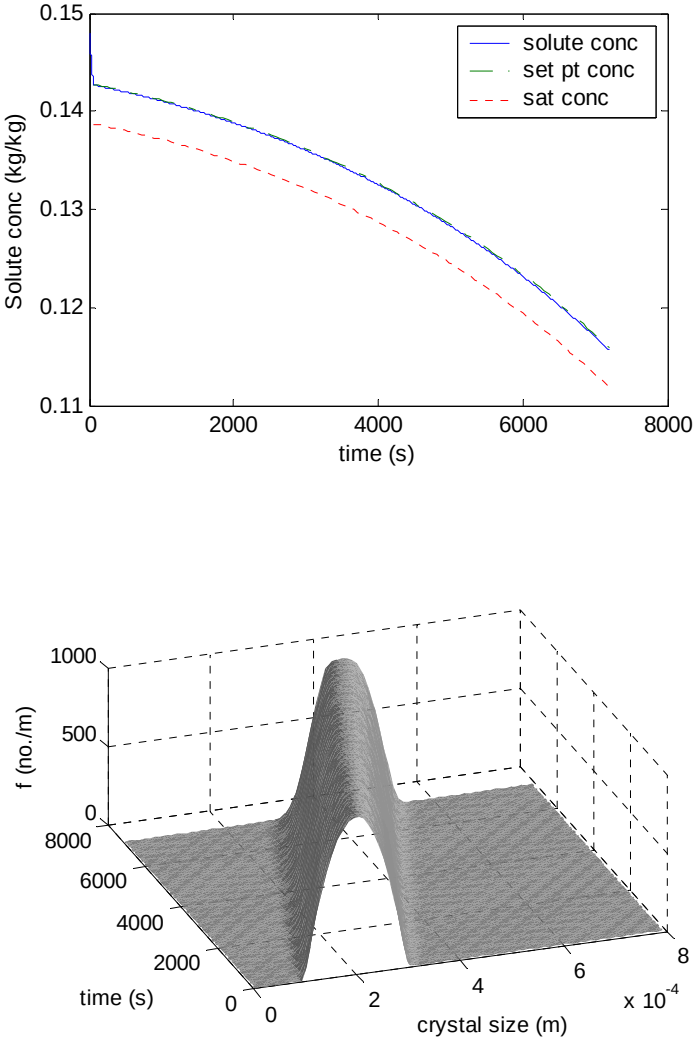

Figure 7.2 The supersaturation and concentration profiles and product crystal size distribution during the simulated seeded antisolvent crystallization of paracetamol in acetone-water mixture with concentration control. The simulation uses a sampling time t_s = 60 s, a constant supersaturation setpoint Δc = 0.004 kg solute/kg solvents, and a seed amount of 1.586 g/kg solvents over a batch time a 2 h.

7.4 Supersaturation Profiles for Antisolvent Crystallization

In this section, the application of different supersaturation profiles for seeded antisolvent crystallization is discussed. The antisolvent crystallization kinetic expressions were inspired from Granberg et al. (1999). The nucleation rates are given by:

$$B(\#\ \text{particles/m}^3\text{-s}) = k_b \Delta c^b \tag{7.6}$$

$$k_b = 4.338\times10^{58}\exp(-1.374w) \quad (60\% \le w \le 80\%) \tag{7.7}$$

$$b = 1.997\times10^{-3}w^2 - 6.237\times10^{-1}w + 4.042\times10^{1} \quad (60\% \le w \le 80\%) \tag{7.8}$$

and the growth rate expressions are:

$$G(\text{m/s}) = k_g \Delta c^g \tag{7.9}$$

$$k_g = -9.6300\times10^{-11}w^3 + 3.3558\times10^{-8}w^2 - 1.2606\times10^{-6}w + 3.6852\times10^{-5} \tag{7.10}$$

$$g = -1.108\times10^{-4}w^2 + 1.024\times10^{-2}w + 1.427, \tag{7.11}$$

where $\Delta c(\text{kg solute/kg solvents}) = c - c^*$, and w is the mass % of antisolvent on a solute free basis. The solubility curve at 16^{o}C is given by (Granberg and Rasmuson, 2000):

$$c^* = 1.302\times10^{-6}w^3 - 1.882\times10^{-4}w^2 - 2.237\times10^{-4}w - 5.746\times10^{-1} \quad (60\% \le w \le 80\%) \tag{7.12}$$

Secondary nucleation is neglected due to low solids density of this particular system.

As both nucleation and growth rates increase with antisolvent composition, operating a seeded crystallizer at constant supersaturation can result in nucleation occurring towards the end of the batch as the antisolvent composition increases (for direct addition mode). Such dependence is also expected for secondary nucleation when solids density is significant. Thus, a supersaturation profile that maximizes growth

and minimizes nucleation for the operating range of antisolvent composition must be determined. By setting a constant tradeoff between growth and nucleation rates, a supersaturation profile can be computed by:

$$K = \frac{G}{B} = \frac{k_g \Delta c^g}{k_b \Delta c^b}, \qquad \Rightarrow \Delta c = \left(K \frac{k_b}{k_g} \right)^{\frac{1}{g-b}}, \tag{7.13}$$

where K is a tradeoff ratio between growth and nucleation rates constrained by achieving a targeted yield within a specified batch time. If the kinetic constants are weak functions of antisolvent composition, a constant supersaturation profile would suffice (Jones and Mullin, 1974; Mullin and Nyvlt, 1971).

Table 7.1 Comparison between four supersaturation profiles. The simulation uses a batch time of 2 h, initial volume of 300 ml, maximum volume of 500 ml, and maximum flow rate of antisolvent of 6 ml/min.

Case	Number-mean size (μm)	Weight-mean size (μm)	Yield (%)
Initial Condition Saturated solution with 60% antisolvent Seed mass = 0.4125 g	187.50	195.65	0.00
Case A 'Low' constant supersaturation Δc = 0.01 kg/kg	450.76	478.65	36.36
Case B 'High' constant supersaturation Δc = 0.01105 kg/kg	465.35	553.11	53.54
Case C Constant tradeoff $K = 7\times10^{-6}$ (μm/s)/(#/m^3-s)	484.94	556.38	53.02
Case D Constant relative supersaturation $\Delta c/c^* = 0.09$	484.01	556.76	53.73

Table 7.1 shows the number- and weight-mean size and yield for antisolvent addition rates based on different constant supersaturations, constant tradeoff, and constant relative supersaturation. The values were computed by the method of moments and mass balances, assuming a well-mixed crystallizer, using Matlab 7.0.1. Figure 7.3 shows the supersaturation profile as a function of antisolvent composition for the case of constant tradeoff and constant relative supersaturation. While a low constant supersaturation profile (Case A) results in small negligible nucleation, the yield is significantly lower and a longer batch time would be required to achieve a higher yield. On increasing the constant supersaturation (Case B), the seeds have grown much larger (greater weight-mean size) and a higher yield is obtained. However, there is more nucleation occurring towards the end of the batch (see the μ_0 plot in Figure 7.4), which can cause problems in the subsequent filtration and drying processes.

The values of the constant supersaturation and constant relative supersaturation (Cases B and D, respectively) in Table 7.1 were chosen to give approximately the same yield as the constant tradeoff case (Case C), so the three cases can be compared on a consistent basis. The supersaturation profile based on constant tradeoff (Case C) decreases with increasing antisolvent composition for this system, which gives a qualitatively different antisolvent addition rate profile compared to constant supersaturation (see Figure 7.5). Maintaining a constant tradeoff (Case C) results in a steady increase in nucleated crystals, but following a constant supersaturation (Case B) gives more nucleation near the end of the batch (Figure 7.4). This results in more fines for Case B (smaller number-mean size in Table 7.1), as the crystals nucleated near the end of the batch have less time to grow. The supersaturation profile obtained based on constant relative supersaturation (Case D) is qualitatively similar to that of a

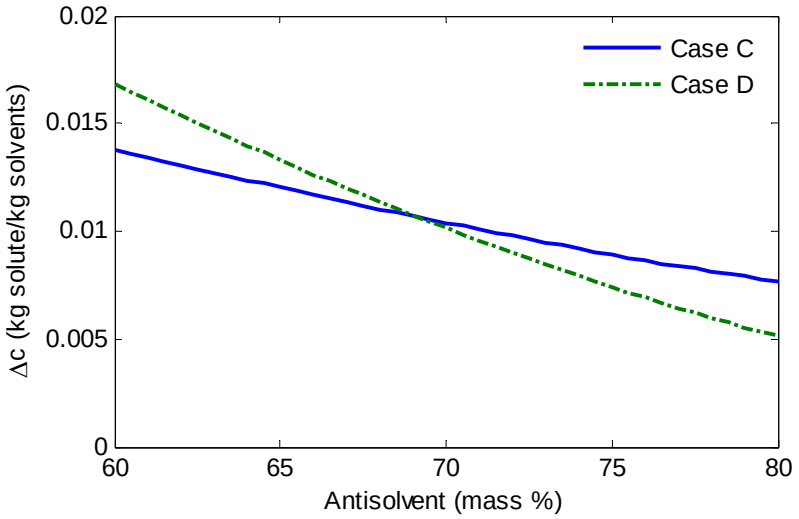

Figure 7.3 Supersaturation profiles based on constant relative supersaturation and constant tradeoff listed in Table 7.1.

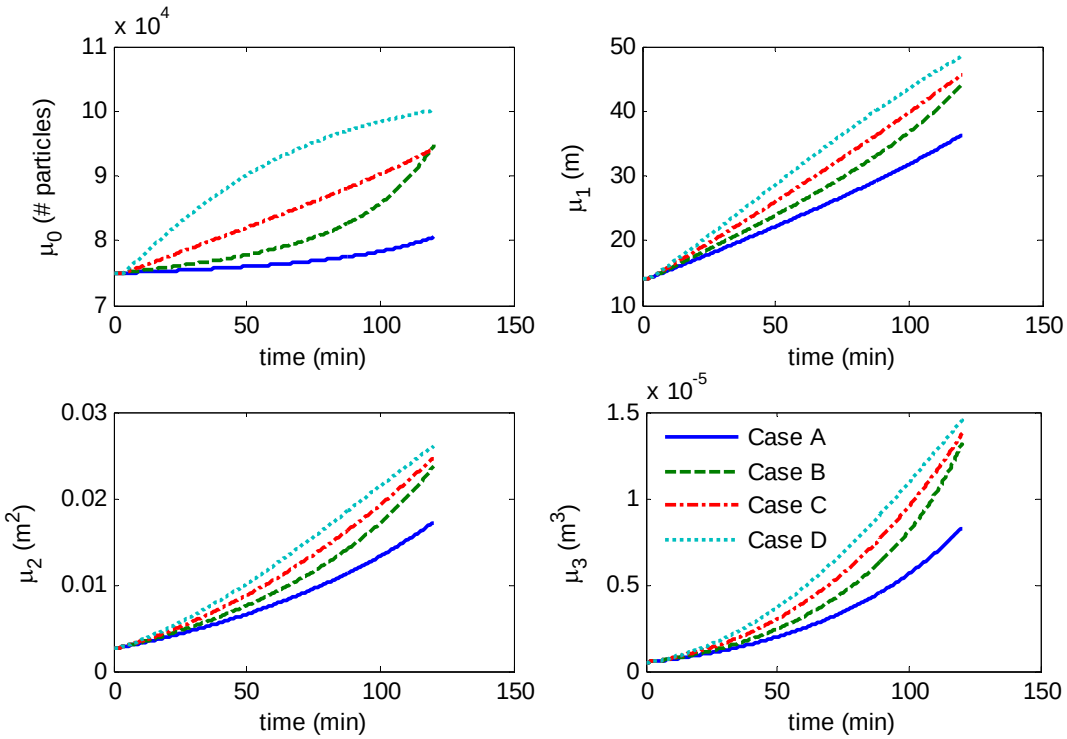

Figure 7.4 First to fourth moments obtained from supersaturation profiles listed in Table 7.1.

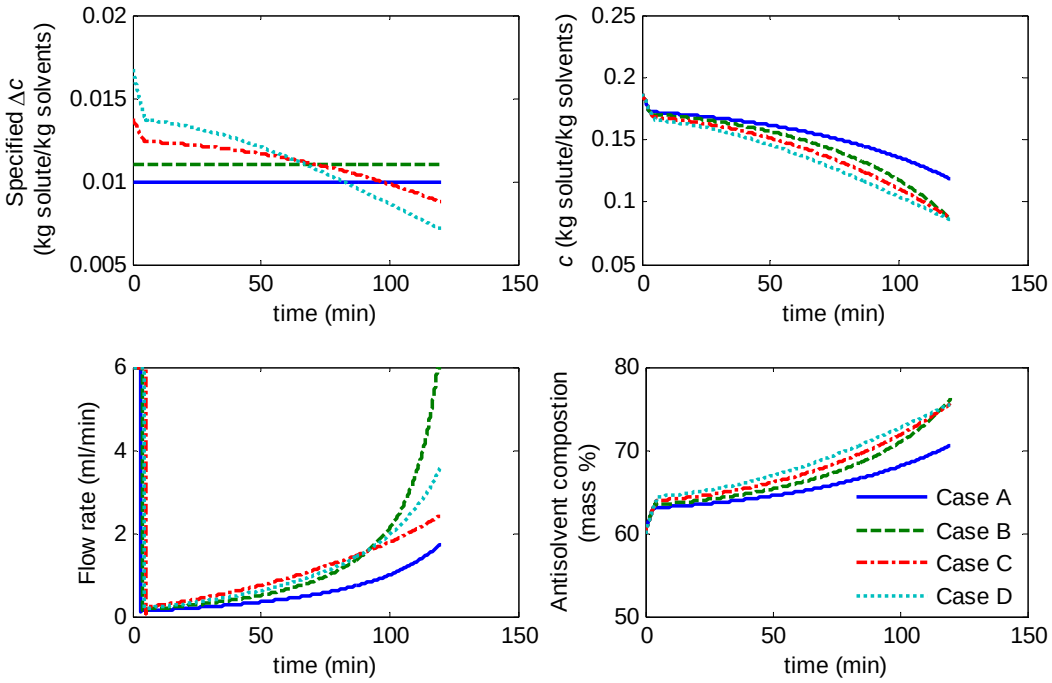

Figure 7.5 Variation of supersaturation Δ*c*, concentration *c*, antisolvent flow rate, and antisolvent composition with time from supersaturation profiles listed in Table 7.1.

constant tradeoff (Case C), but more pronounced, which results in increased nucleation (see the μ_0 plot in Figure 7.4). Crystal product quality based on number and weight-mean size, and yield is similar for both cases. This suggests that, for this system, if the kinetic parameters were not available then several constant relative supersaturation profiles could be evaluated experimentally to converge to nearly the same operations as Case C (which required kinetic parameters to compute). Alternatively, an automated approach (Zhou et al., 2006) could be used to for the systematic experimental convergence of the supersaturation profile towards the optimal setpoint profile.

7.5 Comparison between Direct Operation and Concentration Control Approaches

The antisolvent addition rate profile as a function of time obtained from Case C can be directly implemented to a batch crystallizer as a function of time (referred to as *direct operation*, which is the dominant implementation in industrial practice), or the antisolvent addition rate can be computed according to the desired supersaturation profile based on measured concentrations of solute and solvents at each sampling instance (concentration control). Direct operation is analogous to the T-control strategy for cooling crystallization (Fujiwara et al., 2005). *Direct design* refers to the implementation of concentration control without any prior knowledge of the crystallization kinetics, to experimentally converge to a supersaturation profile that falls between the solubility curve and the metastable limit of the system [2]. The goal of this section is to compare the direction operation and concentration control approaches for operating an antisolvent crystallizer, to assess the relative merits of each.

For the comparison, the supersaturation profile as determined for Case C was used. While the direct operation approach has a fixed batch time of 2 hours, the concentration control approach was set to meet a yield of 53% with a flexible batch time (maximum batch time set at 10 hours). The concentration measurements are assumed to be made every 30 seconds, which is a sufficient sampling time for ATR-FTIR spectroscopy. Figure 7.6 shows the time profile of antisolvent flow rate, antisolvent composition, supersaturation, and solute concentration. There is an initial sharp increase in antisolvent composition to create the high supersaturation from the initial saturated solution. This subsequently results in a drop in the supersaturation setpoint according to the supersaturation profile shown in Figure 7.3. Figure 7.6 also indicates that the concentration control approach can follow the supersaturation setpoint closely. There is slight drift towards the end of the batch due to an increase in growth and nucleation rates at higher antisolvent composition, and an increase in desupersaturation rate as a consequence of the increase in crystal surface area. This drift can be reduced to a negligible value by selecting a shorter sampling time (the sampling time can be reduced to 1 second using a modern FTIR spectrometer using 1 scan per measurement).

Table 7.2 report the sensitivities of both operating strategies to disturbances. For the disturbances in the pump flow rate, the initial mass of antisolvent, and the nucleation prefactor k_b, small deviations in the crystal product quality and yield are observed for direct operation. Concentration control is less sensitive to these disturbances, especially for achieving the target yield. The yield for direction operation is very sensitive to the evaporation of some of the organic solvent, where concentration control produces larger crystals while achieving the target yield. Concentration

control is much less sensitive than direct operation to variations in the growth kinetics (k_g and g).

For a positive shift in the solubility curve or in the nucleation exponent b, which both result in a decrease in nucleation, concentration control results in larger crystals for the targeted yield. However, when there is a negative shift in the solubility curve or in the nucleation exponent b, both direction operation and concentration control result in large deviations in mean crystal size due to excessive nucleation at the early stage of the batch while following the supersaturation setpoint (see Figure 7.7). The lower sensitivity of concentration control to all of the disturbances except for these two motivates the development of a modification of the concentration control approach specifically designed to handle disturbances that create unexpected large nucleation.

One way to reduce the sensitivity of the concentration control approach to such disturbances is to include crystal count measurement to detect the onset of excessive nucleation, commonly used in the detection of the metastable limit (Fujiwara et al., 2002; Zhou et al., 2006). The most commonly used sensor for estimating the number of crystals is laser backscattering, for which the most commonly used sensor used in industry is the FBRM. While crystals much smaller than ~1 micron cannot be measured with the FBRM, the high supersaturation that nucleates such crystals cause the nuclei to grow rapidly, so that they can be detected. An increase in the total number of crystals counted per second, commonly referred to as the “total counts”, indicated that excessive nucleation has been detected (Fujiwara et al., 2002). In the proposed modification to concentration control, once excessive nucleation is detected, the supersaturation setpoint is reduced and the flow rate of the antisolvent is reduced

to zero until the new supersaturation setpoint is reached. The crystal count can be measured at the faster rate of every 5 seconds (e.g., for FBRM). In this study, *excessive* nucleation was detected when the increase in crystal counts is greater than 10% of the expected increase in nuclei (with no disturbances) within the first 12 min. Subsequently, the supersaturation profile was reduced to 40%, in magnitude, of the original supersaturation profile.

The last two rows in Table 7.2 and Figure 7.5 show that this inclusion of the measurement of crystal counts provides a significant reduction in the sensitivity of the concentration control approach to disturbances that cause unexpected nucleation events. Nevertheless, the improvement would vary with different systems, and with the extent in which the supersaturation is reduced. Thus, for solute-solvents systems in which the nucleation kinetics or solubility vary widely due to variations in the contaminant profiles in the chemical feedstocks, it is a good idea to measure the solubility curve and the metastable limit for every new batch to determine the desired supersaturation profile. Such measurements can be done with little effort with software that fully automates the experimental procedure (Zhou et al., 2006).

Although many of the observations in this chapter were obtained for a particular system, these observations are expected to hold for other systems given the inherent ability of concentration control to correct for most disturbances, and the inherent inability of direct operation to suppress the effects of most disturbances. While an experienced control or crystallization expert could have postulated the observations in this section by using intuition, the analysis of a specific system provides some

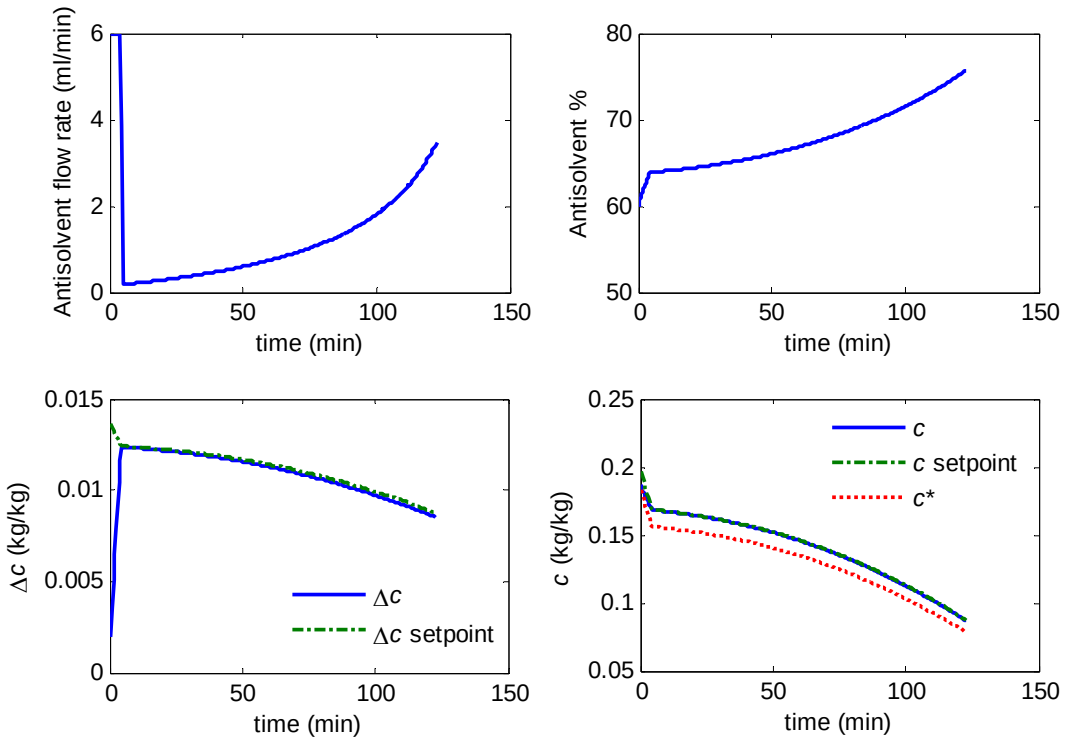

Figure 7.6 Antisolvent flow rate, Antisolvent %, supersaturation and solute concentration as a function time by the concentration control approach following the supersaturation profile for Case C shown in Figure 1. Sampling time = 30 seconds. Maximum flow rate = 6 ml/min.

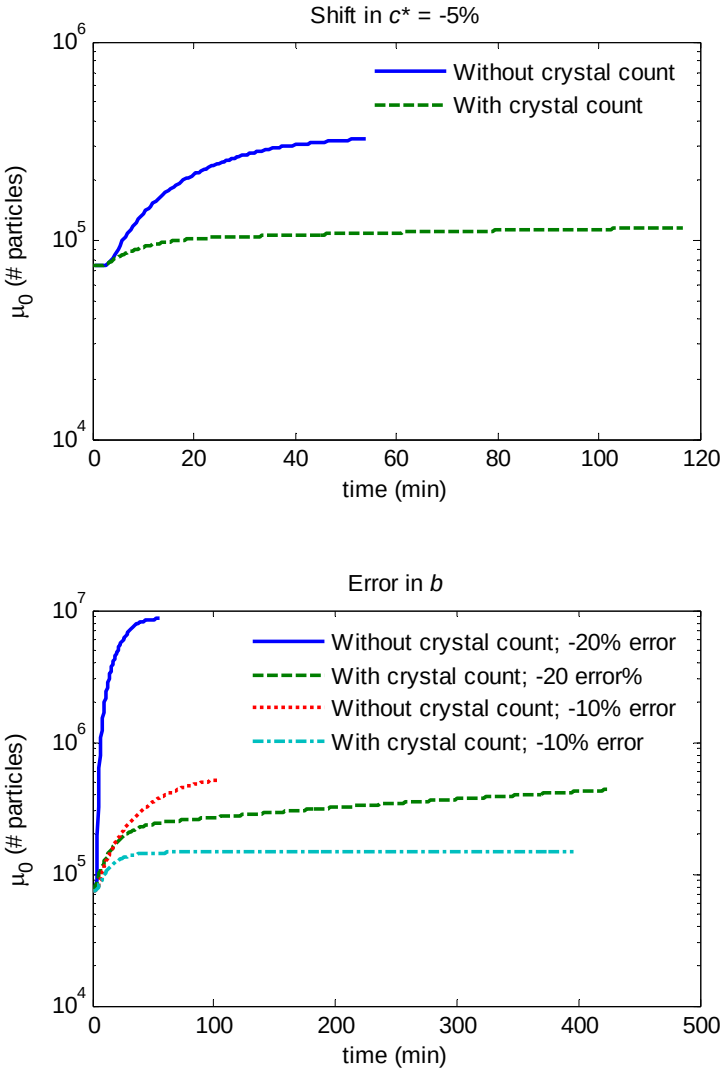

Figure 7.7 Variation of number of particles with time using the concentration control approach with disturbances that causes nucleation, both with and without using crystal count measurement. FBRM measurement time = 5 seconds.

Table 7.2 Sensitivity of direct operation and concentration control to disturbances according to supersaturation profile in Case C.

Disturbance		Relative error (%) Number-mean size		Weight-mean size		Yield		Batch time for C-control (h)
		Direct operation	C-control	Direct operation	C-control	Direct operation	C-control	
Shift in solubility curve	-5%	-17.15	-42.218	-12.63	-23.7589	5.88	0.55	0.9
	+5%	-15.69	15.0594	-6.86	2.0294	-5.57	-0.1333	8.1083
Error in antisolvent flow rate	-5%	-0.61	0.1564	-1.00	0.1335	-2.97	0.2773	2.05
	+5%	0.38	0.0597	0.79	0.1234	2.80	0.3278	2.0417
Variation in initial mass of antisolvent	+5%	-0.68	0.2939	-1.05	0.1025	3.09	0.1435	2.025
Variation in initial mass of solvent	-5%	-35.56	5.2884	-30.11	4.9384	8.55	0.1106	2.15
Evaporation of solvent (organic)	4 g/h	-2.72	7.0586	4.75	7.683	10.00	-0.1708	2.225
	6 g/h	-7.40	4.6532	3.12	8.1689	14.85	0.1387	2.15
Error in k_g	-20%	-16.59	-3.2831	-4.02	-0.5837	-1.50	0.0249	2.525
	-10%	-7.10	-1.4649	-1.67	-0.2163	-0.68	0.1213	2.2583
	+10%	5.16	1.3209	1.24	0.2626	0.57	0.0916	1.8667
	+20%	8.88	2.3381	2.20	0.3404	1.05	-0.17	1.7167
Error in g	-20%	24.68	15.8285	9.43	4.367	6.15	0.3576	0.5833
	-10%	20.29	8.1858	6.39	1.3745	3.69	-0.2003	1.0333
	+10%	-52.71	-12.1326	-19.33	-2.3829	-5.10	-0.017	4.0917
	+20%	-74.27	-27.0125	-47.54	-6.586	-10.56	-0.1873	8.2083
Error in k_b	-20%	2.69	2.5645	0.44	0.3267	-0.02	-0.2304	2.0417
	-10%	1.31	1.257	0.22	0.1628	-0.01	-0.1152	2.0417
	+10%	-1.26	-1.2093	-0.22	-0.1618	0.01	0.1151	2.0417
	+20%	-2.46	-2.3736	-0.43	-0.3225	0.02	0.2302	2.0417
Error in b	-20%	-74.37	-79.0073	-73.20	-75.0598	5.25	0.1636	0.9167
	-10%	-46.56	-53.0406	-31.70	-32.818	2.09	-0.2322	1.7083
	+10%	15.58	14.9769	2.21	2.1859	-0.12	0.097	2.0583
	+20%	16.54	15.8636	2.25	2.235	-0.18	0.0677	2.0583
		Include crystal count measurement for Direct Design						
Shift in solubility curve	-5%	-17.15	-3.6881	-12.63	-2.577	5.88	-0.0139	1.9458
Error in b	-20%	-74.37	-46.3718	-73.20	-29.4008	5.25	-0.0283	7.0639
	-10%	-46.56	-11.1888	-31.70	-10.4301	2.09	-0.0431	6.6056

Direct operation: batch time = 2 h. Concentration control: target yield = 53%; maximum batch time = 10 h; concentration sampling time = 30 s. Concentration control with crystal count: crystal count sampling time = 5 s; once the crystal count is detected to be greater than 7.7×10^4 in less than 12 min, the supersaturation setpoint is set to 40% of the original supersaturation profile and the antisolvent flowrate is set to zero.

estimates of the relative magnitude of the effects of disturbances on the two approaches to operating antisolvent crystallizers, and confirms the relative effectiveness of the use of total count measurement to reducing the sensitivity to disturbances that induce unexpected excessive nucleation. Given that concentration control has already been implemented on antisolvent crystallizers in industry (Zhou et al., 2006), the results indicate that the small additional step of including total count measurement would lead to especially robust industrial implementations of crystallizer control.

7.6 Conclusions

The analysis presented in this chapter shows that the combination of concentration (supersaturation) control with total count measurement provides low sensitivity to process disturbances and variations in solubility and nucleation and growth kinetics, whereas the direct specification of the antisolvent addition rate in batch recipe is inherently sensitive to many disturbances. The batch time is not fixed for concentration control since the rate at which the process moves along the setpoint trajectory is determined by the crystallization kinetics. Although this might result in a change in how the manufacturing process is scheduled, variability in the production time is certainly preferable over variability in product quality in pharmaceutical production. Since it has been shown that an optimal or nearly optimal supersaturation setpoint trajectory can be determined with an automated experimental system (Zhou et al., 2006), the tracking of such a setpoint using the concentration control strategy eliminates the need to develop highly accurate first-principles models by in-situ measurement of the concentrations and particle size distributions. Hence the proposed

approach can significantly reduce the time required for developing a new antisolvent crystallization process.

7.7 References

Barrett, P., Smith, B., Worlitschek, J., Bracken, V., O'Sullivan, B. and O'Grady, D. (2005). A review of the use of process analytical technology for the understanding and optimization of production batch crystallization processes. *Organic Process Research & Development* **9**(3): 348-355.

Birch, M., Fussell, S. J., Higginson, P. D., McDowall, N. and Marziano, I. (2005). Towards a PAT-based strategy for crystallization development. *Organic Process Research & Development* **9**(3): 360-364.

Braatz, R. D. (2002). Advanced control of crystallization processes. *Annual Reviews in Control* **26**(1): 87-99.

Braatz, R. D., Fujiwara, M., Ma, D. L., Togkalidou, T. and Tafti, D. K. (2002). Simulation and new sensor technologies for industrial crystallization: A review. *International Journal of Modern Physics B* **16**(1-2): 346-353.

Chung, S. H., Ma, D. L. and Braatz, R. D. (1999). Optimal seeding in batch crystallization. *Canadian Journal of Chemical Engineering* **77**(3): 590-596.

Chung, S. H., Ma, D. L. and Braatz, R. D. (2000). Optimal model-based experimental design in batch crystallization. *Chemometrics and Intelligent Laboratory Systems* **50**(1): 83-90.

Fujiwara, M., Chow, P. S., Ma, D. L. and Braatz, R. D. (2002). Paracetamol crystallization using laser backscattering and ATR-FTIR spectroscopy: Metastability, agglomeration, and control. *Crystal Growth & Design* **2**(5): 363-370.

Fujiwara, M., Nagy, Z. K., Chew, J. W. and Braatz, R. D. (2005). First-principles and direct design approaches for the control of pharmaceutical crystallization. *Journal of Process Control* **15**(5): 493-504.

Granberg, R. A., Bloch, D. G. and Rasmuson, A. C. (1999). Crystallization of paracetamol in acetone-water mixtures. *Journal of Crystal Growth* **199**: 1287-1293.

Granberg, R. A. and Rasmuson, A. C. (2000). Solubility of paracetamol in binary and ternary mixtures of water plus acetone plus toluene. *Journal of Chemical and Engineering Data* **45**(3): 478-483.

Gron, H., Borissova, A. and Roberts, K. J. (2003). In-process ATR-FTIR spectroscopy for closed-loop supersaturation control of a batch crystallizer producing monosodium glutamate crystals of defined size. *Industrial & Engineering Chemistry Research* **42**(1): 198-206.

Gunawan, R., Ma, D. L., Fujiwara, M. and Braatz, R. D. (2002). Identification of kinetic parameters in multidimensional crystallization processes. *International Journal of Modern Physics B* **16**(1-2): 367-374.

Jones, A. G. and Mullin, J. W. (1974). Programmed cooling crystallization of potassium sulphate solutions. *Chemical Engineering Science* **29**(1): 105.

Larsen, P. A., Patience, D. B. and Rawlings, J. B. (2006). Industrial crystallization process control. *IEEE Control Systems Magazine* **26**(4): 70-80.

Liotta, V. and Sabesan, V. (2004). Monitoring and feedback control of supersaturation using ATR-FTIR to produce an active pharmaceutical ingredient of a desired crystal size. *Organic Process Research & Development* **8**(3): 488-494.

Ma, D. L. and Braatz, R. D. (2003). Robust identification and control of batch processes. *Computers & Chemical Engineering* **27**(8-9): 1175-1184.

Ma, D. L., Chung, S. H. and Braatz, R. D. (1999). Worst-case performance analysis of optimal batch control trajectories. *AIChE Journal* **45**(7): 1469-1476.

Ma, D. L., Tafti, D. K. and Braatz, R. D. (2002). Optimal control and simulation of multidimensional crystallization processes. *Computers & Chemical Engineering* **26**(7-8): 1103-1116.

Mullin, J. W. and Nyvlt, J. (1971). Programmed cooling of batch crystallizers. *Chemical Engineering Science* **26**(3): 369.

Nonoyama, N., Hanaki, K. and Yabuki, Y. (2006). Constant supersaturation control of antisolvent-addition batch crystallization. *Organic Process Research & Development* **10**(4): 727-732.

Paul, E. L., Tung, H. H. and Midler, M. (2005). Organic crystallization processes. *Powder Technology* **150**(2): 133-143.

Sarkar, D., Rohani, S. and Jutan, A. (2006). Multi-objective optimization of seeded batch crystallization processes. *Chemical Engineering Science* **61**(16): 5282-5295.

Togkalidou, T., Tung, H. H., Sun, Y., Andrews, A. T. and Braatz, R. D. (2004). Parameter estimation and optimization of a loosely bound aggregating pharmaceutical crystallization using in situ infrared and laser backscattering measurements. *Industrial & Engineering Chemistry Research* **43**(19): 6168-6181.

Ward, J. D., Mellichamp, D. A. and Doherty, M. F. (2006). Choosing an operating policy for seeded batch crystallization. *AIChE Journal* **52**(6): 2046-2054.

Worlitschek, J. and Mazzotti, M. (2004). Model-based optimization of particle size distribution in batch-cooling crystallization of paracetamol. *Crystal Growth & Design* **4**(5): 891-903.

Yu, L. X., Lionberger, R. A., Raw, A. S., D'Costa, R., Wu, H. Q. and Hussain, A. S. (2004). Applications of process analytical technology to crystallization processes. *Advanced Drug Delivery Reviews* **56**(3): 349-369.

Yu, Z. Q., Chow, P. S. and Tan, R. B. H. (2006). Application of attenuated total reflectance-Fourier transform infrared (ATR-FTIR) technique in the monitoring and control of anti-solvent crystallization. *Industrial & Engineering Chemistry Research* **45**(1): 438-444.

Zhou, G. X., Fujiwara, M., Woo, X. Y., Rusli, E., Tung, H. H., Starbuck, C., Davidson, O., Ge, Z. H. and Braatz, R. D. (2006). Direct design of pharmaceutical antisolvent crystallization through concentration control. *Crystal Growth & Design* **6**(4): 892-898.

Chapter 8
Precise Tailoring of the Crystal Size Distribution By Optimal Control of Impinging Jet Crystallizers

8.1 Introduction

The control of crystal size distribution (CSD) in a batch or continuous crystallization process is critical for efficient downstream processing (Braatz, 2002; Kim et al., 2005), and meeting company internal and governmental regulatory demands for product consistency (Paul et al., 2005). In addition, the desired bioavailability and the method of drug administration and delivery, for example pulmonary delivery, can demand stringent control on the crystal size distribution (Nagao et al., 2005; Rasenack et al., 2003; Shekunov and York, 2000).

The control of a batch crystallization process typically involves following a pre-determined optimal cooling or antisolvent addition trajectory (Ma et al., 2002; Rawlings et al., 1993; Rohani et al., 2005b; Togkalidou et al., 2004; Worlitschek and Mazzotti, 2004), or using feedback control, with on-line monitoring, to maintain a constant supersaturation during the crystallization process (Fujiwara et al., 2002; Gron et al., 2003; Liotta and Sabesan, 2004; Zhou et al., 2006). A review of these control strategies can be found in recently published literature (Fujiwara et al., 2005; Ward et al., 2006). The mass, size distribution, and addition time of the seed crystals, have a significant impact on the final crystal size distribution such that the quantity and distribution of the seed crystals can be optimized for a given product quality (Beckmann, 2000; Choong and Smith, 2004; Chung et al., 1999; Jagadesh et al.,

1999; Jagadesh et al., 1996; Kubota et al., 2002; Kwon and Kim, 2004; Lung-Somarriba et al., 2004; Rohani et al., 2005a; Togkalidou et al., 2004). Usually, these control and optimization strategies are targeted towards the production of large crystals of narrow unimodal size distribution for efficient downstream processing.

The large crystals typically produced from batch crystallizers require further milling to increase the surface area to meet dissolution, tableting, and bioavailability requirements. The undesirable effects from milling (am Ende and Brenek, 2004; Bauer-Brandl, 1996) motivates the development of crystallization processes to directly produce small particles with narrow distribution. Current state-of-the-art includes impinging jet crystallization (am Ende et al., 2003; Dauer et al., 1996; Johnson and Prud'homme, 2003; Lindrud et al., 2001; Marchisio et al., 2006; Midler et al., 1994), and crystallization using supercritical fluids (Jung and Perrut, 2001; Muhrer et al., 2003; Pasquali et al., 2006; Shekunov and York, 2000). The development of these processes to their full potential requires thorough understanding of the processes, which can be achieved by modeling (Henczka et al., 2005; Martin and Cocero, 2004) and experimentation combined with imaging, light scattering, and other measurement techniques (Bell et al., 2005). This chapter will show how to use this understanding to develop an optimal control strategy to precisely tailor the crystal size distribution specific to the bioavailability requirements of an active pharmaceutical ingredient (API).

This chapter proposes three control strategies whose goal is to produce crystals with a *target crystal size distribution* (CSD), that are combinations of optimal control and an impinging jet crystallizer. Analysis of the first control strategy indicates that the CSDs

obtainable by an impinging jet crystallizer by itself are limited. However, the other two control strategies that combine the impinging jet with an aging vessel controlled at constant supersaturation show greatly enhanced controllability of the CSD compared to past academic studies and current industrial practice. Simulation results for the Lovastatin system indicate that even square and multimodal distributions can be obtained, with selected size ranges and distribution peaks and widths. Sensitivity and controllability analyses indicate a high controllability of the CSD with moderate to low sensitivity to disturbances.

8.2 Trying to Tailor CSD by Combining Crystals Produced at Different Jet Velocities

Based on the simulation studies in Chapter 6, the crystal size distributions obtained from confined impinging jets are shown in Figure 8.1. Figure 8.2 shows the increase in the mean crystal size and distribution width with decrease in jet velocity.

8.2.1 Optimization Formulation

The crystal size distributions obtained by varying the jet velocity of a single impinging jet or multiple impinging jets in parallel can be combined to tailor the crystal size distribution. The proportions of crystals produced at each jet velocity can be obtained by solving the optimization:

$$\min_{w_i \geq 0} \left\| F(r) - \sum_i w_i f_i(r) \right\| = \min_{w_i \geq 0} \int_0^\infty \left[F(r) - \sum_i w_i f_i(r) \right]^2 \mathrm{d}r \qquad (8.1)$$

where $F(r)$ is the target crystal size distribution, $f_i(r)$ is the crystal size distribution for a given jet velocity v_i, w_i is the weight of the corresponding crystals, and $\|\cdot\|$ is an

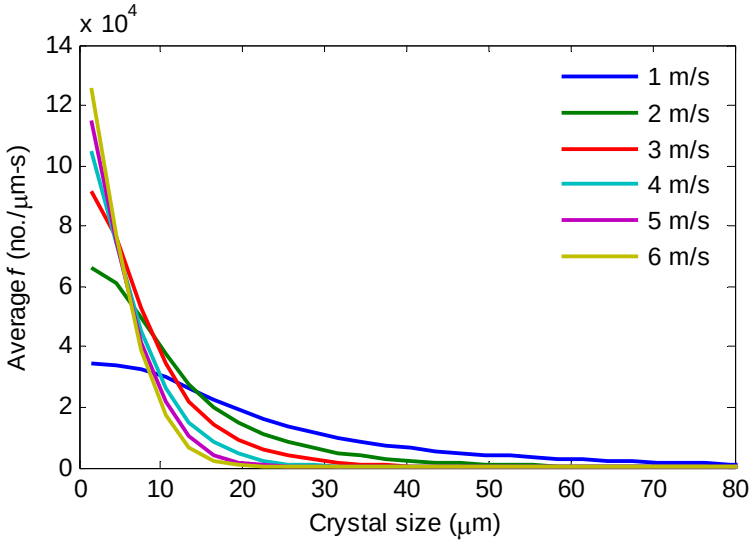

Figure 8.1 Simulated crystal size distribution of Lovastatin from confined impinging jets.

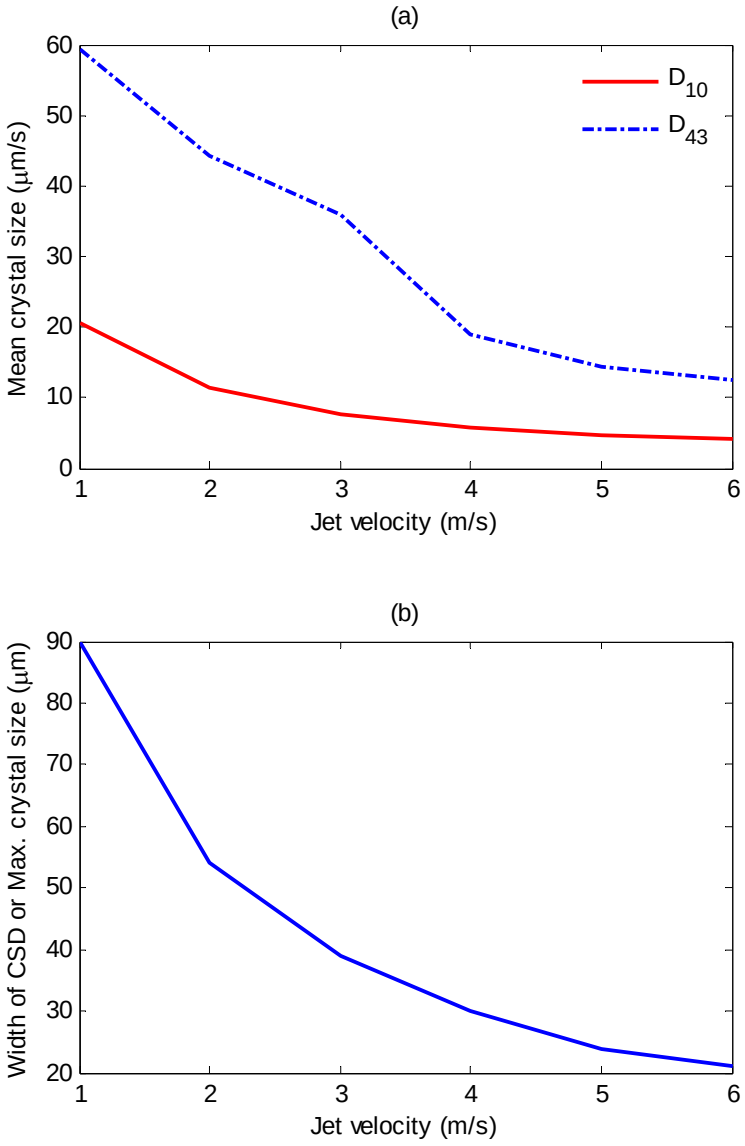

Figure 8.2 Effect of impinging jet velocity on (a) the mean crystal size and (b) the width of the distribution (equals to maximum crystal size since minimum crystal size is 0 μm). The maximum crystal size is defined as the crystal size when f/μ_0 becomes $<1\times10^{-3}$.

appropriately defined norm. The weight w_i is proportional to the time running the impinging jet at velocity v_i, for operations in which the temperature is constant. In this formulation, the set of jet velocities are fixed while the weights of the crystal size distribution corresponding to each jet velocity are optimized. A nearly continuous range of jet velocity can be considered by including a very large number of velocities in the formulation.

Equation (8.1) can be further expressed as

$$\min_{w_i \geq 0} \left\{ \int_0^\infty \left[F(r)\right]^2 \mathrm{d}r - 2\sum_i w_i \int_0^\infty f_i(r)F(r)\mathrm{d}r + \sum_i \sum_j w_i w_j \int_0^\infty f_i(r)\, f_j(r)\mathrm{d}r \right\} \tag{8.2}$$

which can be solved with a quadratic programming solver (e.g., “quadprog” in Matlab) that can be initialized by the unconstrained solution of (8.2):

$$\begin{bmatrix} w_1 \\ \vdots \\ w_{i,\max} \end{bmatrix} = \begin{pmatrix} \int_0^\infty f_1(r)\, f_1(r)\mathrm{d}r & \cdots & \int_0^\infty f_1(r)\, f_{i,\max}(r)\mathrm{d}r \\ \vdots & \ddots & \vdots \\ \int_0^\infty f_{i,\max}(r)\, f_1(r)\mathrm{d}r & \cdots & \int_0^\infty f_{i,\max}(r)\, f_{i,\max}(r)\mathrm{d}r \end{pmatrix}^{-1} \begin{bmatrix} \int_0^\infty f_1(r)F(r)\mathrm{d}r \\ \vdots \\ \int_0^\infty f_{i,\max}(r)F(r)\mathrm{d}r \end{bmatrix} \tag{8.3}$$

8.2.1 Obtainable Crystal Size Distributions

Figure 8.3 shows the crystal size distributions obtained based on two different target distributions and the respective weights of crystals for six different jet velocities. While one target CSD could be obtained approximately by combining crystals from operating the impinging jet at 2 and 3 m/s, the second target CSD could not be obtained. To gain some insight into this, the CSDs obtainable by combining crystals from an impinging jet operating at the six different velocities are quantified in Figure

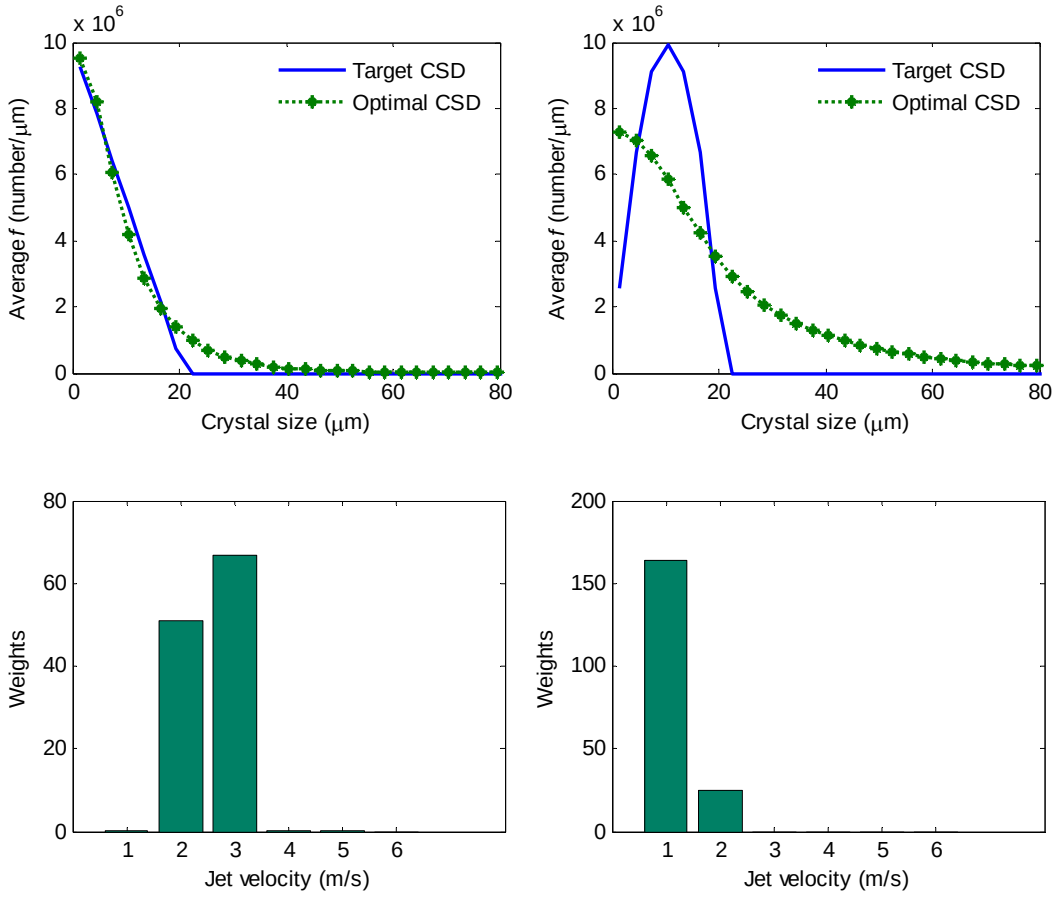

Figure 8.3 Crystal size distributions obtained by combining crystals obtained from impinging jets operating at various jet velocities. The bottom plots are the corresponding weights of crystals from the impinging jets operating at different jet velocities required to achieve the optimal CSD directly above.

8.4. The maximum crystal size, or width of the distribution, is specified by the crystals obtained at the lowest jet velocity. From Figure 8.1 it is seen that this holds regardless of how many different inlet velocities are used within the range of operation (1-6 m/s).

Another general observation is that the shape of the overall CSD is fixed to be monotonically decreasing since all of the CSDs produced by the impinging jet are monotonically decreasing. Combining crystals from different inlet velocities does not give a significantly larger variety of CSDs than obtainable by operating the impinging jet at one inlet velocity (see Figure 8.4b). The second target CSD in Figure 8.3b is not monotonically decreasing and so cannot be obtained by combining crystals from different inlet velocities. Although this study utilized CSDs from a simulation model, experimental studies also have reported similar monotonically decreasing CSDs for the full range of inlet velocities (Mahajan and Kirwan, 1996), in which case there is little control over the target CSD by combining crystals at different inlet velocities.

8.3 Tailoring CSD by Optimal Seeding into an Aging Vessel

An impinging jet crystallizer can be used to tailor the crystal size distribution by sending the crystals to an aging vessel to grow.

8.3.1 Optimization Formulation

In this approach, crystals of the narrowest distribution (jet velocity = 6 m/s) from the impinging jet are quenched, filtered, and dried. Subsequently, various weights of crystals are dropped into an aging vessel for further crystal growth for various times.

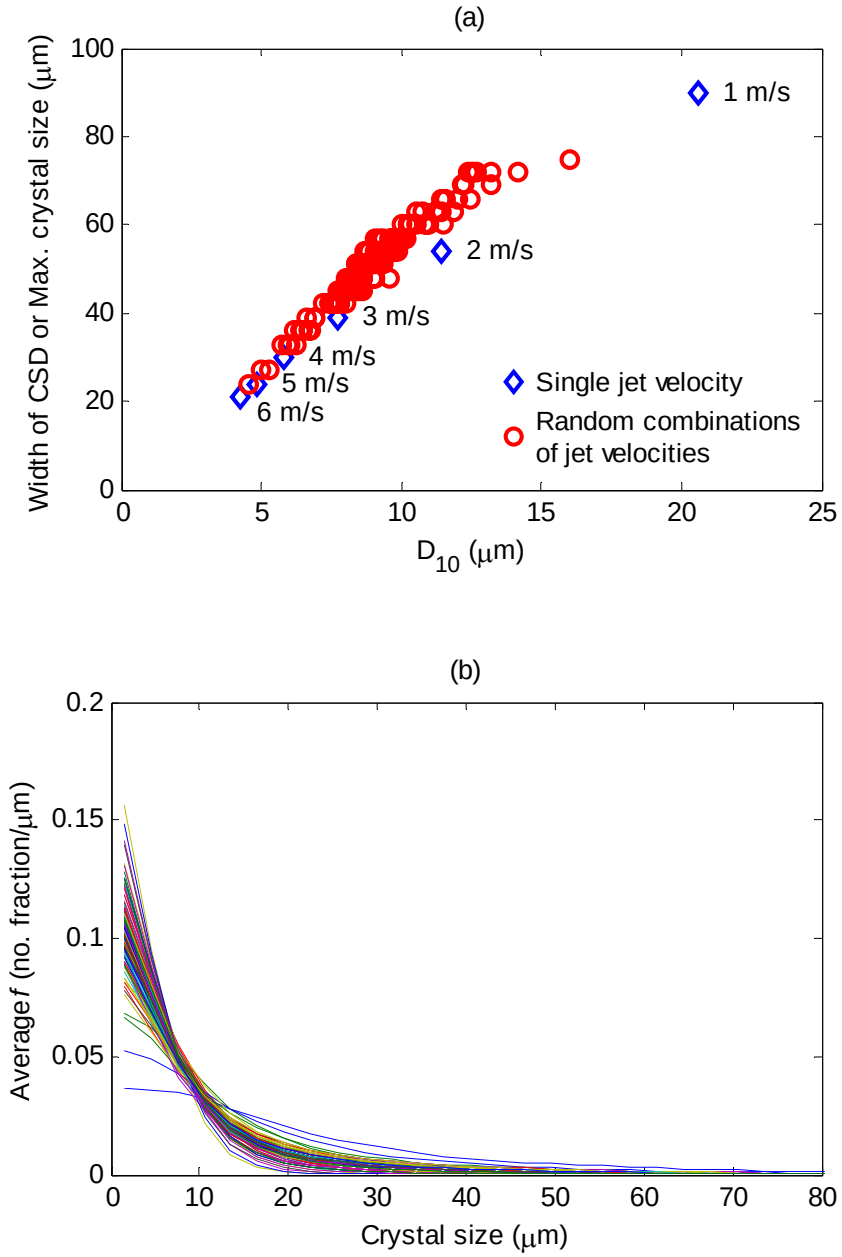

Figure 8.4 (a) Widths of crystal size distribution and mean crystal sizes, and **(b)** crystal size distributions obtainable from single jet velocities or by randomly combining crystals from impinging jets operating at jet velocities between 1 m/s to 6 m/s shown in Figure 8.1.

In other words, this strategy optimizes a seeding time profile based on the target distribution, using seeds of narrow distribution, for the operation of a batch crystallizer confined to operate under constant growth conditions:

$$\min_{\substack{w_i \geq 0 \\ t_{grow,i} \geq 0}} \left\| F(r) - \sum_i w_i f_{grow,i}(r) \right\| = \min_{\substack{w_i \geq 0 \\ t_{grow,i} \geq 0}} \left\| F(r) - \sum_i w_i f(r - Gt_{grow,i}) \right\| \tag{8.4}$$

where $f_{grow,i}$ is the crystal size distribution of the crystals after growing for the time interval $t_{grow,i}$. The time to drop in the crystals, $t_{drop,i}$, of weight w_i can be determined from the total batch time for the crystallizer, t_{batch}:

$$t_{drop,i} = t_{batch} - t_{grow,i} \tag{8.5}$$

The minimum batch time is given by $t_{batch} = \max_i \{t_{grow,i}\}$.

To simplify the formulation, a constant growth rate, *G*, was used, which can be easily achieved by constant supersaturation control (Fujiwara et al., 2005; Zhou et al., 2006), which is optimal or nearly optimal for most crystallizers (Jones and Mullin, 1974). In this case the term, $f(r - Gt_{grow,i})$, can be evaluated by shifting the crystal size distribution along the crystal size axis (that is, the method of characteristics (Morton and Mayers, 1994)). In cases where secondary nucleation, agglomeration, and/or breakage cannot be avoided, a full population balance model that includes the relevant kinetics can be used to determine $f_{grow,i}$. Note that judicious control and adjustment of the supersaturation and agitation rates, as well as choice of solvents, can reduce the extent of secondary nucleation, agglomeration, and breakage (Fujiwara et al., 2002).

Equation (8.4) is similar to the optimization formulation frequently used in the deconvolution of peaks in chromatograms (Steffen et al., 2005; Torres-Lapasio et al.,

1997; Vivo-Truyols et al., 2005), which can be solved using any nonlinear least-squares optimization solver (e.g. "lsqonlin" in Matlab). Thus, equation (8.4) can be re-written for least-squares optimization as follows

$$\min_{\substack{w_i \geq 0 \\ t_{grow,i} \geq 0}} \sum_{j=1}^{N} \left[F(r_j) - \sum_i w_i f_{grow,i}(r_j) \right]^2 \tag{8.6}$$

where $F(r_j)$ and $f_{grow,i}(r_j)$ are the cell-averaged population density discretized along the growth axis of the crystal. The nonlinear solver can be initialized by the unconstrained solution of (8.3) with the approximation $f(r) = \alpha\delta(r)$, where $\delta(r)$ is the Dirac delta function and α is a scalar:

$$\begin{aligned}
&\min_{w_i} \left\| F(r) - \sum_i w_i \alpha \delta(r - Gt_{grow,i}) \right\| \\
&= \min_{w_i} \int_0^\infty \left[F(r) - \sum_i w_i \alpha \delta(r - Gt_{grow,i}) \right]^2 \mathrm{d}r \\
&= \min_{w_i} \left\{ \int_0^\infty \left[F(r) \right]^2 \mathrm{d}r - 2\alpha \sum_i w_i \int_0^\infty \delta(r - Gt_{grow,i}) F(r) \mathrm{d}r \right. \\
&\qquad \left. + \alpha^2 \sum_i \sum_j w_i w_j \int_0^\infty \delta(r - Gt_{grow,i}) \delta(r - Gt_{grow,j}) \mathrm{d}r \right\} \\
&= \min_{w_i} \left\{ \int_0^\infty \left[F(r) \right]^2 \mathrm{d}r - 2\alpha \sum_i w_i F(Gt_{grow,i}) + \alpha^2 \sum_i w_i^2 \right\}
\end{aligned} \tag{8.7}$$

$$\begin{aligned}
&\frac{d}{dw_j} \left\{ \int_0^\infty \left[F(r) \right]^2 \mathrm{d}r - 2\alpha \sum_i w_i F(Gt_{grow,i}) + \alpha^2 \sum_i w_i^2 \right\} = 0 \\
&\Rightarrow \quad -2\alpha F(Gt_{grow,i}) + 2\alpha^2 w_i = 0
\end{aligned} \tag{8.8}$$

$$\Rightarrow \quad w_i = (1/\alpha) F(Gt_{grow,i}) = (1/\alpha) F(G(t_{batch} - t_{drop,i})) \tag{8.9}$$

Hence a plot of the optimal weights w_i as a function of the drop time should look like a "flipped" version of the target crystal size distribution. In addition, $t_{grow,i}$ can be initialized by:

$$t_{grow,i} = r_i / G \quad \text{and} \quad r_i = \frac{i-1}{I-1}(r_{\max} - r_{\min}) + r_{\min} \quad \text{for } i = 1, 2, \ldots, I \tag{8.10}$$

where I is the number of times to drop in the seed crystals, and $r_{\min}$ and $r_{\max}$ are the minimum and maximum crystal size of the target distribution, $F(r)$.

8.3.2 Obtainable Crystal Size Distributions

Figure 8.5 shows the optimal crystal size distribution and times for dropping in the crystals with the corresponding weights for two target distributions. The growth rate was 2 μm/min. Figure 8.5 illustrates how this method allows great flexibility in specifying the shape of the distribution in addition to the crystal size and the width of the distribution. This approach has the capability of tailoring any crystal size distribution of any shape, including flat-top and multi-modal distributions such as shown in Figure 8.5, as long as the target CSD does not have characteristics narrower than the narrowest CSD produced by the impinging jet. More specifically, the CSD based on the optimal seeding profile will deviate from that of the target CSD when the width of the target CSD is narrower that crystal size distribution from the impinging jets, and when there is a decrease in number density along the crystal size axis that is much sharper than that of any CSD produced by the impinging jet (see Figure 8.6 for examples, focusing on where the deviations between the target and optimal CSD occur).

While the CSD obtained on Figure 8.5b is satisfactory, the slight drop in the weights around 53 minutes is probably due to a local minimum. In addition, the recipe in

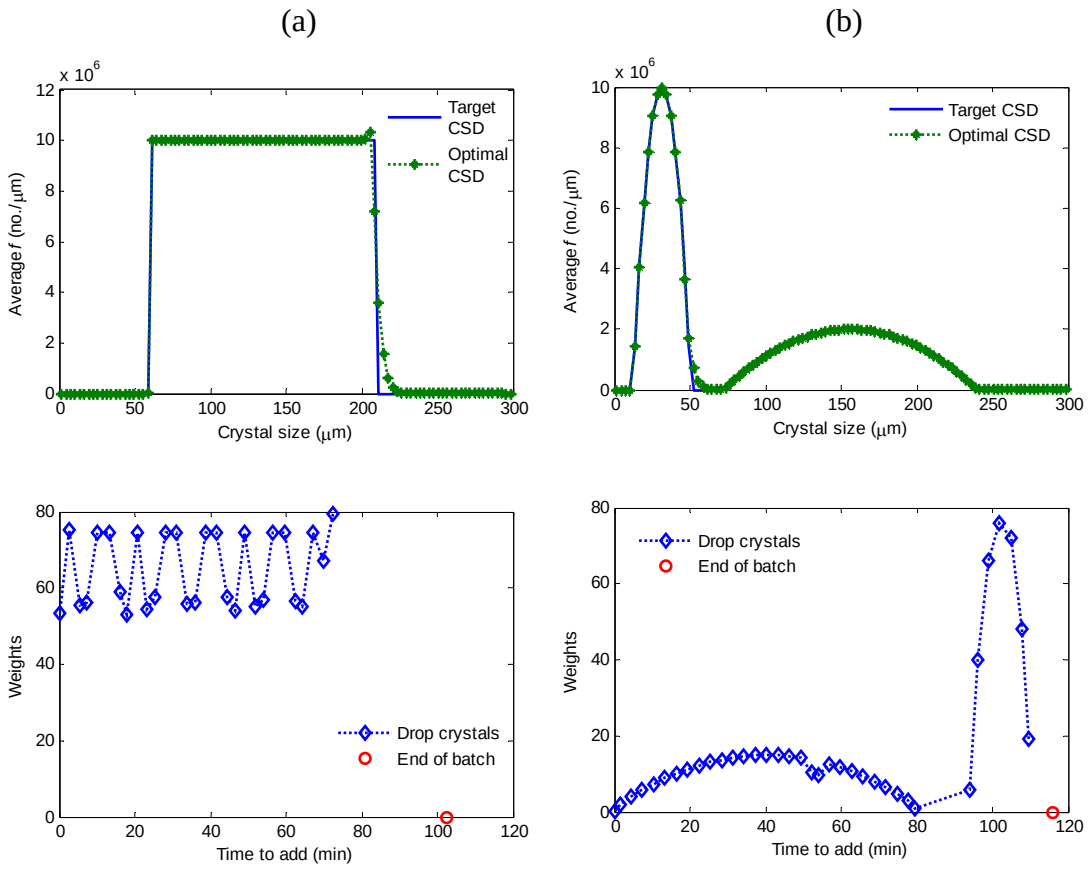

Figure 8.5 Crystal size distributions obtained by dropping crystals obtained from impinging jets into an aging vessel. Growth rate = 2 μm/min. The bottom plots are the corresponding seeding profile required to achieve the optimal CSD directly above.

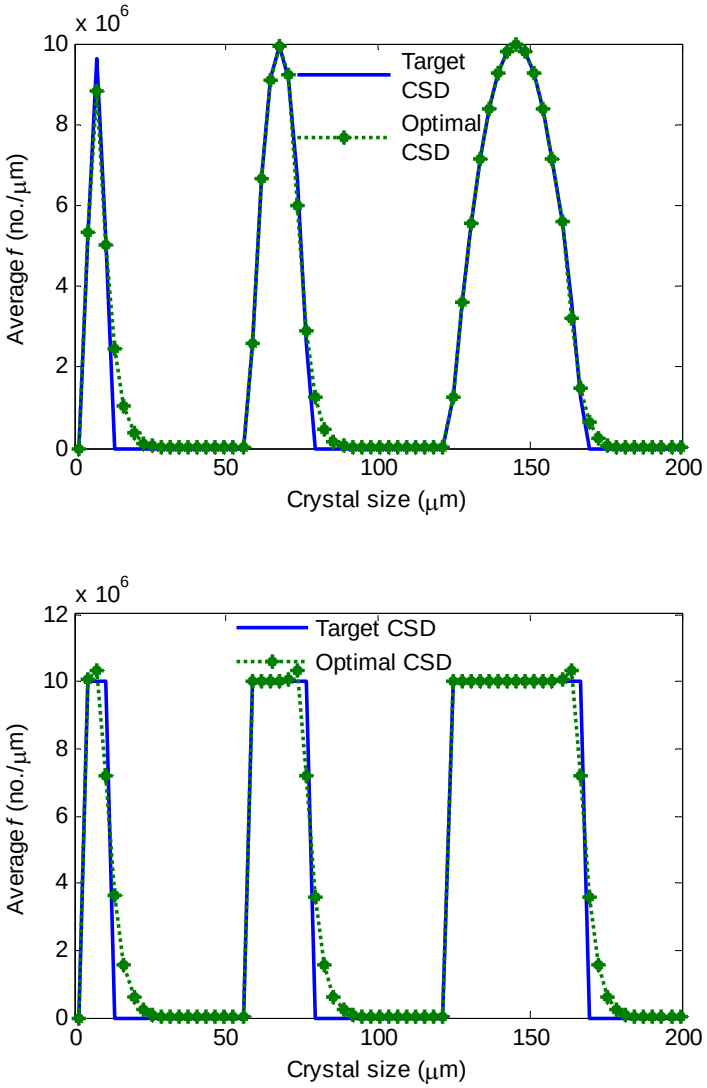

Figure 8.6 Crystal size distributions obtained for various target distribution width and shape by dropping crystals obtained from impinging jets into an aging vessel.

Figure 8.5a can be simplified by fixing a constant weight of seeds to be dropped at different times. These issues can be resolved by further improvements in the objective function and the solution of the optimization problem.

8.4 Tailoring CSD by Optimal Control of Jet Velocity

This section describes another strategy in tailoring the crystal size distribution by combining the operation of an impinging jet crystallizer with a batch aging vessel.

8.4.1 Optimization Formulation

In this approach, the crystals from the impinging jet crystallizer are quenched (to freeze the crystal size distribution) and directly sent to an aging vessel. Similar to the previous strategy, the supersaturation in the aging vessel is controlled such that growth rate remains constant. By varying the velocity from the impinging jets with time, the crystal size distributions that are added into the aging vessel varies with time, and can result in different product size distributions at the end of the aging run. Hence, this offers an opportunity to tailor the crystal size distribution towards a target distribution by optimizing the time-profile of crystal size distributions from the impinging jet crystallizer. In other words, this involves the optimal control of the jet velocity with time:

$$\min_{\substack{v_{jet}(t) \\ \text{Eq. (8.12)}}} \int_0^\infty \left[F(r) - f_{end}(r)\right]^2 \mathrm{d}r \tag{8.11}$$

where f_{end} is the crystal size distribution obtained at the end of the batch, which is evaluated by solving the population balance equation (PBE)

$$\frac{\mathrm{d}f(r,t)}{\mathrm{d}t} + G\frac{\mathrm{d}f(r,t)}{\mathrm{d}r} = f_{jet}(r,t;v_{jet}(t)) \tag{8.12}$$

where f_{jet} is the production rate of crystals from the impinging jet crystallizer in the units of #/μm-s. The distribution f_{jet} can be varied throughout the batch run by adjusting the jet velocity of the impinging jets. To illustrate the approach, the growth rate, *G*, in the batch aging vessel is held at a constant value by using supersaturation control, at a low enough supersaturation that secondary nucleation is negligible. It is also assumed that the mixing blade(s) and baffles are well-designed and the agitation intensity adjusted so that agglomeration and breakage in the batch aging vessel are negligible. Generalization of the approach to include secondary nucleation, agglomeration, and breakage increases the modeling and simulation requirements but is otherwise straightforward.

Equation (8.12) can be solved using any PBE-solver; high-resolution finite-volume methods have the advantage of being easy to implement, having both a low numerical diffusion and dispersion compared to other methods for discretizing the PBE, and in being applicable to other crystallization phenomena such as aggregation, secondary nucleation, and breakage. For $G > 0$, the semidiscrete central scheme (Kurganov and Tadmor, 2000) discretizes the population balance equation along the crystal growth axis as:

$$\frac{d}{dt} f_j(t) = -\frac{G}{\Delta r}\left\{\left[f_j(t) + \frac{\Delta r}{2}(f_r)_j(t)\right] - \left[f_{j-1}(t) + \frac{\Delta r}{2}(f_r)_{j-1}(t)\right]\right\} + f_{jet,j}(t)$$
$$(f_r)_j := \text{minmod}\left(\frac{f_j - f_{j-1}}{\Delta r}, \frac{f_{j+1} - f_{j-1}}{2\Delta r}, \frac{f_{j+1} - f_j}{\Delta r}\right) \tag{8.13}$$

where f_j is the cell-averaged population density, $f_{jet,j}$ is the cell-averaged population density from the impinging jet in the units of #/μm-s, and the minmod limiter is defined by

$$\text{minmod}(\alpha_1, \alpha_2, ...) = \begin{cases} \min\limits_i \{\alpha_i\} & \text{if } \alpha_i > 0 \; \forall i \\ \max\limits_i \{\alpha_i\} & \text{if } \alpha_i < 0 \; \forall i \\ 0 & \text{otherwise} \end{cases} \tag{8.14}$$

Equation (8.13) can be solved by any solver of ordinary differential equations (e.g., ode15s in Matlab 7.0.1).

The optimization (8.11), coupled with the population balance equation (8.12) defines the inlet velocity profile of the impinging jet crystallizer, $v_{jet}(t)$. To solve the optimization (8.11), $v_{jet}(t)$ was parameterized by treating the velocity as constant in variable time intervals with the following constraints:

$$\begin{gathered} v_{jet,min} \le v_{jet,j} \le v_{jet,max} \\ \Delta t_j \gg \text{time for impinging jets to reach steady state} \\ (\approx 10 \times \text{residence time of the impinging jets}) \end{gathered} \tag{8.15}$$

where $v_{jet,min}$ and $v_{jet,max}$ are the minimum and maximum jet velocities for the impinging jets. The constraint that f_{jet} is obtained from an impinging jet crystallizer running at steady-state simplifies the formulation while having an insignificant effect on the optimal solution to equation (8.11). This constraint is that the time interval should be long enough for f_{jet} to be nearly at its steady-state value throughout the time interval. The residence time for an impinging jet is in the order of 0.01 s, hence, setting the second constraint as $\Delta t_j \ge 1$ s is sufficient.

Instead of using discrete velocities as in the previous two formulations, here f_{jet} is fit to a continuous function of jet velocity, e.g., based on the distributions presented in Figure 8.1 or measured for an impinging jet for a range of inlet velocity. Note that

equation (8.11) is written on a number fraction basis (that is, normalizing the number density by the total number of particles, 0^{th} moment). The desired number of particles can be achieved by running multiple impinging jets in parallel or by changing the length of the run.

8.4.2 Obtainable Crystal Size Distributions

Here the above approach is applied to several target crystal size distributions. The growth rate used in the computations was 2 μm/min and the number of time intervals was 15. Figure 8.7 shows that the optimal control of jet velocity is capable of achieving target distributions that have gradual changes in the number density along the crystal size axis. A limitation in the shape of the target crystal size distribution is that the number density must smoothly decrease towards its minimum and maximum size. Similar to the previous optimal seeding approach, the width of the target distribution is limited by the width of the crystal size distribution from the impinging jet crystallizer (compare Figures 8.7a and 8.8).

Overall, the optimal control of impinging jet velocity in a combined impinging jet-batch aging vessel is effective in obtaining a much wider variety of crystal size distributions than for an impinging jet crystallizer on its own. The mean size of the crystals can be further increased by stopping the impinging jet crystallizer and allowing the crystals to grow in the aging vessel. Moreover, multi-modal distributions can be obtained by “switching off” the impinging jet crystallizer for a time period, which determines how far the modes are apart. The optimal velocity-time profile can

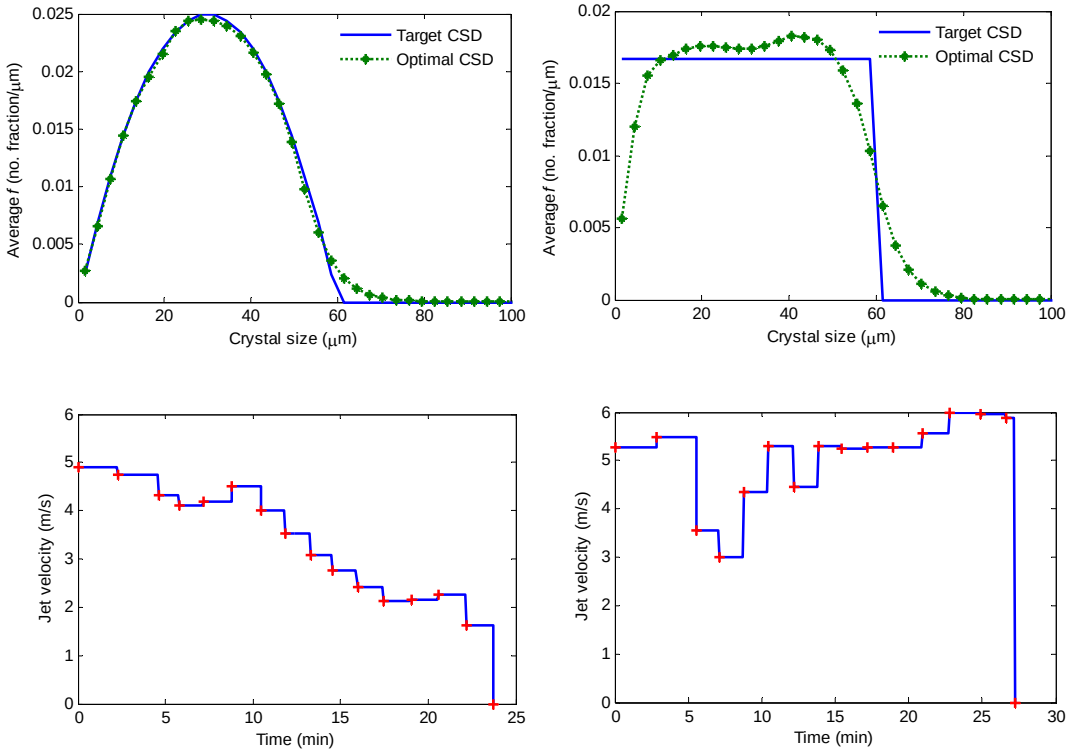

Figure 8.7 Crystal size distributions obtained by optimal control of impinging jet velocity followed by growth in aging vessel. Growth rate = 2 μm/min. The bottom plots are the corresponding jet velocity profile required to achieve the optimal CSD directly above.

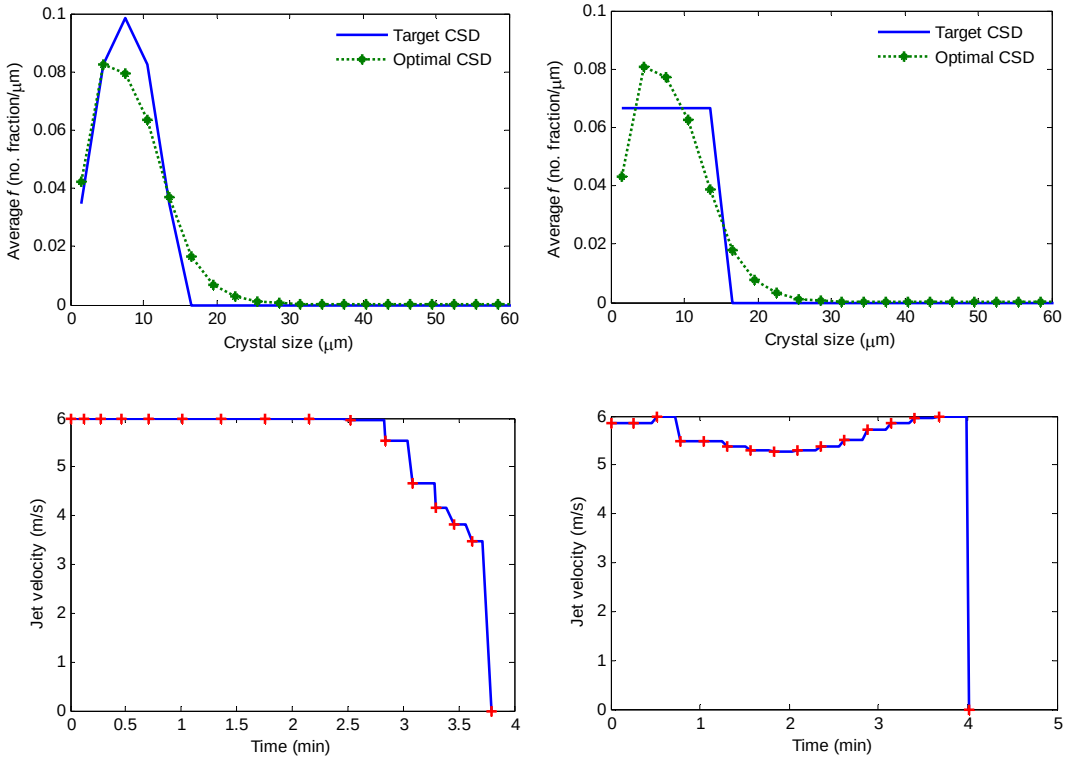

Figure 8.8 Crystal size distributions obtained by optimal control of impinging jet velocity followed by growth in aging vessel for narrow distributions. Growth rate = 2 μm/min. The bottom plots are the corresponding jet velocity profile required to achieve the optimal CSD directly above.

be computed using the above optimal control formulation for each individual mode by shifting the target crystal size distribution of each mode to start at 0 μm.

8.4.3 Controllability and Sensitivity Analysis

The approach of combining impinging jets with an aging vessel to achieve crystals of desired size is commonly used in the pharmaceutical industry. However, the manipulation of jet velocity to target a specific crystal size distribution (CSD), as discussed above, has yet to be implemented. In this section, we will study, in further detail, the controllability and robustness of this process. Figure 8.9 shows the variation of the CSD at the end of the batch of 30 minutes, where crystals from the impinging jet, operating at different jet velocities, are continuously added into the aging vessel with controlled constant growth rate of 2 μm/min. The plot illustrates that operating at different jet velocities does result in a range of size distributions at the end of the batch; given the dynamics of the CSD, these differences would increase as the batch time increases (and larger crystals are obtained). Hence there is significant controllability of the CSD obtained by varying the velocity of the impinging jets from low to high values.

As the CSD from the impinging jet converges at high velocities (see Chapter 6 and Figure 8.1), the distribution at the end of the batch converges to the same distribution at sufficiently high velocities. This indicates that operating at high jet velocity gives very low sensitivity to disturbances in the jet velocity, while having much less controllability. Lower jet velocities result in higher controllability of the crystal size

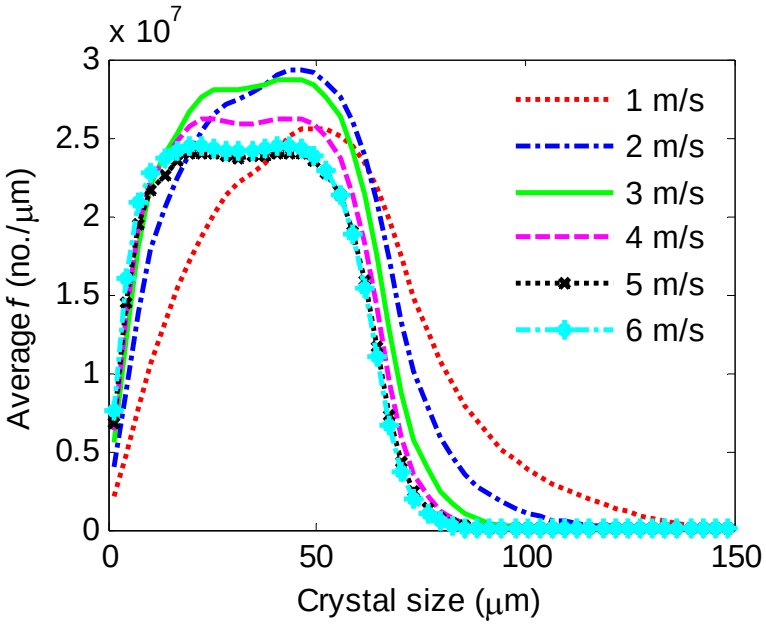

Figure 8.9 CSD at the end of the batch with continuous addition of crystals from the impinging jet, with constant jet velocity, into a supersaturation-controlled tank. Growth rate (in tank) = 2 micron/min, batch time = 30 min.

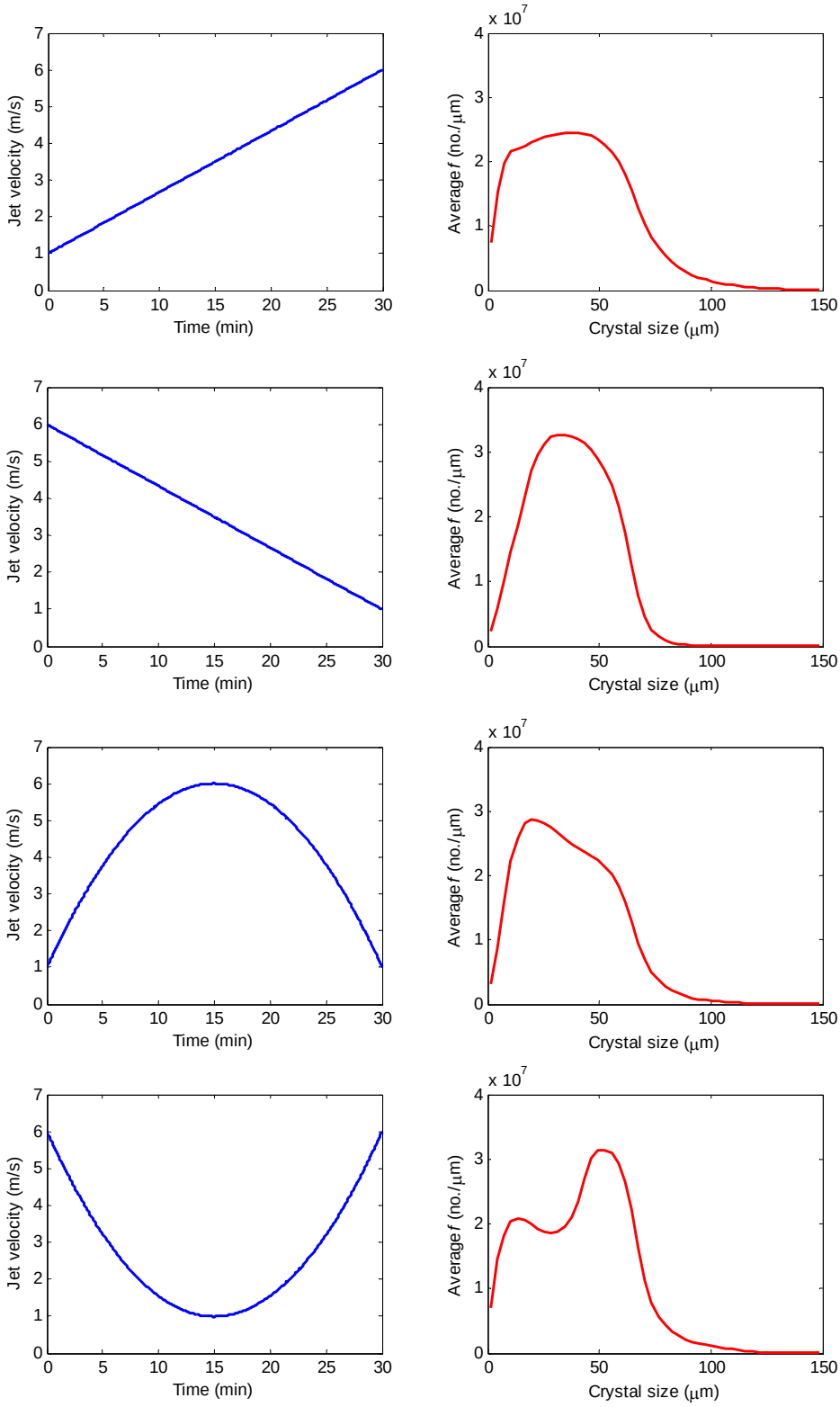

Figure 8.10 CSD at the end of the batch in the aging vessel for different jet velocity profiles of the impinging jet. Growth rate (in tank) = 2 micron/min, batch time = 30 min.

distribution, without having larger, but not excessively large, sensitivity to disturbances in jet velocity. To more fully understand the controllability of the process, various jet velocity profiles were applied to see their effects on the final CSD (see Figure 8.10). As illustrated, the shape of the distribution varies widely with the jet velocity profile. For example, a narrower nearly-parabolic CSD was obtained for a linearly decreasing profile (second plot in Figure 8.10), while a bimodal distribution was obtained for a convex parabolic profile in inlet velocity (fourth plot in Figure 8.10). Thus, it is possible to optimize the control of the jet velocity profile with time to produce crystals with a wide range in size distributions.

The sensitivity of the process to the variations in the supersaturation, hence growth rate, in the aging vessel and the jet velocity based on the optimal profile in Figure 8.7 (left) is shown in Figure 8.11. A slower growth rate results in a narrower size distribution, while a faster growth rate results in a wider size distribution. Note, however, that a 20% variation in growth rate correspond to much larger variations in supersaturation than reported in past studies that demonstrated the robust feedback control of concentration in batch crystallization (Fujiwara et al., 2002; Zhou et al., 2006). Since the solution concentration can be measured quite accurately and reproducibly (Fujiwara et al., 2002), such a large variation would be the result of a significant shift in the kinetics or solubility due to a shift in the contaminant concentrations in the chemical feedstocks. Variation in the solubility can be corrected with each new batch of chemical feedstocks by applying automated solubility measurement (Fujiwara et al., 2002; Liotta and Sabesan, 2004; Zhou et al., 2006) at the production site.

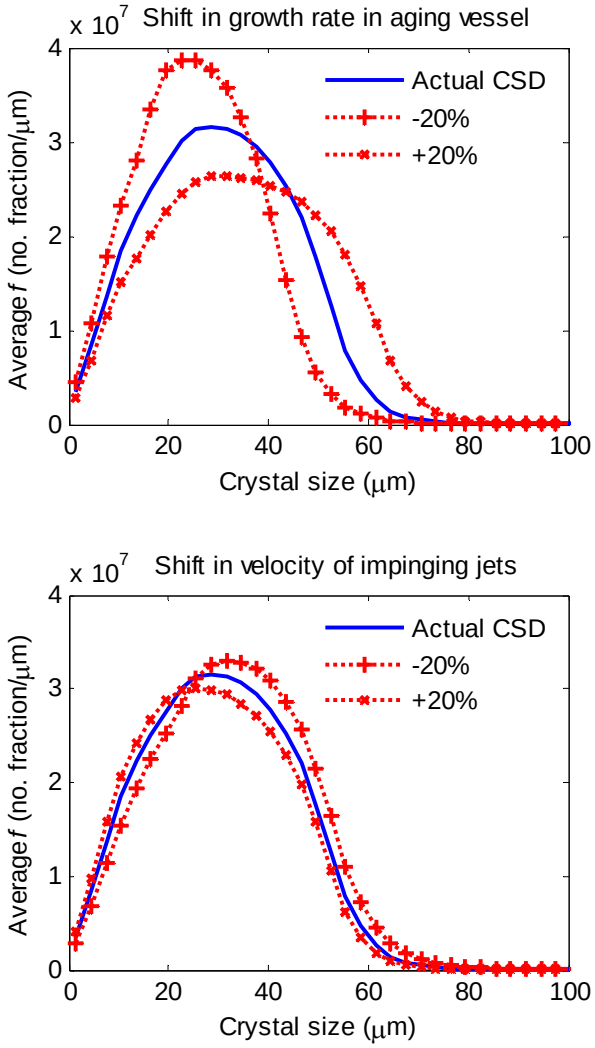

Figure 8.11 Change in CSD due to perturbations of the growth rate in the aging vessel and the velocity of the impinging jets based on the optimal jet velocity profile in Figure 8.7 (left).

Figure 8.11 indicates that the crystal size distribution is relatively insensitive to shifts in jet velocity, since a 20% shift in inlet jet velocity is much larger than what would be obtained in practice. Comparing with Figure 8.10 indicates that the overall shape of the jet velocity versus time profile has a greater impact on the final CSD, than the absolute values of the velocities of the jets. Overall, the analysis in this section shows that this control strategy is a promising approach for tailoring specific crystal size distributions.

8.5 Conclusions and Future Directions

This chapter illustrates, in principle, three different approaches for the optimal control of an impinging jet crystallizer to produce crystals with a target crystal size distribution. The results indicate a great increase in the controllability in targeting crystal size distributions for an impinging jet coupled with an aging vessel. The main limitation is that the minimum width of the distribution is limited by the narrowest crystal size distribution produced from the impinging jet. The key inputs into optimization process are:

1. The crystal size distributions as a function of jet velocity from the impinging jet crystallizer, which can be obtained from experiments (Mahajan and Kirwan, 1996) or simulations validated by experiments (see Chapter 6).
2. The growth kinetics of the crystallization system so as to determine the supersaturation profile to achieve the desired growth rate. Unlike nucleation kinetics, crystal growth rates can be estimated or directly measured with high accuracy (Chung et al., 2000; Gunawan et al., 2002; Ma and Braatz, 2003; Rawlings et al., 1993; Togkalidou et al., 2004; Worlitschek and Mazzotti, 2004).

The seeding crystals used for the optimal seeding method described in Section 8.3 can be obtained from other processes other than the impinging jets, as long as the crystals are sufficiently small with narrow distribution. The required seed proportions have to be prepared before the start of the aging process. In addition, the operation can be manually tedious if seeds have to be added frequently. However, this could be automated using a robotic arm that is programmed to drop the seeds held in different containers (similar to the operation of an autosampler). For the optimal control of velocity of impinging jets described in Section 8.4, the process is continuous in the sense that the crystals goes into the aging vessel directly. Thus, the likelihood of introducing impurities into the process is reduced. This also means that any disturbance that causes a significant change in the CSD from the impinging jets can affect the CSD in the aging vessel. Nevertheless, in-situ particle size distribution measurement techniques (e.g. laser backscattering) can detect such disturbances. For both approaches, the control of the crystallization process becomes two-fold, that is, the control of crystals entering the aging vessel, and the concentration-control of the aging vessel. Despite the increase in control complexity, the increase in the level of CSD control is encouraging. In addition, the implementation of the control schemes need to be done once during process development and can be automated for different crystallization systems subsequently.

The results indicate that the optimal control of impinging jet crystallizers is a promising strategy to produce a target crystal size distribution, providing a level of CSD control far beyond what is achievable using the batch and continuous “well-mixed” vessels that currently dominate the industry. This level of CSD control is good

enough to consider the development of a systematic approach for crystal product engineering, in which a desired crystal product is determined by product design (Costa et al., 2006), when the approach taken in this chapter is used to design the process to manufacture that desired product. The next step is to implement the control strategies on laboratory experiments to fully evaluate the feasibility of the approaches. Constraints identified from experiments can be incorporated into the optimization algorithms and further improvements can be explored.

8.6 References

am Ende, D. J. and Brenek, S. J. (2004). Strategies to control particle size during crystallization processes. *American Pharmaceutical Review* **7**(3): 98-104.

am Ende, D. J., Crawford, T. C. and Weston, N. P. (2003). Reactive crystallization method to improve particle size. Pfizer, Inc. and Pfizer Products, Inc. *Patent, U. S.* 6,558,435.

Bauer-Brandl, A. (1996). Polymorphic transitions of cimetidine during manufacture of solid dosage forms. *International Journal of Pharmaceutics* **140**(2): 195.

Beckmann, W. (2000). Seeding the desired polymorph: Background, possibilities, limitations, and case studies. *Organic Process Research & Development* **4**(5): 372-383.

Bell, P. W., Stephens, A. P., Roberts, C. B. and Duke, S. R. (2005). High-resolution imaging of the supercritical antisolvent process. *Experiments in Fluids* **38**(6): 708-719.

Braatz, R. D. (2002). Advanced control of crystallization processes. *Annual Reviews in Control* **26**(1): 87-99.

Choong, K. L. and Smith, R. (2004). Optimization of batch cooling crystallization. *Chemical Engineering Science* **59**(2): 313-327.

Chung, S. H., Ma, D. L. and Braatz, R. D. (1999). Optimal seeding in batch crystallization. *Canadian Journal of Chemical Engineering* **77**(3): 590-596.

Chung, S. H., Ma, D. L. and Braatz, R. D. (2000). Optimal model-based experimental design in batch crystallization. *Chemometrics and Intelligent Laboratory Systems* **50**(1): 83-90.

Costa, R., Moggridge, G. D. and Saraiva, P. M. (2006). Chemical product engineering: An emerging paradigm within chemical engineering. *AIChE Journal* **52**(6): 1976-1986.

Dauer, R., Mokrauer, J. E. and McKeel, W. J. (1996). Dual jet crystallizer apparatus. Merck & Co., Inc. *Patent, U. S.* 5,578,279.

Fujiwara, M., Chow, P. S., Ma, D. L. and Braatz, R. D. (2002). Paracetamol crystallization using laser backscattering and ATR-FTIR spectroscopy: Metastability, agglomeration, and control. *Crystal Growth & Design* **2**(5): 363-370.

Fujiwara, M., Nagy, Z. K., Chew, J. W. and Braatz, R. D. (2005). First-principles and direct design approaches for the control of pharmaceutical crystallization. *Journal of Process Control* **15**(5): 493-504.

Gron, H., Borissova, A. and Roberts, K. J. (2003). In-process ATR-FTIR spectroscopy for closed-loop supersaturation control of a batch crystallizer producing monosodium glutamate crystals of defined size. *Industrial & Engineering Chemistry Research* **42**(1): 198-206.

Gunawan, R., Ma, D. L., Fujiwara, M. and Braatz, R. D. (2002). Identification of kinetic parameters in multidimensional crystallization processes. *International Journal of Modern Physics B* **16**(1-2): 367-374.

Henczka, M., Baldyga, J. and Shekunov, B. Y. (2005). Particle formation by turbulent mixing with supercritical antisolvent. *Chemical Engineering Science* **60**(8-9): 2193-2201.

Jagadesh, D., Kubota, N., Yokota, M., Doki, N. and Sato, A. (1999). Seeding effect on batch crystallization of potassium sulfate under natural cooling mode and a simple design method of crystallizer. *Journal of Chemical Engineering of Japan* **32**(4): 514-520.

Jagadesh, D., Kubota, N., Yokota, M., Sato, A. and Tavare, N. S. (1996). Large and mono-sized product crystals from natural cooling mode batch crystallizer. *Journal of Chemical Engineering of Japan* **29**(5): 865-873.

Johnson, B. K. and Prud'homme, R. K. (2003). Flash NanoPrecipitation of organic actives and block copolymers using a confined impinging jets mixer. *Australian Journal of Chemistry* **56**(10): 1021-1024.

Jones, A. G. and Mullin, J. W. (1974). Programmed cooling crystallization of potassium sulphate solutions. *Chemical Engineering Science* **29**(1): 105.

Jung, J. and Perrut, M. (2001). Particle design using supercritical fluids: Literature and patent survey. *Journal of Supercritical Fluids* **20**(3): 179-219.

Kim, S., Lotz, B., Lindrud, M., Girard, K., Moore, T., Nagarajan, K., Alvarez, M., Lee, T., Nikfar, F., Davidovich, M., Srivastava, S. and Kiang, S. (2005). Control of the particle properties of a drug substance by crystallization engineering and the effect on drug product formulation. *Organic Process Research & Development* **9**(6): 894-901.

Kubota, N., Doki, N., Yokota, M. and Jagadesh, D. (2002). Seeding effect on product crystal size in batch crystallization. *Journal of Chemical Engineering of Japan* **35**(11): 1063-1071.

Kurganov, A. and Tadmor, E. (2000). New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *Journal of Computational Physics* **160**(1): 241-282.

Kwon, J. H. and Kim, C. W. (2004). A novel insulin microcrystals preparation using a seed zone method. *Journal of Crystal Growth* **263**(1-4): 536-543.

Lindrud, M. D., Kim, S. and Wei, C. (2001). Sonic impinging jet crystallization apparatus and process. Bristol-Myers Squibb Company. *Patent, U. S.* 6,302,958.

Liotta, V. and Sabesan, V. (2004). Monitoring and feedback control of supersaturation using ATR-FTIR to produce an active pharmaceutical ingredient of a desired crystal size. *Organic Process Research & Development* **8**(3): 488-494.

Lung-Somarriba, B. L. M., Moscosa-Santillan, M., Porte, C. and Delacroix, A. (2004). Effect of seeded surface area on crystal size distribution in glycine batch cooling crystallization: a seeding methodology. *Journal of Crystal Growth* **270**(3-4): 624-632.

Ma, D. L. and Braatz, R. D. (2003). Robust identification and control of batch processes. *Computers & Chemical Engineering* **27**(8-9): 1175-1184.

Ma, D. L., Tafti, D. K. and Braatz, R. D. (2002). Optimal control and simulation of multidimensional crystallization processes. *Computers & Chemical Engineering* **26**(7-8): 1103-1116.

Mahajan, A. J. and Kirwan, D. J. (1996). Micromixing effects in a two-impinging-jets precipitator. *AIChE Journal* **42**(7): 1801-1814.

Marchisio, D. L., Rivautella, L. and Barresi, A. A. (2006). Design and scale-up of chemical reactors for nanoparticle precipitation. *AIChE Journal* **52**(5): 1877-1887.

Martin, A. and Cocero, M. J. (2004). Numerical modeling of jet hydrodynamics, mass transfer, and crystallization kinetics in the supercritical antisolvent (SAS) process. *Journal of Supercritical Fluids* **32**(1-3): 203-219.

Midler, M., Paul, E. L., Whittington, E. F., Futran, M., Liu, P. D., Hsu, J. and Pan, S.-H. (1994). Crystallization method to improve crystal structure and size. Merck & Co., Inc. *Patent, U. S.* 5,314,506.

Morton, K. W. and Mayers, D. F. (1994). *Numerical Solution of Partial Differential Equations*. Cambridge, UK, Cambridge University Press.

Muhrer, G., Mazzotti, M. and Muller, M. (2003). Gas antisolvent recrystallization of an organic compound. Tailoring product PSD and scaling-up. *Journal of Supercritical Fluids* **27**(2): 195-203.

Nagao, L. M., Lyapustina, S., Munos, M. K. and Capizzi, M. D. (2005). Aspects of particle science and regulation in pharmaceutical inhalation drug products. *Crystal Growth & Design* **5**(6): 2261-2267.

Pasquali, I., Bettini, R. and Giordano, F. (2006). Solid-state chemistry and particle engineering with supercritical fluids in pharmaceutics. *European Journal of Pharmaceutical Sciences* **27**(4): 299-310.

Paul, E. L., Tung, H. H. and Midler, M. (2005). Organic crystallization processes. *Powder Technology* **150**(2): 133-143.

Rasenack, N., Steckel, H. and Muller, B. W. (2003). Micronization of anti-inflammatory drugs for pulmonary delivery by a controlled crystallization process. *Journal of Pharmaceutical Sciences* **92**(1): 35-44.

Rawlings, J. B., Miller, S. M. and Witkowski, W. R. (1993). Model identification and control of solution crystallization processes - a review. *Industrial & Engineering Chemistry Research* **32**(7): 1275-1296.

Rohani, S., Horne, S. and Murthy, K. (2005a). Control of product quality in batch crystallization of pharmaceuticals and fine chemicals. Part 1: Design of the crystallization process and the effect of solvent. *Organic Process Research & Development* **9**(6): 858-872.

Rohani, S., Horne, S. and Murthy, K. (2005b). Control of product quality in batch crystallization of pharmaceuticals and fine chemicals. Part 2: External control. *Organic Process Research & Development* **9**(6): 873-883.

Shekunov, B. Y. and York, P. (2000). Crystallization processes in pharmaceutical technology and drug delivery design. *Journal of Crystal Growth* **211**(1-4): 122-136.

Steffen, B., Muller, K. P., Komenda, M., Koppmann, R. and Schaub, A. (2005). A new mathematical procedure to evaluate peaks in complex chromatograms. *Journal of Chromatography A* **1071**(1-2): 239-246.

Togkalidou, T., Tung, H. H., Sun, Y., Andrews, A. T. and Braatz, R. D. (2004). Parameter estimation and optimization of a loosely bound aggregating pharmaceutical crystallization using in situ infrared and laser backscattering measurements. *Industrial & Engineering Chemistry Research* **43**(19): 6168-6181.

Torres-Lapasio, J. R., Baeza-Baeza, J. J. and Garcia-Alvarez-Coque, M. C. (1997). A model for the description, simulation, and deconvolution of skewed chromatographic peaks. *Analytical Chemistry* **69**(18): 3822-3831.

Vivo-Truyols, G., Torres-Lapasio, J. R., van Nederkassel, A. M., Vander Heyden, Y. and Massart, D. L. (2005). Automatic program for peak detection and deconvolution of multi-overlapped chromatographic signals - Part II: Peak model and deconvolution algorithms. *Journal of Chromatography A* **1096**(1-2): 146-155.

Ward, J. D., Mellichamp, D. A. and Doherty, M. F. (2006). Choosing an operating policy for seeded batch crystallization. *AIChE Journal* **52**(6): 2046-2054.

Worlitschek, J. and Mazzotti, M. (2004). Model-based optimization of particle size distribution in batch-cooling crystallization of paracetamol. *Crystal Growth & Design* **4**(5): 891-903.

Zhou, G. X., Fujiwara, M., Woo, X. Y., Rusli, E., Tung, H. H., Starbuck, C., Davidson, O., Ge, Z. H. and Braatz, R. D. (2006). Direct design of pharmaceutical antisolvent crystallization through concentration control. *Crystal Growth & Design* **6**(4): 892-898.

Chapter 9
Conclusions and Future Directions

9.1 Key Scientific Accomplishments

The research work presented in this dissertation was carried out to meet the goals and objectives listed in Chapter 1. For the simulation studies of mixing and crystallization, the key scientific accomplishments are summarized as follows.

1. A simulation algorithm that couples macromixing and micromixing models and, the solution of the full population balance equation (PBE) was developed for the modeling of mixing-sensitive crystallization processes (see Appendices). This CFD-micromixing-PBE model is the most advanced simulation code developed to date as it computes the entire crystal size distribution taking into account different mixing scales, while previous work reported in literature only used the method of moments to predict the aggregate and mean properties of the crystal product quality. Crystal nucleation, growth and dissolution kinetics have been included, and the modeling of polymorphic systems was shown to be feasible as well.
2. The CFD-micromixing-PBE model was applied to numerically study the effects of operating variables for various antisolvent crystallization systems for the semibatch agitated tank and impinging jet crystallizers.
3. It was shown, through the sensitivity analysis, that the parameters in the CFD-micromixing model can be adjusted to improve the simulation predictions.
4. This CFD-Micromixing-PBE algorithm can be used by the industry to guide the systematic design and scale-up of crystallizers.

The second section of the dissertation focused on the control of antisolvent crystallization systems, and the key scientific contributions are listed as follows.

1. Simulations and analysis showed the benefits of implementing concentration control over antisolvent flow rate control in a semibatch crystallizer. Further analysis also showed that the robustness of the concentration control can be further improved with laser backscattering measurements in cases where the process disturbance causes excessive nucleation. The theoretical analysis presented enables a process engineer to justify the selection of appropriate control strategies for different crystallization systems.
2. By coupling the impinging jet crystallization with a stirred tank crystallizer operating on concentration control, new control strategies were developed to specifically tailor any target crystal size distributions. This theoretical study offers a high precision level of CSD control which cannot be attained in current batch and continuous crystallization technology. This allows a pharmaceutical crystal product to be designed based on its desired bioavailability and subsequently designing the crystallization process to achieve it.

9.2 Implications on Future Research

While the objectives laid out in the beginning of the dissertation have been met, considerable amount of research work still needs to be done to reach the ultimate research goal, that is, to develop systematic and scientific design approaches for crystallization processes to improve the efficiency of pharmaceutical process development. The following is a suggested list of research problems which remains to

be solved. More details of these new research directions are described in the earlier chapters.

1. As the crystallization kinetics used in the simulations must be independent of mixing and scale, experiments to determine the intrinsic parameters of crystallization kinetics that are independent of hydrodynamics must be designed. One possible way is to design microscale crystallization experiments where the transport processes can be fine-tuned such that there are no transport limitations on the crystallization kinetics. Another suggestion is to use the model CFD-Micromixing-PBE model in the parameter estimation procedure to account for the transport limitations in the experiment.
2. The simulation results from the CFD-micromixing-PBE modeling should be validated, on a three-dimensional basis, with experiments. This requires the use of sophisticated, non-intrusive, sensors to capture the three-dimensional flow, concentration and particles field. For example, using a clear impinging jet crystallizer, the three-dimensional flow and concentration field can be measured using planar laser-induced fluorescence techniques, while the particle size distribution can be measured using forward light scattering techniques at different positions in the impinging jet crystallizer.
3. Multiphase models can be incorporated into the CFD-micromixing-PBE model to simulate the particle flow field more accurately. Aggregation and breakage can also be included to improve the prediction of the particle size distribution.
4. While the coupled simulation algorithm can be used to compute the crystal product quality for a wide range of operating and design variables, optimization methods must be used to identify the optimal variables to give

the desired crystal product quality based on the bioavailability requirements of the pharmaceutical.

5. The new control strategies developed to target specific crystal size distributions have to be implemented in experimental systems to access their feasibility and to identify possible improvements.

The research work highlighted here requires experimental and computational experts to work together. The challenge lies in the fact that different crystallization systems can have vastly different kinetics. Nevertheless, with more experimental and computational tools being developed, the design of crystallization processes using systematic engineering approaches is foreseeable.

Appendix A

User-defined functions for Fluent simulation of semibatch antisolvent crystallization in agitated tank

(Serial computation)

cryst2D500rpmReverse_40cell8micron.c (size-dependent growth)

```
/*-----------------------------------------------------------------------------*/
/* UDF for modeling antisolvent crystallization for a 2D tank.                 */
/*                                                                             */
/* Solves for unseeded crystallization for reverse addition (add saturated     */
/* solution into antisolvent), with primary nucleation, size-dependent growth  */
/* and dissolution.                                                            */
/* Refer to Woo et al., Cryst. Growth & Des., 2006 for details.                */
/* Can be modified to seeded crystallization.                                  */
/*                                                                             */
/* Impeller:                                                                   */
/* The impeller is modeled using fixed velocity data for a A200 impeller.      */
/*                                                                             */
/* Effective viscosity:                                                        */
/* Adjust viscosity using Einstein's equation.                                 */
/*                                                                             */
/* Micromixing:                                                                */
/* The micromixing is solved using a 3-environment micromixing model           */
/* (Fox, Computational Models for Turbulent Reacting Flows, 2003).             */
/* The micromixing rates are computed at every time step.                      */
/*                                                                             */
/* Population balance (mass basis):                                            */
/* The population balance is discretized along the growth axis using           */
/* high-resolution, finite-volume, demidiscrete central scheme (Kurganov       */
/* and Tadmor, J. Comput. Phys., 2000).                                        */
/*                                                                             */
/* Kinetics:                                                                   */
/* Crystallization of paracetamol from acetone-water (Granberg et al.,         */
/* J. Cryst Growth, 1999; Granberg and Rasmuson, J. Chem. Eng. Data, 2000)     */
/* Growth rate computed by diffusion layer growth model (Nyvlt, The Kinetics   */
/* of Industrial Crystallization, 1985).                                       */
/*                                                                             */
/* Inlet:                                                                      */
/* Mass, downward momentum and p1 source                                       */
/*                                                                             */
/* Initial condition:                                                          */
/* p2 = 1                                                                      */
/*                                                                             */
/* UDS:                                                                        */
/* volume fraction (pn) for envt, mixture fraction (mixfrac), and              */
/* weighted species (paracetamol (pa), acetone (ac), water (wa),               */
/* crystals in each bin (fw))                                                  */
/*                                                                             */
/* Assumes constant density (of water) as effect of density on flow field      */
/* is small.                                                                   */
/*                                                                             */
/* Tested on Fluent 6.1.22 in SERIAL mode on a linux cluster.                  */
/* Please refer to Fluent UDF Manual for more information on                   */
/* linking and compiling UDFs, and defining and solving UDSs.                  */
/*                                                                             */
/* Authors:  Xing Yi Woo, Reginald B. H. Tan and Richard D. Braatz             */
/*           University of Illinois at Urbana-Champaign and                    */
/*           National University of Singapore                                  */
/*-----------------------------------------------------------------------------*/

/*  UDS0 = p1
    UDS1 = p2
    UDS2 = p3
    UDS3 = weighted mixfrac3
    UDS4 = mixfrac3
    UDS5 = p3*pa3
    UDS6 = p3*wa3
    UDS7 = p3*ac3
    UDS8 -- UDS47 = p3*fw0_3 -- p3*fw39_3

    UDM0 = mixture variance
    UDM1 = scalar dissipation rate
    UDM2 = gamma
    UDM3 = gamma_s
    UDM4 = gradient of mixfrac3
    UDM5 = p3 = 1 - p1 - p2
    UDM6 = source term for uds2
    UDM7 = pa3
    UDM8 = wa3
```

```
    UDM9 = ac3
    UDM10 = mean conc of pa
    UDM11 = mean conc of wa
    UDM12 = mean conc of ac
    UDM13 = solute (pa) conc in kg/kg solvent
    UDM14 = antisolvent (wa) %
    UDM15 = sat solute conc kg/kg solvent
    UDM16 = supersaturation kg/kg solvent
    UDM17 = nucleation rate, B #particles/m3-s
    UDM18 = kolmogoroff length scale, m
    UDM19 -- UDM59 = G[0] -- G[40], m/s (evaluated at the ends of the grid)
    UDM60 = average growth rate, micron/s
    UDM61 -- UDM100 = f0_3 -- f39_3, #/m-m3
    UDM101 -- UDM142 = fr(-1)_3 -- fr40_3, minmod gradient
    UDM143 -- UDM182 = p3*f0_3 -- p3*f39_3
*/

#include "udf.h"
#include "mem.h"

#define v_tip 1.047197551       /* for 500 rpm, computed by 2*pi*r*rps */
#define r 0.02                  /* impeller radius in m */
#define WEIGHT 1.e20            /* weighting coeff */

#define batcht 3600.0           /* batch time, s */
#define addvol 5.0e-4           /* volume of antisolvent added, m3 */
#define zonevol 4.523893e-7     /* volume of injection zone */
#define zonearea 4.523893e-4    /* area of injection zone */
#define p1_in 1.0               /* p1 in the inlet */
#define vol_init 5.021900e-4    /* initial vol in vessel, m3 */

#define p3_tol 1e-10            /* tolerance on p3 for computing concentrations in
envt 3 */
#define c_phi 2.0               /* micromixing rate constant */
#define sc_t 0.7                /* turbulent schmidt number */
#define diff_lam 1.0e-9         /* laminar diffusivity */
#define rho_crystal 1293.0      /* density of crystal */
#define kv 0.605                /* volume shape factor */
#define ks 4.63                 /* area shape factor */
#define phi_max 0.753           /* maximum packing fraction */
#define epsilon_max 0.32        /* max value of epsilon in computing micromixing rate
*/
#define epsilon_k_max 19.0      /* max value of epsilon/k in computing micromixing
rate */
                                /* max value identified from highest value from steady
state flow field computations
                                   in the flow domain excluding CFD cells next to
walls */

#define mixfrac1 1.0            /* mixture fraction of envt 1 */
#define mixfrac2 0.0            /* mixture fraction of envt 2 */
#define ac2 0.0                 /* concentrations in envt 1 and envt 2, kg/m3; switch
these values for direct addition */
#define ac1 289.7081248
#define wa2 998.2
#define wa1 538.0293746
#define pa2 0.0
#define pa1 122.5051499

#define ncell 40                /* # of bins for population balance */
#define delr 8.0e-6             /* delta r, bin size for discretized PBE */
#define theta 1.5               /* constant in minmod gradient */

/* crystal size, r(j), at the end point of each bin */
real rend[ncell+1] = {0.000E+00,
8.000E-06,
1.600E-05,
2.400E-05,
3.200E-05,
4.000E-05,
4.800E-05,
5.600E-05,
6.400E-05,
7.200E-05,
8.000E-05,
8.800E-05,
```

```
9.600E-05,
1.040E-04,
1.120E-04,
1.200E-04,
1.280E-04,
1.360E-04,
1.440E-04,
1.520E-04,
1.600E-04,
1.680E-04,
1.760E-04,
1.840E-04,
1.920E-04,
2.000E-04,
2.080E-04,
2.160E-04,
2.240E-04,
2.320E-04,
2.400E-04,
2.480E-04,
2.560E-04,
2.640E-04,
2.720E-04,
2.800E-04,
2.880E-04,
2.960E-04,
3.040E-04,
3.120E-04,
3.200E-04};

/* r(j+1)^4 - r(j)^4 */
real r4_diff[ncell] = {4.0960000000E-21,
6.1440000000E-20,
2.6624000000E-19,
7.1680000000E-19,
1.5114240000E-18,
2.7484160000E-18,
4.5260800000E-18,
6.9427200000E-18,
1.0096640000E-17,
1.4086144000E-17,
1.9009536000E-17,
2.4965120000E-17,
3.2051200000E-17,
4.0366080000E-17,
5.0008064000E-17,
6.1075456000E-17,
7.3666560000E-17,
8.7879680000E-17,
1.0381312000E-16,
1.2156518400E-16,
1.4123417600E-16,
1.6291840000E-16,
1.8671616000E-16,
2.1272576000E-16,
2.4104550400E-16,
2.7177369600E-16,
3.0500864000E-16,
3.4084864000E-16,
3.7939200000E-16,
4.2073702400E-16,
4.6498201600E-16,
5.1222528000E-16,
5.6256512000E-16,
6.1609984000E-16,
6.7292774400E-16,
7.3314713600E-16,
7.9685632000E-16,
8.6415360000E-16,
9.3513728000E-16,
1.0099056640E-15};

real Get_gamma_s(cell_t c, Thread *t, real p1, real p2, real p3, real gamma);
real minmod(real fL, real f, real fR);
real Get_crystalsource(cell_t c, Thread *t, int j);
```

```
/* Velocity data for Lightnin A200 impeller */
Thread *imp_in_id;

/*  profile for x-velocity    */
DEFINE_SOURCE(axial_velocity, cell, thread, dS, eqn)
{
  real u,y, x[ND_ND], source;              /* variable declarations */
  cell_t c;

      C_CENTROID(x,c,thread);
      y = x[1];
      u=(-((0.01123080244+1.22510144547*(y/r)-
8.60434585806*(y*y/(r*r))+27.53735609731*(y*y*y/(r*r*r))-
31.75536179043*(y*y*y*y/(r*r*r*r))+11.77596324737*(y*y*y*y*y/(r*r*r*r*r)))*v_tip) -
C_U(cell,thread));
/*      printf("%f,%f\n",u,y); */
      source=WEIGHT*u;
      dS[eqn] = -WEIGHT;

  return source;
}

/*  profile for radial-velocity    */
DEFINE_SOURCE(radial_velocity, cell, thread, dS, eqn)
{
  real v,y, x[ND_ND], source;              /* variable declarations */
  cell_t c;

      C_CENTROID(x,c,thread);
      y = x[1];
      v=(((-0.02970810029+0.35999288645*(y/r)-
1.1461133392*(y*y/(r*r))+3.10023214341*(y*y*y/(r*r*r))-
3.52010393338*(y*y*y*y/(r*r*r*r))+1.27889295812*(y*y*y*y*y/(r*r*r*r*r)))*v_tip) -
C_V(cell,thread));
      source=WEIGHT*v;
      dS[eqn] = -WEIGHT;

  return source;
}

/*  profile for swirl-velocity    */
DEFINE_SOURCE(swirl_velocity, cell, thread, dS, eqn)
{
  real w,y, x[ND_ND], source;              /* variable declarations */
  cell_t c;

      C_CENTROID(x,c,thread);
      y = x[1];
      w=(((-0.0883910586-0.42752363520*(y/r)+3.69606167595*(y*y/(r*r))-
11.98588383459*(y*y*y/(r*r*r))+12.83209749972*(y*y*y*y/(r*r*r*r))-
4.32537440960*(y*y*y*y*y/(r*r*r*r*r)))*v_tip) - C_W(cell,thread));
      source=WEIGHT*w;
      dS[eqn] = -WEIGHT;

  return source;
}

/*  profile for kinetic energy  */
DEFINE_SOURCE(tke, cell, thread, dS, eqn)
{
  real k,y, x[ND_ND], source;              /* variable declarations */
  cell_t c;

      C_CENTROID(x,c,thread);
      y = x[1];
      k=(((0.05022854980-0.42922101743*(y/r)+1.78327697464*(y*y/(r*r))-
3.72521997377*(y*y*y/(r*r*r))+3.88359439021*(y*y*y*y/(r*r*r*r))-
1.50868633454*(y*y*y*y*y/(r*r*r*r*r)))*v_tip*v_tip) - C_K(cell,thread));
      source=WEIGHT*k;
      dS[eqn] = -WEIGHT;

  return source;
}

/* profile for dissipation rate  */
DEFINE_SOURCE(diss, cell, thread, dS, eqn)
```

```
{
  real e, y, x[ND_ND], source;              /* variable declarations */
  cell_t c;

      C_CENTROID(x,c,thread);
      y = x[1];
      e=(((0.02646128841-0.25298662904*(y/r)+1.01977316775*(y*y/(r*r))-
2.06580451784*(y*y*y/(r*r*r))+2.08808135349*(y*y*y*y/(r*r*r*r))-
0.78469728463*(y*y*y*y*y/(r*r*r*r*r)))*v_tip*v_tip*v_tip/r) - C_D(cell,thread));
      source=WEIGHT*e;
      dS[eqn] = -WEIGHT;

  return source;
}

/* computes density for mixture, kg/m3 */
/* not used */
DEFINE_PROPERTY(rho_mix, c, t)
{
    real rho, rho_init, as_vol;

    rho_init =  pa2 + wa2 + ac2;
    as_vol = (addvol / batcht) * CURRENT_TIME;
    rho = (rho_init * vol_init + wa1 * as_vol) / (vol_init + as_vol);

    return rho;
}

/* Effective viscosity */
DEFINE_PROPERTY(eff_mu, c, t)
{
    real effmu, fw_sum, fwj_mean, volfrac_cry;
    real mu_pa=0.001681, mu_ac=0.000331, mu_wa=0.001003, intrisic_mu;
    int j;

    /* Get volume fraction of all crystals */
    fw_sum = 0.0;
    for (j=0; j<ncell; j++) {
        fwj_mean = C_UDSI(c,t,8+j);
        if (fwj_mean > 0.0) fw_sum += fwj_mean;
    }
    volfrac_cry = fw_sum / rho_crystal;

    /* Compute effective viscosity */
    effmu = mu_wa * (1.0 + 2.5 * volfrac_cry);
    /* effmu = mu_wa * (1.0 + volfrac_cry / phi_max); */

    return effmu;
}

/* diffusivity for UDS eqn */
DEFINE_DIFFUSIVITY(pn_turb_diff, c, t, i)
{
    real turbdiff_rho;

    turbdiff_rho = C_MU_T(c,t) / sc_t;  /* (turbulent diffusivity * density) */

    return turbdiff_rho;
}

/* compute the minmod gradient, nucleation, growth and dissolution rates at every
iteration */
DEFINE_ADJUST(micromix, d)
{
    Thread *t;
    cell_t c;
    real p1, p2, p3, pa3, wa3, ac3, fwj_3, fj_3[ncell+4], frj_3[ncell+2];
    real c_pa, w, csat_pa, supersat, csat_v, c1, c2, B, supersat_v;
    real ki, i, G[ncell+1], kolmo_l, k_mu, epsilon, c3, schmidt, reynolds, sherwood,
kd, G_avg, dp;
    int j;

    thread_loop_c(t,d)
    {
        begin_c_loop(c,t)
            {
```

```
                p1 = C_UDSI(c,t,0);
                p2 = C_UDSI(c,t,1);
                p3 = C_UDSI(c,t,2);

                k_mu = C_MU_L(c,t) / C_R(c,t);  /* kinematic viscosity */
                epsilon = C_D(c,t);
                if (epsilon > epsilon_max) epsilon = epsilon_max;
                c3 = (k_mu*k_mu*k_mu) / epsilon;
                kolmo_l = pow(c3, 0.25);    /* kolmogoroff lengtn scale */
                C_UDMI(c,t,18) = kolmo_l;


                if (p3 > p3_tol) {
                    /* conc in p3 */
                    pa3 = C_UDSI(c,t,5) / p3;
                    wa3 = C_UDSI(c,t,6) / p3;
                    ac3 = C_UDSI(c,t,7) / p3;

                    /* # distribution (f) in p3 */
                    for (j=0; j<ncell; j++) {
                        fwj_3 = C_UDSI(c,t,(8+j)) / p3;
                        if (fwj_3 < 0.0) fwj_3 = 0.0;
                        fj_3[j+2] = (4.0/(rho_crystal*kv)) * fwj_3 / r4_diff[j];
                    }
                    fj_3[0] = 0.0;
                    fj_3[1] = 0.0;
                    fj_3[ncell+2] = fj_3[ncell+1];
                    fj_3[ncell+3] = fj_3[ncell+1];
                    /* minmod gradient of f in p3 */
                    for (j=0; j<=(ncell+1); j++) frj_3[j] = minmod(fj_3[j], fj_3[j+1],
fj_3[j+2]);

                    /* mass fraction of solute on solute free basis in envt 3 */
                    c_pa =  pa3 / (wa3 + ac3);

                    /* water mass % on solute free basis in envt 3 */
                    w = wa3 / (wa3 + ac3) * 100.0;

                    /* sat solute conc, mass fraction on solute free basis in envt 3
*/
                    csat_pa = 0.001 * (-5.01902e-9 * (w*w*w*w*w*w) + 1.69767e-6 *
(w*w*w*w*w) - 2.46765e-4 * (w*w*w*w) + 2.19262e-2 * (w*w*w) - 1.27018 * (w*w) +
3.42614e1 * w + 7.96086e1);

                    /* supersaturation, mass fraction on solute free basis in envt 3
*/
                    supersat = c_pa - csat_pa;

                    if (supersat > 0.0) {
                        /* Nucleation rate, #particles/m3-s */
                        csat_v = -7.56719e-9*(w*w*w*w*w*w) + 2.52296e-6*(w*w*w*w*w) -
3.32604e-4*(w*w*w*w) + 2.33867e-2*(w*w*w) - 1.01740*(w*w) + 2.33555e1*w + 6.08849e1;
/* kg solute/m3 */
                        c1 = log(rho_crystal/csat_v);
                        c2 = log(c_pa/csat_pa);
                        B = 8.56080e8 * exp(-1.22850e-3 * (c1*c1*c1) / (c2*c2));

                        /* Growth rate, m/s */
                        supersat_v = pa3 - csat_v;
                        if (supersat_v < 0.0) supersat_v = 0.0;
                        /* kg = -1.60e-10 * (w*w*w) + 5.59e-08 * (w*w) - 2.10e-06 * w
+ 6.14e-05;
                        g = -1.11e-04 * (w*w) + 1.02e-02 * w + 1.43;
                        G = kg * pow(supersat, g); */
                        ki = 1.95e-07 * w - 7.35e-06;
                        /* G[0] = ki * supersat_v * supersat_v;
                        printf("G[0] = %e; ki = %e; supersatv = %e; \n", G[0], ki,
supersatv); */
                        schmidt = k_mu / diff_lam;
                        for (j=0; j<=ncell; j++) {
                            if (j == 0) dp = 1.0e-9;
                            else dp = rend[j];
                            reynolds = pow(epsilon, 0.333333) * pow(dp,1.333333) /
k_mu;
                            if (dp <= kolmo_l)  sherwood = 2.0 + 0.52 * pow(reynolds,
0.52) * pow(schmidt, 0.333333);
```

```
                            else sherwood = 2.0 + 0.5 * pow(reynolds, 0.62) *
pow(schmidt, 0.333333);
                            kd = sherwood * diff_lam / dp;
                            G[j] = (ks/(6.0*kv*rho_crystal)) * (kd*kd/ki) * (1.0 +
(2.0*ki/kd)*supersat_v - sqrt(1.0 + (4.0*ki/kd)*supersat_v));
                        }
                    }
                    else if (supersat < 0.0) {
                        B = 0.0;
                        /* Dissolution rate, m/s */
                        csat_v = -7.56719e-9*(w*w*w*w*w*w) + 2.52296e-6*(w*w*w*w*w) -
3.32604e-4*(w*w*w*w) + 2.33867e-2*(w*w*w) - 1.01740*(w*w) + 2.33555e1*w + 6.08849e1;
/* kg solute/m3 */
                        supersat_v = pa3 - csat_v;
                        schmidt = k_mu / diff_lam;
                        for (j=0; j<=ncell; j++) {
                            if (j == 0) dp = 1.0e-7;
                            else dp = rend[j];
                            reynolds = pow(epsilon, 0.333333) * pow(dp,1.333333) /
k_mu;
                            if (dp <= kolmo_l)  sherwood = 2.0 + 0.52 * pow(reynolds,
0.52) * pow(schmidt, 0.333333);
                            else sherwood = 2.0 + 0.5 * pow(reynolds, 0.62) *
pow(schmidt, 0.333333);
                            kd = sherwood * diff_lam / dp;
                            G[j] = (ks/(3.0*kv*rho_crystal)) * kd * supersat_v;
                        }
                    }
                    else {
                        B = 0.0;
                        supersat_v = 0.0;
                        for (j=0; j<=ncell; j++) G[j] = 0.0;
                    }
                }
                else {
                    pa3 = 0.0;
                    wa3 = 0.0;
                    ac3 = 0.0;
                    for (j=0; j<(ncell+3); j++) fj_3[j] = 0.0;
                    for (j=0; j<=ncell+1; j++) frj_3[j] = 0.0;
                    c_pa = 0.0;
                    w = 0.0;
                    csat_pa = 0.0;
                    supersat = 0.0;
                    supersat_v = 0.0;
                    B = 0.0;
                    for (j=0; j<=ncell; j++) G[j] = 0.0;
                }
                /* assign values into user-defined memory */
                C_UDMI(c,t,7) = pa3;
                C_UDMI(c,t,8) = wa3;
                C_UDMI(c,t,9) = ac3;
                C_UDMI(c,t,13) = c_pa;
                C_UDMI(c,t,14) = w;
                C_UDMI(c,t,15) = csat_pa;
                C_UDMI(c,t,16) = supersat;
                C_UDMI(c,t,6) = supersat_v;
                C_UDMI(c,t,17) = B;
                G_avg = 0.0;
                for (j=0; j<=ncell; j++) {
                    G_avg += G[j];
                    C_UDMI(c,t,19+j) = G[j];
                }
                G_avg = G_avg * 1e6 / (ncell+1);
                C_UDMI(c,t,(19+ncell+1)) = G_avg;
                for (j=0; j<ncell; j++)     C_UDMI(c,t,(61+j)) = fj_3[j+2];
                for (j=0; j<=(ncell+1); j++)    C_UDMI(c,t,(101+j)) = frj_3[j];
            }
        end_c_loop(c,t)
    }
}

/* compute micromixng rates, and mean conc values at every time step  */
DEFINE_EXECUTE_AT_END(mean_conc)
{
    Domain *d;
```

```
    Thread *t;
    cell_t c;
    real p1, p2, p3, mixfrac3, gammaD, gamma, mixvar, epsilon_k;
    real massj[ncell+3], current_vol, current_mass, sum_vol, cellvol, csdj[ncell];
    real p1_avg=0.0, p2_avg=0.0, p3_avg=0.0, gamma_avg=0.0, mixfrac3_avg=0.0,
psum_err_avg = 0.0;
    real p3_vol = 0.0, W_avg = 0.0, S_avg = 0.0, G_avg = 0.0, B_avg = 0.0;

    int j, current_tstep;
    FILE *fd;

    d = Get_Domain(1);
    sum_vol = 0.0;

    thread_loop_c(t,d)
    {
        begin_c_loop(c,t)
        {
            p1 = C_UDSI(c,t,0);
            p2 = C_UDSI(c,t,1);
            p3 = C_UDSI(c,t,2);

            /* error in sum of vol fraction in all envt*/
            C_UDMI(c,t,5) = (p1 + p2 + p3) - 1.0;

            /* mixture fraction of envt 3 */
            if (p3 > p3_tol) {
                mixfrac3 = C_UDSI(c,t,3) / p3;
                if (mixfrac3 < 0.0) mixfrac3 = 1e-6;
            }
            else mixfrac3 = 0.5;
            C_UDSI(c,t,4) = mixfrac3;

            /* mixture variance */
            mixvar = p1 * (1.0-p1) - 2.0 * p1 * p3 * mixfrac3 + p3 * (1.0-p3) *
mixfrac3 * mixfrac3;
            C_UDMI(c,t,0) = mixvar;

            /* scalar dissipation rate */
            epsilon_k = C_D(c,t) / C_K(c,t);
            if (epsilon_k > epsilon_k_max) epsilon_k = epsilon_k_max;
            C_UDMI(c,t,1) = c_phi * C_UDMI(c,t,0) * epsilon_k;

            /* micromixing rate */
            /* micromixing occurs when there is a minimal amt of p1 or p2 present and
p1 or p2 < 1 */
            if (p1 > 1e-10) {
                /* p1 always <1 for the given initial condition */
                gammaD = ((p1 * (1.0-p1) * (1.0-mixfrac3) * (1.0-mixfrac3) + p2 *
(1.0-p2) * mixfrac3 * mixfrac3));
                gamma = mixvar / gammaD;
                gamma = gamma * c_phi * epsilon_k;
                if (gamma < 0.0) gamma = 0.0;
            }
            else if ((p2 > 1e-10) && (p2 < (1.0-1e-10))) {
                gammaD = ((p1 * (1.0-p1) * (1.0-mixfrac3) * (1.0-mixfrac3) + p2 *
(1.0-p2) * mixfrac3 * mixfrac3));
                gamma = mixvar / gammaD;
                gamma = gamma * c_phi * epsilon_k;
                if (gamma < 0.0) gamma = 0.0;
            }
            else gamma = 0.0;
            C_UDMI(c,t,2) = gamma;

            /* mean conc in CFD cell */
            C_UDMI(c,t,10) = p1 * pa1 + p2 * pa2 + C_UDSI(c,t,5);
            C_UDMI(c,t,11) = p1 * wa1 + p2 * wa2 + C_UDSI(c,t,6);
            C_UDMI(c,t,12) = p1 * ac1 + p2 * ac2 + C_UDSI(c,t,7);
            for (j=0; j<ncell; j++) {
                if (C_UDSI(c,t,8+j) > 0.0)  C_UDMI(c,t,143+j) = 1.0e-6 *
(4.0/(rho_crystal*kv)) * C_UDSI(c,t,8+j) / r4_diff[j];
                else C_UDMI(c,t,143+j) = 0.0;
            }

            sum_vol += C_VOLUME(c,t);
        }
```

```
        end_c_loop(c,t)
    }

    sum_vol = sum_vol * (2.0*3.141593654) * 1.0e3;  /* liters */
    current_vol = ((addvol/batcht) * CURRENT_TIME + vol_init) * 1.0e3;
    printf("sum_vol = %14.8e, current_vol = %14.8e \n", sum_vol, current_vol);

    /* write to file every 10 time steps */
    current_tstep = N_TIME;
    if ((current_tstep%10) == NULL) {
        for (j=0; j<(ncell+3); j++) massj[j] = 0.0;
        for (j=0; j<ncell; j++) csdj[j] = 0.0;

        current_mass = (pa2 + wa2 + ac2) * vol_init + (pa1 + wa1 + ac1) *
(addvol/batcht) * CURRENT_TIME;
        thread_loop_c(t,d)
        {
            begin_c_loop(c,t)
            {
                cellvol = C_VOLUME(c,t) * (2.0*3.141593654);
                for (j=0; j<ncell; j++) {
                    massj[j] += C_UDSI(c,t,8+j) * cellvol;
                    csdj[j] += C_UDMI(c,t,143+j) * cellvol;
                }
                massj[ncell] += C_UDMI(c,t,10) * cellvol;
                massj[ncell+1] += C_UDMI(c,t,11) * cellvol;
                massj[ncell+2] += C_UDMI(c,t,12) * cellvol;
                p3 = C_UDSI(c,t,2);
                if (p3 > p3_tol) {
                    p3_vol += p3 * cellvol;
                    W_avg += C_UDMI(c,t,14) * p3 * cellvol;
                    S_avg += C_UDMI(c,t,16) * p3 * cellvol;
                    B_avg += C_UDMI(c,t,17) * p3 * cellvol;
                    G_avg += C_UDMI(c,t,60) * p3 * cellvol;
                }
                p1_avg += C_UDSI(c,t,0) * cellvol;
                p2_avg += C_UDSI(c,t,1) * cellvol;
                p3_avg += C_UDSI(c,t,2) * cellvol;
                mixfrac3_avg += C_UDSI(c,t,4) * cellvol;
                gamma_avg += C_UDMI(c,t,2) * cellvol;
                psum_err_avg += fabs(C_UDMI(c,t,5)) * cellvol;
            }
            end_c_loop(c,t)
        }
        for (j=0; j<ncell; j++) csdj[j] = csdj[j] / sum_vol;
        W_avg = W_avg / p3_vol;
        S_avg = S_avg / p3_vol;
        B_avg = B_avg * 1.0e-3 / p3_vol;    /* per liter */
        G_avg = G_avg / p3_vol;
        p1_avg = p1_avg * 1.0e3 / sum_vol;
        p2_avg = p2_avg * 1.0e3 / sum_vol;
        p3_avg = p3_avg * 1.0e3 / sum_vol;
        mixfrac3_avg = mixfrac3_avg * 1.0e3 / sum_vol;
        gamma_avg = gamma_avg * 1.0e3 / sum_vol;
        psum_err_avg = psum_err_avg * 1.0e3 / sum_vol;
        fd = fopen("Rcry2d500rpm_MassCsd.txt", "a");
        fprintf(fd, "%d %f %14.8e %14.8e ", N_TIME, CURRENT_TIME, sum_vol,
current_mass);
        for (j=0; j<(ncell+3); j++) fprintf(fd, "%e ", massj[j]);
        for (j=0; j<ncell; j++) fprintf(fd, "%e ", csdj[j]);
        fprintf(fd, "\n");
        fclose(fd);
        fd = fopen("Rcry2d500rpm_KineticsMicromix.txt", "a");
        fprintf(fd, "%d %f ", N_TIME, CURRENT_TIME);
        fprintf(fd, "%e %e %e %e ", W_avg, S_avg, B_avg, G_avg);
        fprintf(fd, "%e %e %e %e %e %e \n", p1_avg, p2_avg, p3_avg, mixfrac3_avg,
gamma_avg, psum_err_avg);
        fclose(fd);
    }
}

/* mass source terms for inlet */
DEFINE_SOURCE(mass_inlet, c, t, dS, eqn)
{
    real source, m_rate, v_rate;
```

```
    v_rate = addvol / batcht;
    m_rate = v_rate * 998.2;    /* mass flow rate in kg/s */

    source = m_rate / zonevol;
    dS[eqn] = 0.0;
    return source;
}

/* x-momentum source for inlet */
DEFINE_SOURCE(mom_inlet, c, t, dS, eqn)
{
    real source, velocity, v_rate;

    v_rate = addvol / batcht;
    velocity = -v_rate / zonearea;

    source = (v_rate * 998.2 * velocity) / (zonevol);
    dS[eqn] = 0.0;
    return source;
}

/* p1 source for inlet */
/* not used */
DEFINE_SOURCE(p1_inlet, c, t, dS, eqn)
{
    real source, source_inlet, v_rate, gamma, gamma_s, p1, p2, p3;

    v_rate = addvol / batcht;
    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);

    /* micromixing terms to eliminate spurious dissipation */
    if (gamma > 0.0) gamma_s = Get_gamma_s(c,t,p1,p2,p3,gamma);
    else gamma_s = 0.0;

    source_inlet = (v_rate * 998.2 * p1_in) / (zonevol);
    source = C_R(c,t) * (-gamma * p1 * (1.0-p1) + gamma_s * p3) + source_inlet;
    dS[eqn] = C_R(c,t) * (-gamma * (1.0 - 2.0*p1) - gamma_s);
    return source;
}

/* source terms for pn UDS eqn, based on Fox multienvironment micromixing model */
DEFINE_SOURCE(p1_source, c, t, dS, eqn)
{
    real source, gamma, gamma_s, p1, p2, p3, source_inlet, v_rate;
    int inletID = 5;
    Domain *d;
    Thread *inlet_t;

    d = Get_Domain(1);
    inlet_t = Lookup_Thread(d, inletID);
    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);

    /* micromixing terms to eliminate spurious dissipation */
    if (gamma > 0.0) gamma_s = Get_gamma_s(c,t,p1,p2,p3,gamma);
    else gamma_s = 0.0;
    C_UDMI(c,t,3) = gamma_s;
    C_UDMI(c,t,4) = NV_MAG2(C_UDSI_G(c,t,4));

    source = C_R(c,t) * (-gamma * p1 * (1.0-p1) + gamma_s * p3);
    /* addition at inlet */
    if (t == inlet_t) {
        v_rate = addvol / batcht;
        source_inlet = (v_rate * 998.2 * p1_in) / (zonevol);
        source += source_inlet;
    }
    /* C_UDMI(c,t,6) = source; */
    dS[eqn] = C_R(c,t) * (-gamma * (1.0 - 2.0*p1) - gamma_s);
    return source;
}
```

```
DEFINE_SOURCE(p2_source, c, t, dS, eqn)
{
    real source, gamma, gamma_s, p1, p2, p3;

    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);

    if (gamma > 0.0) gamma_s = Get_gamma_s(c,t,p1,p2,p3,gamma);
    else gamma_s = 0.0;

    source = C_R(c,t) * (-gamma * p2 * (1.0-p2) + gamma_s * p3);
    dS[eqn] = C_R(c,t) * (-gamma * (1.0 - 2.0*p2) - gamma_s);
    return source;
}

DEFINE_SOURCE(p3_source, c, t, dS, eqn)
{
    real source, p1, p2, p3, gamma, gamma_s;

    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);

    if (gamma > 0.0) gamma_s = Get_gamma_s(c,t,p1,p2,p3,gamma);
    else gamma_s = 0.0;

    source = C_R(c,t) * (gamma * (p1*(1.0-p1) + p2*(1.0-p2)) - (p3*2.0*gamma_s));
    /* C_UDMI(c,t,6) = source; */
    dS[eqn] = C_R(c,t) * (gamma * (2.0 - 4.0*p2 - 4.0*p3) - 2.0*gamma_s);
    return source;
}

/* source term for mixture fraction and species in envt 3 */
DEFINE_SOURCE(wtmixfrac3_source, c, t, dS, eqn)
{
    real source, p1, p2, p3, gamma, gamma_s;

    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);

    if (gamma > 0.0) gamma_s = Get_gamma_s(c,t,p1,p2,p3,gamma);
    else gamma_s = 0.0;

    source = C_R(c,t) * (gamma * (p1*(1.0-p1)*mixfrac1 + p2*(1.0-p2)*mixfrac2) - p3 *
gamma_s * (mixfrac1+mixfrac2));
    /* C_UDMI(c,t,6) = source; */
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(pa3_source, c, t, dS, eqn)
{
    real source, p1, p2, p3, gamma, gamma_s, cry_source;
    int j;

    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);

    if (gamma > 0.0) gamma_s = Get_gamma_s(c,t,p1,p2,p3,gamma);
    else gamma_s = 0.0;

    cry_source = 0.0;
    for (j=0; j<ncell; j++) cry_source += Get_crystalsource(c,t,j);

    source = C_R(c,t) * (gamma * (p1*(1.0-p1)*pa1 + p2*(1.0-p2)*pa2) - p3 * gamma_s *
(pa1+pa2)) - cry_source;
    dS[eqn] = 0.0;
    return source;
}
```

```
DEFINE_SOURCE(wa3_source, c, t, dS, eqn)
{
    real source, p1, p2, p3, gamma, gamma_s;

    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);

    if (gamma > 0.0) gamma_s = Get_gamma_s(c,t,p1,p2,p3,gamma);
    else gamma_s = 0.0;

    source = C_R(c,t) * (gamma * (p1*(1.0-p1)*wa1 + p2*(1.0-p2)*wa2) - p3 * gamma_s *
(wa1+wa2));
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(ac3_source, c, t, dS, eqn)
{
    real source, p1, p2, p3, gamma, gamma_s;

    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);

    if (gamma > 0.0) gamma_s = Get_gamma_s(c,t,p1,p2,p3,gamma);
    else gamma_s = 0.0;

    source = C_R(c,t) * (gamma * (p1*(1.0-p1)*ac1 + p2*(1.0-p2)*ac2) - p3 * gamma_s *
(ac1+ac2));
    dS[eqn] = 0.0;
    return source;
}

/* source terms of crystals in envt 3 */
DEFINE_SOURCE(fw0_source, c, t, dS, eqn)
{
    real source;
    int j=0;

    source = Get_crystalsource(c,t,j);
    /* C_UDMI(c,t,6) = source; */
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw1_source, c, t, dS, eqn)
{
    real source;
    int j=1;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw2_source, c, t, dS, eqn)
{
    real source;
    int j=2;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw3_source, c, t, dS, eqn)
{
    real source;
    int j=3;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
```

```
    return source;
}

DEFINE_SOURCE(fw4_source, c, t, dS, eqn)
{
    real source;
    int j=4;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw5_source, c, t, dS, eqn)
{
    real source;
    int j=5;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw6_source, c, t, dS, eqn)
{
    real source;
    int j=6;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw7_source, c, t, dS, eqn)
{
    real source;
    int j=7;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw8_source, c, t, dS, eqn)
{
    real source;
    int j=8;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw9_source, c, t, dS, eqn)
{
    real source;
    int j=9;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw10_source, c, t, dS, eqn)
{
    real source;
    int j=10;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw11_source, c, t, dS, eqn)
{
    real source;
```

```
    int j=11;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw12_source, c, t, dS, eqn)
{
    real source;
    int j=12;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw13_source, c, t, dS, eqn)
{
    real source;
    int j=13;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw14_source, c, t, dS, eqn)
{
    real source;
    int j=14;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw15_source, c, t, dS, eqn)
{
    real source;
    int j=15;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw16_source, c, t, dS, eqn)
{
    real source;
    int j=16;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw17_source, c, t, dS, eqn)
{
    real source;
    int j=17;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw18_source, c, t, dS, eqn)
{
    real source;
    int j=18;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}
```

```
DEFINE_SOURCE(fw19_source, c, t, dS, eqn)
{
    real source;
    int j=19;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw20_source, c, t, dS, eqn)
{
    real source;
    int j=20;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw21_source, c, t, dS, eqn)
{
    real source;
    int j=21;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw22_source, c, t, dS, eqn)
{
    real source;
    int j=22;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw23_source, c, t, dS, eqn)
{
    real source;
    int j=23;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw24_source, c, t, dS, eqn)
{
    real source;
    int j=24;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw25_source, c, t, dS, eqn)
{
    real source;
    int j=25;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw26_source, c, t, dS, eqn)
{
    real source;
    int j=26;
```

```
    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw27_source, c, t, dS, eqn)
{
    real source;
    int j=27;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw28_source, c, t, dS, eqn)
{
    real source;
    int j=28;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw29_source, c, t, dS, eqn)
{
    real source;
    int j=29;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw30_source, c, t, dS, eqn)
{
    real source;
    int j=30;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw31_source, c, t, dS, eqn)
{
    real source;
    int j=31;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw32_source, c, t, dS, eqn)
{
    real source;
    int j=32;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw33_source, c, t, dS, eqn)
{
    real source;
    int j=33;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw34_source, c, t, dS, eqn)
```

```
{
    real source;
    int j=34;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw35_source, c, t, dS, eqn)
{
    real source;
    int j=35;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw36_source, c, t, dS, eqn)
{
    real source;
    int j=36;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw37_source, c, t, dS, eqn)
{
    real source;
    int j=37;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw38_source, c, t, dS, eqn)
{
    real source;
    int j=38;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

DEFINE_SOURCE(fw39_source, c, t, dS, eqn)
{
    real source;
    int j=39;

    source = Get_crystalsource(c,t,j);
    dS[eqn] = 0.0;
    return source;
}

real Get_gamma_s(cell_t c, Thread *t, real p1, real p2, real p3, real gamma) {

    real gamma_s, mixfrac3, grad_sq, term1, term2;

    mixfrac3 = C_UDSI(c,t,4);
    grad_sq = NV_MAG2(C_UDSI_G(c,t,4));
    gamma_s = 2.0 * C_UDSI_DIFF(c,t,0) * grad_sq / ((1.0-mixfrac3) * (1.0-mixfrac3) +
mixfrac3 * mixfrac3);
    term1 =  gamma * p1 * (1.0-p1) - gamma_s * p3;
    term2 = gamma * p2 * (1.0-p2) - gamma_s * p3;
    if ((term1 < 0.0) || (term2 < 0.0)) {
        if (p3 > p3_tol) {
            gamma_s =  gamma * p1 * (1.0-p1) / p3;
            if (term2 < term1) gamma_s = gamma * p2 * (1.0-p2) / p3;
        }
        else gamma_s = 0.0;
    }
```

```
    if (gamma_s < 0.0) gamma_s = 0.0;
    /* gamma_s = 0.0; */
    return gamma_s;
}

real minmod(real fL, real f, real fR) {

    real grad1, grad2, grad3, fr;

    grad1 = theta * (f - fL) / delr;
    grad2 = 0.5 * (fR - fL) / delr;
    grad3 = theta * (fR - f) / delr;

    if ((grad1>0.0) && (grad2>0.0) && (grad3>0.0)) {
        fr = grad1;
        if (grad2 < fr) fr = grad2;
        if (grad3 < fr) fr = grad3;
    }
    else if ((grad1<0.0) && (grad2<0.0) && (grad3<0.0)) {
        fr = grad1;
        if (grad2 > fr) fr = grad2;
        if (grad3 > fr) fr = grad3;
    }
    else fr = 0.0;

    return fr;
}

real Get_crystalsource(cell_t c, Thread *t, int j) {

    real crystalsource, p3, GL, GR, f, fL, fr, frL, B, supersat, frR, fR;

    p3 = C_UDSI(c,t,2);
    GL = C_UDMI(c,t,19+j);
    GR = C_UDMI(c,t,19+j+1);
    f = C_UDMI(c,t,61+j);
    fr = C_UDMI(c,t,101+j+1);
    supersat = C_UDMI(c,t,16);

    /* growth */
    if (supersat > 0.0) {
        frL = C_UDMI(c,t,101+j);
        if (j == 0) {
        fL = 0.0;
        B = C_UDMI(c,t,17);
        crystalsource = C_R(c,t) * (p3*rho_crystal*kv*0.25/delr) * r4_diff[j] * (-GR *
(f + 0.5*delr*fr) + GL * (fL + 0.5*delr*frL) + B);
        }
        else {
            fL = C_UDMI(c,t,61+j-1);
            crystalsource = C_R(c,t) * (p3*rho_crystal*kv*0.25/delr) * r4_diff[j] * (-
GR * (f + 0.5*delr*fr) + GL * (fL + 0.5*delr*frL));
        }
    }
    /* dissolution */
    else if (supersat < 0.0) {
        frR = (c,t,101+j+2);
        if (j == (ncell-1)) fR = f;
        else fR = C_UDMI(c,t,61+j+1);
            crystalsource = C_R(c,t) * (p3*rho_crystal*kv*0.25/delr) * r4_diff[j] * (-
GR * (fR - 0.5*delr*frR) + GL * (f - 0.5*delr*fr));
    }
    else crystalsource = 0.0;

    return crystalsource;
}
```

cryst2DtankB_dmesh.c

```
/*----------------------------------------------------------------------*/
/* UDF for dynamic mesh for 2D tank.                                    */
/*                                                                      */
/* Compute velocity at which the liquid level rises based on            */
/* constant addition rate.                                              */
/*                                                                      */
/* Tested on Fluent 6.1.22 in serial mode on a linux cluster.           */
/* Please refer to Fluent UDF Manual for more information on            */
/* linking and compiling UDFs.                                          */
/*                                                                      */
/* Authors:  Xing Yi Woo, Reginald B. H. Tan and Richard D. Braatz      */
/*           University of Illinois at Urbana-Champaign and             */
/*           National University of Singapore                           */
/*----------------------------------------------------------------------*/

#include "udf.h"

#define batcht 3600.0       /* batch time in s */
#define addvol 5e-4         /* added volume, m3 */
#define area 0.0078414157   /* area of top surface of tank in m2 */
#define pi_const 3.141593654

DEFINE_CG_MOTION(surf_vel, dt, vel, omega, time, dtime)
{
    real v_rate, rise;

    v_rate = addvol / batcht;
    rise = v_rate / area;

    vel[0] = rise;
}
```

Appendix B

User-defined functions for Fluent simulation of antisolvent crystallization in 3D confined impinging jets

(Parallel computation)

para-3envCIJ-lovastatin30.c (size-independent growth)

```
/*-----------------------------------------------------------------------------*/
/* UDF for modeling antisolvent crystallization for a 3D confined impinging jet.*/
/*                                                                             */
/* Details of geometry (500A-Y2X) from Johnson and Prud'homme (AIChE J., 2003). */
/*                                                                             */
/* Solves for unseeded crystallization with primary nucleation and             */
/* size-independent growth                                                     */
/* Refer to Woo et al., Cryst. Growth Des., (submitted) for details.           */
/*                                                                             */
/* Micromixing:                                                                */
/* The micromixing is solved using a 3-environment micromixing model           */
/* (Fox, Computational Models for Turbulent Reacting Flows, 2003).             */
/* The micromixing rates are computed at every time step.                      */
/*                                                                             */
/* Population balance (mass basis):                                            */
/* The population balance is discretized along the growth axis using           */
/* high-resolution, finite-volume, demidiscrete central scheme (Kurganov       */
/* and Tadmor, J. Comput. Phys., 2000).                                        */
/*                                                                             */
/* Kinetics:                                                                   */
/* Crystallization of lovastatin from water-methanol (Mahajan and Kirwan,      */
/* J. Cryst Growth, 1994)                                                      */
/*                                                                             */
/* Inlet pipes:                                                                */
/* One inlet for saturated solution (envt 1) and one for antisolvent (envt 2). */
/*                                                                             */
/* Initial condition:                                                          */
/* p1 = p2 = 0.5                                                               */
/*                                                                             */
/* UDS:                                                                        */
/* volume fraction (pn) for envt, mixture fraction (mixfrac), and              */
/* weighted species (solute, antisolvent, solvent and                          */
/* crystals in each bin (fw)).                                                 */
/*                                                                             */
/* Assumes constant density (of mixture) as effect of density on flow field    */
/* is small.                                                                   */
/*                                                                             */
/* Tested on Fluent 6.2.16 in PARALLEL mode on a linux cluster.                */
/* Please refer to Fluent UDF Manual for more information on                   */
/* linking and compiling UDFs, and defining and solving UDSs.                  */
/*                                                                             */
/* Authors:  Xing Yi Woo, Reginald B. H. Tan and Richard D. Braatz             */
/*           University of Illinois at Urbana-Champaign and                    */
/*           National University of Singapore                                  */
/*-----------------------------------------------------------------------------*/

/*  YI, UDM and UDS
    ===============

    Volume fractions of envt
    ------------------------
    UDS0 = envt 1, p_1
    UDS1 = envt 2, p_2
    UDS2 = envt 3 ,p_3 (by solving transport eqn)
    UDS3 = mixture fraction in envt 3
    UDS4 = weighted mixture fraction in envt 3

    Weighted concentrations in envt 3
    ---------------------------------
    UDS5 = Solute
    UDS6 = Antisolvent
    UDS7 = solvent
    UDS8 -- UDS37 = fw0 - fw29

    Micromixing parameters
    ----------------------
    UDM0 = mixture variance
    UDM1 = scalar dissipation rate
    UDM2 = gamma
    UDM3 = gradient of mixfrac3
    UDM4 = gamma_s
    UDM5 = p3 = 1.0 - p1 -p2
    UDM6 = p3(UDS2, from transport equation) - p3(UDM5, from volume fraction balance)
```

```
    Concentrations and kinetic parameters in envt 3
    -----------------------
    UDM7 = solute
    UDM8 = antisolvent
    UDM9 = solvent
    UDM10 = local c (kg/kg solvents)
    UDM11 = local antisolvent % (solute free basis)
    UDM12 = local c* (kg/kg solvents)
    UDM13 = local supersaturation (kg/kg solvents)
    UDM14 = local relative supersaturation, c/c*
    UDM15 = nucleation rate, B (#particles/m^3-s)
    UDM16 = kolmogoroff length scale (m) (not calculated)
    UDM17 = size independent growth rate, G (m/s)
    UDM18 = G (micron/s)
    UDM19 -- 48 = f0 -- f29 (#particles/m-m^3)
    UDM49 -- 80 = fr-1 -- fr30, minmod gradient

    Mean concentrations in CFD cell
    ------------------------------
    UDM81 = solute
    UDM82 = antisolvent
    UDM83 = solvent
    UDM84 -- 113 = f0 -- f29
*/

#include "udf.h"
#include "mem.h"
#include "prf.h"

/* constants */
#define p_tol 1.0e-5           /* minimum tolerance for volume fraction of envt 3 */
#define c_phi 4.0              /* micromixing constant */
#define sc_t 1.40              /* turbulent schmidt number */
#define diff_lam 1.0e-9        /* laminar diffusion coefficient */
#define turb_t_min 2.2e-4      /* min tolerance for turbulent time scale to calculate
scalar dissipation rate, changes with velocity */
                               /* determined from flow domain of steady-state flow
field, excluding cells next to walls */
#define rho_crystal 1273.0     /* density of crystal */
#define kv 0.0006250           /* volume shape factor of crystals */
#define ka 0.101250            /* area shape factor of crystals */
#define numspecies 33          /* solute, solvents, crystals */
#define ncell 30               /* # bins for population balance equation */
#define delr 3.0e-6            /* size of each bin, m */
#define theta 1.5              /* minmod constant */
#define n_uds 38               /* # of UDS */
#define n_udm 114              /* # of UDM */

/* boundary/zones ID*/
#define outletID 14
#define inlet1ID 16
#define inlet2ID 15
#define fluid_inlet1ID 5
#define fluid_inlet2ID 4
#define fluid_chamberID 2
#define fluid_outletID 3
#define wall_chamberID 6
#define wall_outletID 7
#define wall_inlet1ID 9
#define wall_inlet2ID 8

/*  mass concentrations (kg/m3) in envt 1 and 2
    Envt 1 = saturated solution, Envt 2 = antisolvent
    First entry is mixture fraction */
real phi1[numspecies+1] = {1.0, 25.45558, 0.0, 769.30273, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
real phi2[numspecies+1] = {0.0, 0.0, 998.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
/* r (m) at ends of cell */
real rend[ncell+1] = {0.00E+00,
3.00E-06,
6.00E-06,
9.00E-06,
```

```
1.20E-05,
1.50E-05,
1.80E-05,
2.10E-05,
2.40E-05,
2.70E-05,
3.00E-05,
3.30E-05,
3.60E-05,
3.90E-05,
4.20E-05,
4.50E-05,
4.80E-05,
5.10E-05,
5.40E-05,
5.70E-05,
6.00E-05,
6.30E-05,
6.60E-05,
6.90E-05,
7.20E-05,
7.50E-05,
7.80E-05,
8.10E-05,
8.40E-05,
8.70E-05,
9.00E-05};
/* r(j+1)^4 - r(j)^4 */
real r4_diff[ncell] = {8.10000E-23,
1.21500E-21,
5.26500E-21,
1.41750E-20,
2.98890E-20,
5.43510E-20,
8.95050E-20,
1.37295E-19,
1.99665E-19,
2.78559E-19,
3.75921E-19,
4.93695E-19,
6.33825E-19,
7.98255E-19,
9.88929E-19,
1.20779E-18,
1.45679E-18,
1.73786E-18,
2.05295E-18,
2.40400E-18,
2.79296E-18,
3.22178E-18,
3.69239E-18,
4.20674E-18,
4.76677E-18,
5.37443E-18,
6.03167E-18,
6.74042E-18,
7.50263E-18,
8.32024E-18};

/* ------------------------------------------------------------------------------
- */

real Get_gamma_s(cell_t c, Thread *t, real p1, real p2, real p3, real gamma);
real Get_source(cell_t c, Thread *t, int j);
real Get_rxn(cell_t c, Thread *t, int j);
real minmod(real fL, real f, real fR);
real Get_crystal(cell_t c, Thread *t, int j);
real Get_solubility(real as);

/* diffusivity for UDS eqn */
DEFINE_DIFFUSIVITY(pn_turb_diff, c, t, i)
{
    real turbdiff_rho;

#if !RP_HOST
```

```
    turbdiff_rho = C_MU_T(c,t) / sc_t + C_R(c,t) * diff_lam;    /* (turbulent
diffusivity * density) */

    return turbdiff_rho;
#endif
}

/* computes the crystallization kinetics and vol fraction of envt 3 for every
iteration */
DEFINE_ADJUST(micromix, d)
{
    Thread *t, *tf;
    cell_t c;
    face_t f;
    real p1, p2, p3, mixfrac3;
    real solute3, antisolvent3, solvent3, fwj_3, fj_3[ncell+4], frj_3[ncell+2];
    real cs, as, csat, supersat, rel_supersat, csat_v, B, G_sizeindep, B_homo,
B_hetero;
    int j, zoneID, n, crys_j;

#if PARALLEL
    PRF_GSYNC();
#endif
#if !RP_HOST
    thread_loop_f(tf, d)
    {
        zoneID = THREAD_ID(tf);

        if (THREAD_STORAGE(tf,SV_UDS_I(0))!= NULL) {
            begin_f_loop(f, tf)
            {
                p1 = F_UDSI(f,tf,0);
                p2 = F_UDSI(f,tf,1);
                p3 = F_UDSI(f,tf,2);

                if ((zoneID == wall_inlet1ID) || (zoneID == wall_inlet2ID)) mixfrac3 =
0.5;
                else {
                    if (p3 > p_tol) {
                        mixfrac3 = F_UDSI(f,tf,4) / p3;
                        if (mixfrac3 < 0.0) mixfrac3 = 0.0;
                    }
                    else mixfrac3 = 0.5;
                }
                F_UDSI(f,tf,3) = mixfrac3;
            }
            end_f_loop(f, tf)
        }
    }

    thread_loop_c(t,d)
    {
        zoneID = THREAD_ID(t);

        if ((zoneID == fluid_chamberID) || (zoneID == fluid_outletID)) {
            begin_c_loop_int(c,t)
            {
                p1 = C_UDSI(c,t,0);
                p2 = C_UDSI(c,t,1);
                p3 = C_UDSI(c,t,2);

                /* mixture fraction of envt 2 and 3 */
                if (p3 > p_tol) {
                    mixfrac3 = C_UDSI(c,t,4) / p3;
                    if (mixfrac3 < 0.0) mixfrac3 = 0.0;
                }
                    else mixfrac3 = 0.5;
                C_UDSI(c,t,3) = mixfrac3;

                /* Concentration and crystalllization rates in envt 3 */

                /*C_UDMI(c,t,16) = kolmo_l; */

                if (p3 > p_tol) {
                    /* conc in p3 */
                    solute3 = C_UDSI(c,t,5) / p3;
```

```
                    antisolvent3 = C_UDSI(c,t,6) / p3;
                    solvent3 = C_UDSI(c,t,7) / p3;

                    /* # distribution (f) envt 3 */
                    for (crys_j=0; crys_j<ncell; crys_j++) {
                        fwj_3 = C_UDSI(c,t,(8+crys_j)) / p3;
                        /* if (fwj_3 < 0.0) fwj_3 = 0.0; */
                        fj_3[crys_j+2] = (4.0/(rho_crystal*kv)) * fwj_3 /
r4_diff[crys_j];
                    }
                }
                else {
                    solute3 = (phi1[1] + phi2[1]) / 2.0;
                    antisolvent3 = (phi1[2] + phi2[2]) / 2.0;
                    solvent3 = (phi1[3] + phi2[3]) / 2.0;
                    for (crys_j=0; crys_j<ncell; crys_j++) {
                        fwj_3 = (phi1[crys_j+4] + phi2[crys_j+4]) / 2.0;
                        fj_3[crys_j+2] = (4.0/(rho_crystal*kv)) * fwj_3 /
r4_diff[crys_j];
                    }
                }

                fj_3[0] = 0.0;
                fj_3[1] = 0.0;
                fj_3[ncell+2] = fj_3[ncell+1];
                fj_3[ncell+3] = fj_3[ncell+1];
                /* minmod gradient of f in p3 */
                for (crys_j=0; crys_j<=(ncell+1); crys_j++) frj_3[crys_j] =
minmod(fj_3[crys_j], fj_3[crys_j+1], fj_3[crys_j+2]);

                /* mass fraction of solute on solute free basis in envt 3 */
                cs =  solute3 / (antisolvent3 + solvent3);

                /* antisolvent mass % on solute free basis in envt 3 */
                as = antisolvent3 / (antisolvent3 + solvent3) * 100.0;

                /* sat solute conc, mass fraction on solute free basis in envt 3 */
                csat = Get_solubility(as);

                /* supersaturation, mass fraction on solute free basis in envt 3 */
                supersat = cs - csat;

                /* relative supersaturation, in envt 3 */
                rel_supersat = cs / csat;

                if (supersat > 0.0) {
                    /* Nucleation rate, #particles/m3-s */
                    B_homo = 6.9656e14 * exp(-15.7647/((log(rel_supersat)) *
(log(rel_supersat))));
                    B_hetero = 2.19196e8 * exp(-0.994387/((log(rel_supersat)) *
(log(rel_supersat))));
                    B = B_homo + B_hetero;

                    G_sizeindep = 8.3333e-30 * (pow((2.4623e3 * log(rel_supersat)),
6.7));
                }
                else if (supersat < 0.0) {
                    B = 0.0;
                    G_sizeindep = 0.0;
                }
                else {
                    B = 0.0;
                    G_sizeindep = 0.0;
                }
                /* assign values into user-defined memory */
                C_UDMI(c,t,7) = solute3;
                C_UDMI(c,t,8) = antisolvent3;
                C_UDMI(c,t,9) = solvent3;
                C_UDMI(c,t,10) = cs;
                C_UDMI(c,t,11) = as;
                C_UDMI(c,t,12) = csat;
                C_UDMI(c,t,13) = supersat;
                C_UDMI(c,t,14) = rel_supersat;
                C_UDMI(c,t,15) = B;
                C_UDMI(c,t,17) = G_sizeindep;
                C_UDMI(c,t,18) = G_sizeindep * 1.0e6;
```

```
                for (crys_j=0; crys_j<ncell; crys_j++)      C_UDMI(c,t,(19+crys_j)) =
fj_3[crys_j+2];
                for (crys_j=0; crys_j<=(ncell+1); crys_j++) C_UDMI(c,t,(49+crys_j)) =
frj_3[crys_j];
            }
            end_c_loop_int(c,t)
        }
    }
#endif
#if PARALLEL
    PRF_GSYNC();
    EXCHANGE_UDMI(d,0,(n_udm-1));
#endif
}

/* computes p_3, conversion of DMP in each cell, weighted conc in envt 3, and
micromixing rate at every time step. */
DEFINE_EXECUTE_AT_END(conversion)
{
    Domain *d;
    Thread *t, *tf;
    cell_t c;
    face_t f;
    real p1, p2, p3, mixfrac3;
    real gammaD, gamma, mixvar, gamma_s, turb_t;
    real fj_out[ncell], flow_out, fwj_out, flow_out_sum, solute_out, antisolvent_out,
solvent_out;
    real solute_out_conc, antisolvent_out_conc, solvent_out_conc, cs_out, as_out,
csat_out, supersat_out, rel_supersat_out;
    real time, timestep;
    int j, crys_j, zoneID, ntime;
    FILE *fd;

#if PARALLEL
    PRF_GSYNC();
#endif
#if !RP_HOST
    d = Get_Domain(1);

    thread_loop_f(tf, d)
    {
        zoneID = THREAD_ID(tf);

        if (THREAD_STORAGE(tf,SV_UDS_I(0))!=NULL) {
            begin_f_loop(f, tf)
            {
                p1 = F_UDSI(f,tf,0);
                p2 = F_UDSI(f,tf,1);
                p3 = F_UDSI(f,tf,2);

                if ((zoneID == wall_inlet1ID) || (zoneID == wall_inlet2ID)) mixfrac3 =
0.5;
                else {
                    if (p3 > p_tol) {
                        mixfrac3 = F_UDSI(f,tf,4) / p3;
                        if (mixfrac3 < 0.0) mixfrac3 = 0.0;
                    }
                    else mixfrac3 = 0.5;
                }
                F_UDSI(f,tf,3) = mixfrac3;
            }
            end_f_loop(f, tf)
        }
    }
    thread_loop_c(t,d)
    {
        zoneID = THREAD_ID(t);

        if ((zoneID == fluid_chamberID) || (zoneID == fluid_outletID)) {
            begin_c_loop_int(c,t)
            {
                p1 = C_UDSI(c,t,0);
                p2 = C_UDSI(c,t,1);
                p3 = C_UDSI(c,t,2);
                C_UDMI(c,t,5) = 1.0 - p1 - p2;
```

```
                /* error in p3 */
                C_UDMI(c,t,6) = (p3 - C_UDMI(c,t,5));

                /* mixture fraction of envt 2 and 3 */
                if (p3 > p_tol) {
                    mixfrac3 = C_UDSI(c,t,4) / p3;
                    if (mixfrac3 < 0.0) mixfrac3 = 0.0;
                }
                else mixfrac3 = 0.5;
                C_UDSI(c,t,3) = mixfrac3;

                /* mixture variance */
                mixvar = p1 * (1.0-p1) - 2.0 * p1 * p3 * mixfrac3 + p3 * (1.0-p3) *
mixfrac3 * mixfrac3;
                C_UDMI(c,t,0) = mixvar;

                /* scalar dissipation rate */
                turb_t = C_K(c,t) / C_D(c,t);
                if (turb_t < turb_t_min) turb_t = turb_t_min;
                C_UDMI(c,t,1) = 0.5 * c_phi * mixvar / turb_t;

                /* micromixing rate */
                if (((p1 > p_tol)&&(p1<(1.0-p_tol))) || ((p2 > p_tol)&&(p2<(1.0-
p_tol))))  {
                    gammaD = ((p1 * (1.0-p1) * (1.0-mixfrac3) * (1.0-mixfrac3) + p2 *
(1.0-p2) * mixfrac3 * mixfrac3));
                    gamma = mixvar / gammaD;
                    gamma = gamma * 0.5 * c_phi / turb_t;
                    if (gamma < 0.0) gamma = 0.0;
                }
                else gamma = 0.0;
                if (p3 < 0.0) gamma = 0.0;
                /* gamma = c_phi * 0.5/turb_t;*/
                C_UDMI(c,t,2) = gamma;

                /* compute gradient of mixture fraction */
                if (T_STORAGE_R_NV(t,SV_UDSI_G(3)) != NULL) {
                    C_UDMI(c,t,3) = NV_MAG2(C_UDSI_G(c,t,3));
                }

                /* micromixing terms to eliminate spurious dissipation */
                gamma_s = Get_gamma_s(c,t,p1,p2,p3,gamma);
                C_UDMI(c,t,4) = gamma_s;

                /* Mean concentration in CFD cell */
                for (j=0; j<numspecies; j++) C_UDMI(c,t,(j+81)) = p1 * phi1[j+1] + p2
* phi2[j+1] + C_UDSI(c,t,(5+j));
                for (crys_j=0; crys_j<ncell; crys_j++)  C_UDMI(c,t,(crys_j+84)) =
1.0e-6 * (4.0/(rho_crystal*kv)) * C_UDMI(c,t,(crys_j+84)) / r4_diff[crys_j];
            }
            end_c_loop_int(c,t)
        }
        if ((zoneID == fluid_inlet1ID) || (zoneID == fluid_inlet2ID)) {
            begin_c_loop_int(c,t)
            {
                p1 = C_UDSI(c,t,0);
                p2 = C_UDSI(c,t,1);
                p3 = C_UDSI(c,t,2);

                /* Mean concentration in CFD cell */
                for (j=0; j<numspecies; j++) C_UDMI(c,t,(j+81)) = p1 * phi1[j+1] + p2
* phi2[j+1] + C_UDSI(c,t,(5+j));
                for (crys_j=0; crys_j<ncell; crys_j++)  C_UDMI(c,t,(crys_j+84)) =
1.0e-6 * (4.0/(rho_crystal*kv)) * C_UDMI(c,t,(crys_j+84)) / r4_diff[crys_j];
            }
            end_c_loop_int(c,t)
        }
    }
#if PARALLEL
    PRF_GSYNC();
#endif

    /* compute CSD (flow rate) at the outlet and write to file */
    tf = Lookup_Thread(d, outletID);

    for (crys_j=0; crys_j<ncell; crys_j++) fj_out[crys_j]=0.0;
```

```
    flow_out_sum = 0.0;
    solute_out = 0.0;
    antisolvent_out = 0.0;
    solvent_out = 0.0;

    begin_f_loop(f,tf)
    {
        p1 = F_UDSI(f,tf,0);
        p2 = F_UDSI(f,tf,1);
        flow_out = F_FLUX(f,tf) / F_R(f,tf);
        flow_out_sum += flow_out;
        solute_out += flow_out * (p1 * phi1[1] + p2 * phi2[1] + F_UDSI(f,tf,(5)));
        antisolvent_out += flow_out * (p1 * phi1[2] + p2 * phi2[2] +
F_UDSI(f,tf,(6)));
        solvent_out += flow_out * (p1 * phi1[3] + p2 * phi2[3] + F_UDSI(f,tf,(7)));
        for (crys_j=0; crys_j<ncell; crys_j++) {
            fwj_out = flow_out * (p1 * phi1[crys_j+4] + p2 * phi2[crys_j+4] +
F_UDSI(f,tf,(8+crys_j)));
            fj_out[crys_j] += 1.0e-6 * (4.0/(rho_crystal*kv)) * fwj_out /
r4_diff[crys_j];
        }
    }
    end_f_loop(f,tf)

#if RP_NODE

    for (crys_j=0; crys_j<ncell; crys_j++)  fj_out[crys_j] =
PRF_GRSUM1(fj_out[crys_j]);
    solute_out = PRF_GRSUM1(solute_out);
    antisolvent_out = PRF_GRSUM1(antisolvent_out);
    solvent_out = PRF_GRSUM1(solvent_out);
    flow_out_sum = PRF_GRSUM1(flow_out_sum);

#endif

    solute_out_conc = solute_out / flow_out_sum;
    antisolvent_out_conc = antisolvent_out / flow_out_sum;
    solvent_out_conc = solvent_out / flow_out_sum;
    cs_out = solute_out_conc / (antisolvent_out_conc + solvent_out_conc);
    as_out = 100.0 * antisolvent_out_conc / (antisolvent_out_conc + solvent_out_conc);
    csat_out = Get_solubility(as_out);
    supersat_out = cs_out - csat_out;
    rel_supersat_out = cs_out / csat_out;

    time = CURRENT_TIME;
    timestep = CURRENT_TIMESTEP;
    ntime = N_TIME;

#endif

    node_to_host_real(fj_out, ncell);
    node_to_host_real_6(flow_out_sum, cs_out, as_out, csat_out, supersat_out,
rel_supersat_out);
    node_to_host_real_2(time, timestep);
    node_to_host_int_1(ntime);

#if !RP_NODE
    fd = fopen("CSDoutlet.txt", "a");
    fprintf(fd, "%d %e %e %e ", ntime, timestep, time, flow_out_sum);
    for (crys_j=0; crys_j<ncell; crys_j++)  fprintf(fd, "%e ", fj_out[crys_j]);
    fprintf(fd, "\n");
    fclose(fd);

    fd = fopen("SUPERSAToutlet.txt", "a");
    fprintf(fd, "%d %e %e %e %e %e %e %e %e \n", ntime, timestep, time, flow_out_sum,
cs_out, as_out, csat_out, supersat_out, rel_supersat_out);
    fclose(fd);
#endif
#if PARALLEL
    PRF_GSYNC();
    EXCHANGE_UDMI(d,0,(n_udm-1));
#endif
}

/* source term for pn and mixture fraction UDS eqn, based on Fox multienvironment
micromixing model */
```

```
DEFINE_SOURCE(p1_source, c, t, dS, eqn)
{
    real source, gamma, gamma_s, p1, p2, p3;

#if !RP_HOST
    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);
    gamma_s = C_UDMI(c,t,4);

    source = C_R(c,t) * (-gamma * p1 * (1.0-p1) + gamma_s * p3);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(p2_source, c, t, dS, eqn)
{
    real source, gamma, gamma_s, p1, p2, p3;

#if !RP_HOST
    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);
    gamma_s = C_UDMI(c,t,4);

    source = C_R(c,t) * (-gamma * p2 * (1.0-p2) + gamma_s * p3);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(p3_source, c, t, dS, eqn)
{
    real source, p1, p2, p3, gamma, gamma_s;

#if !RP_HOST
    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);
    gamma_s = C_UDMI(c,t,4);

    source = C_R(c,t) * (gamma * (p1*(1.0-p1) + p2*(1.0-p2)) - (p3*2.0*gamma_s));
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(mixfrac3_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=0;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

/* source term for weighted conc in envt 3 UDS eqn */
DEFINE_SOURCE(solute_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=1;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
```

```
}

DEFINE_SOURCE(antisolvent_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=2;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(solvent_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=3;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw0_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=4;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw1_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=5;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw2_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=6;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw3_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=7;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
```

```
    return source;
#endif
}

DEFINE_SOURCE(fw4_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=8;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw5_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=9;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw6_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=10;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw7_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=11;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw8_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=12;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw9_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=13;
```

```
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw10_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=14;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw11_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=15;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw12_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=16;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw13_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=17;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw14_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=18;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw15_source, c, t, dS, eqn)
{
    real source;
    int j;
```

```
#if !RP_HOST
    j=19;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw16_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=20;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw17_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=21;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw18_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=22;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw19_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=23;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw20_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=24;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw21_source, c, t, dS, eqn)
{
    real source;
```

```
    int j;

#if !RP_HOST
    j=25;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw22_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=26;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw23_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=27;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw24_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=28;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw25_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=29;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw26_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=30;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw27_source, c, t, dS, eqn)
```

```
{
    real source;
    int j;

#if !RP_HOST
    j=31;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw28_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=32;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw29_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=33;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

real Get_gamma_s(cell_t c, Thread *t, real p1, real p2, real p3, real gamma) {

    real gamma_s, mixfrac3, grad_sq, term1, term2;

    if (p3 > p_tol) {
        mixfrac3 = C_UDSI(c,t,3);
        grad_sq = C_UDMI(c,t,3);
        gamma_s = 2.0 * C_UDSI_DIFF(c,t,0) * grad_sq / ((1.0-mixfrac3) * (1.0-
mixfrac3) + mixfrac3 * mixfrac3);
        term1 =  gamma * p1 * (1.0-p1) - gamma_s * p3;
        term2 = gamma * p2 * (1.0-p2) - gamma_s * p3;
        if ((term1 < 0.0) || (term2 < 0.0)) {
            gamma_s =  gamma * p1 * (1.0-p1) / p3;
            if (term2 < term1) gamma_s = gamma * p2 * (1.0-p2) / p3;
        }
    }
    else gamma_s = 0.0;
    if (gamma_s < 0.0) gamma_s = 0.0;
    /* gamma_s = 0.0; */
    return gamma_s;
}

/* micromixing source terms */
real Get_source(cell_t c, Thread *t, int j) {

    real source, gamma, gamma_s, p1, p2, p3, rxn;

    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);
    gamma_s = C_UDMI(c,t,4);

    if (p3 > p_tol) rxn = Get_rxn(c,t,j);
    else rxn = 0.0;

    source =  C_R(c,t) * (gamma * (p1*(1.0-p1)*phi1[j] + p2*(1.0-p2)*phi2[j]) -
gamma_s*p3*(phi1[j] + phi2[j]) + p3*rxn);
```

```
    return source;
}

/* species source terms */
real Get_rxn(cell_t c, Thread *t, int j) {

    real rxn;
    int crys_j, crys_jj;

    if (j >= 4) crys_j = j - 4;

    if (j == 0) rxn = 0.0;  /* mixture fraction */
    if (j == 1) {           /* solute */
        rxn = 0.0;
        for (crys_jj=0; crys_jj<ncell; crys_jj++) rxn += -Get_crystal(c,t,crys_jj);
    }
    if (j == 2) rxn = 0.0;  /* antisolvent */
    if (j == 3) rxn = 0.0;  /* solvent */
    if (j >= 4) rxn = Get_crystal(c,t,crys_j);  /* fw0 - fw29 */

    return rxn;
}

/* source term for population balance */
real Get_crystal(cell_t c, Thread *t, int j) {

    real crystalsource, p3, GL, GR, f, fL, fr, frL, B, supersat, frR, fR;

    p3 = C_UDSI(c,t,2);
    GL = C_UDMI(c,t,17);
    GR = C_UDMI(c,t,17);
    f = C_UDMI(c,t,19+j);
    fr = C_UDMI(c,t,49+j+1);
    supersat = C_UDMI(c,t,13);

    if (supersat > 0.0) {
        frL = C_UDMI(c,t,49+j);
        if (j == 0) {
        fL = 0.0;
        B = C_UDMI(c,t,15);
        crystalsource = (rho_crystal*kv*0.25/delr) * r4_diff[j] * (-GR * (f +
0.5*delr*fr) + GL * (fL + 0.5*delr*frL) + B);
        }
        else {
            fL = C_UDMI(c,t,19+j-1);
            crystalsource = (rho_crystal*kv*0.25/delr) * r4_diff[j] * (-GR * (f +
0.5*delr*fr) + GL * (fL + 0.5*delr*frL));
        }
    }
    /* else if (supersat < 0.0) {
        frR = C_UDMI(c,t,49+j+2);
        if (j == (ncell-1)) fR = f;
        else fR = C_UDMI(c,t,19+j+1);
            crystalsource = (rho_crystal*kv*0.25/delr) * r4_diff[j] * (-GR * (fR -
0.5*delr*frR) + GL * (f - 0.5*delr*fr));
    } */
    else crystalsource = 0.0;

    return crystalsource;
}

real minmod(real fL, real f, real fR) {

    real grad1, grad2, grad3, fr;

    grad1 = theta * (f - fL) / delr;
    grad2 = 0.5 * (fR - fL) / delr;
    grad3 = theta * (fR - f) / delr;

    if ((grad1>0.0) && (grad2>0.0) && (grad3>0.0)) {
        fr = grad1;
        if (grad2 < fr) fr = grad2;
        if (grad3 < fr) fr = grad3;
    }
    else if ((grad1<0.0) && (grad2<0.0) && (grad3<0.0)) {
```

```
        fr = grad1;
        if (grad2 > fr) fr = grad2;
        if (grad3 > fr) fr = grad3;
    }
    else fr = 0.0;

    return fr;
}

real Get_solubility(real as) {

    real csat;

    if (as <=45.67) csat = 0.001 * (-2.7455e-4 * (as*as*as) + 3.3716e-2 * (as*as) -
1.6704 * as + 3.3089e1);
    else csat = 0.001 * (-1.7884e-2 * as + 1.7888);

    return csat;
}
```

3envCIJ-histidine.c (size-independent growth, 2 polymorphs without transformation)

```
/*-------------------------------------------------------------------------------*/
/* UDF for modeling antisolvent crystallization for a 3D confined impinging jet.*/
/* Includes 2 sets of population balance for formation of 2 polymorphs,         */
/* without transformation.                                                       */
/*                                                                               */
/* Details of geometry (500A-Y2X) from Johnson and Prud'homme (AIChE J., 2003). */
/*                                                                               */
/* Solves for unseeded crystallization with primary nucleation and               */
/* size-independent growth                                                       */
/* Refer to Woo et al., Cryst. Growth Des., (submitted) for details.             */
/*                                                                               */
/* Micromixing:                                                                  */
/* The micromixing is solved using a 3-environment micromixing model             */
/* (Fox, Computational Models for Turbulent Reacting Flows, 2003).               */
/* The micromixing rates are computed at every time step.                        */
/*                                                                               */
/* Population balance (mass basis):                                              */
/* The population balance is discretized along the growth axis using             */
/* high-resolution, finite-volume, demidiscrete central scheme (Kurganov         */
/* and Tadmor, J. Comput. Phys., 2000).                                          */
/*                                                                               */
/* Kinetics:                                                                     */
/* Crystallization of l-histidine from water-isopropanol (Kitamura et al.,       */
/* J. Cryst Growth, 1994; Roelands et al., Cryst. Growth Des., 2006).            */
/*                                                                               */
/* Inlet pipes:                                                                  */
/* One inlet for saturated solution (envt 1) and one for antisolvent (envt 2).   */
/*                                                                               */
/* Initial condition:                                                            */
/* p1 = p2 = 0.5                                                                 */
/*                                                                               */
/* UDS:                                                                          */
/* volume fraction (pn) for envt, mixture fraction (mixfrac), and                */
/* weighted species (solute, antisolvent, solvent and                            */
/* crystals in each bin (fw)).                                                   */
/*                                                                               */
/* Assumes constant density (of mixture) as effect of density on flow field      */
/* is small.                                                                     */
/*                                                                               */
/* Tested on Fluent 6.2.16 in PARALLEL mode on a linux cluster.                  */
/* Please refer to Fluent UDF Manual for more information on                     */
/* linking and compiling UDFs, and defining and solving UDSs.                    */
/*                                                                               */
/* Authors:  Xing Yi Woo, Reginald B. H. Tan and Richard D. Braatz               */
/*           University of Illinois at Urbana-Champaign and                      */
/*           National University of Singapore                                    */
/*-------------------------------------------------------------------------------*/

/*  YI, UDM and UDS
    ===============

    Volume fractions of envt
    ------------------------
    UDS0 = envt 1, p_1
    UDS1 = envt 2, p_2
    UDS2 = envt 3 ,p_3 (by solving transport eqn)
    UDS3 = mixture fraction in envt 3
    UDS4 = weighted mixture fraction in envt 3

    Weighted concentrations in envt 3
    ---------------------------------
    UDS5 = Solute
    UDS6 = Antisolvent
    UDS7 = solvent
    UDS8 -- UDS27 = fw0 - fw19 (Polymorph A)
    UDS28 -- UDS47 = fw0 - fw19 (Polymorph B)

    Micromixing parameters
    ----------------------
    UDM0 = mixture variance
    UDM1 = scalar dissipation rate
    UDM2 = gamma
```

```
    UDM3 = gradient of mixfrac3
    UDM4 = gamma_s
    UDM5 = p3 = 1.0 - p1 -p2
    UDM6 = p3(UDS2, from transport equation) - p3(UDM5, from volume fraction balance)

    Concentrations and kinetic parameters in envt 3
    -----------------------
    UDM7 = solute
    UDM8 = antisolvent
    UDM9 = solvent
    UDM10 = local c (kmol/m^3)
    UDM11 = local antisolvent volume fraction (solute free basis)
    UDM12 = local c* of A (kmol/m^3)
    UDM13 = local c* of B (kmol/m^3)
    UDM14 = local supersaturation, S=c/c* of A
    UDM15 = local supersaturation, S=c/c* of B
    UDM16 = nucleation rate, J of A (#/m^3-s)
    UDM17 = nucleation rate, J of B (#/m^3-s)
    UDM18 = size independent growth rate, G of A (m/s)
    UDM19 = size independent growth rate, G of B (m/s)
    UDM20 = G of A (micron/s)
    UDM21 = G of B (micron/s)
    UDM22 -- 41 = f0 -- f29 (A) (#/m-m^3)
    UDM42 -- 63 = fr-1 -- fr30, minmod gradient (A)
    UDM64 -- 83 = f0 -- f29 (B) (#/m-m^3)
    UDM84 -- 105 = fr-1 -- fr30, minmod gradient (B)

    Mean concentrations in CFD cell
    -------------------------------
    UDM106 = solute
    UDM107 = antisolvent
    UDM108 = solvent
    UDM109 -- 128 = f0 -- f29 (A) (#/micron-m^3)
    UDM129 -- 148 = f0 -- f29 (B) (#/micron-m^3)
*/

#include "udf.h"
#include "mem.h"
#include "prf.h"

/* constants */
#define p_tol 1.0e-5            /* minimum tolerance for volume fraction of envt 3 */
#define c_phi 4.0               /* micromixing constant */
#define sc_t 1.40               /* turbulent schmidt number */
#define diff_lam 1.0e-9         /* laminar diffusion coefficient */
#define turb_t_min 6.2e-5       /* minimum tolerance for turbulent time scale to
calculate scalar dissipation rate, changes with velocity */
#define rho_crystal 1440.0      /* density of crystal */
#define solute_MW 155.160       /* molecular weight for solute */
#define rho_antisolvent 790.0   /* density of antisolvent */
#define rho_solvent 998.20      /* density of solvent */
#define kv 0.6666666            /* volume shape factor of crystals */
#define ka 1.0                  /* area shape factor of crystals */
#define numspecies 43           /* solute, solvents, crystals */
#define ncell 20                /* # bins for population balance equation */
#define delr 2.0e-10            /* size of each bin, m */
#define theta 1.5               /* minmod constant */
#define n_uds 48                /* # of UDS */
#define n_udm 150               /* # of UDM */

/* boundary/zones ID*/
#define outletID 14
#define inlet1ID 16
#define inlet2ID 15
#define fluid_inlet1ID 5
#define fluid_inlet2ID 4
#define fluid_chamberID 2
#define fluid_outletID 3
#define wall_chamberID 6
#define wall_outletID 7
#define wall_inlet1ID 9
#define wall_inlet2ID 8

/*  mass concentrations (kg/m3) in envt 1 and 2
    Envt 1 = saturated solution, Envt 2 = antisolvent
    First entry is mixture fraction */
```

```
real phi1[numspecies+1] = {1.0, 42.51384, 0.0, 998.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
real phi2[numspecies+1] = {0.0, 9.61992, 316.0, 598.92, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
/* r (m) at ends of cell */
real rend[ncell+1] = {0.00E+00,
2.00E-10,
4.00E-10,
6.00E-10,
8.00E-10,
1.00E-09,
1.20E-09,
1.40E-09,
1.60E-09,
1.80E-09,
2.00E-09,
2.20E-09,
2.40E-09,
2.60E-09,
2.80E-09,
3.00E-09,
3.20E-09,
3.40E-09,
3.60E-09,
3.80E-09,
4.00E-09};
/* r(j+1)^4 - r(j)^4 */
real r4_diff[ncell] = {1.60000E-39,
2.40000E-38,
1.04000E-37,
2.80000E-37,
5.90400E-37,
1.07360E-36,
1.76800E-36,
2.71200E-36,
3.94400E-36,
5.50240E-36,
7.42560E-36,
9.75200E-36,
1.25200E-35,
1.57680E-35,
1.95344E-35,
2.38576E-35,
2.87760E-35,
3.43280E-35,
4.05520E-35,
4.74864E-35};

/* ------------------------------------------------------------------------------------
- */

real Get_gamma_s(cell_t c, Thread *t, real p1, real p2, real p3, real gamma);
real Get_source(cell_t c, Thread *t, int j);
real Get_rxn(cell_t c, Thread *t, int j);
real minmod(real fL, real f, real fR);
real Get_crystal_A(cell_t c, Thread *t, int j);
real Get_crystal_B(cell_t c, Thread *t, int j);
real Get_solubility_A(real as);
real Get_solubility_B(real as);
real Get_nucleation_A(real csat, real S, real T);
real Get_nucleation_B(real csat, real S, real T);
real Get_growth_A(real S, real as);
real Get_growth_B(real S, real as);


/* diffusivity for pn UDS eqn */
DEFINE_DIFFUSIVITY(pn_turb_diff, c, t, i)
{
    real turbdiff_rho;

#if !RP_HOST
    turbdiff_rho = C_MU_T(c,t) / sc_t + C_R(c,t) * diff_lam;    /* (turbulent
diffusivity * density) */
```

```
    return turbdiff_rho;
#endif
}

/* computes the crystallization rates and vol fraction of envt 3 for every iteration
*/
DEFINE_ADJUST(micromix, d)
{
    Thread *t, *tf;
    cell_t c;
    face_t f;
    real p1, p2, p3, mixfrac3;
    real solute3, antisolvent3, solvent3, fwj_3A, fj_3A[ncell+4], frj_3A[ncell+2],
fwj_3B, fj_3B[ncell+4], frj_3B[ncell+2];
    real cs, vol_solvent, vol_antisolvent, as, csat_A, csat_B, S_A, S_B, J_A, J_B,
G_A, G_B;
    int j, zoneID, n, crys_j;

#if PARALLEL
    PRF_GSYNC();
#endif
#if !RP_HOST
    thread_loop_f(tf, d)
    {
        zoneID = THREAD_ID(tf);

        if (THREAD_STORAGE(tf,SV_UDS_I(0))!= NULL) {
            begin_f_loop(f, tf)
            {
                p1 = F_UDSI(f,tf,0);
                p2 = F_UDSI(f,tf,1);
                p3 = F_UDSI(f,tf,2);

                if ((zoneID == wall_inlet1ID) || (zoneID == wall_inlet2ID)) mixfrac3 =
0.5;
                else {
                    if (p3 > p_tol) {
                        mixfrac3 = F_UDSI(f,tf,4) / p3;
                        if (mixfrac3 < 0.0) mixfrac3 = 0.0;
                    }
                    else mixfrac3 = 0.5;
                }
                F_UDSI(f,tf,3) = mixfrac3;
            }
            end_f_loop(f, tf)
        }
    }

    thread_loop_c(t,d)
    {
        zoneID = THREAD_ID(t);

        if ((zoneID == fluid_chamberID) || (zoneID == fluid_outletID)) {
            begin_c_loop_int(c,t)
            {
                p1 = C_UDSI(c,t,0);
                p2 = C_UDSI(c,t,1);
                p3 = C_UDSI(c,t,2);

                /* mixture fraction of envt 2 and 3 */
                if (p3 > p_tol) {
                    mixfrac3 = C_UDSI(c,t,4) / p3;
                    if (mixfrac3 < 0.0) mixfrac3 = 0.0;
                }
                    else mixfrac3 = 0.5;
                C_UDSI(c,t,3) = mixfrac3;

                /* Concentration and crystalllization rates in envt 3 */

                if (p3 > p_tol) {
                    /* conc in p3 */
                    solute3 = C_UDSI(c,t,5) / p3;
                    antisolvent3 = C_UDSI(c,t,6) / p3;
                    solvent3 = C_UDSI(c,t,7) / p3;

                    /* # distribution (f) of polymorph A and B envt 3 */
```

```
                    for (crys_j=0; crys_j<ncell; crys_j++) {
                        fwj_3A = C_UDSI(c,t,(8+crys_j)) / p3;
                        fwj_3B = C_UDSI(c,t,(28+crys_j)) / p3;
                        if (fwj_3A < 0.0) fwj_3A = 0.0;
                        if (fwj_3B < 0.0) fwj_3B = 0.0;
                        fj_3A[crys_j+2] = (4.0/(rho_crystal*kv)) * fwj_3A /
r4_diff[crys_j];
                        fj_3B[crys_j+2] = (4.0/(rho_crystal*kv)) * fwj_3B /
r4_diff[crys_j];
                    }
                }
                else {
                    solute3 = (phi1[1] + phi2[1]) / 2.0;
                    antisolvent3 = (phi1[2] + phi2[2]) / 2.0;
                    solvent3 = (phi1[3] + phi2[3]) / 2.0;
                    for (crys_j=0; crys_j<ncell; crys_j++) {
                        fwj_3A = (phi1[crys_j+4] + phi2[crys_j+4]) / 2.0;
                        fwj_3B = (phi1[crys_j+24] + phi2[crys_j+24]) / 2.0;
                        fj_3A[crys_j+2] = (4.0/(rho_crystal*kv)) * fwj_3A /
r4_diff[crys_j];
                        fj_3B[crys_j+2] = (4.0/(rho_crystal*kv)) * fwj_3B /
r4_diff[crys_j];
                    }
                }

                fj_3A[0] = 0.0;
                fj_3A[1] = 0.0;
                fj_3A[ncell+2] = fj_3A[ncell+1];
                fj_3A[ncell+3] = fj_3A[ncell+1];
                fj_3B[0] = 0.0;
                fj_3B[1] = 0.0;
                fj_3B[ncell+2] = fj_3B[ncell+1];
                fj_3B[ncell+3] = fj_3B[ncell+1];
                /* minmod gradient of f in p3 */
                for (crys_j=0; crys_j<=(ncell+1); crys_j++) {
                    frj_3A[crys_j] = minmod(fj_3A[crys_j], fj_3A[crys_j+1],
fj_3A[crys_j+2]);
                    frj_3B[crys_j] = minmod(fj_3B[crys_j], fj_3B[crys_j+1],
fj_3B[crys_j+2]);
                }

                /* concentration of solute in envt 3 kmol/m^3 */
                cs =  solute3 / solute_MW;

                /* antisolvent volume fraction on solute free basis in envt 3 */
                vol_solvent = solvent3 / rho_solvent;
                vol_antisolvent= antisolvent3 / rho_antisolvent;
                as = vol_antisolvent / (vol_antisolvent + vol_solvent);

                /* sat solute conc of A and B, kmol/m^3 in envt 3 */
                csat_A = Get_solubility_A(as);
                csat_B = Get_solubility_B(as);

                /* supersaturation in envt 3, concentration ratio */
                S_A = cs / csat_A;
                S_B = cs / csat_B;

                if (S_A > 1.0) {
                    J_A = Get_nucleation_A(csat_A, S_A, C_T(c,t));  /* Nucleation
rate, #/m3-s */
                    G_A = Get_growth_A(S_A, as);    /* Growth rate, m/s */
                }
                else {
                    J_A = 0.0;
                    G_A = 0.0;
                }

                if (S_B > 1.0) {
                    J_B = Get_nucleation_B(csat_B, S_B, C_T(c,t));
                    G_B = Get_growth_B(S_B, as);
                }
                else {
                    J_B = 0.0;
                    G_B = 0.0;
                }
```

```
                /* assign values into user-defined memory */
                C_UDMI(c,t,7) = solute3;
                C_UDMI(c,t,8) = antisolvent3;
                C_UDMI(c,t,9) = solvent3;
                C_UDMI(c,t,10) = cs;
                C_UDMI(c,t,11) = as;
                C_UDMI(c,t,12) = csat_A;
                C_UDMI(c,t,13) = csat_B;
                C_UDMI(c,t,14) = S_A;
                C_UDMI(c,t,15) = S_B;
                C_UDMI(c,t,16) = J_A;
                C_UDMI(c,t,17) = J_B;
                C_UDMI(c,t,18) = G_A;
                C_UDMI(c,t,19) = G_B;
                C_UDMI(c,t,20) = G_A * 1.0e6;
                C_UDMI(c,t,21) = G_B * 1.0e6;
                for (crys_j=0; crys_j<ncell; crys_j++) {
                    C_UDMI(c,t,(22+crys_j)) = fj_3A[crys_j+2];
                    C_UDMI(c,t,(64+crys_j)) = fj_3B[crys_j+2];
                }
                for (crys_j=0; crys_j<=(ncell+1); crys_j++) {
                    C_UDMI(c,t,(42+crys_j)) = frj_3A[crys_j];
                    C_UDMI(c,t,(84+crys_j)) = frj_3B[crys_j];
                }
            }
            end_c_loop_int(c,t)
        }
    }
#endif
#if PARALLEL
    PRF_GSYNC();
    EXCHANGE_UDMI(d,0,(n_udm-1));
#endif
}

/* computes p_3, conversion of DMP in each cell, weighted conc in envt 3, and
micromixing at every iteration.  */
DEFINE_EXECUTE_AT_END(conversion)
{
    Domain *d;
    Thread *t, *tf;
    cell_t c;
    face_t f;
    real p1, p2, p3, mixfrac3;
    real gammaD, gamma, mixvar, gamma_s, turb_t;
    real fj_outA[ncell], fj_outB[ncell], flow_out, fwj_outA, fwj_outB, flow_out_sum,
solute_out, antisolvent_out, solvent_out;
    real solute_out_conc, antisolvent_out_conc, solvent_out_conc, cs_out, as_out;
    real time, timestep;
    int j, crys_j, zoneID, ntime;
    FILE *fd;

#if PARALLEL
    PRF_GSYNC();
#endif
#if !RP_HOST
    d = Get_Domain(1);

    thread_loop_f(tf, d)
    {
        zoneID = THREAD_ID(tf);

        if (THREAD_STORAGE(tf,SV_UDS_I(0))!=NULL) {
            begin_f_loop(f, tf)
            {
                p1 = F_UDSI(f,tf,0);
                p2 = F_UDSI(f,tf,1);
                p3 = F_UDSI(f,tf,2);

                if ((zoneID == wall_inlet1ID) || (zoneID == wall_inlet2ID)) mixfrac3 =
0.5;
                else {
                    if (p3 > p_tol) {
                        mixfrac3 = F_UDSI(f,tf,4) / p3;
                        if (mixfrac3 < 0.0) mixfrac3 = 0.0;
                    }
```

```
                    else mixfrac3 = 0.5;
                }
                F_UDSI(f,tf,3) = mixfrac3;
            }
            end_f_loop(f, tf)
        }
    }
    thread_loop_c(t,d)
    {
        zoneID = THREAD_ID(t);

        if ((zoneID == fluid_chamberID) || (zoneID == fluid_outletID)) {
            begin_c_loop_int(c,t)
            {
                p1 = C_UDSI(c,t,0);
                p2 = C_UDSI(c,t,1);
                p3 = C_UDSI(c,t,2);
                C_UDMI(c,t,5) = 1.0 - p1 - p2;

                /* error in p3 */
                C_UDMI(c,t,6) = (p3 - C_UDMI(c,t,5));

                /* mixture fraction of envt 2 and 3 */
                if (p3 > p_tol) {
                    mixfrac3 = C_UDSI(c,t,4) / p3;
                    if (mixfrac3 < 0.0) mixfrac3 = 0.0;
                }
                else mixfrac3 = 0.5;
                C_UDSI(c,t,3) = mixfrac3;

                /* mixture variance */
                mixvar = p1 * (1.0-p1) - 2.0 * p1 * p3 * mixfrac3 + p3 * (1.0-p3) *
mixfrac3 * mixfrac3;
                C_UDMI(c,t,0) = mixvar;

                /* scalar dissipation rate */
                turb_t = C_K(c,t) / C_D(c,t);
                if (turb_t < turb_t_min) turb_t = turb_t_min;
                C_UDMI(c,t,1) = 0.5 * c_phi * mixvar / turb_t;

                /* micromixing rate */
                if (((p1 > p_tol)&&(p1<(1.0-p_tol))) || ((p2 > p_tol)&&(p2<(1.0-
p_tol))))  {
                    gammaD = ((p1 * (1.0-p1) * (1.0-mixfrac3) * (1.0-mixfrac3) + p2 *
(1.0-p2) * mixfrac3 * mixfrac3));
                    gamma = mixvar / gammaD;
                    gamma = gamma * 0.5 * c_phi / turb_t;
                    if (gamma < 0.0) gamma = 0.0;
                }
                else gamma = 0.0;
                if (p3 < 0.0) gamma = 0.0;
                /* gamma = c_phi * 0.5/turb_t;*/
                C_UDMI(c,t,2) = gamma;

                /* compute gradient of mixture fraction */
                if (T_STORAGE_R_NV(t,SV_UDSI_G(3)) != NULL) {
                    C_UDMI(c,t,3) = NV_MAG2(C_UDSI_G(c,t,3));
                }

                /* micromixing terms to eliminate spurious dissipation */
                gamma_s = Get_gamma_s(c,t,p1,p2,p3,gamma);
                C_UDMI(c,t,4) = gamma_s;

                /* Mean concentration in CFD cell */
                for (j=0; j<numspecies; j++) C_UDMI(c,t,(j+106)) = p1 * phi1[j+1] + p2
* phi2[j+1] + C_UDSI(c,t,(5+j));
                for (crys_j=0; crys_j<ncell; crys_j++) {
                    C_UDMI(c,t,(crys_j+109)) = 1.0e-6 * (4.0/(rho_crystal*kv)) *
C_UDMI(c,t,(crys_j+109)) / r4_diff[crys_j];
                    C_UDMI(c,t,(crys_j+129)) = 1.0e-6 * (4.0/(rho_crystal*kv)) *
C_UDMI(c,t,(crys_j+129)) / r4_diff[crys_j];
                }
            }
            end_c_loop_int(c,t)
        }
        if ((zoneID == fluid_inlet1ID) || (zoneID == fluid_inlet2ID)) {
```

```
            begin_c_loop_int(c,t)
            {
                p1 = C_UDSI(c,t,0);
                p2 = C_UDSI(c,t,1);
                p3 = C_UDSI(c,t,2);

                /* Mean concentration in CFD cell */
                for (j=0; j<numspecies; j++) C_UDMI(c,t,(j+106)) = p1 * phi1[j+1] + p2
* phi2[j+1] + C_UDSI(c,t,(5+j));
                for (crys_j=0; crys_j<ncell; crys_j++) {
                    C_UDMI(c,t,(crys_j+109)) = 1.0e-6 * (4.0/(rho_crystal*kv)) *
C_UDMI(c,t,(crys_j+109)) / r4_diff[crys_j];
                    C_UDMI(c,t,(crys_j+129)) = 1.0e-6 * (4.0/(rho_crystal*kv)) *
C_UDMI(c,t,(crys_j+129)) / r4_diff[crys_j];
                }
            }
            end_c_loop_int(c,t)
        }
    }
#if PARALLEL
    PRF_GSYNC();
#endif

    /* compute CSD (flow rate) at the outlet and write to file */
    tf = Lookup_Thread(d, outletID);

    for (crys_j=0; crys_j<ncell; crys_j++) {
        fj_outA[crys_j]=0.0;
        fj_outB[crys_j]=0.0;
    }
    flow_out_sum = 0.0;
    solute_out = 0.0;
    antisolvent_out = 0.0;
    solvent_out = 0.0;

    begin_f_loop(f,tf)
    {
        p1 = F_UDSI(f,tf,0);
        p2 = F_UDSI(f,tf,1);
        flow_out = F_FLUX(f,tf) / F_R(f,tf);
        flow_out_sum += flow_out;
        solute_out += flow_out * (p1 * phi1[1] + p2 * phi2[1] + F_UDSI(f,tf,(5)));
        antisolvent_out += flow_out * (p1 * phi1[2] + p2 * phi2[2] +
F_UDSI(f,tf,(6)));
        solvent_out += flow_out * (p1 * phi1[3] + p2 * phi2[3] + F_UDSI(f,tf,(7)));
        for (crys_j=0; crys_j<ncell; crys_j++) {
            fwj_outA = flow_out * (p1 * phi1[crys_j+4] + p2 * phi2[crys_j+4] +
F_UDSI(f,tf,(8+crys_j)));
            fwj_outB = flow_out * (p1 * phi1[crys_j+24] + p2 * phi2[crys_j+24] +
F_UDSI(f,tf,(28+crys_j)));
            fj_outA[crys_j] += 1.0e-6 * (4.0/(rho_crystal*kv)) * fwj_outA /
r4_diff[crys_j];
            fj_outB[crys_j] += 1.0e-6 * (4.0/(rho_crystal*kv)) * fwj_outB /
r4_diff[crys_j];
        }
    }
    end_f_loop(f,tf)

#if RP_NODE

    for (crys_j=0; crys_j<ncell; crys_j++) {
        fj_outA[crys_j] = PRF_GRSUM1(fj_outA[crys_j]);
        fj_outB[crys_j] = PRF_GRSUM1(fj_outB[crys_j]);
    }
    solute_out = PRF_GRSUM1(solute_out);
    antisolvent_out = PRF_GRSUM1(antisolvent_out);
    solvent_out = PRF_GRSUM1(solvent_out);
    flow_out_sum = PRF_GRSUM1(flow_out_sum);

#endif

    solute_out_conc = solute_out / flow_out_sum;
    antisolvent_out_conc = antisolvent_out / flow_out_sum;
    solvent_out_conc = solvent_out / flow_out_sum;
    cs_out = solute_out_conc / solute_MW;
```

```
    as_out = (antisolvent_out_conc/rho_antisolvent) /
((antisolvent_out_conc/rho_antisolvent) + (solvent_out_conc/rho_solvent));
    /* csat_out = Get_solubility(as_out);
    supersat_out = cs_out - csat_out;
    rel_supersat_out = cs_out / csat_out; */

    time = CURRENT_TIME;
    timestep = CURRENT_TIMESTEP;
    ntime = N_TIME;

#endif

    node_to_host_real(fj_outA, ncell);
    node_to_host_real(fj_outB, ncell);
    node_to_host_real_3(flow_out_sum, cs_out, as_out);
    node_to_host_real_2(time, timestep);
    node_to_host_int_1(ntime);

#if !RP_NODE
    fd = fopen("CSDoutlet.txt", "a");
    fprintf(fd, "%d %e %e %e A ", ntime, timestep, time, flow_out_sum);
    for (crys_j=0; crys_j<ncell; crys_j++)  fprintf(fd, "%e ", fj_outA[crys_j]);
    fprintf(fd, "B ");
    for (crys_j=0; crys_j<ncell; crys_j++)  fprintf(fd, "%e ", fj_outB[crys_j]);
    fprintf(fd, "\n");
    fclose(fd);

    fd = fopen("CONCoutlet.txt", "a");
    fprintf(fd, "%d %e %e %e %e %e \n", ntime, timestep, time, flow_out_sum, cs_out,
as_out);
    fclose(fd);
#endif
#if PARALLEL
    PRF_GSYNC();
    EXCHANGE_UDMI(d,0,(n_udm-1));
#endif
}

/* source term for pn and mixture fraction UDS eqn, based on Fox multienvironment
micromixing model */
DEFINE_SOURCE(p1_source, c, t, dS, eqn)
{
    real source, gamma, gamma_s, p1, p2, p3;

#if !RP_HOST
    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);
    gamma_s = C_UDMI(c,t,4);

    source = C_R(c,t) * (-gamma * p1 * (1.0-p1) + gamma_s * p3);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(p2_source, c, t, dS, eqn)
{
    real source, gamma, gamma_s, p1, p2, p3;

#if !RP_HOST
    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);
    gamma_s = C_UDMI(c,t,4);

    source = C_R(c,t) * (-gamma * p2 * (1.0-p2) + gamma_s * p3);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(p3_source, c, t, dS, eqn)
{
```

```
    real source, p1, p2, p3, gamma, gamma_s;

#if !RP_HOST
    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);
    gamma_s = C_UDMI(c,t,4);

    source = C_R(c,t) * (gamma * (p1*(1.0-p1) + p2*(1.0-p2)) - (p3*2.0*gamma_s));
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(mixfrac3_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=0;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

/* source term for weighted conc in envt 3 UDS eqn */
DEFINE_SOURCE(solute_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=1;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(antisolvent_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=2;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(solvent_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=3;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw0A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=4;
    source = Get_source(c,t,j);
```

```
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw1A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=5;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw2A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=6;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw3A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=7;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw4A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=8;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw5A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=9;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw6A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
```

```
    j=10;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw7A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=11;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw8A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=12;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw9A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=13;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw10A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=14;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw11A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=15;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw12A_source, c, t, dS, eqn)
{
    real source;
    int j;
```

```
#if !RP_HOST
    j=16;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw13A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=17;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw14A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=18;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw15A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=19;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw16A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=20;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw17A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=21;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw18A_source, c, t, dS, eqn)
{
```

```
    real source;
    int j;

#if !RP_HOST
    j=22;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw19A_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=23;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw0B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=24;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw1B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=25;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw2B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=26;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw3B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=27;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}
```

```
DEFINE_SOURCE(fw4B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=28;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw5B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=29;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw6B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=30;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw7B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=31;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw8B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=32;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw9B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=33;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
```

```
}

DEFINE_SOURCE(fw10B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=34;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw11B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=35;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw12B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=36;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw13B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=37;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw14B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=38;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw15B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=39;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
```

```
    return source;
#endif
}

DEFINE_SOURCE(fw16B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=40;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw17B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=41;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw18B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=42;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

DEFINE_SOURCE(fw19B_source, c, t, dS, eqn)
{
    real source;
    int j;

#if !RP_HOST
    j=43;
    source = Get_source(c,t,j);
    dS[eqn] = 0.0;
    return source;
#endif
}

real Get_gamma_s(cell_t c, Thread *t, real p1, real p2, real p3, real gamma) {

    real gamma_s, mixfrac3, grad_sq, term1, term2;

    if (p3 > p_tol) {
        mixfrac3 = C_UDSI(c,t,3);
        grad_sq = C_UDMI(c,t,3);
        gamma_s = 2.0 * C_UDSI_DIFF(c,t,0) * grad_sq / ((1.0-mixfrac3) * (1.0-
mixfrac3) + mixfrac3 * mixfrac3);
        term1 =  gamma * p1 * (1.0-p1) - gamma_s * p3;
        term2 = gamma * p2 * (1.0-p2) - gamma_s * p3;
        if ((term1 < 0.0) || (term2 < 0.0)) {
            gamma_s =  gamma * p1 * (1.0-p1) / p3;
            if (term2 < term1) gamma_s = gamma * p2 * (1.0-p2) / p3;
        }
    }
    else gamma_s = 0.0;
    if (gamma_s < 0.0) gamma_s = 0.0;
    /* gamma_s = 0.0; */
    return gamma_s;
```

```
}

/* Micromixing terms */
real Get_source(cell_t c, Thread *t, int j) {

    real source, gamma, gamma_s, p1, p2, p3, rxn;

    p1 = C_UDSI(c,t,0);
    p2 = C_UDSI(c,t,1);
    p3 = C_UDSI(c,t,2);
    gamma = C_UDMI(c,t,2);
    gamma_s = C_UDMI(c,t,4);

    if (p3 > p_tol) rxn = Get_rxn(c,t,j);
    else rxn = 0.0;

    source =  C_R(c,t) * (gamma * (p1*(1.0-p1)*phi1[j] + p2*(1.0-p2)*phi2[j]) -
gamma_s*p3*(phi1[j] + phi2[j]) + p3*rxn);

    return source;
}

/* Species source terms */
real Get_rxn(cell_t c, Thread *t, int j) {

    real rxn;
    int crys_j, crys_jj;

    if (j == 0) rxn = 0.0;  /* mixture fraction */
    if (j == 1) {           /* solute */
        rxn = 0.0;
        for (crys_jj=0; crys_jj<ncell; crys_jj++) rxn += -(Get_crystal_A(c,t,crys_jj)
+ Get_crystal_B(c,t,crys_jj));
    }
    if (j == 2) rxn = 0.0;  /* antisolvent */
    if (j == 3) rxn = 0.0;  /* solvent */
    if ((j >= 4) && (j < 24)) {
        crys_j = j - 4;
        rxn = Get_crystal_A(c,t,crys_j);    /* fw0 - fw29 of A */
    }
    if (j >= 24) {
        crys_j = j - 24;
        rxn = Get_crystal_B(c,t,crys_j);    /* fw0 - fw29 of B */
    }

    return rxn;
}

/* source terms for population balance for polymorph A */
real Get_crystal_A(cell_t c, Thread *t, int j) {

    real crystalsource, GL, GR, f, fL, fr, frL, J, supersat, frR, fR;

    GL = C_UDMI(c,t,18);
    GR = C_UDMI(c,t,18);
    f = C_UDMI(c,t,22+j);
    fr = C_UDMI(c,t,42+j+1);
    supersat = C_UDMI(c,t,14);

    if (supersat > 1.0) {
        frL = C_UDMI(c,t,42+j);
        if (j == 0) {
        fL = 0.0;
        J = C_UDMI(c,t,16);
        crystalsource = (rho_crystal*kv*0.25/delr) * r4_diff[j] * (-GR * (f +
0.5*delr*fr) + GL * (fL + 0.5*delr*frL) + J);
        }
        else {
            fL = C_UDMI(c,t,22+j-1);
            crystalsource = (rho_crystal*kv*0.25/delr) * r4_diff[j] * (-GR * (f +
0.5*delr*fr) + GL * (fL + 0.5*delr*frL));
        }
    }
    /*else if (supersat <= 1.0) {
        frR = C_UDMI(c,t,42+j+2);
        if (j == (ncell-1)) fR = f;
```

```
        else fR = C_UDMI(c,t,22+j+1);
            crystalsource = (rho_crystal*kv*0.25/delr) * r4_diff[j] * (-GR * (fR -
0.5*delr*frR) + GL * (f - 0.5*delr*fr));
    } */
    else crystalsource = 0.0;

    /* crystalsource = 0.0; */
    return crystalsource;
}

/* source terms for population balance for polymorph A */
real Get_crystal_B(cell_t c, Thread *t, int j) {

    real crystalsource, GL, GR, f, fL, fr, frL, J, supersat, frR, fR;

    GL = C_UDMI(c,t,19);
    GR = C_UDMI(c,t,19);
    f = C_UDMI(c,t,64+j);
    fr = C_UDMI(c,t,84+j+1);
    supersat = C_UDMI(c,t,15);

    if (supersat > 1.0) {
        frL = C_UDMI(c,t,84+j);
        if (j == 0) {
        fL = 0.0;
        J = C_UDMI(c,t,17);
        crystalsource = (rho_crystal*kv*0.25/delr) * r4_diff[j] * (-GR * (f +
0.5*delr*fr) + GL * (fL + 0.5*delr*frL) + J);
        }
        else {
            fL = C_UDMI(c,t,64+j-1);
            crystalsource = (rho_crystal*kv*0.25/delr) * r4_diff[j] * (-GR * (f +
0.5*delr*fr) + GL * (fL + 0.5*delr*frL));
        }
    }
    /* else if (supersat <= 1.0) {
        frR = C_UDMI(c,t,84+j+2);
        if (j == (ncell-1)) fR = f;
        else fR = C_UDMI(c,t,64+j+1);
            crystalsource = (rho_crystal*kv*0.25/delr) * r4_diff[j] * (-GR * (fR -
0.5*delr*frR) + GL * (f - 0.5*delr*fr));
    } */
    else crystalsource = 0.0;

    /* crystalsource = 0.0; */
    return crystalsource;
}

real minmod(real fL, real f, real fR) {

    real grad1, grad2, grad3, fr;

    grad1 = theta * (f - fL) / delr;
    grad2 = 0.5 * (fR - fL) / delr;
    grad3 = theta * (fR - f) / delr;

    if ((grad1>0.0) && (grad2>0.0) && (grad3>0.0)) {
        fr = grad1;
        if (grad2 < fr) fr = grad2;
        if (grad3 < fr) fr = grad3;
    }
    else if ((grad1<0.0) && (grad2<0.0) && (grad3<0.0)) {
        fr = grad1;
        if (grad2 > fr) fr = grad2;
        if (grad3 > fr) fr = grad3;
    }
    else fr = 0.0;

    return fr;
}

real Get_solubility_A(real as) {

    real csat;

    csat = -1.3333 * (as*as*as) + 1.8625 * (as*as) - 1.0242 * as + 0.256;
```

```
    return csat;
}

real Get_solubility_B(real as) {

    real csat;

    csat = -1.2083 * (as*as*as) + 1.775 * (as*as) - 1.0467 * as + 0.274;

    return csat;
}

real Get_nucleation_A(real csat, real S, real T) {

    real vo, AN, k, phi, A_hen;
    real ST, ST_eff, w, J;

    vo = 1.7892e-28;    /* m^3/molecule */
    AN = 6.0221415e23;  /* Avogadro's number */
    k = 1.3806503e-23;  /* Boltzmann constant */
    phi = 0.3;
    A_hen = 1.0e20;

    ST = 0.414 * k * T * (pow(vo, (-2.0/3.0))) * (log (1.0 / (1000.0*AN*vo*csat)));
    ST_eff = phi * ST;
    w = (16.0*3.141592654/3.0) * vo * vo * ST_eff * ST_eff * ST_eff / (k * T * k * T *
log(S) * log(S));
    J = A_hen * exp(-w / (k * T));

    return J;
}

real Get_nucleation_B(real csat, real S, real T) {

    real vo, AN, k, phi, A_hen;
    real ST, ST_eff, w, J;

    vo = 1.7892e-28;    /* m^3/molecule */
    AN = 6.0221415e23;  /* Avogadro's number */
    k = 1.3806503e-23;  /* Boltzmann constant */
    phi = 0.85*0.3;
    A_hen = 1.0e20;

    ST = 0.414 * k * T * (pow(vo, (-2.0/3.0))) * (log (1.0 / (1000.0*AN*vo*csat)));
    ST_eff = phi * ST;
    w = (16.0*3.141592654/3.0) * vo * vo * ST_eff * ST_eff * ST_eff / (k * T * k * T *
log(S) * log(S));
    J = A_hen * exp(-w / (k * T));

    return J;
}

real Get_growth_A(real S, real as) {

    real kg, g, G;

    kg = 3.676e-5 * as * as - 2.895e-5 * as + 5.769e-6;
    g = -47.0 * as * as + 16.458 * as + 3.3983;
    G = kg * pow((S-1),g);

    return G;
}

real Get_growth_B(real S, real as) {

    real kg, g, G;

    kg = -1.365e-6 * as * as - 2.736e-7 * as + 5.365e-7;
    g = -11.398 * as * as + 6.9075 * as + 1.4004;
    G = kg * pow((S-1),g);

    return G;
}
```