CFD analysis of reactive crystallisation in stirred tanks

a focus on nickel hydroxide

Andrew Richard Krige
BSc(Eng) University of Cape Town (1998)

A dissertation submitted to
Faculty of Engineering and the Built Environment at University of Cape Town
in partial fulfilment of the requirements for the degree
Master of Science
in
Chemical Engineering

March 2006

Abstract

The recovery of metal values from aqueous streams via precipitation, or reactive crystallisation, is both an economically and environmentally significant unit operation in a wide spectrum of industries. However, the control and operability of these systems is hampered by the limited fundamental basis for their design, frequently resulting in downstream processing difficulties.

Previous work by Dustan (2001) considered precipitation and subsequent dewatering as an integral system, and quantified, to a first order, the relationship between the various sub-processes. The current study is aimed at further resolving the upstream components of this integrated model (i.e. the precipitation itself), with a particular focus, due principally to the rapid kinetics typically exhibited by precipitation systems, on mixing and representation of time-dependent spatial inhomogeneities. Specifically, the primary objective is to develop a computational fluid dynamics (CFD) modelling capability, which, in addition to those conserving momentum and mass, incorporates models describing the solution chemistry and a discretised population balance (DPB).

The DPB of Hounslow et al. (1988), which is able to account for the processes of nucleation, growth and aggregation in detailing the evolution of the particle size distribution, is developed and modified. The dependence of the rates of these kinetic processes on the chemical environment demands accurate characterisation of the solution chemistry, requiring consideration of the full range of aqueous species. The non-linear system of thermodynamic relationships and mass balances (employing the Davies activity coefficient model to relate species concentrations and activities) which define the intraphase equilibrium state of the system is formulated. The methods of solution of the ordinary differential DPB and of the speciation calculations are established.

A new method for the integration of such models with the standard CFD calculations (predicting fluid flow and associated behaviours through numerical solution of the governing differential equations) is promoted in this work. Dictated largely by the inclusion of the modelling of complex solution chemistry, the approach is centred on the use of a user-defined Fortran subroutine which is called by the CFD solver prior to the run and at the end of each time step of the transient calculation. The extensive speciation and solids formation computations are encoded within this subroutine and performed in each of the several thousand cells in the finite volume mesh representing the flow domain, as are mass balances observing deposited material. The local, transient conditions required to perform these calculations dynamically are supplied by specifying an in-built CFD solver scheme such that transport equations for species molarities and particle number concentrations within each size class are solved either on their own using a fixed pre-calculated flow field or alongside those of the

hydrodynamic variables if a time-dependent flow is modelled. It is argued that transfer of the essential precipitation calculations to a user-defined subroutine currently allows increased model control and flexibility.

The preliminary application of the resultant modelling framework to the simulation of a real, non-ideal test system, viz. the semi-batch precipitation of nickel hydroxide from chloride solution, is demonstrated. The necessary model calibration and generation of computational representations of the experimental geometry and flow fields are effected using the laboratory data of Dustan (2001). Importantly, due to the fact that Dustan's experimental design was not conceived with the intention of supporting CFD modelling efforts, the extent and quality of the data is such that validation of the model was never the expectation. Nevertheless, the exercise enables some interesting insights into the system to be gained, two of the most notable being confirmation that, for practical modelling purposes, it is not reasonable to assume spatial uniformity and that an equilibrium description of nucleation is appropriate. It also provides information on the relative merits and deficiencies of both model and experiment, thereby facilitating an improved understanding of the way forward, and, significantly, it establishes the robustness, structural integrity and internal consistency of the overall model. It is reiterated that the fact that the current utility of the model does not necessarily exceed that of simpler approaches amounts to a limitation of the experimental data and not of the CFD modelling approach.

The contribution of this dissertation is considered to lie primarily in the development of a novel model architecture, encompassing the non-trivial integration of various model features envisaged as being of particular value in the study of multi-component precipitation systems which require rigorous aqueous thermodynamic modelling, similar to the one investigated here. The framework is of distinct generic value and is simple to refine, easily accommodating upgrades as greater fundamental understanding or processing power becomes available or as increased experimental resolution allows improved system characterisation. With specific respect to the test system, such experimental work is identified as a priority for future endeavours if the detailed knowledge of the spatial dynamics afforded by CFD modelling is to be exploited meaningfully.

Table of contents

List of figures

List of tables

Acknowledgements

I would like to extend my sincere appreciation to the following:

My supervisors, Professor Jim Petrie and Associate-Professor David Fletcher, for their vision and guidance, their expertise and practical assistance and their inspiration, patience and friendship. My postgraduate experience has been a truly challenging and ultimately enriching and rewarding odyssey in which not only my knowledge and critical thinking faculties, but also my horizons, have been broadened. I have benefited from opportunities for international travel and exposure and, tellingly, holistic consideration of my personal development has been favoured over expedience. I am aware that not all postgraduate students are as fortunate in this regard.

The vibrant and multicultural community of postgraduate students at both the University of Cape Town and the University of Sydney, for the education and the memories. Special mention must go to my fellow members of the respective Environmental Process Engineering Research Groups for the supportive, stimulating and, most importantly, fun environment in which to work; here, it would be remiss of me not to single out the roles of the likes of Andrew Dustan, Mary Stewart and Lauren Basson.

Yvonne, for your (almost) unwavering patience and support, both practical and emotional. Without your encouragement, belief and input I may well not have seen this journey through.

The financial assistance of the NRF is also gratefully acknowledged, as are the contributions of those whom I have no doubt overlooked here.

Chapter

1

Introduction

1.1 Background

Due to the toxic nature of many metal ions, the recovery of metal values from aqueous streams via precipitation, or reactive crystallisation, is both an economically and environmentally significant technology in a wide spectrum of industries. However, the limited fundamental basis for the design of these processes leads to frequent difficulties in subsequent processing. For example, solid-liquid separation difficulties following precipitation (particularly of hydroxides) are common. Problems, the result of undesirable precipitate characteristics such as particle morphology and particle size distribution (PSD), include slow and unpredictable settling and filtration and high moisture content sediments. These, in turn, translate to higher equipment and operating costs, solids handling difficulties and loss of process water.

It is well accepted that the choice of precipitation operating conditions affects the product characteristics and hence determines the dewatering behaviour (Franke & Mersmann 1995). As intimated above, however, there is an absence of a fundamental description of the manner in which this occurs. Typically, consideration of, amongst other things, the relative rates of the various precipitation processes, merely allows the establishment of general operational strategies for both bulk and continuous installations. Such qualitative guidelines (e.g. Mersmann 1999) aim to exploit these intrinsic features in an effort to deliver the desired product attributes, an essentially heuristic device of limited value in meeting various design and operability objectives. Previous work by co-worker Andrew Dustan (Dustan 2001, Dustan et al. 2005a, Dustan et al. 2005b) recognised the need to quantify explicitly the relationships between design and operating variables and the overall system performance. Dustan's approach considered precipitation and subsequent dewatering as an integral system, arguing that the level of understanding within hitherto discrete disciplines was under-utilised

for practical attempts at modelling the overall system. Through identification of the different component processes and exploration of the inter-relationships between them, an integrated model framework was developed aimed at providing a tool to inform design and facilitate the optimisation, operation and control of new and existing reactor-separator units.

1.2 Objectives

The objective of this study is to further resolve the upstream components of the integrated model of Dustan (2001) (i.e. the precipitation itself, whose task it is ultimately to define a set of suspension characteristics) and, in so doing, to establish the potential benefits thereof. Particular focus is placed on mixing and the desired representation of time-dependent spatial inhomogeneities. As such, the precise aims are to develop a computational fluid dynamics (CFD) modelling capability, to integrate within the CFD platform a fully speciated kinetic model and discretised population balance (specific to the requirements of a test system) and to demonstrate the application of the resultant model superstructure to this system, thereby permitting its efficacy, practical utility and, more importantly within the context of this work, its structural integrity and internal consistency to be established. The intent is that the superstructure have distinct generic qualities and be of value in the study of similar systems, which typically exhibit complex solution chemistry. To the best of the author's knowledge, such an integrated modelling platform has not been available hitherto (though the research group of Mike Hounslow and colleagues at the University of Sheffield in the UK is actively pursuing this goal using test systems (Hounslow et al. 2005)).

The test system, a 'difficult' one and one which is of considerable interest in its own right, is the same as that employed by Dustan (2001), viz. the semi-batch precipitation of nickel hydroxide. It is important to note here that Dustan's experimental design was not conceived with CFD validation or detailed system characterisation in mind and that the extent and quality of the associated data is hence unlikely to be adequate to support CFD modelling efforts satisfactorily. Having recourse only to this laboratory data, it is thus not the expectation that the model be validated. Since the particulars of the test system necessarily inform the model features, the development of the model framework is, however, presented, for the sake of simplicity, alongside the model calibration and application (as outlined in Sec. 1.3). Although, realistically, it is not the intention to perfectly resolve the system under investigation or provide significant additional insight into it through comprehensive reconciliation of simulation and experimental results, in juxtaposition, information on the relative merits and deficiencies of model and experiment, as well as greater appreciation of the way forward, may hopefully be gained.

1.3 Dissertation structure

A review of the relevant theory and literature is presented in Chap. 2. This incorporates content covering the topics of solution thermodynamics, precipitation kinetics, the population balance and mixing, including a focus on CFD and its application to stirred tanks and precipitation systems. Where applicable and/or available, the material is discussed within the context of metal hydroxide precipitation.

Chap. 3 effectively addresses the issues surrounding the availability, quality and necessary manipulation of the experimental data at hand. What follows in Chaps. 4 and 5 is a documentation of the development of an integrated model structure aimed at ultimately being of value in endeavours to simulate precipitation systems similar to the one investigated in this work. Chap. 4 details the models of solution chemistry and the population balance and the specifics and results of the rate extraction exercise, all of which are then integrated with the CFD modelling efforts of Chap. 5. It should be noted here that the extensively annotated code files presented in App. A are considered to be a resource of central importance to this chapter.

Selected, meaningful simulation results are recorded and discussed in Chap. 6, thereby establishing the efficacy and structural integrity of the developed models. Finally, the various findings and developments of value are summarised in Chap. 7, together with recommendations for continued study in the field.

Chapter

2

Review of relevant theory and literature

2.1 Complexities of metal hydroxide precipitation

Precipitation, or reactive crystallisation, is used widely in the chemical processing industries. Precipitation of metal hydroxides specifically is also prevalent, either as a means to produce a solid product with an intrinsic economic value (e.g. the production of $Al(OH)_3$ from bauxite ores) or as part of waste water treatment processes concerned with the removal of unwanted metal species, often as a hydroxide, from process streams. In both cases, the properties of the precipitated particles are of key importance and operators must pay considerable attention to 'particle design' (Söhnel & Garside 1992).

Design criteria include particle size distribution, density and shape of product particles, crystal form (including hydration state), particulate shear and compressive yield stresses, as well as particle surface properties, including surface potential. Of these, the particle size distribution and particle morphology are particularly significant in determining the performance of downstream particle processing. The ease of solid-liquid separation including filterability, suspension rheological behaviour and settling and drying rates, for example, are all critically dependent on these properties. In addition, the chemical composition of the precipitate in terms of purity and crystal form can also be significant, especially if the product has economic value.

The dearth of fundamental understanding outlined in Sec. 1.1 means meeting 'particle design' criteria in metal hydroxide systems is (as it is in other precipitation systems) particularly challenging. There are frequently deviations in the performance of precipitation technology in time and space, in specific applications and in comparison with chemical theory (Patterson 1987). This can be attributed, in part,

to the fact that precipitation processes are still poorly characterised in comparison with more 'ideal' crystallisation systems – $BaSO_4$ being the classic example and the subject of many studies in the literature. A distinguishing feature of precipitation processes, that to a certain extent accounts for their 'non-ideality', is that they are generally initiated at high supersaturation, resulting in fast nucleation that yields large numbers of very small primary crystals (Mullin 2001). This rapid nucleation occurs simultaneously with particle growth and the process is complicated further by the presence of secondary processes of aggregation and ageing. These fundamental processes are difficult to separate and investigate independently. This is only exacerbated by the lack of experimental techniques available to accurately detect particles smaller than 1μm (van Leeuwen 1998). Typical features of metal hydroxide systems specifically, which further complicate their study, are complex solution chemistry presenting a wide range of aqueous species and the precipitation of amorphous, highly disordered hydroxide matrices which result in sticky, colloidal suspensions (Patterson 1987).

Furthermore, precipitation processes and resulting particle properties are greatly influenced by the manner and intensity of reactant solution mixing. Frequently the precipitation time scales are of the order of mixing time scales resulting in local regions of high supersaturation. The strong dependence of nucleation, growth and aggregation rates on solution conditions means that spatial heterogeneities must be quantified for the precipitation behaviour to be fully understood and 'particle design' to be effected. Accordingly, given that plant operators can usually control only liquid feed rate, concentration, residence time and agitation in their system, there is a real need to explore the fundamental way in which these operational variables define system performance.

The modelling of metal hydroxide precipitation systems thus requires, at the very least, an understanding of mixing, solution speciation and nucleation, growth and aggregation mechanisms of solids formation. The latter two concepts are discussed below. It should be noted that topics in the remaining sections, as far as possible and where relevant, are addressed with specific reference to metal hydroxides.

2.2 Precipitation thermodynamics and kinetics

The characteristic features of precipitation, which are also particular to metal hydroxide precipitation systems, can be summarised as follows (Söhnel & Garside 1992):

- precipitation systems generally involve only sparingly soluble materials and their low solubility leads to the development of high levels of supersaturation;
- nucleation plays a major role in the precipitation process since the characteristically elevated supersaturation levels result in very high nucleation rates;
- high nucleation rates, in turn, give rise to the production of large numbers of relatively small crystals;
- the small size of the precipitated particles may lead to the occurrence of several secondary

processes, including aggregation and ageing, which can further lead to major changes in the precipitate size distribution and the particle morphology;
the supersaturation necessary for precipitation and the nucleation itself both result from chemical reactions which are typically very fast, and, as such, the role of mixing cannot be overstated.

This section considers the thermodynamic driving force for precipitation, expressed in terms of supersaturation, and the kinetics of the various precipitation sub-processes that determine the final precipitate product characteristics. Mixing is addressed in Sec. 2.3.

2.2.1 The importance of speciation and supersaturation

The chemistry in precipitation systems is complex, the result of interactions between the components in the reacting solutions. Ions in solution will interact with each other to form complex species and with solution molecules to form a range of hydrated species. Even for apparently simple reaction systems, many complexation reactions exist (Dustan 2001). It is thus important to pay close attention to the speciation of the system as the solubility of the precipitating species may be significantly affected by these reactions.

2.2.1.1 Estimation of species activity

In order to quantify the solubility of the precipitating species, and to determine the thermodynamic driving force for precipitation, it is necessary to calculate the activity coefficients of species in solution. In ideal systems, the activity may be taken as the concentration in solution. The asymmetric convention means that the activity coefficient of the species approaches unity as the concentration of the species approaches zero (infinite dilution).

In solutions containing ionic species, the forces between charged solute ions operate over a much longer range than those associated with the interactions of neutral molecules. Consequently, even at very low concentrations, electrolyte solutions are very non-ideal. As such, the activities of the components can not be taken as equal to their concentrations. Thus, in electrolyte systems, the activities of dissolved species, a_i, and molalities, M_i, are related through activity coefficients, γ_i, to account for thermodynamic non-idealities, such that

$$a_i = \gamma_i M_i . \tag{2.1}$$

For sparingly soluble compounds at supersaturation, experimental estimation of activity coefficients is hampered by the instability of the solution (Söhnel & Garside 1992). It is therefore necessary to default to theoretical estimates or evaluations based on appropriate correlations. These methods exhibit varying degrees of sophistication with greatest divergence occurring at higher ionic strengths (Dustan 2001). Ionic strength, I, is defined as:

$$I = 0.5\sum_i M_i z_i^2 \,. \tag{2.2}$$

Here, z_i is species charge and the summation is over all ions in solution.

Many activity coefficient models are based on the theory of interionic interaction put forward by Debye & Hückel (1923). This theory, and associated Debye-Hückel activity models, account for 'long-range' interactions between species where the constituent species are relatively far apart and the activity coefficient is mostly influenced by the general charged environment (Rafal et al. 1994). The Debye-Hückel limiting law is given by

$$\log \gamma_i = -A|z_+ z_-|\sqrt{I}\,, \tag{2.3}$$

where A is the temperature dependent Debye-Hückel constant, which is defined as

$$A = \frac{1}{2.303}\left(\frac{e}{\sqrt{D_e k_B T}}\right)^3 \sqrt{\frac{2\pi\rho_0 N_A}{1000}} = 1.327757\times 10^5 \frac{\sqrt{\rho_0}}{(D_e T)^{3/2}}\,. \tag{2.4}$$

This equation contains the constants e, the electronic charge, k_B, Boltzmann's constant and N_A, Avogadro's number, whilst ρ_0 is the solvent density, D_e the dielectric constant and T absolute temperature.

As this limiting law is only valid for very dilute solutions (typically $I < 0.001$ mol.kg$^{-1}_{H_2O}$) it was extended in an attempt to improve the representativeness of this equation. The extended Debye-Hückel relationship, which yields satisfactory estimates up to ionic strengths of 0.1 mol.kg$^{-1}_{H_2O}$, is of the form:

$$\log \gamma_i = -\frac{A|z_+ z_-|\sqrt{I}}{1 + aB\sqrt{I}}\,. \tag{2.5}$$

Further correlations applicable to higher ionic strengths are detailed extensively in Zemaitis et al. (1986) and are summarised in Cairncross (1998). These expressions include a 'short-range' term which typically sums a number of individual ion-ion and ion-molecule pair interaction terms (Rafal et al. 1994). State-of-the-art speciation models in the form of commercially available software packages, such as OLI™, represent the pinnacle of speciation model sophistication and comprehensiveness and are applicable to the high ionic strengths and complex solutions typically encountered in industrial processes. These more sophisticated models come complete with a database of thermodynamic data, including equilibrium constants, coefficients and ion interaction parameters.

2.2.1.2 Supersaturation

The thermodynamic driving force for precipitation is the difference between the chemical potentials of two chemical states: the supersaturated solution (where the solute is present at levels exceeding equilibrium solubility) and the solution at equilibrium with the precipitated solute. While the thermodynamics are covered comprehensively elsewhere, e.g. Sandler (1989), it is sufficient to state

here that central to the determination of this driving force is the solute supersaturation, S, which for the precipitation reaction $\nu_+A^{z+}(aq) + \nu_-B^{z-}(aq) \leftrightarrow A_{\nu_+}B_{\nu_-}(s)$ may be defined as

$$S = \frac{(a_{A^{z+}})^{\nu_+}(a_{B^{z-}})^{\nu_-}}{(a_{A^{z+}})^{\nu_+}(a_{B^{z-}})^{\nu_-}\Big|_{eq}} = \frac{(a_{A^{z+}})^{\nu_+}(a_{B^{z-}})^{\nu_-}}{K_{sp}}. \tag{2.6}$$

Here, the temperature-dependent constant K_{sp} is referred to as the solubility product.

Indeed, supersaturation is considered the key variable – even the 'supreme governing parameter' (Demopoulos 1993) – for the precipitation process. Fig. 2.1 shows the extent of influence of supersaturation on the precipitation process. Nucleation, growth and transformation kinetics, which largely control precipitation and determine the resulting particle size distribution (PSD) of the precipitate product, are all dependent on supersaturation to varying degrees. Further, transformation kinetics dictate the conversion of precipitate from polymorphs or hydrates to more crystalline forms, while the position of the system in the appropriate phase diagram also depends on the saturation level (Söhnel & Garside 1992).

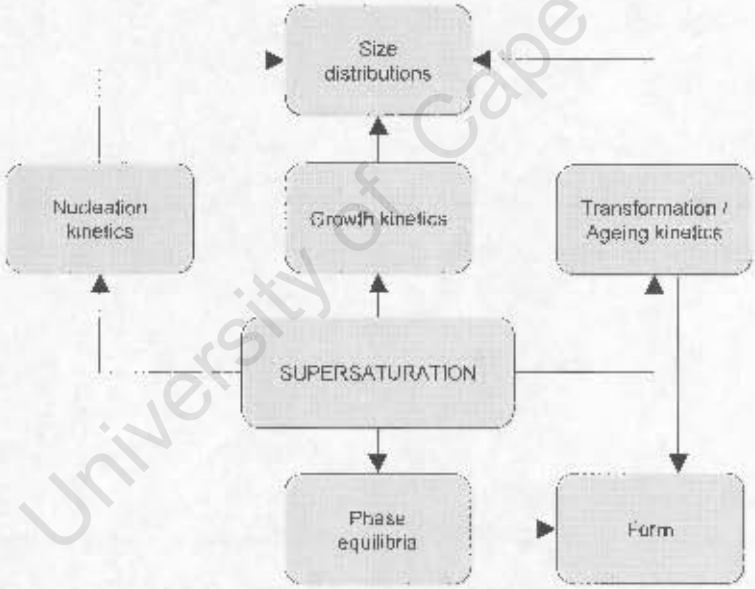

Figure 2.1 The role of supersaturation in precipitation processes (adapted from Söhnel & Garside (1992))

Due to its influence on all aspects of the precipitation process, there exists considerable motivation for the accurate and dynamic estimation of the solution supersaturation, and hence the solution chemistry, in any attempt to effectively simulate precipitation behaviour. This is particularly true for the precipitation of sparingly soluble compounds where large variations in supersaturation (either over time or spatially due to mixing) are encountered.

2.2.2 Nucleation and growth

As stated in the introduction to this section, precipitation in metal hydroxide systems is characterised by high rates of nucleation brought about by the high supersaturation encountered upon mixing. However, supersaturation alone, although necessary, is not sufficient to cause the onset of precipitation. An additional requirement is the presence in solution of minute embryos or nuclei that act as centres of crystallisation (Mullin 2001).

2.2.2.1 Nucleation

Nucleation may be either defined as primary, occurring at high supersaturations in systems with no crystalline matter initially present, or secondary, occurring at low to moderate supersaturations and requiring the presence of parent crystals of the solute. Primary nucleation may be further delineated into homogeneous and heterogeneous nucleation. Heterogeneous nucleation is induced by the presence of a foreign substrate, often microscopic dust particles or trace impurities, while in homogeneous nucleation such a substrate is absent and new phase formation is spontaneous – induced by statistical fluctuations of solute entities clustering together (van Leeuwen 1998). In the precipitation of sparingly soluble substances, secondary nucleation is either absent or insignificant and median particle size is mainly affected by the primary nucleation rate which this discussion is limited to (Franke & Mersmann 1995).

Fundamental theory describing the kinetics of nucleation is detailed in Söhnel & Garside (1992). In general, the nucleation rate, J with units of $m^{-3}.s^{-1}$, is a function of surface energy, γ^*, temperature and supersaturation expressed in the form of an Arrhenius type equation, which for primary nucleation is represented similarly to the following equation:

$$J = AS \exp\left(-\frac{16\pi(\gamma^*)^3 V_m^2}{3k_B^3 T^3 (\ln S)^2}\right). \tag{2.7}$$

Typically, changes in S in the pre-exponential factor AS are regarded to be of minor influence compared to changes in S in the exponential factor. Different expressions for the pre-exponential factor developed by different researchers, for both homogeneous and heterogeneous primary nucleation, can be found in Mersmann (1995) and Söhnel & Garside (1992).

When plotted against supersaturation, the above equation for nucleation rate, with appropriate modifications to represent heterogeneous and homogeneous primary nucleation, yields the following trends exhibited in Fig. 2.2.

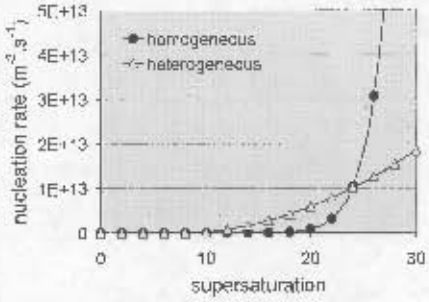

Figure 2.2 An example of relative homogeneous and heterogeneous nucleation rates (adapted from van Leeuwen (1998) for the ideal barium sulphate system)

While the fundamental theory regarding nucleation is extensive, evaluation of the quantities in the rate equations is difficult, particularly for the highly non-ideal precipitating systems where disordered, amorphous or metastable phases may form first (Dustan 2001). In addition, Mullin (2001) questions the dependence of the nucleation rate on surface energy, a quantity that has little perceived significance, particularly for small molecular aggregates of critical nucleus size. In general, little more than qualitative agreement with these fundamental descriptions can be found in the literature. On a more practical level, the strongly non-linear behaviour of the exponential relation for nucleation (Fig. 2.2) can sometimes pose computational difficulties and instabilities (van Leeuwen 1998).

Whilst bearing this in mind, in practical applications use is often made of a semi-empirical description of nucleation rates, typically with a power law dependence on supersaturation, such that

$$J = k_n (S-1)^n \,. \tag{2.8}$$

In the above equation, the nucleation constant, k_n, and the kinetic order for nucleation, n, are empirically determined. It should be further noted that k_n has no physical significance, but can be expanded to reflect its functional dependence on temperature, hydrodynamics, presence of impurities, particle number or surface area (Tavare 1987).

As nucleation only occurs above a certain critical supersaturation level (see Fig 2.2), it is prudent to determine the range over which the above power law equation is a valid approximation of the nucleation rate, especially in systems which exhibit large variations in supersaturation. The critical supersaturation level is often defined as the value of S at which $J = 1$, 10 or 100 cm^{-3}.s^{-1}.

2.2.2.2 Growth

Crystal growth occurs as solute is deposited onto existing nuclei or particles. Growth involves a number of diffusion, adsorption, reaction and counter diffusion steps which are described fully in Dirksen and Ring (1991). Usually the kinetic growth rate expression is based on the rate-limiting step in the growth mechanism chain. While various growth models have been developed, in most practical

applications, a power law relation is again appropriate and convenient to use as an approximation of growth rate:

$$G = k_g (S-1)^g . \tag{2.9}$$

Here, once again, the constant k_g may be correlated to a number of applicable parameters, including temperature, measures of hydrodynamic conditions and, in some cases, particle size (Tavare 1987). The order in the growth rate power law relationship, g, is typically between 0 and 2.5, with values of unity commonplace. This approximation is seen to deliver a satisfactory mathematical description of growth over ranges of supersaturation where the rate-limiting step in the growth mechanism is unchanged (Dirksen & Ring 1991).

2.2.2.3 Relative nucleation and growth rates in metal hydroxide systems

It has already been stated that the high supersaturation levels encountered in precipitation systems results in the formation of large numbers of small primary particles. This suggests that precipitation is not only characterised by high nucleation rates, as already established, but that growth rates are relatively insignificant. Moreover, Demopoulos (1993) states that for sparingly soluble metal hydroxides, the dependency of nucleation on supersaturation is of the order 6 – 12, and, for growth rates, only 1 – 2. By superimposing these characteristic rate profiles on to the same set of axes, as demonstrated in Fig. 2.3, it is possible to identify regions of nucleation-controlled and growth-controlled precipitation. Plots such as these for real systems may guide the control of supersaturation for operation in the growth-controlled regime where necessary to meet specified particle design requirements (Demopoulos 1993, Demopoulos et al. 1995).

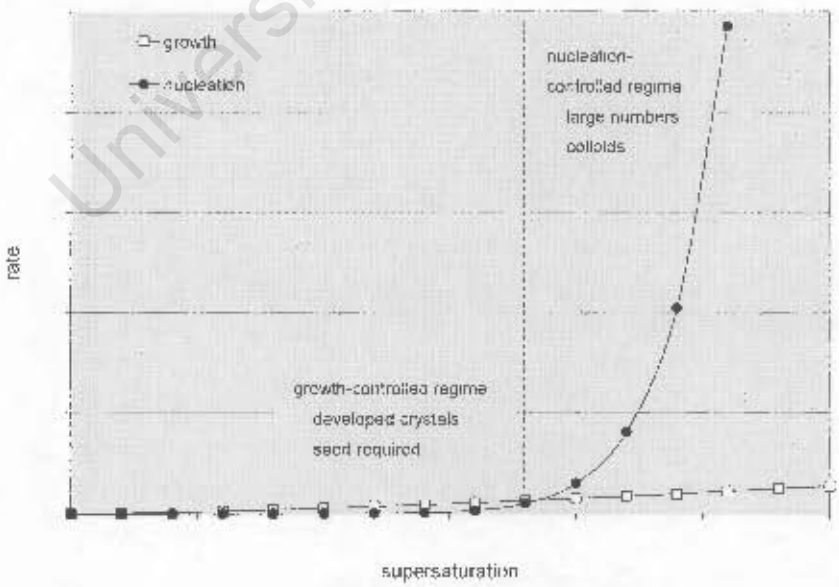

Figure 2.3 Characteristic nucleation and growth regimes during precipitation of metal hydroxides (adapted from Demopoulos (1993)).

However, this plot also provides further evidence to support the notion that growth is insignificant in metal hydroxide precipitation systems; this for the following reasons (as summarised by Dustan (2001)) which make operation in the growth-controlled regime difficult to implement:

> Firstly, the solubility of metal hydroxides is generally very low, making the generation of high supersaturation, above critical nucleation values, hard to avoid.
>
> Secondly, growth is very slow and thus responsible for correspondingly slow reduction of solution supersaturation, a particular problem when the critical values for nucleation have been exceeded.
>
> Finally, nucleation rates can be very high, of the same order, or even shorter, than mixing times. This results in the generation of localised pockets of high supersaturation – even in cases where calculated 'mean' values predict a supersaturation level below those critical for nucleation. In fact, Franke & Mersmann (1995) report variations in nucleation rates of up to 10 orders of magnitude as a result of local supersaturation fluctuations alone.

2.2.3 Aggregation and other secondary processes

This chapter has thus far discussed the three basic features common to crystallisation processes: supersaturation, nucleation and growth (Mullin 2001). In addition to these are a number of secondary processes that may have a profound effect on the final precipitate properties. The first is aggregation, which generally occurs soon after nucleation and has a significant effect on the resulting particle size distribution. The number of other processes with the potential to further alter the chemical and physical properties of the precipitate include recrystallisation, ripening and ageing (the latter often being the collective term for these processes).

2.2.3.1 Aggregation and the aggregation kernel

Aggregation is a reversible process in which particles dispersed in solution collide, adhere to one another and become cemented together to form new larger particles, which may then aggregate further. Within the literature, the terms aggregation, agglomeration and coagulation are often used interchangeably, and while the terminology for these processes has yet to be unified, the following distinctions can be made:

> Aggregation refers to the clustering of separate particles to form larger particles. The mechanisms controlling the rate of this process are varied and the particles may be held together by a number of forces.
>
> Agglomeration refers to the clustering of particles already formed to form relatively strong secondary particles held together by crystalline bridges but also occasionally by physical forces.

Coagulation is a specific case of agglomeration involving very small particles with physical forces holding the clusters together. Flocculation is a type of coagulation where the bridging forces are artificially induced through, for example, the addition of flocculants.

Collisions between particles occur as a result of their motion relative to each other due to Brownian motion induced by thermal energy; velocity gradients in mechanically agitated suspensions; and differential settling of individual particles (Hogg 2000). Mechanisms of particle collision are size dependent, with Brownian motion being the dominant mechanism for very small particles (i.e. those less than 0.5μm). As particles aggregate to form larger units, agitation effects begin to dominate and collisions occur as a result of particle movement with the fluid in laminar or turbulent regimes or relative to the fluid due to gravitational or inertial forces. The tendency for particles to aggregate also depends on the attractive and repulsive forces at play; the main attractive force between particles is the van der Waals force, while the main repulsive force arises as a consequence of the electrostatic interactions within the double layer surrounding the particles (the so-called electrical double layer).

Fundamental aggregation theory stems from the work of von Smoluchowski (1916) who considered aggregation to be the result of binary interactions, the rate of which can be taken as proportional to the number of interacting particles. While von Smoluchowski's theory provides insight into the behaviour of aggregating systems, its application, in the form of semi-empirical relationships, is restricted to a few simple 'limiting' cases. It has also been argued that the deterministic models developed are incapable of fully describing the complexities associated with particle aggregation – a process which involves transport and collisions, interaction of particles, rupture of aggregates and cementation of aggregates into stable agglomerates – satisfactorily (Söhnel and Garside 1992). Furthermore, in precipitating systems, the particles often carry a surface charge, resulting in a repulsive force not accounted for in the classical theories (van Leeuwen 1998).

As a result, it is again necessary to take a more empirical approach to aggregation when modelling precipitation systems, through the application of what is known as the aggregation kernel. In this approach, the rate of aggregation of particles of size $[L, L + dL]$ with particles of size $[\lambda, \lambda + d\lambda]$ in a multi-particulate system is characterised by the aggregation kernel, $\beta(L,\lambda)$, which is a measure of the frequency with which a particle of size L collides with one of size λ to form a viable aggregate. The aggregation kernel is often expressed as the product of two factors – a size-independent kernel, β_o, and a size-dependent kernel (Sastry 1975). It is generally assumed that the size-dependent kernel is constant and not a function of solution conditions. In contrast, the size-independent kernel, while not being a function of size, is a function of the chemical and/or physical environment (i.e. variables such as supersaturation, local energy dissipation, temperature etc.).

$$\beta(L,\lambda) = \beta_o \times f(L,\lambda). \qquad (2.10)$$

Common aggregation kernels that reflect different collision mechanisms are listed in Table 2.1. In addition, other kernels are presented that are purely empirical or analytically convenient, including the

size-independent kernel and the linear, quadratic and cubic sum kernels. The choice of aggregation kernel is usually based on comparison of kernel performance with experimental data. Interestingly, the size-independent kernel, which coincidentally is the simplest to implement, has been found in some precipitation studies to give the best agreement with experimental results (e.g. Bramley et al. 1996, Ilievski & White 1994).

Table 2.1 Common aggregation kernels (adapted from Bramley et al. (1996) and Ginter & Loyalka (1996))

Mechanism	Aggregation kernel
Size independent	β_0
Brownian motion	$\beta_0 (L+\lambda)(L^{-1}+\lambda^{-1})$
Gravitational	$\beta_0 (L+\lambda)^2 \lvert L-\lambda \rvert$
Shear	$\beta_0 (L+\lambda)^3$
Particle inertia	$\beta_0 (L+\lambda)^2 \lvert L^2-\lambda^2 \rvert$
Thompson kernel (empirical)	$\beta_0 (L^3+\lambda^3)^2 / (L^3-\lambda^3)$
Linear sum	$\beta_0 (L+\lambda)$
Quadratic sum	$\beta_0 (L^2+\lambda^2)$
Cubic sum	$\beta_0 (L^3+\lambda^3)$

In some circumstances, the chosen aggregation kernel can lead to the prediction of particles with infinite size known as gels (Smit et al. 1994, 1995). This occurs when a kernel predicts an increasing aggregation rate with increasing particle size. Thus, as aggregation proceeds both the size of the particles and the frequency with which they collide increase, leading eventually to the formation of particles of 'infinite' size. The cubic kernel has been seen to produce such results. While essentially a consequence of the mathematical formulation of the aggregation kernel, gelation has been observed experimentally for the aggregation of colloids (Spanhel & Anderson 1991).

2.2.3.2 Aggregation in metal hydroxide systems

In metal hydroxide systems, where growth is negligible, aggregation is the most important size enlargement mechanism. For example, Pavlides (1995) reported that primary growth played an insignificant role in particle size enlargement compared with aggregative growth for the iron(III) hydroxide precipitation system. Similarly, Ang & Mullin (1979) observed rapid aggregation during the precipitation of $Mg(OH)_2$ from $MgCl_2$ solutions. Ageing and deaggregation, the latter resulting from shear and turbulent stresses, followed this initial period of aggregation.

Ilievski & White (1994) identified agglomeration as the main process of particle enlargement for industrial $Al(OH)_3$ precipitation. They proposed the following mechanism for the development of

stable agglomerates during $Al(OH)_3$ precipitation: loosely bound flocs form due to shear induced collisions; subsequent cementation of floes occurs through deposition of $Al(OH)_3$ from solution; stable agglomerates are eventually formed through continued cementation.

Dustan (2001) identified, through experimental observation and analysis, aggregation of nuclei as the primary method of size enlargement for the precipitation of $Ni(OH)_2$ from both chloride and sulphate solutions. However, in contrast to the mechanism proposed by Ilievski & White (1994), substantial cementation of the aggregates is not observed in this system. Rather, the aggregates formed are generally fragile and highly susceptible to breakage in the presence of turbulence or shear. In addition, aggregate structure does not conform to the ideal conservation of measured volume. Dustan's experimental observations with respect to aggregate structure and aggregate rate thus indicated a system which has much more in common with a flocculation process than with the growth-agglomeration of standard crystallisation.

A prerequisite for aggregation in flocculating systems is the elimination of any interparticle repulsive forces that oppose aggregation (Hogg 2000). Destabilisation of the suspended fine particles is usually achieved through the addition of flocculants, which serve to eliminate the surface charge on particles dispersed in solution. In the $Ni(OH)_2$ system of Dustan (2001), destabilisation can occur as a result of the relatively high pH levels encountered (through its inverse relationship with electrostatic potential) and of ionic strength. In the presence of high concentrations of ions in solution, the charges on neighbouring particles are shielded from one another. This effectively compresses the electrical double layers surrounding the particles allowing them to approach one another closely enough to fall into the range where attractive forces dominate (Hogg 2000).

2.2.3.3 Ageing and resultant solid phase character

As mentioned in the introduction to this section, Sec. 2.2.3, particles formed by nucleation and enlarged through aggregation may undergo physical and chemical changes that affect the final product character provided they remain in contact with the mother liquor; these processes are often collectively termed ageing. One such process is ripening, also called isothermal recrystallisation or Ostwald ripening. This process describes the tendency, within a dispersed solid-liquid system, for dissolution of smaller particles and subsequent deposition of solute onto larger particles. This shift of the PSD to larger sizes is driven by the minimisation of total surface free energy.

Another ageing process is phase transformation. According to Ostwald's law of stages, which is applicable to precipitation systems where solids formation proceeds rapidly, the formation of a thermodynamically stable phase is preceded by that of a number of less stable, or metastable, phases possessing greater energy than the stable phase (Söhnel and Garside 1992). Thus, phase transformation involves the gradual transformation, entailing both physical and chemical changes, from the initially formed metastable solid to a more stable form. Such changes may include the conversion of amorphous precipitates to more crystalline forms, the reduction of both structural and non-structural

water, the conversion of metastable phases to more stable or equilibrium phases and the recrystallisation of non-equilibrium shapes of primary particles to form more compact shapes. The rate of these changes is a function of precipitation conditions including concentration of reactant solutions, solution ionic strength, the degree of agitation during ageing, the solution temperature and the presence of admixtures (Söhnel & Garside 1992).

For metal hydroxide systems, the initial formation of amorphous phases and subsequent phase transformation is common, for example, the sequential conversion of amorphous $Al(OH)_3$ to increasingly more stable solid phases. In particular, the reduction of the content of both non-structural (adsorbed or interstitial in the lattice) and structural water (bound to metal atoms through a chemical bond) is characteristic of ageing in metal hydroxide precipitation systems (Söhnel & Garside 1992 and references contained therein). The reduction of non-structural water, a large amount of which is contained in freshly precipitated metal hydroxides, can be attributed to lattice rearrangement and the growth of crystalline particles. In general, the tendency to form amorphous precipitates in metal hydroxide systems increases with the valency of the metal ion (Patterson et al. 1990).

In his study of nickel hydroxide precipitation from chloride solutions, Dustan (2001) suggests that a number of chemical and physical rearrangements take place during ageing and these have a significant impact on aggregate shear strength and apparent density. Although no attempt is made to resolve the dominant mechanisms involved in the ageing process, a number of observations and hypotheses are put forward. During ageing of nickel hydroxide precipitates Dustan observes an increase in aggregate density accompanied by a decrease in particle size and slight increase in particle number. These changes in particle characteristics can be attributed to a loss of non-structural water and associated particle rearrangements to more compact packing together with a loss of structural water during molecular rearrangements of the metastable form to more stable phases.

2.2.4 Solids population balance

As noted in Sec. 2.1, the particle size distribution or PSD is one of the most important design criteria to be considered in the operation of precipitation processes (due in large to its influence on the performance of downstream particle processing technologies), yet it is one of the most difficult properties to predict. Unlike systems comprised of one or more bulk phases which may be sufficiently described by a consideration of temperature, composition and relevant rate or equilibrium processes, particulate dispersed-phase systems require additional information to fully characterise them, since one or more independent properties can be specified for each individual particle entity in the dispersed system (Randolph & Larson 1971).

A system of particles simultaneously undergoing nucleation, aggregation and growth may be described mathematically by a solids population balance – a statement of continuity for particulate systems. In its most general form the solids population balance may be given as

$$\frac{\partial n}{\partial t} + \nabla \cdot (\mathbf{v}n) = B - D \,. \tag{2.11}$$

Kinetic expressions for the various precipitation processes are then used in conjunction with the above population balance equation (PBE) to describe the evolution of the PSD.

Eq. (2.11) is formulated in terms of both internal and external coordinates, where internal coordinates relate to independent particle properties (most notably particle size) and external coordinates relate to the position of the particle in space. In the above equation, n is the population density function and $\mathbf{v}$ is the vector particle phase-space velocity with both external and internal components. Here, the external velocity $\mathbf{v}_e$ is related to, but not necessarily identical to the fluid velocity $\mathbf{u}$. An example of internal particle velocity $\mathbf{v}_i$ applicable to precipitation is crystal growth. B and D are birth and death functions which describe the appearance and disappearance of particles at a particular point in the particle phase space as a result of, for example, particle aggregation or breakage.

For a well-mixed batch system of constant volume, with length, L, as the internal particle coordinate, the population balance reduces to (Randolph & Larson 1971)

$$\frac{\partial n}{\partial t} + G\frac{\partial n}{\partial L} = B - D \,. \tag{2.12}$$

Here again, n is the number density function. This function is defined such that, if in the system there are dN particles per unit volume of suspension in the size range L to $L + dL$, then at that size and time, $n(L) = dN/dL$ (Bramley et al. 1996). Thus, Eq. (2.12) relates the rate of change of the number of particles in the differential size range dL to the rates of birth and death in the size range and the rates of growth into and out of the size range (Hounslow et al. 1988).

Note that in the above equation the particle growth rate, G, is not a function of L – a consequence of the McCabe ΔL law (McCabe 1929) which assumes a linear rate of crystal growth independent of particle length. The ease with which growth may be described using length promotes the use of length as the internal particle coordinate as opposed to, for example, volume. The length-based forms of the birth and death terms for aggregation are (Hulbert & Katz 1964):

$$B(L) = \frac{L^2}{2}\int_0^L \frac{\beta[(L^3-\lambda^3)^{1/3},\lambda)]n[(L^3-\lambda^3)^{1/3}]n(\lambda)d\lambda}{(L^3-\lambda^3)^{2/3}} \tag{2.13}$$

$$D(L) = n(L)\int_0^\infty \beta(L,\lambda)n(\lambda)d\lambda \,. \tag{2.14}$$

Nucleation is commonly thought of as the appearance or birth of particles of zero size, from which it follows that

$$B = B_0\delta(L) \,, \tag{2.15}$$

where B_0 is the nucleation rate and $\delta(L)$ is the Dirac delta function (Hounslow et al. 1988).

to or longer than the reaction time scale. Mixing times are discussed in detail in Baldyga et al. (1995). However it should be noted that, of the mixing processes, only micromixing has the potential to affect the reaction directly, while meso- and macromixing affect the reaction indirectly by influencing the rate of micromixing (Baldyga & Orciuch 2001).

2.3.1.1 Macromixing

Macromixing acts on the scale of the whole vessel and serves to convey fluid through regions where the turbulent properties vary (Zauner & Jones 2002). This, the largest scale of mixing, incorporates the flow processes that control the mean concentration and the residence time distribution in the vessel. The characteristic time scale for the mean convective flow is given by the circulation time, t_C, which for a stirred tank is

$$t_C = a\frac{V_t}{Nd_{imp}^3}. \tag{2.21}$$

Here, V_t refers to total volume of the vessel contents, N the stirrer speed and d_{imp} the impeller diameter, whilst a is a constant, the value of which depends on the pumping capacity of the impeller.

2.3.1.2 Mesomixing

Mesomixing reflects turbulent diffusion and the inertial-convective disintegration of large eddies that contain partially segregated fluid. Baldyga & Bourne (1992) define the characteristic time for turbulent diffusion, t_D, as follows:

$$t_D = \frac{Q_{feed}}{\bar{u}D_T}. \tag{2.22}$$

In this equation, Q_{feed} is the feed addition rate, $\bar{u}$ is the magnitude of the velocity close to the feed point and D_T is the turbulent diffusivity.

For the inertial-convective disintegration of eddies of size r (where r is greater than the Kolmogorov microscale, η, but smaller than the large energy-containing eddies of size L), Baldyga et al. (1997) estimate the inertial-convective characteristic mixing time, t_S, as

$$t_S = \frac{k}{2\varepsilon} = \frac{3}{4}\frac{L^{2/3}}{\varepsilon^{1/3}}, \tag{2.23}$$

where k is the turbulent kinetic energy and ε is the turbulent kinetic energy dissipation rate.

2.3.1.3 Micromixing

As with mesomixing, two mechanisms make up micromixing effects, viz. viscous-convective deformation of fluid elements and molecular diffusion. These mechanisms operate over length scales smaller than the Kolmogorov length scale, η.

For viscous-convective mixing, the characteristic time scale is represented by what is known as the engulfment time constant, t_E:

$$t_E = \frac{1}{E} = 17.24\left(\frac{\nu}{\varepsilon}\right)^{1/2}. \tag{2.24}$$

E is referred to as the engulfment parameter and ν is the kinematic viscosity (Baldyga & Bourne 1989a).

The molecular diffusion time scale, denoted t_G, is given by

$$t_G = t_E\left(0.030 + \frac{17050}{Sc}\right)^{-1}, \tag{2.25}$$

where Sc is the Schmidt number of the fluid (Baldyga & Bourne 1989b).

2.3.2 Modelling approaches

Mixing of fluids in agitated vessels is a common unit operation in many industrial processes. Control of mixing efficiency is important, as selectivity, product yield and product quality may all be influenced by mixing effects. It is therefore critical to include mixing considerations in any attempt to design, scale-up or represent (through modelling) such equipment. Many mesomixing and micromixing models have been developed over the last 50 years to describe and quantify the role of mixing in reacting systems (Zauner & Jones 2002). According to Villermaux & Falk (1994), these models may be classified into three categories: phenomenological models, physical models and detailed numerical models.

2.3.2.1 Phenomenological models

Phenomenological models extend macroscale residence time distribution (RTD) concepts to the local microscopic scale through the visualisation of idealised segregated zones with exchange fluxes. A characteristic of these types of models, and perhaps their main disadvantage, is that the model parameters, including mixing times, are not known a priori and have to be fitted to experimental data (Villermaux & Falk 1994). They are thus of limited use in predicting behaviour and developing rules for reactor scale up.

The most common example of a phenomenological model is the IEM or 'interaction by exchange with the mean' model, attributed to Harada et al. (1962). This model assumes that micromixing takes place by a reversible exchange between feed regions (represented by well-mixed batch zones) and a mean environment or bulk according to a single mixing time constant (Zauner & Jones 2002). Similar models include the 3E and 4E (three and four environment) models developed by Ritchie & Togby (1979) and Mehta & Tarbell (1983) respectively. These models divide the reactor into two segregated environments and one or two well-mixed leaving environments with transfer coefficients accounting

for the mixing between environments. Villermaux & Falk (1994) developed a so-called generalised mixing model (GMM), which is seen to reduce to simpler compartment models in certain cases.

2.3.2.2 Physical models

Physical models are mechanistic in that they are based on fundamental descriptions of diffusion and convection occurring during mixing. While simplifications are necessary – usually the mixing vessel is divided into zones based on experimentally observed flow patterns – model parameters can be independently determined from an analysis of fluid dynamics or laboratory measurements (Villermaux & Falk 1994).

The characteristic model here is the EDD model developed by Baldyga & Bourne (1984), which is derived from the engulfment-deformation-diffusion (EDD) theory of turbulent mixing. This theory is based on the observation that during mixing entering material is engulfed by bulk fluid to form vortices. The fluid elements imbedded in the vortices then deform and stretch to form slabs, while, within these laminated substructures, molecular diffusion occurs to bring about local homogenisation. Baldyga and Bourne later simplified the EDD model, after recognising that in some cases, in particular when $Sc > 4000$, the engulfment frequency is the rate-determining step of micromixing and thus molecular diffusion and deformation can be neglected. This led to the development of the engulfment or E model (Baldyga & Bourne 1989a).

Physical models are capable of making a priori predictions but only within the range of validity of the assumptions and with full cognisance of the limits of the flow model employed (Villermaux & Falk 1994).

2.3.2.3 Detailed numerical models

Detailed numerical or analytical models are based on a comprehensive description of fluid dynamics in mixing vessels (Villermaux & Falk 1994). Due to the rapid development in the field of computational fluid dynamics (CFD), several commercial and private computer packages are now available to facilitate the solution of the relevant mass, momentum and energy conservation equations. Owing to the significant role CFD plays in this study, the approach to turbulence and micromixing modelling in CFD is the focus of the next section.

2.3.3 A focus on CFD

Mixing tank operations typically exhibit complex three-dimensional and unsteady fluid flow, which leads to considerable uncertainty in design and scale-up. Significant improvement can only be achieved from a detailed description of the fluid flow characteristics (Middleton et al. 1986). Recent progress and continuing improvements in CFD techniques and digital computing allow complex flow situations to be modelled with a reasonable amount of detail and accuracy. CFD essentially involves the prediction of the fluid flow behaviour through the solution of the mathematical equations governing

fluid flow and heat and mass transfer, i.e. those describing the conservation of mass (the continuity equation), momentum, energy and component mass. Each of these partial differential transport equations may be generically expressed as a scalar advection-diffusion equation, given in coordinate-free notation by

$$\frac{\partial}{\partial t}(\rho\phi) + \nabla\cdot(\rho\mathbf{v}\phi) = \nabla\cdot(\Gamma_\phi\nabla\phi) + S_\phi . \tag{2.26}$$

The first term in Eq. (2.26) is known as the unsteady term and represents transient effects, the second term is referred to as the convective term and describes the convective transport of ϕ ($\mathbf{v}$ is the velocity vector) and the third term, the diffusive term, reflects diffusion. Γ_ϕ is the diffusion coefficient. The final source (or sink) term represents creation and destruction of ϕ. In principle, not only laminar flows, but also a turbulent mixing process, could be fully described by directly solving the equations symbolised above; this, since turbulent flows are merely extremely complex, unsteady laminar flows. However, when considering turbulence, this would require a grid size that can encompass the smallest relevant length scale. In turbulent mixing systems this length scale corresponds to the smallest scale of concentration fluctuations described by the Batchelor microscale, which is impractically small. The computational requirements of such direct numerical simulation (DNS) of turbulence is enormous and the technique is therefore currently limited to moderate Reynolds number flows in very simple geometries (Vicum et al. 2004).

Thus, in order to solve these equations for practical applications and with the finite computational resources available, it is necessary to make a number of approximations to account for the complexities introduced by turbulence and micromixing (Fletcher et al. 1998). In most situations the above equations are averaged and coupled to various empirical models (see Sec. 2.3.3.1). The resulting partial differential equations are then discretised on a finite mesh, and with the application of appropriate boundary and initial conditions, solved iteratively to produce quantities like velocity, pressure, turbulence, temperature and concentration for each mesh point, for every time step (Fletcher et al. 1998).

2.3.3.1 Turbulence modelling in CFD

As intimated above, under turbulent conditions, such as those most frequently encountered in stirred tanks, turbulence models are used to solve the transport equations for the Reynolds-averaged quantities:

$$\overline{\phi} = \frac{1}{\delta t}\int_{t}^{t+\delta t}\phi\, dt . \tag{2.27}$$

Here, δt is a time scale large relative to the time scale of turbulent fluctuations, and small relative to the time scale to be resolved. Such a definition enables the instantaneous flow variables to be split into their mean and fluctuating components:

$$\phi = \overline{\phi} + \phi' . \tag{2.28}$$

Observing the mathematical properties of Reynolds averaging, dropping the overbar notation from Reynolds-averaged mean quantities for all primitive variables and retaining the bars for averages involving products of fluctuating quantities, Eq. (2.26) may then be written in Reynolds-averaged form as

$$\frac{\partial}{\partial t}(\rho\phi) + \nabla\cdot(\rho\mathbf{v}\phi) = \nabla\cdot(\Gamma_\phi\nabla\phi - \rho\overline{\mathbf{v}'\phi'}) + S_\phi\,. \tag{2.29}$$

Thus, Eq. (2.26) is unchanged except for the inclusion of a turbulent flux term additional to the molecular diffusive flux. These are denoted the Reynolds fluxes or, if the variable under consideration is a velocity (i.e. the relevant transport equation is that of momentum), the Reynolds stresses, and they arise not from the linear diffusive term in Eq. (2.26), but from the non-linear convective term. They reflect the fact that convective transport due to turbulent velocity fluctuations will act to enhance mixing over and above that caused by thermal fluctuations at the molecular level (Fletcher 1997). Eq. (2.29) now requires closure by an appropriate turbulence model, which provides models for the computation of the Reynolds stresses and fluxes. Such models may be broadly categorised as being either eddy viscosity models (Reynolds stresses and fluxes are modelled algebraically in terms of known mean quantities) or second order closure models (differential transport equations are solved for the turbulent fluxes).

The most widely applied turbulence closure model in CFD is the standard k–ε model – a two equation, eddy viscosity model. The eddy viscosity hypothesis assumes that the Reynolds stresses can be linearly related to the mean velocity gradients via an eddy (or turbulent) viscosity, μ_T, in a manner analogous to the relationship between the stress and strain tensors in laminar Newtonian flow. Similarly, the eddy diffusivity hypothesis states that the Reynolds fluxes of a scalar are linearly related, via an eddy (or turbulent) diffusivity, $\Gamma_{\phi,T}$, to the mean scalar gradient. The generic Eq. (2.29) thus becomes:

$$\frac{\partial}{\partial t}(\rho\phi) + \nabla\cdot(\rho\mathbf{v}\phi) = \nabla\cdot(\Gamma_{\phi,eff}\nabla\phi) + S_\phi\,, \tag{2.30}$$

where

$$\Gamma_{\phi,eff} = \Gamma_\phi + \Gamma_{\phi,T}\,. \tag{2.31}$$

Focussing on the k–ε model, the turbulent viscosity is given by

$$\mu_T = \rho c_\mu \frac{k^2}{\varepsilon}\,. \tag{2.32}$$

In the above equation, c_μ is a constant, k is the turbulent kinetic energy and ε is the turbulent kinetic energy dissipation rate. The value of $\Gamma_{\phi,T}$ is then simply related to that of μ_T as prescribed by the model and the relevant flow variable of interest. Two further equations governing the conservation of k and ε are necessarily derived in this model. These are obtained by a combination of the manipulation of the fundamental equations, to construct equations for the transport of these quantities, followed by

approximation to remove the unknown terms this process introduces. Details of these equations, which contain a number of tuned constants, as well as those covering the Reynolds stresses and fluxes and other particulars, are offered in Fletcher (1997) and Hjertager et al. (2002) or indeed most texts on turbulence modelling.

Given its simplicity, the k–ε model performs well in a wide range of turbulent flows. One drawback though is that it assumes turbulent diffusion to be isotropic or directionally independent. For flows involving strong anisotropy, such as those close to walls (a problem overcome by the application of near-wall functions) or in swirl flows (van Leeuwen 1998), such an assumption is clearly invalid, and the model is overly diffusive. Nevertheless, although modifications to the standard k–ε model (such as that based on Renormalisation Group (RNG) theory) and even second order turbulence closure models such as the differential stress and differential flux models claim superior performance, none at present has conclusively been shown to outperform consistently the standard k–ε model (Fletcher 1997). Jenne & Reuss (1999) present a critical assessment on the use of k–ε turbulence models for simulation of Rushton-turbine induced turbulent flow in baffled stirred tank reactors.

2.3.3.2 Micromixing modelling in CFD

In order to model turbulent flows in mixing vessels, it is necessary to default to the Reynolds-averaged equations outlined in Sec. 2.3.3.1 above. However, in such a representation, while macromixing and mesomixing are solved together in terms of convection and turbulent diffusion, micromixing is neglected (Marchisio & Barresi 2003). Essentially, this problem occurs because the chemical reaction rates are dependent on the instantaneous species concentrations, which vary rapidly due to turbulence, but in the numerical simulations only time (ensemble) averaged values are readily available. The problem is well described for precipitation by Baldyga & Orciuch (1997).

To overcome this problem, a specific micromixing model needs to be added. There are several approaches to micromixing that can be coupled with CFD codes. A popular strategy, and one that is commonly an elective feature of CFD codes, is the eddy dissipation concept (EDC), which was developed for the prediction of gaseous combustion reactions in turbulent flows. It is based on the premise that reaction time scales can be related to the dissipation of turbulent eddies containing reacting solutions (Magnussen & Hjertager 1976). Hjertager et al. (2002) have attempted to modify the EDC model to make it applicable to fast chemical reactions occurring in the liquid phase.

A more rigorous and increasingly common approach is to account statistically for the fluctuations via the introduction of additional models for the various scales of mixing behaviour (varying from the inertial-convective to the viscous-diffusive) and probability density function (PDF) models that seek to represent the turbulence fluctuations. However, much simplification is necessary, which often takes the form of solving transport equations for a mixture fraction and its variance, and using a prescribed PDF form to calculate the fluctuations. This approach is used by Baldyga & Orciuch (1997) to study precipitation via a single reaction. More complex models adopt a Lagrangian approach to account for

the micromixing and integrate this with Eulerian transport models (Fox 1998). However, application is still limited to simple reaction systems.

2.3.3.3 Flow prediction in stirred tanks

Mixing of fluids in stirred tanks is an important and well-established unit operation that finds application is a wide range of industrial processes. As a result, numerous studies are detailed in the literature that describe the effect of impeller and vessel geometry, fluid properties and operating parameters on power consumption and fluid flow patterns (Rigby et al. 1998). The most widely applied and studied mixing configuration is the baffled tank in combination with the standard Rushton impeller.

Numerical simulation of the flow field in stirred tanks is complicated by the interaction between the stationary baffles and the rotating impeller, which produces a periodic unsteady flow (Koh & Wu 1998). A number of approaches have been developed to account for the effect of the impeller in the stirred tank. Early approaches generally exclude the impeller region from the computational domain and prescribe, on the basis of simple models or experimental data, the flow field near the impeller as a stationary boundary condition (Brucato et al. 1998). Alternatively, the impeller is modelled as a distributed source of empirically determined momentum and turbulence quantities (e.g. Ranade 1997). Although effective, these approaches have limited predictive capabilities due to their reliance on experimental data.

Recently, the development of sliding mesh techniques has allowed the explicit simulation of the entire flow field in baffled stirred tanks without the need for empirical data (Brucato et al. 1998). In this approach, the flow field is divided into two discreet domains, an inner domain that rotates with the impeller and a stationary outer domain. At each time step, the additional velocity component due to the motion of the mesh is taken into account. The sliding mesh approach can now be implemented as a standard option in most commercially available CFD codes.

2.4 Application of CFD to precipitation systems

Over the past decade the application of computational fluid dynamics (CFD) to complex flow and reaction problems has increasingly generated considerable interest. Precipitation processes frequently exhibit very fast reaction kinetics, making mixing of great importance in determining product quality. It is speculated that the detailed knowledge of spatial dynamics afforded by CFD modelling would add substantial value to the analysis of such systems, yet there remain relatively few published attempts reported on its application to the field of precipitation. Efforts to model precipitation with a detailed description of fluid dynamics via CFD are hampered by the significant challenges that arise when simultaneously considering complex precipitation kinetics and PSD evolution together with micromixing phenomena (Falk & Schaer 2001). As a result there tends to be a division in the literature

between works that attempt to bring together precipitation with detailed micromixing modelling and works that focus on representing the precipitation process more accurately.

The original work in this area was reported by Seckler et al. (1995) who developed a model describing the precipitation of calcium phosphate on a local scale in a double-jet continuous reactor. This work was further developed by van Leeuwen et al. (1996) who studied the steady-state two-dimensional precipitation of barium sulphate in a continuous rectangular flat reactor with jet mixing. Wei & Garside (1997) developed two-dimensional and three-dimensional descriptions of steady-state barium sulphate precipitation in jet mixer pipelines, and later extended these simulations to explore the continuous precipitation of barium sulphate in stirred tanks. All these pioneering works, and more recent studies including those by Al-Rashed & Jones (1999) and Rousseaux et al. (2001), assume perfect micromixing within each computational cell and yet still achieve good agreement with experimental data.

Other researchers maintain the importance of micromixing effects in precipitation systems. Falk & Schaer (2001), for example, present a PDF method for simulating the precipitation process, which considers nucleation, growth and agglomeration sub-processes together with micromixing. However, their method is demonstrated for a very simple case of silica aggregation (in the absence of nucleation, growth and supersaturation) in a simple reactor configuration, due to the acknowledged complexity of considering all processes simultaneously. In contrast, Jarworski & Nienow (2003) assert that in stirred tanks the contribution of mixing to overall precipitation kinetics is marginal. Thus, in their study of continuous precipitation of barium sulphate micromixing effects are not included in the model framework. There is thus a clear lack of agreement over the role of micromixing in precipitating systems, which can be attributed to the still-limited empirical knowledge and fundamental understanding of precipitation processes.

It is because of these limitations that even works that concentrate primarily on the integration of precipitation processes with CFD (i.e. those that assume perfect micromixing) are subject to a number of necessary simplifying assumptions. Often only the mechanisms of nucleation and growth are considered; represented typically by empirical power law or Arrhenius type relationships. Aggregation is rarely modelled. This can be attributed to the fact that researchers often choose to represent the behaviour of 'model' systems, such as the precipitation reaction of barium chloride and sodium sulphate, where aggregation is thought to be minimal. Besides simplifying the modelling effort substantially, these ideal systems are chosen because they have been the most widely investigated experimentally and reported in the literature. For much the same reasons, there are no reported attempts to model metal hydroxide precipitation in conjunction with CFD (i.e. these systems do not behave ideally and are not often the focus of experimental studies).

A further simplification in the representation of precipitation systems is the use of the moment transformation of the population balance to represent the evolution of the particle size distribution. Indeed, this practice is almost ubiquitous. The only examples of a full population balance are in the works of Urban & Liberis (1999) and Madec et al. (2001). In the former case, however, the flow field

predicted by CFD is used to divide the crystalliser up into a network of uniformly mixed regions and to determine the interconnections between these regions. Aggregation is again not considered. Madec et al. (2001) use a direct Monte-Carlo method to simulate the full population balance for the case of agglomeration only, but this necessitates an extremely coarse CFD mesh of only nine cells.

Perhaps the most significant shortcoming, as it relates to the system under investigation here, in all the reported attempts to model precipitation, is the cursory attention given to the supersaturation calculation. The aqueous chemistry in the 'ideal' systems studied has been such that rigorous modelling of solution thermodynamics was not undertaken. However, in metal hydroxide systems, consideration of the critical role of solution speciation in the control of supersaturation, the precipitation driving force, is paramount.

Chapter

3

Experimental system and available data

3.1 Introduction

The experimental data supporting this work is gleaned from a portion of the laboratory component of the study of the semi-batch precipitation of nickel hydroxide by Dustan (2001). Although the intention was not to resolve the particular system in great detail, Dustan's investigation was extensive in its scope. Accordingly, the objective of this chapter is briefly to familiarise the reader with only the relevant particulars of Dustan's experimental system, procedures and program; this is considered prerequisite to the detailing, in the succeeding chapters, of this study's own modelling efforts. Additionally, and perhaps more importantly, the chapter serves to highlight the associated issues covering the extent and quality of the resultant data, which are pertinent to model development (in particular parameter extraction) and subsequent evaluation of model efficacy. It must be stated that no definitive, detailed display or analysis of experimental results is offered. For this and for further details regarding laboratory methods please refer to Dustan (2001), a work which self-evidently forms the basis of the information presented here.

3.2 Experimental system and procedures

The applicable operating procedure for the semi-batch precipitation of nickel hydroxide from chloride solution may be summarised as follows. Within a baffled 5 dm^3 cylindrical glass vessel of diameter (T) 0.17 m, an initial charge solution of 3 dm^3 of 0.01 $mol.dm^{-3}$ $NiCl_2$ is controlled at 25 °C and agitated

with a six-bladed Rushton turbine. To this, a solution of 0.075 mol.dm^{-3} NaOH is added at a constant rate. Once a target pH of 10 is reached, NaOH addition switches to a control mode to maintain this pH. The tank is configured in the standard Rushton geometry, such that the other dimensions may be related to T thus (Lane & Koh 1997):

impeller clearance, $T/3$;
impeller diameter, $T/3$;
blade height, $T/15$;
blade length, $T/12$;
disc diameter, $T/4$;
baffle width, $T/10$.

The feed point is located at the surface of the initial liquid volume, midway between the impeller shaft and the vessel wall and midway between two of the four equally-spaced baffles.

During the course of a precipitation run, the pH is continuously monitored using a fast response glass pH probe, connected to a PC. Following the period of constant NaOH addition, the incremental volumes of additional hydroxide required to maintain the setpoint pH are recorded. Online particle size and solids volume concentration are measured using a Malvern Mastersizer S™ instrument. A Masterflex™ peristaltic pump is employed to recycle the tank contents through the analyser and back into the reaction vessel, with the suspension characterisation measurements logged at intervals throughout the process. In addition, nickel concentrations in solution are quantified by periodic, discrete sampling of the suspension – a small sample is withdrawn and immediately filtered through a 0.45 μm membrane prior to ICP analysis of the filtrate. It should be noted that, with respect to all readings, efforts are made to collect as accurate and representative a measure as possible of what is evidently a spatially non-uniform system, whilst also minimising disruption to it.

3.3 Data availability and quality

This section seeks to call attention to various issues defining the extent of the available experimental data and to aspects surrounding the quality thereof.

3.3.1 Summary of relevant experimental program

A vast array of possible design and operating variables exist for the control and optimisation of stirred tank precipitation processes. However, definition of the system as in Sec. 3.2 reveals that only a very limited subset of these are of interest here. Accordingly, only variations in the volumetric NaOH feed rate and the stirrer speed, both, incidentally, readily controllable by plant operators, are explored. Low, median and high values for these variables are selected, as illustrated in Table 3.1. A combination of the two median values, together with the constant variable values specified in Sec. 3.2, constitutes what is defined as a 'base case' scenario. The remaining four scenarios involve the independent variation of

these two variable values and are identified in the remainder of this work using the logical notation presented in Table 3.1.

Table 3.1 Specification of precipitation operating conditions for each of the experimental runs, with respect to investigated variables (non-'base' values in italics).

	NaOH addition rate ($cm^3.s^{-1}$)	Stirrer speed (rpm)
base	0.2	150
rate L	0.08	150
rate H	2	150
stir L	0.2	90
stir H	0.2	330

It should be mentioned that the addition rates listed above are approximate since there was some difficulty maintaining constant target values – more precise measured values are recorded by the instrumentation. Furthermore, no particular significance is attached to any of these scenarios. Indeed, their selection was fairly arbitrary; however, they are expected to sufficiently demonstrate meaningful trends and facilitate the investigation of the influence of these key variables in the modelling work to follow.

3.3.2 Measurement techniques and difficulties

Relying on the fact that the diffraction angle of laser light passed through the suspension sample is inversely proportional to particle size, the Malvern Mastersizer S™ employs the technique of laser diffraction (more correctly called Low Angle Laser Light Scattering) to determine particle volume distributions from analysed light energy data. Use is made of the full Mie theory to completely solve the equations for interaction of light with matter, reportedly allowing accurate results over the size range 0.05 – 900 μm (Rawle 2001). The same particle and suspending medium optical properties required by Mie theory are also applied by the analyser, together with the Beer-Lambert Law and the measured volume-based size distribution, to calculate the solids volume concentration.

Although the volume distributions reported by the sizing apparatus are generally deemed the preferred metric for chemical engineers, since they reflect where the mass of a system lies, other distributions or mean diameters of interest may be calculated from the measured data as required. An awareness of which is an original measurement and which a derived value is, however, of some import. Besides risking the possibility of error compounding as a consequence of conversion, significantly lower confidence is typically associated with the accuracy of derived measurements (Rawle 2001). Accordingly, it is simply mentioned here that caution should be exercised when exploiting the resultant transformations.

For this particular application, particle numbers are of great importance, both practically, through their relationship with system operability and the stated long-term objective of optimised downstream performance, and theoretically, through their required explication for parameter estimation and subsequent modelling efforts. As such, the measured volume distributions are converted to ones based upon particle number, employing assumptions of particle sphericity and some form of distribution within each size interval (in this case, a constant number-density function). The above-mentioned caution regarding the reliability of derived quantities is duly noted; the fact that tiny absolute errors in measured volume in the smaller size classes could represent the equivalent of disregarding millions of particles, serves as an illustration of the inherent difficulties. However, these potential inaccuracies, whilst hampering efforts at system characterisation, should not detract from the methodology developed and demonstrated in Chaps. 4 – 6. In light of the additional deficiencies and uncertainties evident in the process description, and which are outlined in this dissertation in sections to follow, it is argued that precise experimental quantification is not paramount. The previously articulated primary goals of the study support the notion that a consistent, comparable measure of the system is sufficient. Further, Dustan (2001) contends that the recurrent nature of the particle size analysis, at intervals throughout the course of a run, increases the confidence in the calculated particle number distributions.

3.3.3 Selected observations of results

Those features of the data deemed contextually relevant are included below. Prior to considering these it is worth remarking that the suspensions produced experimentally, like many metal hydroxide precipitate phases, exhibit low crystallinity, being amorphous and colloidal in nature; thus compounding the difficulties of particulate measurements. Indeed, as noted by Dustan (2001), the process has many characteristics more in common with flocculation or sol-gel applications than with classic crystallisation. In this regard, it is also valuable to note up-front that, again in line with other similar systems, growth phenomena are concluded to be insignificant by Dustan (2001). As such, Dustan's proposal – that aggregation is the predominant size-enlargement mechanism, whilst solute deposition occurs solely via nucleation – is adopted here.

In exploring the issues to follow, it is also important to recognise that, whilst being mindful of spatial heterogeneity, in the absence of spatially distributed data it can only be assumed that, for analysis and parameter extraction purposes, measured values (and their derivatives) represent the variables' uniform distributions.

3.3.3.1 Variability of precipitate density

Knowledge of the density of the precipitate phase in solution is a prerequisite for the performance of mass balances (relating precipitate volume to yield and, in turn, to dissolved concentrations) in the analyses of experimental data and in the development of models describing the process. For the poorly defined and significantly hydrated precipitates encountered here, the listed literature values for nickel hydroxide density (ed. Linde 1996) of 4150 kg.m^{-3} for crystalline solids and 3650 kg.m^{-3} for

amorphous material do not apply. Dustan (2001) therefore introduced the concept of an experimentally determined precipitate molar density, defined as the quantity of precipitated nickel per unit volume of detected solid. He proposed that during the semi-batch precipitation, an ageing period could be considered subsequent to an initial precipitation period in which essentially all solute deposition assumedly occurs. In this work, we concern ourselves solely with this precipitation period, the end of which is taken to be demarcated by an observed maximum in the measured total solids volume. The appropriate solids volume fraction profiles are represented in Fig. 3.1.

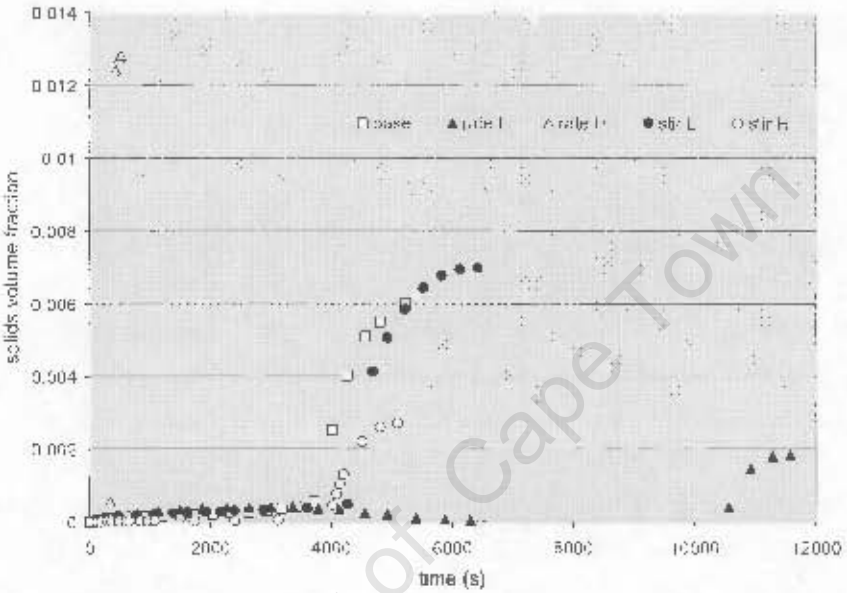

Figure 3.1 Profiles of measured solids volume fraction for the precipitation period for each of the experimental runs, culminating in the display of the maximum measured solids concentration.

In all cases it is fairly shortly after the setpoint pH of 10 has been attained that these characteristic maxima (evident in both solids volume and solids volume fraction) occur. The assumption, alluded to above, that at this point all nickel exists in solution at its equilibrium concentration (which, to all intents and purposes, is zero at this elevated pH), is thus supported by the ICP measurements indicating that, for each run, the realisation of the setpoint pH very closely coincides with the practically complete depletion of total dissolved nickel. Mass conservation implies that the corresponding nickel content of the maximum measured solids volumes is that of the total system, making it a simple undertaking then to determine the required precipitate molar density values. Since the various final solids volume fraction values plotted in Fig. 3.1 represent the same quantity of nickel, and since great similarity exists between the final values of total volume accompanying each series above, it is apparent that the calculated density values vary markedly under different operating conditions; they are tabulated below.

Table 3.2 Experimentally determined precipitate molar densities for each of the experimental runs.

	Precipitate molar density ($kmol_{Ni}.m^{-3}$)
base	1.26
rate L	4.15
rate H	0.61
stir L	1.15
stir H	2.95

Throughout the precipitation period, Dustan (2001) assumes the above quantities to be constant and characteristic of the particular set of operating conditions. Basic trends in relating the precipitate molar density to operational variables are clearly evident from Table 3.2, and although Dustan (2001) notes that density in solution appears to be a dynamic quantity – a function of, amongst other possibilities, both solids age (based on observations during the ageing period) and applied shear – no correlation was found for this important parameter after his examination of the full suite of experimental data. The variable nature of the density value from one scenario to another therefore has obvious detrimental implications for the predictive capability of modelling efforts, since for each operating point of interest results from the equivalent experimental run are necessary to determine a model input. However, for want of a superior practical alternative, the same approach is adopted in this study.

The corresponding molar density of hydroxide in the precipitate phase is estimated through its stoichiometric relationship to nickel. It is sufficient to state at this stage that Dustan (2001), whilst conceding that the quantity may be a time-dependent variable and that its value may be disputed, suggests that the ratio of Ni:OH in the solid is very close to that expected from pure $Ni(OH)_2$ formation, i.e. hydroxide density values are double those presented in the preceding table.

3.3.3.2 Incongruence of results

The precipitate molar densities defined and enumerated above are central to illustrating an interesting feature of the experimental results. They are used in the estimation of the various solids concentration profiles as inferred by mass balance from the ICP analyses of total nickel in solution. Comparison of the profiles thus generated with those measured by the Malvern Mastersizer S™ (appearing in Fig. 3.1) reveals a significant discrepancy. This is graphically presented in Fig. 3.2 for the 'base' scenario. Each of the remaining scenarios evince similarly anomalous behaviour, with an approximately linear decrease in total dissolved nickel corresponding, in essence, to a constant rate of volume deposition.

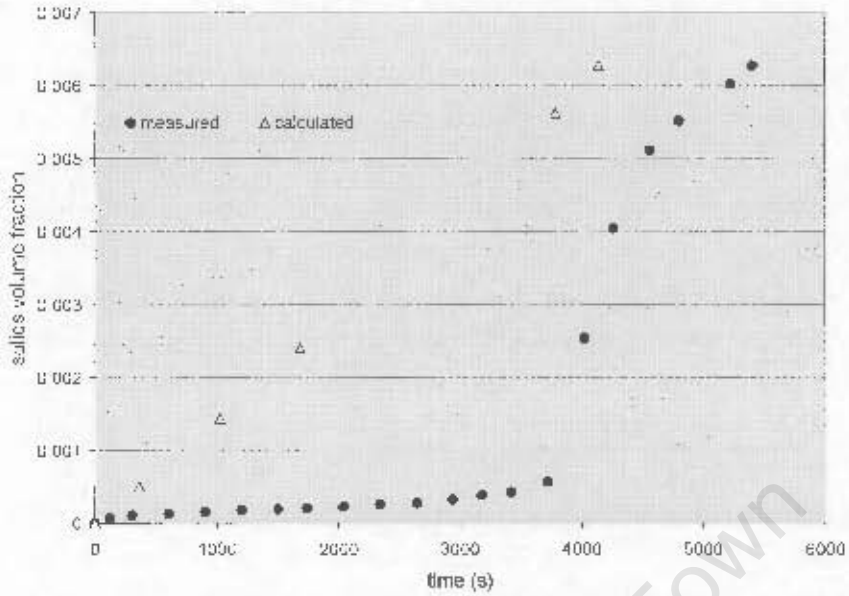

Figure 3.2 Profiles of solids volume fraction for 'base' run as described by solids concentration measurements (measured) and by mass balance calculations based on total dissolved nickel concentration measurements (calculated).

This phenomenon effectually illustrates why the precipitate molar density is only calculated at the conclusion of the precipitation period (Sec. 3.3.3.1) and is not tested at earlier times: the measurements of solids volume are seemingly unreliable. The matter of making sense of the above results is addressed in Sec. 3.4.

3.3.3.3 Reproducibility of results

Uncertainties and experimental difficulties notwithstanding, repeats of the 'base' experimental run (although not presented here) demonstrate good reproducibility in the data of the various measured properties. This serves to build some confidence in the results, suggesting that, although the accuracy or precision may at times be questionable, the instrumentation is not providing random output, but a reliably consistent measure of the system behaviour. The consequent faith in the capacity to capture experimentally the distinctive features or characteristics of the different scenarios permits, inter alia, comparative analyses to be made with greater assurance.

3.4 Manipulation of raw solids volume data

There exists a clear need to reconcile the contradictory data presented in Fig 3.2. Deliberation to this end yields two alternative, plausible theories, the first of which is adopted and implemented in this study.

3.4.1 Accounting for undetected solids volume

Dustan (2001) contends that a reasonable explanation for the lack of derived agreement between solids volume and nickel concentration measurements is a systematic failure of the particle sizing equipment. Although admittedly sporadic, the ICP analyses are deemed a more dependable gauge of the extent of solute deposition than is the suspension characterisation. Reservations surrounding the probity of the size analyser output have previously been expressed, with system-specific phenomena such as the formation of oxy-hydroxide species (leading to possible conformational changes in particle clusters) and aggregate densification with time (through secondary crystallisation processes) identified as compounding difficulties concerning laser diffraction readings. This stance (i.e. rejection of solids volume concentration measurements in favour of those of total nickel in solution) is further supported by the observations that:

- pH profiles (not shown) indicate substantial early hydroxide consumption;
- neither pH profiles nor cumulative profiles of volumetric NaOH addition (not shown) exhibit an inflection that might be expected to accompany the period of apparent sudden and rapid deposition reflected in the solids concentration measurements; and,
- the results of experimental runs by Dustan (2001) using alternative anions all display approximately linear deposition profiles (as inferred from both solids concentration and dissolved nickel measurements), whilst showing responses in other, connected properties (e.g. pH) which are comparable to those of the corresponding chloride runs.

In addition, although it is conceded that filtration of samples can never be instantaneous, the discrepancy in the extent of deposition implied by the two measurements is so vast that for this practical limitation to be used as counter-argument would require that a very slow nucleation rate accelerate alarmingly prior to and during filtration. Indeed, the interpretation presented here of these contradictory measurements is published (Petrie et al. 2002) and supported by leading authorities in the field, including Dr Alan Rawle, presently Divisional Manager Applications Support with Malvern Instruments, Inc. in the USA (Dustan 2005, personal communication)[†].

As such, it is posited that the presence of very small particles – diameters of nuclei have been previously estimated at 11.5 nm for this particular system (Suoninen et al. 1973) – is not detected by the Malvern Mastersizer S™ until they have aggregated to a certain minimum 'visible' size. Clearly then, it is required that some critical solution and/or surface condition exist before significant aggregation can occur.

The measured size distributions reveal that solids volume is only consistently detected in the thirteen size intervals spanning lower bounds of 2.016 – 32.254 μm, intimating that the mooted 'cut-off' size

---

[†] Nevertheless, the possibility that the interpretation, selection and subsequent manipulation of experimental data may be incorrect is acknowledged; alternative explanations are recognised and even suggested (Sec. 3.4.2). Ultimately, however, whether or not this interpretation is correct is not of primary importance within the context of the aims of the work and should not detract from the core contribution, which is (as has been mentioned and will be seen in the following chapters) the development of a model architecture of generic value.

below which particles are invisible to the size and volume measurements is in the region of 2 μm. The seemingly obscure dimensions referenced merely correspond to the discrete output channels of the sizing apparatus, which, beginning at 0.05 μm, divides the length domain into intervals in a geometric series such that the upper and lower bounds of each size interval are in a ratio of $2^{1/3}$. Recalling that nickel assays are preceded by filtration of the sample through a 0.45 μm membrane immediately after collection, it is therefore evident that, theoretically, the unseen solids volume represented by the difference in Fig 3.2 is associated with particles in the size range 0.45 – 2.016 μm. Considering this fact, the selected manipulation of the raw experimental data, whereby the 'lost' solids volume is simply assigned to the smallest of the thirteen particle size classes (with due consideration of precipitate molar density and 'nucleus' volume), is not thought to be an unreasonable approximation.

It is of interest to remark, again theoretically, that, if allowances are made for particles between 0.0115 μm and 0.45 μm in size, the undetected volume could feasibly be even greater. However, it is later demonstrated in Chaps. 4 and 6 that, whether assuming spatial uniformity or modelling the system's detailed spatial dependence via CFD, calculations indicate the deposition predicted by the total dissolved nickel concentrations closely approaches the maximum thermodynamically permissible. This, in turn, implies that the quantity of precipitate composed of these smallest of particles is persistently very limited. Such a situation is arguably unrealistic, casting doubt on the theoretical assertion that all particles less than 0.45 μm in diameter will pass through the membrane (and filter cake). In practice, the attribution of a strict classification capability to such an elementary filtration operation is likely inaccurate.

3.4.2 Size-dependent aggregate density

The notion that the Malvern Mastersizer S™ is failing to discern the presence of particles as large as 2 μm clearly flies contrary to the conservative claim cited in Sec. 3.3.2 that the device enables measurement of particles at least as small as 0.05 μm. In light of these contradictory detection limits, an alternative and perhaps more credible explanation for the apparent conflict between solution and suspension measurements is that there exists, in fact, no conflict. Rather, it may be postulated that the full solids volume (or very close to it) is detected and reported and that the precipitate molar density is accordingly a function of particle size, i.e. larger aggregates occupy a greater specific volume than smaller ones. Hence, it may be defensibly conceived that, although their presence is not necessarily reflected in the measured size distributions, small, dense particles predominate during the early stages of the precipitation; these do not contribute significantly to the solids volume, but do constitute a substantial quantity of nickel. Measurement difficulties borne of the minimal volume associated with vast numbers of smaller particles were mentioned previously in Sec. 3.3.2. The consequences of potential errors are clearly exacerbated by the proposed relationship between particle size and density, reinforcing the ideal requirement of higher resolution particle number distributions and complicating the task of appropriate manipulation of the raw solids data.

There are a selection of factors which suggest that this idea of a size-dependent density might be the superior rationalisation of 'incongruent' experimental results.

Foremost, the argument does not rely upon an assumption of glaring hardware malfunction. As a result of this, it is better able to justify convincingly the various suspension measurements. Irrespective of the particle sizes involved, the premise, required by the reasoning presented in Sec. 3.4.1, that such a substantial solids volume remains invisible seems intuitively unreasonable. Conversely, failure merely to resolve consistently the negligible fractions of total solids volume resident in the smaller particle size classes is somewhat understandable.

Further, the alternative explanation is supported by experimental observations. Although the accuracy of measured average particle diameters is questionable, empirical evidence of a consistent relationship between particle size and density in solution is provided by profiles of solids volume and average size for the different runs spanning the range of the precipitation program initiated by Dustan (2001). (The reader is directed to this work for full documentation of results.) These profiles clearly suggest that the larger the aggregate, the lower its density. This is most compellingly demonstrated by the behaviour exhibited during the ageing period, not only because of the greater certainty that the measured solids volumes for each of the experimental scenarios during this period represent both comparable nickel quantities and the true total volumes, but also due to the low initial solids concentrations (below the recommended measurement range of the Malvern Mastersizer S™) which render early size measurements particularly unreliable.

In addition, whilst the density values presented in Table 3.2 may adequately reflect the physical character of large, loosely-bound flocs, the assignment to nuclei of a density value closer to those supplied in the literature for nickel hydroxide (see Sec. 3.3.3.1) is certainly considered more appropriate. Even the 'low' density of 3650 $kg.m^{-3}$ listed for amorphous solid translates to a precipitate molar density of approximately 40 $kmol_{Ni}.m^{-3}$.

Finally, the non-conservation of measured aggregate volume is a recognised theoretical phenomenon (e.g. Gardner & Theis 1996, Hogg 2000). Indeed, such a model is largely consistent with the description of aggregate structure proposed (but not implemented) by Dustan (2001) based on his observations of the same system.

Another feature which makes this description an attractive one is that it painlessly overcomes the existing problem of the deleterious consequences for model predictive capability of the absence of a correlation for the different densities measured under distinct operating conditions. Moreover, it does not oblige that the core behavioural deductions established upon an assumption of undetected solid (Dustan 2001) be readdressed. Chiefly here, the explanations of measured solids volume data offered both in Sec. 3.4.1 and in this section each demand the sudden onset of a period of rapid aggregation: the former, for aggregates to move into the instrument's field of view; the latter, to 'create' volume through the non-volume-conserving formation of loosely-bound aggregates. All things considered, a

combination of the arguments presented in these two sections probably represents the most suitable account of reality.

As intimated above, the concept of a size-dependent density of fractal aggregates is touched on by Dustan (2001), although not embraced with much enthusiasm in its unadulterated form. Dustan does deconstruct the density dependence to a number of potential functional components, but does not submit the possibility that density could be, for example, not a function of shear, but essentially a function of particle size alone, which is, in turn, controlled by shear-dependent aggregation and breakage functions. The probability that reality is more complex than thus described is not dismissed; indeed, the fractal dimension of colloidal particles is often found to change with solution and fluid mechanical conditions (Gardner & Theis 1996). However, whilst conceding that the relationship between density and particle size could itself be a function of stirring intensity, the simplicity and elegance of the approach outlined here is perhaps worth pursuing. It is certainly an improvement on the current, implemented model.

Ultimately, though, such a description is not explored due in large to the difficulties inherent in the modelling of non-conservation of particle volume upon aggregation. Furthermore, difficulties still exist in establishing the mathematical specifics of the relationship between density and particle size. However, some thoughts on potential strategies for resolving these problems are offered in Sec. 7.2. Having taken the luxury of digression in detailing an important alternative paradigm in this section, it is worth reiterating that in the remainder of the work the positions outlined in Secs. 3.3.3.1 and 3.4.1 apply for all practical purposes.

3.5 Concluding remarks

The material presented in this chapter has served to acquaint the reader with the laboratory system and analytical methods which are the source of the experimental data utilised in this study. In so doing, the manipulation of solids measurements has been justified and detailed. Additionally, insights have been provided into data availability and quality, which reveal that the information afforded by the existing data is insufficient to characterise the system in great detail. These features have important consequences for the accuracy and generality of key extracted parameters and the resultant assessment of the applicability of the computational model (discussed in Chap. 6). Firstly, however, the specifics of the experimental system clearly inform the development of such a model to describe the process, an exercise which is documented in Chaps. 4 and 5.

Chapter

4

Model development and implementation (I): solution thermo-dynamics and the population balance

4.1 Introduction

Arguably the most essential component of any precipitation model is the description, by a population balance, of the evolution of the particle size distribution. Integral to such a description is a representation both of the kinetics of the pertinent defining processes and of the solution chemistry which governs them. Indeed, these considerations effectively constitute the sole necessary elements, in one form or another, of many widely-applied computational models, such as the MSMPR model for continuous crystallisers. Adoption of an approach of this type is, of course, only appropriate if, for all practical modelling purposes, perfect and instantaneous mixing may be assumed.

Having discussed in Chap. 3 the various experimental data at hand and the details of the system from which they were obtained, this chapter serves to provide the mathematical specifics of the sub-models selected to resolve the key population balance and aqueous thermodynamics calculations. Details of the relevant computational methods used in their implementation are also supplied. Further, the particulars of the procedure employed to extract rate functions from the experimental measurements (which draws on these two models) are presented, as is a discussion on the resulting kinetic expressions.

4.2 Modelling of solution chemistry

Apparent from Sec. 2.2.1 is the central importance of solution thermodynamics as the driving force for solids formation. So too is the related dependence on the chemical environment (in particular supersaturation) of the rates of the various kinetic processes controlling precipitation. Consequently, accurate and dynamic description of the solution chemistry is a necessary element in any model aimed at effectively simulating precipitation behaviour. Prior to this even, quantification of solution variables is required in the very formulation of appropriate rate functions, a procedure outlined in Sec. 4.4.

4.2.1 Model equations

Requisite characterisation of the chemical environment demands that the system be fully speciated, since, even for apparently simple reaction systems (and particularly those involving electrolytes), many reacting species may exist with the potential to significantly affect the solution properties. For the bulk of the simulations performed in this work, the only reaction not assumed to be instantaneous is the solids formation reaction itself, which proceeds according to the nucleation kinetics. Even when the notion of equilibrium nucleation is explored, speciation calculations assume that all intraphase solution reactions are fast relative to precipitation, and, as such, may be described entirely by equilibrium reactions.

The state of chemical equilibrium for $\mathcal{M}$ independent reactions occurring in a single phase comprising a total of $\mathcal{N}$ species is that state which satisfies the constraints on the system, viz. the set of $\mathcal{N}$ stoichiometric relations

$$n_i = n_{i,0} + \sum_{j=1}^{\mathcal{M}} \nu_{ij} X_j \qquad i = 1, 2, \ldots, \mathcal{N} \tag{4.1}$$

and the $\mathcal{M}$ equilibrium relations

$$K_{a,j} = \prod_{i=1}^{\mathcal{N}} a_i^{\nu_{ij}} \qquad j = 1, 2, \ldots, \mathcal{M}. \tag{4.2}$$

Here, n_i and $n_{i,0}$ are the equilibrium and initial number of moles of species i respectively, X_j is the molar extent of the j^{th} reaction, where X is defined as

$$X = \frac{n_i - n_{i,0}}{\nu_i}, \tag{4.3}$$

and ν_{ij} is the stoichiometric coefficient for species i in the j^{th} reaction, defined such that it is positive for reaction products, negative for reactants and equal to zero for inert species. As the above equations suggest, equilibrium calculations for multiple reaction systems are complicated.

For the precipitation system under consideration (i.e. precipitation of $Ni(OH)_2$ from $NiCl_2$ solution by the addition of NaOH), the main solution equilibria identified are presented in Table 4.1. In representing these reactions, a distinction is made between components and species – components, themselves species, are the independent basic units of which a variety of species may be comprised. Thus, $(N-M)$ components react in M independent reactions to form an additional M species, yielding a total of N species, with the j^{th} reaction producing species $(j+N-M)$.

Table 4.1 Main solution equilibria, with accompanying thermodynamic data, identified for the $NiCl_2/NaOH/H_2O$ system.

Reaction	Equilibrium constant, K_a
$H_2O \leftrightarrow OH^- + H^+$	1.0046×10^{-14}
$Ni^{2+} + H_2O \leftrightarrow NiOH^+ + H^+$	1.3804×10^{-10}
$Ni^{2+} + 2H_2O \leftrightarrow Ni(OH)_2(aq) + 2H^+$	1.0000×10^{-18}
$Ni^{2+} + 3H_2O \leftrightarrow Ni(OH)_3^- + 3H^+$	1.0000×10^{-30}
$2Ni^{2+} + H_2O \leftrightarrow Ni_2OH^{3+} + H^+$	1.9953×10^{-11}
$Ni^{2+} + Cl^- \leftrightarrow NiCl^+$	2.5061×10^{0}
$Ni^{2+} + 2Cl^- \leftrightarrow NiCl_2(aq)$	9.1201×10^{0}

Generation of the above list assumes the complete dissociation of the remaining molecular aqueous species HCl(aq), NaCl(aq) and NaOH(aq). In addition, it must be mentioned that the tabulated K_a values are those at the temperature of interest (25 °C) and for the reaction schemes as provided. The selection of H_2O as a component over the seemingly more logical OH^- ion, although to some extent motivated by the available thermodynamic data, is essentially an arbitrary one. However, an unintentional and minor advantage of this action will become apparent in Sec. 4.2.2. The approach and most of the quantitative information used here are similar to those employed by the solution speciation program MINTEQA2 (Allison et al. 1991), with additional inputs for the $Ni(OH)_2$ system taken from Baes & Mesmer (1976).

With particular application to the system of interest, adopting the component/species convention and introducing the notation n_i^j to denote the number of moles of component i, allows the mass balance equations (Eq. (4.1)) to be reformulated as

$$n_i^j = n_{i,0}^j \qquad i = 1, 2, \ldots, 5 \tag{4.4}$$

or

$$n_i - \sum_{j=1}^{2} \nu_{ij} n_{j,15} = n_{i,0} - \sum_{j=1}^{5} \nu_{ij} n_{j+5,0} \qquad i = 1, 2, \ldots, 5. \tag{4.5}$$

In the absence of redox phenomena, such balances also ensure the conservation of charge, thereby maintaining electroneutrality of the solution, an important consideration in the study of electrolytes. Further, Eq. (4.2) may now be simplified:

$$K_{a,j} = a_{j+5}\prod_{i=1}^{5} a_i^{\nu_{ij}} \qquad j = 1, 2, \ldots, 7. \tag{4.6}$$

In this analysis the Na^+ ion is included as component and species despite the fact that it is the only sodium species. Notwithstanding the above modifications, it is clear that, even for the relatively simple aqueous system of $NiCl_2$ and NaOH, a collection of several non-linear, coupled equations are to be solved simultaneously if the equilibrium state is to be determined.

These calculations are further complicated by thermodynamic non-idealities, which, as identified in Sec. 2.2.1.1, necessitate the use of activity coefficients to relate the molalities and activities of dissolved species. Varying degrees of sophistication exist for the prediction of these coefficients in electrolyte systems; the accurate application of simpler models being reserved for low ionic strength scenarios. In this work, the Davies equation, an extension of the Debye-Hückel limiting law (Eq. (2.3)), is the selected model:

$$\gamma_i = 10^{-Az_i^2\left(\frac{\sqrt{I}}{1+\sqrt{I}} - 0.3I\right)}. \tag{4.7}$$

Here, A is the Debye-Hückel constant and I the ionic strength both as defined in Sec. 2.2.1.1 and z_i is the charge on species i. For neutral solutes (i.e. undissociated electrolytes), the quantity z_i^2 is replaced by $|z_+ z_-|$, the absolute value of the product of ion valences of the dissociated electrolyte. Although developed for dilute aqueous solutions of single strong electrolytes in which the ions are non-associative, the Davies equation has proved valid for mixed electrolyte solutions and is utilised here in the absence of the species specific and/or species pair interaction parameters required for more advanced activity coefficient models. Reported limitations on the Davies equation's applicability vary from a maximum ionic strength of 0.1 $mol.kg^{-1}_{H_2O}$ (Zemaitis et al. 1986) to one of 0.5 $mol.kg^{-1}_{H_2O}$ (Stumm & Morgan 1981).

The solids formation reaction is represented by

$$Ni^{2+} + 2OH^- \leftrightarrow Ni(OH)_2(s) \qquad K_a = \frac{1}{K_{sp}} = 5.0118\times10^{14}, \tag{4.8}$$

for which the supersaturation may then be defined as

$$S = \frac{a_{Ni^{2+}} a_{OH^-}^2}{a_{Ni^{2+}} a_{OH^-}^2\big|_{eq}} = \frac{\gamma_{Ni^{2+}} M_{Ni^{2+}} (\gamma_{OH^-} M_{OH^-})^2}{K_{sp}}. \tag{4.9}$$

For the relevant temperature, a wide range of values for the solubility product (K_{sp}) have been reported, a fact which can be attributed to the general uncertainty regarding the physical state of the precipitate. The value selected above is that reasoned by Dustan (2001) to be most appropriate for use in this study.

Consideration of equilibrium precipitation requires a slight amendment to the approach described above. In this situation, modelling attempts demand the inclusion in the array of equations to be solved of an additional equilibrium expression for the above solids formation reaction. Further, presence of the solid species must be accounted for in the relevant component balance equations, an undertaking which calls for the effective substitution of OH^- in Eq.(4.8) with $(H_2O - H^+)$. Since the supplied equilibrium constant is for the reaction as stated, such action is unnecessary for the equilibrium relation, the satisfaction of which imposes a value of one upon the supersaturation as calculated by Eq.(4.9). As is standard practice for pure solids and pure liquids, the activity of the solid species, like that of the solvent (water) in the intraphase equilibria of Table 4.1, is simply estimated at unity.

4.2.2 Implementation

Computationally, the substantial task of resolving the equilibrium state of the system amounts to solution of a system of several non-linear algebraic equations or functions in an equal number of unknowns. A double-precision subroutine (NS11AD) designed for this purpose and compiled in the Fortran programming language was acquired, upon application and free of charge, from the HSL Archive. Fortran is the language of choice since the CFX® software used here in the integration of CFD modelling permits the description of additional or complex model features via user-defined Fortran subroutines. The solution method of subroutine NS11AD employs the ideas of Newton-Raphson and Steepest Descent, coupled with Broyden's method for improving Jacobian matrices. In executing these calculations, an additional subroutine MB01CD, obtained together with NS11AD, is called to evaluate the inverse of the Jacobian approximation using a technique essentially of simple Gaussian elimination with row interchanges. Among the necessary information passed to subroutine NS11AD is an initial estimate of the solution and a reasonable step size to be used for approximating the partial derivatives comprising the Jacobian matrix by finite differences. The subroutine CALFUN is also supplied to compute the values of the functions for which zeroes are to be found. A solution is accepted by NS11AD when a desired accuracy is achieved, i.e. the sum of the square errors in the resultant function values decrease to a specified maximum acceptable level.

With respect to this particular equation system, the functions calculated in subroutine CALFUN represent the equilibrium relationships and mass balances, variously detailed in the preceding section, which are to be satisfied. Formulated as in Eq. (4.5), mass balances conserve total component quantity, whilst the solution sought (i.e. the unknowns) are the molar quantities of each of the species, from which the concentrations of dissolved species (expressed as molality) may be determined. However, in recognition of the nature and requirements of the CFD analysis to follow, a slightly altered representation of the model equations is encoded in subroutine CALFUN. The solution is the set of equilibrium species concentrations (expressed as molarity) and the mass balances, in turn, conserve the molarities of the components. Accordingly, the equilibrium condition of species H_20 need not be solved and the mass balance relationship covering component H_2O may be neglected. Molarity then also replaces molality in the calculations of ionic strength and dissolved species activity; considering

the diluteness of the reagent solutions utilised experimentally, ignoring the distinction between these two measures of concentration is deemed justifiable. In addition, since the performance of subroutine NS11AD is improved if both all function and all (species concentration) variable values are similar in magnitude, logarithms are used extensively in the algorithm, in essence as a scaling operation.

As regards the required initial estimate of the equilibrium concentrations, when the routine is called repeatedly as part of the simulation of an actual precipitation process, the modelled time-step size is such that either the set of existing non-equilibrated concentration values or that of the equilibrated values at the previous time-step is sufficient. An 'isolated' calculation, such as a stand-alone speciation or that necessary at the launch of a precipitation simulation, demands a strategy whereby the requisite accuracy of the initial estimate is generated separately by a method which does not itself require a good initial guess, if at all. This can be achieved either by application to the equivalent equation system of a more powerful and robust solver (e.g. that offered by the gPROMS modelling platform) or by intelligent simplification of the model described by the Fortran code.

For further details on the computational implementation of the solution chemistry calculations, refer to the commented code file appearing in Sec. A.2.2, which represents the full extent of the user-defined Fortran programming supplementing the CFX® model developed later in this work to simulate the precipitation of nickel hydroxide. As an integral component of this overall structure, subroutine NS11AD is called and its argument list evaluated and specified at several different points within the subroutine USRTRN. The subroutine NS11AD itself is located on p. 149 and subroutines CALFUN and MB01CD are to be found on pp. 146 and 155 respectively. A schematic representation of the algorithm is also offered in Fig. 5.6.

4.2.3 Validation

Prior to application of the model depicted above to the simulation of the current precipitation system, a thorough investigation was conducted aimed at determining its performance over a range of electrolyte concentrations. This was achieved through a comparison of the generated output with the speciation behaviour predicted by the OLI™ software (OLI Systems, Inc.) for a variety of hypothetical input conditions, thereby allowing a 'point of departure' to be established. The models employed by OLI™, a recognised leader in the field of aqueous thermodynamic modelling, represent the high end in the range of sophistication and comprehensiveness. Supported by an extensive thermodynamic property databank, the models are reportedly appropriate for the complex and/or high ionic strength solutions typical of industrial processes. It should perhaps be noted that the differences in the descriptions of the models compared extend not only to the details of both the thermodynamic model and data, but also to the identity of the very species considered. This underlines the complications associated with the modelling of aqueous electrolyte systems.

Notwithstanding, in briefly summarising the results of this investigation it is sufficient to say that the predictions of the relatively simple model detailed here are in very close agreement with those of the

OLI™ software for moderate ionic strengths ($< 0.2\,\mathrm{mol.kg}^{-1}_{H_2O}$). Deviation, however, grows significantly with increasing ionic strength, findings which are loosely consistent with the ranges of applicability for the Davies equation cited in Sec. 4.2.1. Here it is worth stating that an original concession made in achieving the degree of correspondence for dilute solutions was the exclusion from the selected model of the tetra nickel species $Ni_4(OH)_4^{4+}$ which was included in the analysis of Dustan (2001).

Only relatively low ionic strengths are encountered in the experimental study used for model calibration and validation efforts (see Chap. 3), conditions under which the use of simpler activity coefficient models is justified. Indeed, it can now be asserted with some confidence that the chemical characterisation model employed in subsequent simulations is expected to give satisfactory quantitative description of the solution chemistry.

4.3 The discretised population balance of Hounslow et al.

Successful simulation of particulate processes such as precipitation demands that a statement of continuity for the evolving particulate phase be observed. As stated in Sec. 2.2.4, this amounts to solution of the partial differential population balance equation. Since it is generally entirely intractable analytically, a number of transformations and numerical methods have been developed to this end (see Sec. 2.2.4).

In this study the discretised population balance (DPB) of Hounslow et al. (1988) for simultaneous nucleation, growth and aggregation is employed to solve the population balance and hence describe the progression of the particle size distribution (PSD). The advantage of using a DPB, which involves the discretisation of the size domain into intervals and the assumption of some form for $n(L)$ (the number-density function) within each interval, is that it transforms the population balance equation into a set of ordinary differential equations. This drastically reduces the complexity of its solution. An additional advantage of this specific DPB over alternative numerical methods is the replacement of integral terms, which complicate solution by traditional finite difference techniques, with summations, further reducing the computational requirements. The easy accommodation of the modelling of size-dependent aggregation kernels also represents an improvement on the previous approach of Dustan (2001), who employed a moment transformation of the population balance using volume as internal co-ordinate. Also then of added value is the higher resolution of the DPB model: knowledge of the full PSD, as opposed to simply that of selected moments, is advantageous for it is this property of the precipitate phase, along with particle morphology, which often controls product quality.

The discretisation scheme associated with this particular DPB is selected for a combination of practical and theoretical reasons and divides the length domain into intervals in a geometric series such that the

upper and lower bounds of each size interval are in a ratio (r) of $2^{1/3}$. The idealised number-density function within each of these intervals is assumed to be constant.

Before detailing the model, it is necessary to rewrite the definition of the moments of a PSD (Eq. (2.16)) in discrete form if it is to be of any subsequent value (Hounslow et al. 1988):

$$m_j = \sum_{i=1}^{\infty} \overline{L_i^j} N_i \,, \tag{4.10}$$

where N_i is the particle number concentration in the i^{th} interval and $\overline{L_i^j}$ is the appropriate mean size in the i^{th} interval for calculating the j^{th} moment, and which, for a constant number-density function within each size interval, can be calculated as

$$\overline{L_i^j} = \frac{1}{j+1}\left(\frac{r^{j+1}-1}{r-1}\right) L_i^j \,. \tag{4.11}$$

L_i is the lower bound of size interval i. It follows from Eq. (4.10) that

$$\frac{dm_j}{dt} = \sum_{i=1}^{\infty} \overline{L_i^j} \frac{dN_i}{dt} \,. \tag{4.12}$$

Hounslow and co-workers (1988) propose the following equation to calculate the rate of change of particle number concentration in the i^{th} interval due to aggregation:

$$\left.\frac{dN_i}{dt}\right|_{agg} = N_{i-1}\sum_{j=1}^{i-2} 2^{j-i+1}\beta(\overline{L_{i-1}^1}, \overline{L_j^1})N_j + \frac{1}{2}\beta(\overline{L_{i-1}^1}, \overline{L_{i-1}^1})N_{i-1}^2 - N_i\sum_{j=1}^{i-1} 2^{j-i}\beta(\overline{L_i^1}, \overline{L_j^1})N_j$$
$$- N_i\sum_{j=i}^{\infty}\beta(\overline{L_i^1}, \overline{L_j^1})N_j \qquad i = 1, 2, \ldots, \infty \,. \tag{4.13}$$

Invoking the McCabe ΔL law, which assumes a size-independent linear rate of particle growth, the discretised equation developed for the description of growth is

$$\left.\frac{dN_i}{dt}\right|_{g} = \frac{2G}{(1+r)L_i}\left(\frac{r}{r^2-1}N_{i-1} + N_i - \frac{r}{r^2-1}N_{i+1}\right) \qquad i = 1, 2, \ldots, \infty \,, \tag{4.14}$$

whilst for nucleation it is simply

$$\left.\frac{dN_1}{dt}\right|_{nuc} = B_0 \,. \tag{4.15}$$

In formulating the above three equations, B_0 and G are the nucleation and linear growth rates respectively, and $\beta(L,\lambda)$ is the aggregation kernel, comprising a size-independent portion, β_0, and a second factor, $f(L,\lambda)$, which is some function of particle size.

4.3.1 Discussion of, and necessary modifications to, the DPB

The technique used to generate Eq. (4.13) considers four binary interaction mechanisms, each of which is then represented by a single mathematical term. The first term in Eq. (4.13) accounts for birth in interval i as a result of collisions between particles in the $(i-1)^{th}$ interval and particles in the first to $(i-2)^{th}$ intervals. The second term represents aggregate formation in the i^{th} interval by collisions between particles both in the $(i-1)^{th}$ interval. Death of particles in interval i as a consequence of aggregation with particles in the first to $(i-1)^{th}$ intervals is reflected in the third term, whilst the fourth term accounts for death in interval i resulting from collision and adherence of particles in the i^{th} interval to particles from that or higher intervals. Clearly, each of the first, second and third terms are null for $i = 1$ and the first term again null for $i = 2$.

Such a mechanistic approach helps to ensure the correct prediction of both particle number and volume – a key feature of the equation. However, this is only achieved in the limit as the number of particle size intervals modelled tends to infinity, or, simply stated, if there exist just an insignificant number of particles in the largest size interval. Since in this work a finite number of thirteen size intervals, selected from the experimental data as those in which solids were consistently detected (see Sec.3.4.1), are modelled, a couple of modifications to Eq. (4.13) are necessary. For obvious reasons, death in the largest size class is not permissible. Further, a correction must be made for the loss of particles in smaller size classes due to collisions with particles in the largest size class which produce aggregates larger in size than the upper bound of the largest interval. Opting to conserve precipitate quantity (or volume) in preference to correctly reflecting the change in particle number upon aggregation, one of two possible alterations is required:

$$\left.\frac{dN_{13}}{dt}\right|_{agg} = N_{12}\sum_{j=1}^{11} 2^{j-13+1}\,\beta(\overline{L_{12}^1},\overline{L_j^1})N_j + \frac{1}{2}\beta(\overline{L_{12}^1},\overline{L_{12}^1})N_{12}^2 + N_{13}\sum_{j=1}^{12} 2^{j-13}\,\beta(\overline{L_{13}^1},\overline{L_j^1})N_j \tag{4.16}$$

and

$$\left.\frac{dN_i}{dt}\right|_{agg} = N_{i-1}\sum_{j=1}^{i-2} 2^{j-i+1}\,\beta(\overline{L_{i-1}^1},\overline{L_j^1})N_j + \frac{1}{2}\beta(\overline{L_{i-1}^1},\overline{L_{i-1}^1})N_{i-1}^2 - N_i\sum_{j=1}^{i-1} 2^{j-i}\,\beta(\overline{L_i^1},\overline{L_j^1})N_j - N_i\sum_{j=i}^{13}\beta(\overline{L_i^1},\overline{L_j^1})N_j \qquad i=1,2,\ldots,12 \tag{4.17}$$

for the remaining intervals, or

$$\left.\frac{dN_{13}}{dt}\right|_{agg} = N_{12}\sum_{j=1}^{11} 2^{j-13+1}\,\beta(\overline{L_{12}^1},\overline{L_j^1})N_j + \frac{1}{2}\beta(\overline{L_{12}^1},\overline{L_{12}^1})N_{12}^2 \tag{4.18}$$

and

$$\left.\frac{dN_i}{dt}\right|_{agg} = N_{i-1}\sum_{j=1}^{i-2} 2^{j-i+1}\,\beta(\overline{L_{i-1}^1},\overline{L_j^1})N_j + \frac{1}{2}\beta(\overline{L_{i-1}^1},\overline{L_{i-1}^1})N_{i-1}^2 - N_i\sum_{j=1}^{i-1} 2^{j-i}\,\beta(\overline{L_i^1},\overline{L_j^1})N_j - N_i\sum_{j=i}^{12}\beta(\overline{L_i^1},\overline{L_j^1})N_j \qquad i=1,2,\ldots,12 \tag{4.19}$$

for the remaining intervals. Although it could be argued that the most appropriate answer to the problem is simply modelling a greater number of particle size intervals, maintaining a manageable number reduces the computational load. As a result of this practical consideration and the observed capacity of Eqs. (4.16) and (4.17) to produce an unrealistic concentration of precipitate in the highest size interval, the latter of the solutions provided above (i.e. Eqs. (4.18) and (4.19)) is selected and used throughout the remainder of this work. The chosen correction effectively disregards aggregate formation involving particles resident in the largest size class.

Focussing on the family of ordinary differential equations represented by Eq. (4.14), its most valuable attribute is that it is guaranteed to predict the first three moments of the PSD correctly. Indeed, this was the criterion used in its derivation, a condition that could equally well have applied to a combination of any three of the moments. However, problems again arise with size domain 'end effects', prompting Hounslow (1990) to offer the following amendment to ensure the total number of particles is preserved:

$$\left.\frac{dN_1}{dt}\right|_g = \frac{2G}{(1+r)L_i}\left(\left(1-\frac{r^2}{r^2-1}\right)N_1 - \frac{r}{r^2-1}N_2\right). \tag{4.20}$$

Ultimately though, the lack of fundamental basis to the formulation of Eq. (4.14) and corresponding absence of physical interpretation of its terms can complicate attempts to make such corrections and result in absurd situations such as a net loss of particles in a particular size class due solely to the existence of particles in the immediately higher interval. Though not an insurmountable problem, it is fortunate that addressing of these difficulties is made unnecessary by the observations of co-worker Dustan, who conducted the experimental program. As mentioned in Sec. 3.3.3, his rigorous analysis of the experimental measurements (Dustan 2001) reveals no evidence of particle growth for the precipitation system under consideration, a finding which is supported by a number of other studies of metal hydroxide precipitation indicating primary growth to be negligible (e.g. Domingo et al. 1994, Ilievski & White 1994, Mullin et al. 1989, Pavlides 1995). As such, it is proposed that particle enlargement is as a result of aggregation alone and that nucleation is the sole mechanism of solute deposition.

Nucleation is commonly defined in precipitation studies to occur at zero size, with nuclei then acting primarily as sites for particle growth. However, nuclei of finite size are certainly required in this analysis (i.e. in the absence of growth) to enable solute deposition to occur by nucleation. With respect to the DPB, if the size domain is discretised into a sufficient number of intervals it is usually possible to allow nucleation to occur in the smallest size interval, in which case it can be accounted for by Eq. (4.15). As is evident in Chap. 3, however, estimates of true nucleus size for the current system fall below the minimum detected by the apparatus under experimental measurement conditions. In this situation it is appropriate to introduce the notion of a source function, analytically indistinguishable from the nucleation rate, which can be considered to be the rate of appearance of particles, via growth and/or aggregation, in the smallest size interval modelled.

The overall rate of change of particle number concentration may thus be computed by summing the relevant rates presented in Eqs. (4.15), (4.18) and (4.19):

$$\frac{dN_1}{dt} = \left.\frac{dN_1}{dt}\right|_{nuc} + \left.\frac{dN_1}{dt}\right|_{agg} \tag{4.21}$$

and

$$\frac{dN_i}{dt} = \left.\frac{dN_i}{dt}\right|_{agg} \qquad i = 2,3,\ldots,13 \tag{4.22}$$

for the remaining twelve size intervals simulated. This system of coupled ordinary differential equations may be solved numerically using conventional ODE techniques, but only once the embedded rate constants (β_0 and B_0) have been determined in some way.

4.3.2 Implementation

As with the implementation of the equilibrium speciation sub-model, the computational methods employed in the solution of the discretised population balance are displayed in Sec. A.2.2 encased within an extensively annotated transcript of the Fortran requirements for the integrated CFD precipitation model. It can be observed at various points in the code of subroutine USRTRN (p. 113) that, in this study, the system of ordinary differential equations represented by Eqs. (4.21) and (4.22) is solved using a simple reverse Euler numerical integration technique – the small size of the time-step ultimately selected for precipitation simulation purposes militates against the necessary employment of more advanced and time-consuming numerical methods. Note, too, that the option exists to perform the aggregation-related population balance calculations only after the effects of nucleation on the PSD have been established. This feature is intended to be invoked when modelling equilibrium solute deposition, and is mentioned again in Sec. 4.4.2.2.

It should also be remarked that, although an array of different aggregation kernels is encoded for investigation, the scope of this work only ever entails the application of the size-independent kernel. Significantly, however, besides being mathematically the simplest to implement (a consideration which is typically of reduced significance when employing a DPB), such a description of aggregation has been recognised as being the most appropriate in a number of analyses (e.g. Bramley et al. 1996, Ilievski & White 1994, Seyssiecq et al. 2000).

Of further relevance is the fact that the DPB model detailed in this chapter was developed for a batch system. Application therefore to a semi-batch precipitation process demands that the feed be accounted for (both in the modelling of dilution effects on species and particle number concentrations and in mass balances which, together with those describing solute deposition, determine the inputs to the solution chemistry model). This is discussed in greater detail, within the context of the CFD precipitation model, in Chap. 5. As indicated above, meaningful solution of Eqs. (4.21) and (4.22) additionally

requires that correlations relating the parameters β_0 and B_0 to the outputs of the solution chemistry model are available. The approach adopted to address this need is documented immediately below.

4.4 Development of rate functions

The rate constants central to the DPB equations are themselves recognised as being functions of their local chemical and often, in the case of aggregation, physical environments; this was stated in Chap. 2. Prior to the development of expressions describing these relationships, it is necessary to establish from experiment a record, throughout the process, of both these kinetic values and the corresponding solution conditions which gave rise to them.

4.4.1 Rate extraction from experimental data

A method developed by Bramley et al. (1996) for extracting rate constants for simultaneous growth and aggregation from experimental data is followed in this study. The technique is a differential one which is used by Bramley and co-workers for illustrative purposes in conjunction with the DPB of Hounslow et al. (1988) to calculate the requisite rates. A differential method is preferred to the commonly employed integral approach both because of its relative simplicity and since it does not require some prior knowledge of what constitutes an appropriate kinetic model. Additional merits of the particular differential technique outlined here are the inclusion of the source function or nucleation rate (B_0) to account for the movement of crystals into the field of view of the particle size analyser, and the fact that it allows aggregation to be described by a size-dependent kernel. Thus, the method can be used to determine which, if any, of the existing aggregation kernels might be appropriate.

With due consideration to the nature of particle nucleation, particle growth and volume-conserving aggregation, the rates of change of the zeroth moment (or total particle number), third moment (proportional to total particle volume) and particle number in the first size interval may be written as three equations in terms of the three unknown rate constants:

$$\frac{dm_0}{dt} = \beta_0 \Phi_1 + B_0 \tag{4.23}$$

$$\frac{dm_3}{dt} = G\Phi_2 + B_0 \overline{L_1^3} \tag{4.24}$$

$$\frac{dN_1}{dt} = \beta_0 \Phi_3 + G\Phi_4 + B_0 \,. \tag{4.25}$$

Here, the following have been defined:

$$\Phi_1 = \frac{\left.\frac{dm_0}{dt}\right|_{agg}}{\beta_0} = \frac{\left.\sum_{i=1}^{\infty} \overline{L_i^0} \frac{dN_i}{dt}\right|_{agg}}{\beta_0} \tag{4.26}$$

$$\Phi_2 = \frac{\left.\frac{dm_3}{dt}\right|_{g}}{G} = \frac{\left.\sum_{i=1}^{\infty} \overline{L_i^3} \frac{dN_i}{dt}\right|_{g}}{G} \tag{4.27}$$

$$\Phi_3 = \frac{\left.\frac{dN_1}{dt}\right|_{agg}}{\beta_0} \tag{4.28}$$

$$\Phi_4 = \frac{\left.\frac{dN_1}{dt}\right|_{g}}{G}. \tag{4.29}$$

The experimental PSDs, available at discrete times, yield values for m_0, m_3 and N_1 very simply, and, though there exist difficulties generating error-free estimates (particularly if the concentration of data points is low), the rates of change can be calculated graphically or numerically from these values. Further, using the appropriate experimental data, the values of Φ_1, Φ_2, Φ_3 and Φ_4 can all be readily calculated from the relevant DPB equations, enabling Eqs. (4.23) – (4.25) to be solved to obtain the desired rate constants. Cairncross (1998) notes that these equations can also be written using matrix notation ($\mathbf{Ax} = \mathbf{b}$) as

$$\begin{pmatrix} \Phi_1 & 0 & 1 \\ 0 & \Phi_2 & \overline{L_1^3} \\ \Phi_3 & \Phi_4 & 1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ G \\ B_0 \end{pmatrix} = \begin{pmatrix} \frac{dm_0}{dt} \\ \frac{dm_3}{dt} \\ \frac{dN_1}{dt} \end{pmatrix}. \tag{4.30}$$

Since the condition for the existence of a unique solution is $\det(\mathbf{A}) \neq 0$, such a representation facilitates a simple check on the consistency and independence of the constituent linear equations.

In light of the changes made to the DPB of Hounslow et al. (Sec. 4.3.1) and which are applied to upcoming modelling attempts, associated modifications to the differential rate extraction technique detailed above are required. In the absence of growth, Eqs. (4.23) – (4.25) may be replaced by

$$\frac{dm_0}{dt} = \beta_0 \Phi + B_0 \tag{4.31}$$

$$\frac{dm_3}{dt} = B_0 \overline{L_1^3}. \tag{4.32}$$

In formulating this set of equations, the rates of change of the zeroth and third moments are selected as the measures of precipitation behaviour since, as alluded to in Sec. 3.3.2, the greatest potential relative error lies in the N_1 experimental data. In this way, it is again fortunate that particle growth may be neglected. Simultaneous solution of these equations produces expressions for the rate constants:

$$B_0 = \frac{\dfrac{dm_3}{dt}}{\overline{L_1^3}} \tag{4.33}$$

$$\beta_0 = \frac{\dfrac{dm_0}{dt} - \dfrac{\overline{dm_3}}{dt}}{\Phi}, \tag{4.34}$$

where, with application to the specific case of the modified aggregation equations just developed, and recalling that $\beta(L,\lambda) = \beta_0 f(L,\lambda)$,

$$\begin{aligned}\Phi &= \frac{\left.\dfrac{dm_0}{dt}\right|_{agg}}{\beta_0} = \frac{\left.\displaystyle\sum_{i=1}^{\infty} \overline{L_i^0} \frac{dN_i}{dt}\right|_{agg}}{\beta_0} \\ &= \sum_{i=3}^{13} N_{i-3} \sum_{j=1}^{i-2} 2^{j-i+1} f(\overline{L_{i-1}^1}, \overline{L_j^1}) N_j + \sum_{i=2}^{13} \frac{1}{2} f(L_{i-1}^1, L_{i-1}^1) N_{i-1}^2 - \sum_{i=2}^{12} N_i \sum_{j=1}^{i-1} 2^{j-i} f(\overline{L_i^1}, \overline{L_j^1}) N_j \\ &\quad - \sum_{i=1}^{12} N_i \sum_{j=i}^{12} f(\overline{L_i^1}, \overline{L_j^1}) N_j .\end{aligned} \tag{4.35}$$

4.4.2 Fitting of models to rate constants

Supported by the discrete experimental measurements of particle size and solids volume concentration, the preceding parameter extraction exercise allows profiles of the rate constants over the course of a particular run to be established. Profiles too of various solution conditions can be calculated by the repeated application to the system (dynamically defined by experimentally-derived mass balance observing both feed and solute deposition) of the aqueous chemistry model detailed in Sec. 4.2. Giving due consideration to widely recognised functional relationships (see Secs. 2.2.2 and 2.2.3), the dependence of B_0 and β_0 on selected solution conditions can then be investigated and correlations empirically determined. In so doing, no change in these correlations during the course of the precipitation (which could perhaps be indicative of a switch in mechanism) is considered. Furthermore, unless otherwise stated, the rate and chemistry profiles invoked in this exercise are those generated from experimental solids data engineered to accord with the results of total dissolved nickel assays (see Sec. 3.4.1)[†].

It is important here to reiterate that, having recourse only to the results of bulk experimental sampling, all measured values are assumed to reflect the variables' uniform distribution throughout the stirred tank; ipso facto, derived rate constants and solution properties too can only be representative of a

[†] It should be noted that the rate extraction technique was also applied to the unmanipulated solids data from the Malvern Mastersizer S™. Although the results are not presented, this exercise produced rate functions which were not only inconsistent across the various data sets collected under different operating conditions (as will be seen of the expressions developed in Secs. 4.4.2.1 and 4.4.2.2 immediately following), but which were also absurd in formulation.

practically unattainable homogeneous state. Such an abstraction can have serious implications for the efficacy of the rate functions ultimately elicited. The technique employed here was originally developed for a batch process, a mode of operation in which spatial inhomogeneities may be expected to be a great deal less significant.

4.4.2.1 Nucleation

The nucleation rate expression ultimately gleaned from 'base' experimental data of extracted nucleation rates and calculated solution properties is

$$B_n = 1.81\times10^{11}(S-1)^{0.97}. \tag{4.36}$$

'base' data alone is employed in the development of this equation, since the ideal of extracting fundamental, consistent relationships independent of operating point proved unattainable. To aid illustration of this point, profiles of precipitate yield and supersaturation for each of the experimental runs are plotted below. The precipitate yield simply reflects the percentage of nickel removed from solution; as such, it is proportional to the third moment of the PSD and its calculation takes account of the total nickel content of the system and the relevant experimentally determined precipitate molar density.

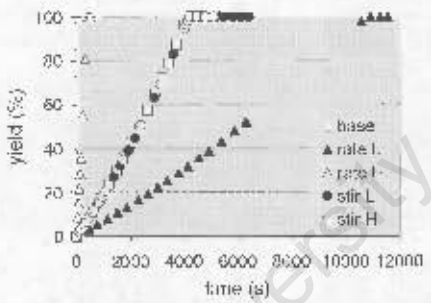

Figure 4.1 Profiles of experimentally measured precipitate yield under each of the operating conditions investigated.

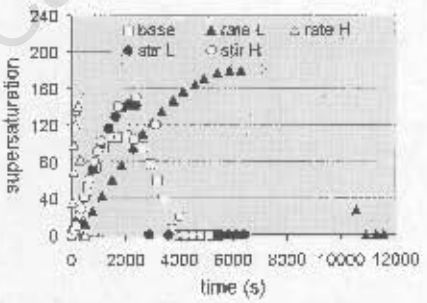

Figure 4.2 Profiles of supersaturation calculated from experimental measurements under each of the operating conditions investigated.

It is evident from Fig. 4.1 that the rates of nucleation (as the sole means of removing nickel from solution) exhibited in, for example, the 'base', 'rate L' and 'rate H' scenarios differ markedly, yet that the solution conditions at each of the data points are such that the calculated supersaturation values in Fig. 4.2 (and, indeed, those of other properties too) cover approximately the same range for the various runs. Attempts to extract a single, appropriate number-based function are further thwarted by the variability of the precipitate molar density under different operating conditions (see Table 3.2).

Perhaps the most obvious feature of Eq. (4.36) is the apparent inability to educe the high order dependence of the nucleation rate upon supersaturation typically associated with hydroxide precipitation processes (see Sec. 2.2.2). This does not suggest that either the semi-empirical power-law nucleation rate equation formulation or a high order dependence is inappropriate, but simply that such

an expression could not be derived from the available experimental data. In addition to the point made in the previous paragraph concerning the relative ranges of nucleation rate and supersaturation values, this is largely due to the further observation from Figs. 4.1 and 4.2 that there exists for each of the experimental scenarios an essentially constant rate of solute deposition irrespective of the calculated supersaturation levels. This is precisely the relationship described by Eq. (4.36). Thus the nucleation rate function extracted from the data sets of other scenarios would reveal only a change (based on the differences in molar rate of deposition exhibited in Fig. 4.1, and those in particle density) in the initial constant factor of Eq. (4.36), which dominates the predicted behaviour. Such a separate development for each operating point would be of no ultimate benefit to the model description (nor, it could be argued though, is the modelling of diverse precipitate molar densities for which no correlation is reported). Yield profiles being as they are, the extraction of the prior anticipated consistent, high order function would require flat supersaturation profiles, each of a different appropriate magnitude.

Failure of this exercise to distinguish, for example, the elevated supersaturation levels clearly responsible for 'rate II' nucleation behaviour amounts to a critical limitation with respect to isolating the controlling solution condition values in a system exhibiting high rates. This is also most likely linked to the quantity and resolution of available experimental data and the resultant assumptions of spatial uniformity underpinning the entire process aimed at the development of appropriate rate functions. In fact, the expectation that a single, apposite relationship be extracted for both nucleation and aggregation is certainly optimistic and would imply that a homogeneous representation is in all probability sufficient for modelling attempts. In light of the above discussion, Eq. (4.36) cannot be expected to perform adequately in its task of accurately capturing the dependence of the nucleation rate under a range of operating conditions.

Exploration of the potential validity of equilibrium-controlled solute deposition is motivated by the rapid nucleation rates observed (Fig. 4.1) and the relatively low calculated supersaturation values suggestive of a system verging on interphase equilibrium (Fig. 4.2). Experimental equilibrium yield profiles (Fig. 4.3, below) and attendant profiles of solution properties such as supersaturation (Fig. 4.4, below) may be generated by further reworking of the manipulated experimental data sets. Such a procedure simply entails inclusion of the solid species and its equilibrium relationship in the speciation calculations (see Sec. 4.2) associated with each of the experimental data points and the adjustment of the particle numbers accordingly (in the same manner outlined in Sec. 3.4.1). The constrained data presented in Fig. 4.4 merely manifest, by definition, the satisfaction of the equilibrium expression for the solids formation reaction.

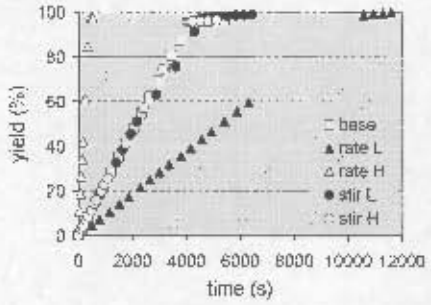

Figure 4.3 Profiles of experimentally-defined equilibrium precipitate yield under each of the operating conditions investigated.

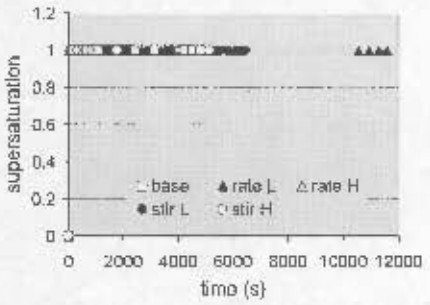

Figure 4.4 Profiles of supersaturation calculated from 'equilibrium precipitation' experimental data for each of the operating conditions investigated.

Again assuming homogeneity, Fig. 4.3 clearly further supports the notion that the system exhibits rates of solute deposition very closely approaching that prescribed by equilibrium considerations. Indeed the various corresponding differences in the yield profiles of Figs. 4.1 and 4.3 are considered likely to fall within the probable error bounds of the dissolved nickel measurements upon which the profiles in Fig. 4.1 are based. Here, besides the standard sources of random error, reference is made to the possible additional systemic source related to the sample filtration identified in Sec. 3.4.1 – one that might imply extents of deposition greater than those inferred from the measured values of total nickel in solution and therefore closer to those of Fig. 4.3. The value of the solubility product employed in the development of the equilibrium yield profiles of Fig. 4.3 is itself, however, a matter of uncertainty, as mentioned earlier in this chapter. As such, if nucleation is indeed equilibrium-controlled, too small a value of K_{sp} may well also be responsible in part or whole for the minor differences evident between Figs. 4.1 and 4.3. Moreover, comparison of the experimentally measured pH profiles in Fig. 4.5 (below) with the equilibrium pH profiles in Fig. 4.6 (below, corresponding to the yield and supersaturation profiles of Figs. 4.3 and 4.4 respectively) suggests that the predicted equilibrium extent of hydroxide removal requires limiting if agreement is to be improved. Were it not for the fact that the Ni:OH ratio in the precipitate phase is, in addition, a contentious issue (as alluded to in Sec. 3.3.3.1), this observation would have wholeheartedly endorsed the position that an increase in K_{sp} is a necessary measure to curtail solute deposition and so boost calculated equilibrium pH levels to those measured experimentally.

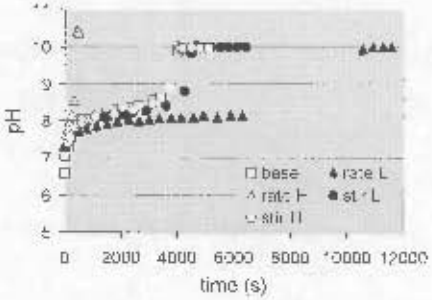

Figure 4.5 Profiles of experimentally measured pH under each of the operating conditions investigated.

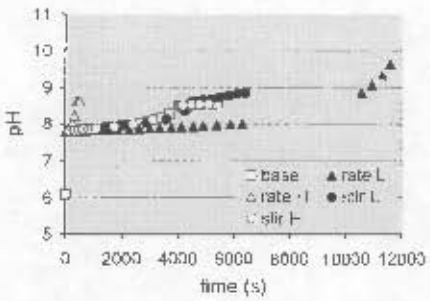

Figure 4.6 Profiles of pH calculated from 'equilibrium precipitation' experimental data for each of the operating conditions investigated.

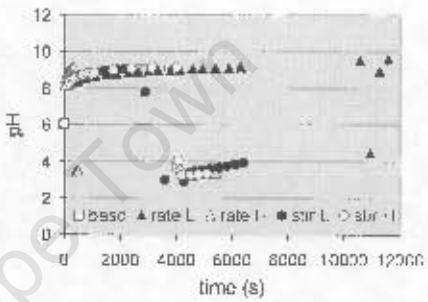

Figure 4.7 Profiles of pH calculated from experimental measurements under each of the operating conditions investigated.

The characters of the pH responses plotted in Figs. 4.5 – 4.7[†] indicate the irregular sensitivity of pH to significant relative changes in cumulative net hydroxide addition effected by small variations in yield. Such behaviour is dependent upon the existing buffering capacity of the solution and, though not unusual, betrays the complex interdependent nature of the underlying equilibrium calculations and pH definition. Recognising this sensitivity (which is of some relevance, as referred to in Chap. 6), only slight alterations in the model description may be expected to result in improved agreement between Figs. 4.5 and 4.6. Indeed, the precision of the thermodynamic model and data as a whole could even be questioned. As it stands, if the nucleation rate does represent equilibrium deposition, the problem of discerning the relative merits of the above arguments and establishing which deficiencies, whether in combination or alone, are responsible for the discrepancies both between Figs. 4.1 and 4.3 and between Figs. 4.5 and 4.6 is inextricable given the current available data. Even if this were not so, the preceding reasoning is based on assumptions of uniformity of measured and calculated values, thus making the potential insights of limited applicability. In better approximating reality, simulation of spatial heterogeneities via CFD (to be explored in Chaps. 5 and 6), in tandem with equilibrium nucleation, is

[†] Although not displayed here, it is simply mentioned as a point of interest that the various pH profiles produced by application of the speciation model to the system as defined by the unmanipulated experimental measurements of solids volume are entirely irreconcilable with those of Fig. 4.5. If the hypothesis submitted in Sec. 3.4.2 is disregarded, this constitutes further corroborating evidence of the gross errors in these measurements.

expected to reveal different overall deposition rates from those apparent in Fig. 4.3 (which effectively represent the results of a modelling approach presupposing perfect mixedness). With respect to the operating conditions studied in this work, it may be imagined that these rates would be higher, particularly in the absence of dissolution, thereby widening the gap with the bulk experimental measurements. Nevertheless, since the aforementioned uncertainties surrounding the chemical characterisation of the system persist, the notion of equilibrium nucleation is still credible and can not be discounted. At the very least, it can be posited that nucleation displays extremely fast kinetics, which, if not able to be described by equilibrium modelling, come very close to this status, and which are worthy of further investigation.

As regards the computational study of equilibrium nucleation, the modelled rate of formation of nuclei is then simply an appropriate translation (accounting for 'nucleus' volume, precipitate molar density and simulation time-step size) of the extent of deposition dictated by equilibrium chemistry considerations. Dissolution is deemed to be a relatively slow process and, as such, is not modelled; i.e. the existence of a supersaturated solution is a necessary condition (and the mere presence of solids is insufficient) for the solids equilibrium relation to be imposed. In lieu of a satisfactory rate function, such an equilibrium description is the favoured model of nucleation kinetics.

However, the considerable added expense of including the solid species in the speciation calculations can, within an integrated model structure, become prohibitive (see Sec. 6.3.4). Accordingly, an attempt has been made to develop a nucleation rate equation which mimics equilibrium-controlled solids formation behaviour. To this end, the plot featured in Fig. 4.8 (next page) may be generated by exploiting information gained through comparison, for each of the experimental runs, of the alternately manipulated data points. In an effort to represent a range of local solution compositions as similar as possible to those encountered in the upcoming modelling work (the results of which are presented in Chap. 6), the graph also includes data gathered from a CFD simulation of the precipitation process.

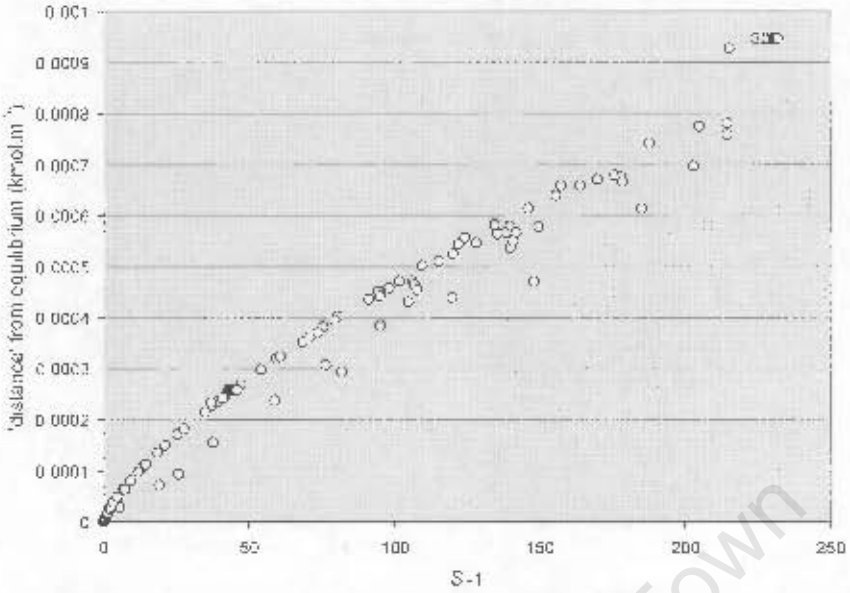

Figure 4.8 Relationship between supersaturation and the corresponding extent of nucleation required to attain equilibrium with respect to the solid species (data garnered from points defined by experiment and simulation).

Propitiously, an obvious trendline may be fitted to Fig. 4.8, yielding a power-law expression which may ultimately be converted to an equivalent number-based nucleation rate:

$$B_0 = \frac{1.265\times10^{-3}(S-1)^{0.77}}{[\mathrm{Ni}]_{ppt}\, v_0 \Delta t} \,. \tag{4.37}$$

Here, $[\mathrm{Ni}]_{ppt}$ is the relevant precipitate molar density, v_0 is the 'nucleus' volume and Δt is the simulation time-step size.

The above nucleation rate equation appears similar to a conventional one, except in relating solid quantity rather than particle number to supersaturation, and in necessarily taking explicit account of the simulation time-step size. However, in critically assessing the formulation of Eq. (4.37), it must be stressed that it does not define a true rate dependence. Rather, through the inclusion of Δt, it is merely a practical contrivance aimed at exploiting the systemic relationship depicted in Fig. 4.8 as a convenient means of reproducing the mathematical specifics of the equilibrium modelling technique (which itself ultimately employs an improvised derivative rate dictated by Δt). Accordingly, the value of the exponent in the above equation, although nonetheless of interest, cannot be interpreted as the order of the nucleation rate in the classical sense. It is important to note that there does indeed exist a rate function defining the actual (potentially high order) dependence of nucleation rate on supersaturation and/or other solution properties; it simply cannot be captured given the available experimental data and the fact that a high order dependence in particular would only be readily discernible in a system exhibiting lower rates. For practical modelling purposes though, this is not of due concern apropos the specific system of interest here, since, as mentioned earlier in this section, nucleation proceeds sufficiently fast that the reaction products seem effectively equilibrium-controlled.

An additional important feature of Eq. (4.37) is that, as with the description of true equilibrium solids formation, such a model of nucleation successfully overcomes some of the complications allied to the variability of precipitate density with operating point. Specifically, the derived number-based rate equation (incorporating the precipitate density), as opposed to a purely number-based representation such as Eq. (4.36), ensures that identical solution conditions result in the same rate of solute deposition rather than the same rate of particle formation. This is considered to be consistent with the governing theoretical principles discussed in Sec. 2.2. The 'equilibrium mimic' kinetic construction of Eq. (4.37) is used in the vast majority of simulations conducted for this study; its performance in replicating equilibrium behaviour is evaluated in Chap. 6.

4.4.2.2 Aggregation

It was previously mentioned that the most appropriate mathematical representation of aggregation behaviour is often found to be provided by the size-independent aggregation kernel, a model in accordance with which the aggregation kernel (β) comprises only the size-independent factor β_0. In establishing the correlation below, the size-independent kernel alone was considered; hence it is only appropriate for use with such a kernel:

$$\begin{aligned} \beta_0 &= 8.33\times10^{-78} e^{14pH+1050I} \\ &= \beta . \end{aligned} \quad (4.38)$$

As with the extraction of Eq. (4.36), and for the same core reasons outlined in Sec. 4.4.2.1, only data related to the 'base' experimental run was employed in the development of Eq. (4.38). Moreover, the experimental profiles selected for use in educing the above relationship are those resulting from the manipulation of the data to reflect equilibrium solute deposition under conditions of spatial uniformity, i.e. the set of profiles including Figs. 4.3, 4.4 and 4.6. The primary motivation for this decision is the slight exaggeration (with respect to a homogeneous system) of the terminal and near-terminal yields presented by the measurements of total dissolved nickel; or more specifically, the absurd calculated pH values (see Fig. 4.7) which are the most noticeable manifestation thereof.

The formulation of the aggregation expression is of some interest, insofar as aggregation is theoretically favoured (via a reduction in electrostatic potential) at high pH and ionic strength. Superficially, at least, this could be construed as a measure of endorsement. However, notwithstanding this observation, Eq. (4.38) is not considered likely to represent a true and accurate dependence of the aggregation kernel on solution conditions. This would not be any different for the functional relationships similarly educed for the other, size-dependent kernels. Ignorance of the potential influence of hydrodynamics aside, the nature of the available experimental data, the resultant necessary assumptions of spatial uniformity inherent in attempts to characterise the system kinetics and the associated inability to extract a consistent function under the various operating conditions entail that expectations of the equation's general applicability and wide-ranging predictive capability are unrealistic.

It is also instructive to note here that in practically implementing Eq. (4.38) there exists, as evident in the Fortran code of Sec. A.2.2, the option of supplying as input those solution property values calculated only after the effects on solute levels of 'simultaneous' nucleation have been computationally actualised for a particular time-step. Theoretically, such a strategy is appropriate when either of the equilibrium approaches to nucleation modelling detailed in Sec. 4.4.2.1 is employed, since the underlying concept is one of instantaneous deposition.

4.5 Concluding remarks

The selection and development of the final models of solution speciation, population balance and rate dependence, as specifically applied to the system of interest, have been reported in this chapter, as have their methods of solution. These descriptions form a central part of the overall, integrated model established in this work. The remaining model component, necessary for processes exhibiting rapid kinetics, is a representation of the relationships governing the time-dependent spatial distributions of the system variables; this is introduced in Chap. 5. However, in recognising the need to account for local variations within the reactor, it has been demonstrated in this chapter that the extracted rate functions, particularly that covering aggregation, are unlikely to offer a satisfactory description of reality. This shortcoming is due in large part to the limitations of the available experimental data.

Chapter

5

Model development and implementation (II): computational fluid dynamics

5.1 Introduction

As mentioned in the previous chapter, the mathematical descriptions submitted in this work thus far are sufficient only for the modelling of spatially invariant precipitation. CFD modelling affords the detailed knowledge of a process's spatial dynamics which is argued as being essential to the simulation of the nickel hydroxide system. The purpose of this chapter is to acquaint the reader with the structural and algorithmic details of the CFD models developed to provide ultimately a spatially-dependent description of the evolution of the solution and particulate properties. In introducing this information, the various schematic flow diagrams presented here play a central role. So too do the code file reproductions and the comprehensive annotation accompanying them (available in App. A), a resource which is considered to be an integral component of this chapter and to which repeated reference is made. Note that using the files of Sec. A.2 as a basis, it is a fairly simple exercise to generate stand-alone Fortran code for the simulation of a precipitation process in which spatially uniform conditions are assumed to exist throughout the reaction vessel.

5.2 CFX® basics

In this study, the chosen CFD modelling platform is the CFX® software package (CFX International, AEA Technology plc). The CFX® code employs a finite volume method to predict fluid flow and associated behaviours through solution of the governing differential equations. Broadly, the approach involves the geometrical definition of the region of interest, the splitting of this domain into typically thousands of small sub-volumes or cells and then the derivation and solution of a discretised representation of the problem. The final step mentioned in this sequence is worthy of some elaboration here, and may be summarised as follows:

The appropriate transport equations identified (of the form of Eq. (2.26) and potentially spanning those covering the velocity components, pressure, turbulence variables, enthalpy and other scalars such as species concentrations) are integrated over control volumes (cells) using specified procedures. This process yields a set of coupled linear algebraic equations. Each variable to be solved has its own equation in each cell which describes the effect on that variable in that cell of every other variable in that cell and every variable in neighbouring cells. The solution technique then comprises taking each variable in turn, regarding all other variables as fixed, formulating a discrete transport equation for that variable for every cell in the flow domain (as just outlined) and solving for the updated values of the variable to a desired accuracy using a specified simultaneous linear equation solver. Such a solver uses an iterative solution method in an exercise known as the inner iteration. A cycle through each of the variables to be solved defines a single outer iteration, which is repeated until the problem has satisfactorily converged. The non-linearity of the underlying coupled partial differential equations is simulated by reforming the coefficients of the discrete linearised equations, using the most recently calculated values of the variables, at each outer iteration. Thus, iteration is employed at two levels: an inner iteration to solve for the spatial coupling for each variable and an outer iteration to solve for the coupling between variables. The technique is summarised diagrammatically (and quite simplistically) in Fig. 5.5, with respect to the solution of user scalars as part of the integrated precipitation simulation discussed in Sec. 5.4

It should be noted that the treatment of pressure is slightly different from the foregoing description, since it does not obey a transport equation. Here, it is sufficient to state that the closely coupled nature of velocities and pressure directs the iterative updating of pressure and correcting of velocity components to satisfy continuity, in a procedure known as pressure-correction. The exact implementation of this method is defined by a specified velocity-pressure coupling algorithm.

The structure of the CFX® software is such that the necessary data defining the geometrical domain of the calculation and the finite difference grid is contained within a geometry file, created by the CFX® graphical pre-processor. A command file and, as required, a user-defined Fortran file provide the input specification of the flow problem. These files are submitted together to the CFX® solver module, which implements the solution algorithms to solve the various equations describing the problem.

5.3 Flow simulation

Before even considering the integration of the modelling of species and particle number concentrations (detailed in Chap. 4) necessary for the simulation of the precipitation process, it is prudent to develop an adequate description alone of the flow field generated in the laboratory stirred tank. The application of the CFX® code to this problem is documented here. Discussed in turn below are the requisite inputs to the CFX-4.3™ Solver, a double-precision run of which yields a steady-state depiction of the initial flow field established experimentally.

5.3.1 Geometry creation

A representation of the experimental geometry (defined in Sec. 3.2) is built within the CFX® graphical pre-processor; in so doing, the initial fluid volume of 3 dm^3 is modelled. The tank sides and bottom and the liquid surface are specified as boundary walls, and the four baffles are defined as walls of zero thickness. Further, since the same geometrical information described here is to be employed in the simulations of the precipitation process itself, a small three-dimensional region is identified for the later introduction of feed. Current detailed simulations of impeller-driven flow typically require subdivision of the flow domain into several hundred thousand cells. Consideration of transient precipitation modelling in such an environment suggests that the computational expense, not only of resolving the transport equations (even if using a fixed flow field), but of dynamically executing the calculations related to speciation and the population balance, would be substantial (see Sec. 6.3.4). Consequently, the generation of a fairly coarse grid is considered appropriate as a first-order measure. Such a grid does not warrant a fully predictive solution method that directly simulates the impeller geometry, and, accordingly, only patches isolating the impeller disc (defined as a zero thickness wall) and the impeller-swept volume (its function is elucidated in Sec. 5.3.2.1) are specified; the existence of the impeller shaft is neglected due to its minimal influence on the flow field.

In identifying the location of the impeller-related patches it is necessary to mention that there exists an exception to the standard geometry detailed in Sec. 3.2. Miscommunication with co-worker Dustan (who recall conducted the experimental work) over the specifics of the surface-to-impeller distance provided, resulted in what is now recognised as an error in the modelled impeller clearance. Whilst this constitutes a significant inaccuracy (a clearance of 0.0182 m as opposed to the true value of 0.057 m), it is important to note that it in no way compromises the integrity of the overall methodology, the development of which forms the core of the primary objective of the current work.

As previously intimated, a reasonably coarse volume mesh is generated, progressively slightly finer towards regions where solution variables may be expected to have high spatial gradients and therefore greater resolution is required. These areas include the feed and impeller zones, and, although not intended to resolve near-wall effects, the boundary walls. The resulting grid contains 16, 17, and 37 cell divisions in the axial, radial and azimuthal directions respectively. It is worthwhile remarking here

that the entire initial volume is modelled, rather than merely a segment of the cylinder; this is due largely to the non-axisymmetric positioning of the feed point.

Note additionally that, within the pre-processor, the system geometry is represented graphically as a rectangular 'block' by directly mapping the logical (x,r,θ) dimensions of the cylindrical tank onto (x,y,z) space; thus the z-dimension of this block is 2π. Accordingly, further definition is required: a symmetry patch is applied to the $y = 0$ plane (the axis) and the set of the two extreme z surfaces are identified as periodic boundaries, indicating the cyclic continuation of the flow domain. When the geometrical data is outputted by the pre-processor to a geometry file, it is done so using cylindrical coordinates which do not describe the rectangular construction, i.e. the mapping exercise is reversed, whereby the coordinate values specifying the topology and grid created are interpreted and reported as those of the cylindrical coordinate system. The purpose of such a coordinate reference switch is in simplifying the meshing task and helping to ensure the orthogonality of the grid, the quality of which facilitates improved simulation convergence.

5.3.2 Problem specification

The problem is defined for the solver primarily within the command file, which is reproduced in Sec. A.1.1 (as previously stated, all transcripts of code files in App. A appear together with detailed instructive comments). Note that only non-default inputs require declaration in this file.

5.3.2.1 Summary of Information

The transport equations are solved in cylindrical coordinates for steady-state, incompressible, isothermal and turbulent flow. The widely applied standard k–ε turbulence model (Sec. 2.3.3.1), which is based upon an eddy viscosity hypothesis, is employed. Notwithstanding the fact that it assumes isotropic turbulent diffusion, and is hence not ideally suited to swirling flows, the model is selected, after consideration of the low resolution of the grid, due to its comparatively limited computational requirements. Further, no other model has been conclusively shown to perform consistently better. Set as a convergence criterion controlling the outer solver iteration, the tolerance on the mass source residual (continuity error) is 1×10^{-6} kg.s^{-1}. The fluid properties supplied are those of pure water at a temperature of 25 °C. An appropriate angular velocity boundary condition is specified for the impeller disc, a free slip (zero shear stress) boundary condition is applied to the flat free liquid surface, and standard no slip conditions are specified for the remaining walls (i.e. the baffles and tank wall and bottom).

Observing the flow generated by a Rushton turbine, the stirrer action is simply modelled through the introduction of momentum sources within the impeller-swept volume (i.e. source terms are added to the velocity equations in each of the three directions as required). Note that the modelled presence of the impeller disc prevents unobstructed axial flow through the impeller region. This simplified method, one of a number of similar non-explicit techniques (Sec. 2.3.3.3), clearly fails to capture full details of

the flow within the impeller zone, and inherently neglects the characteristic unsteady flow originating from the interaction of rotating impeller blades and fixed baffles. However, considering the overall complexity of the system, the nature of the grid, the various other inaccuracies and uncertainties in the process description and, again, given the primary objective of the study, such an approach is deemed acceptable and of a consistent and suitable resolution. Approximate values for the source terms are determined through repeated 'tuning' and comparison of flow results with those of a 'snapshot' produced from a transient simulation using a comprehensive, high resolution sliding mesh technique (Sec. 2.3.3.3). The particulars of this detailed simulation are briefly outlined in Sec. 5.3.2.2 below. Here, it is not only a corresponding circulatory flow pattern that is desired, but, when investigating turbulence-controlled reaction rates in Sec. 5.4, accurate descriptions of the magnitude and distribution of turbulence variables (k and ε) are also needed. The inclusion of source terms in the modelled turbulence equations, which may be effected to achieve this end, is established as unnecessary in this instance. It should also be noted that the detailed flow simulation is only performed for the median stirrer speed of 150 rpm; when exploring alternative agitation rates, the momentum sources are merely scaled proportionately (admittedly, again a fairly crude simplification) and the impeller disc angular velocity boundary condition altered appropriately.

Supplementing the information contained within the command file is a Fortran file, which appears in Sec. A.1.2. The function of user defined Fortran routines is to describe features that are too complex to be set in the command file; here, subroutine USRTRN is called, once the steady-state flow field has been determined, to calculate a measure of the turbulent mixing time scale which is exploited later when considering the effect of micromixing limitations on the nucleation reaction rates (Sec. 5.4). The definition of the mixing time corresponds to the viscous mixing variant of the eddy break-up combustion model:

$$\tau = 0.04237\left(\frac{\rho k^2}{\mu\varepsilon}\right)^{0.25}\frac{k}{\varepsilon}. \qquad (5.1)$$

5.3.2.2 Detailed calibration simulation

As mentioned in the preceding section, the momentum sources inputted to the approximate CFD flow model used in this work are calibrated through reference to the results of a simulation employing a more rigorous modelling approach. The geometry, command and Fortran files required for this run of CFX-4.3™ Solver are generated by the program CFX-ProMixus™, a tool developed purposely for the detailed analysis of mixing vessel processes. The creation of these files (transcripts of which are not included in App. A) demands only the specification of some basic (chiefly geometrical) information, an exercise in the process of which the configurational error highlighted in Sec. 5.3.2.1 is repeated.

The resulting simulation employs a very finely meshed geometry and a high-fidelity sliding mesh technique whereby parameters are supplied and models are invoked to describe the relative motions of a rotating inner grid (incorporating the impeller) and a stationary outer grid (containing the baffles).

All stationary and rotating parts are modelled explicitly, allowing their periodic interactions to be resolved. Turbulence closure is again by the two equation k–ε model. The outcome is an accurate prediction of the swirling turbulent, time-dependent flow induced by a six-bladed Rushton turbine in the standard geometry (but for the impeller clearance) rotating at 150 rpm in 3 dm^3 of water at 25 °C. Although, for example, the development of a free surface is not considered, this approach provides one of best available representations of the 'base' flow field established in the laboratory prior to any feed addition.

An important necessary modification to the command file automatically generated by CFX-ProMixus™ is an increase in the number of modelled time steps, and hence the simulated duration. This permits the development of the flow such that a quasi (cyclical) steady state is attained, i.e. equivalent impeller-baffle alignments yield steady variable values at the computational monitoring point. It is this final 'steady' field which then guides the calibration of the cruder model. Other changes to the files include the specification of a realistic tolerance on the mass source residual, an increase in the maximum permitted number of outer solver iterations to allow this convergence criterion to be met, and calculation (as in Eq. (5.1)) of the turbulent mixing time scale. As a point of interest, since the flow is symmetrical and the results are not to be applied to the simulation of the precipitation, only half the azimuthal extent of the tank is modelled; the total number of cells constituting the finite difference grid is 134940.

5.4 Integration of model components for precipitation process simulation

The integration of the solution chemistry, precipitation kinetics and population balance models with CFD flow modelling capabilities is one of the major objectives of this work. This permits the desired dynamic description of the distributions throughout the tank of the key model variables controlling the rates of the precipitation reactions and of the resultant particulate properties.

5.4.1 Model approach and structure

As is evident from the material presented in Sec. 2.4, application of CFD modelling to precipitation typically involves the simple moment transformation of the population balance, accounting for nucleation and particle growth only. Transport equations for the moments of the PSD (which incorporate the precipitation-related source terms) are then solved alongside the chemical species transport equations, and, if employing a transient flow field, those of the hydrodynamic variables too. Significantly, aqueous speciation reactions are not considered. In this study, the nature of the particular system of interest demands that certain model features be included. Of these, it is primarily the inclusion of the modelling of complex solution chemistry, but also that of aggregation, via a discretised population balance, which, if possible at all, would clearly greatly complicate the standard approach

just outlined and which hence motivate that a novel model architecture is developed and implemented here.

Consequently, a different method of integration, still based within the CFX® modelling platform, is advanced in this work. This is achieved through the use of the user-defined Fortran subroutine USRTRN, which is called by the CFX-4.3™ Solver prior to the run, and at the end of each time step of the transient calculation. Speciation and solids formation computations are encoded within this subroutine and performed in each of the several thousand cells representing the fluid volume. The requisite local, transient conditions are supplied by specifying a CFX-4.3™ Solver scheme such that transport equations for species molarities and number concentrations of particles within each size class (which are simply modelled as 'unidentified' user scalars) are solved on their own using a fixed pre-calculated flow field (see Sec. 5.3). Under such a scenario, it is essentially only the temporally and spatially dependent advection and diffusion of particles and dissolved species which is resolved external to subroutine USRTRN. A summary of the contents and function of the various data and code files submitted to the double-precision CFX-4.3™ Solver for the simulation of the semi-batch precipitation process is provided in Fig. 5.1. Extensively annotated transcripts of the command and Fortran files are presented in Secs. A.2.1 and A.2.2 respectively. A broad representation of the overall solution strategy is portrayed in Fig. 5.2. The grey circles apparent in Fig. 5.2 illustrate (as they do in Figs. 5.3 – 5.6 as well) the initial and final states in the progression of information, and assist in reflecting the core function and ultimate objective of the series of calculations depicted. The inputs of external files and/or routines are also clearly identified.

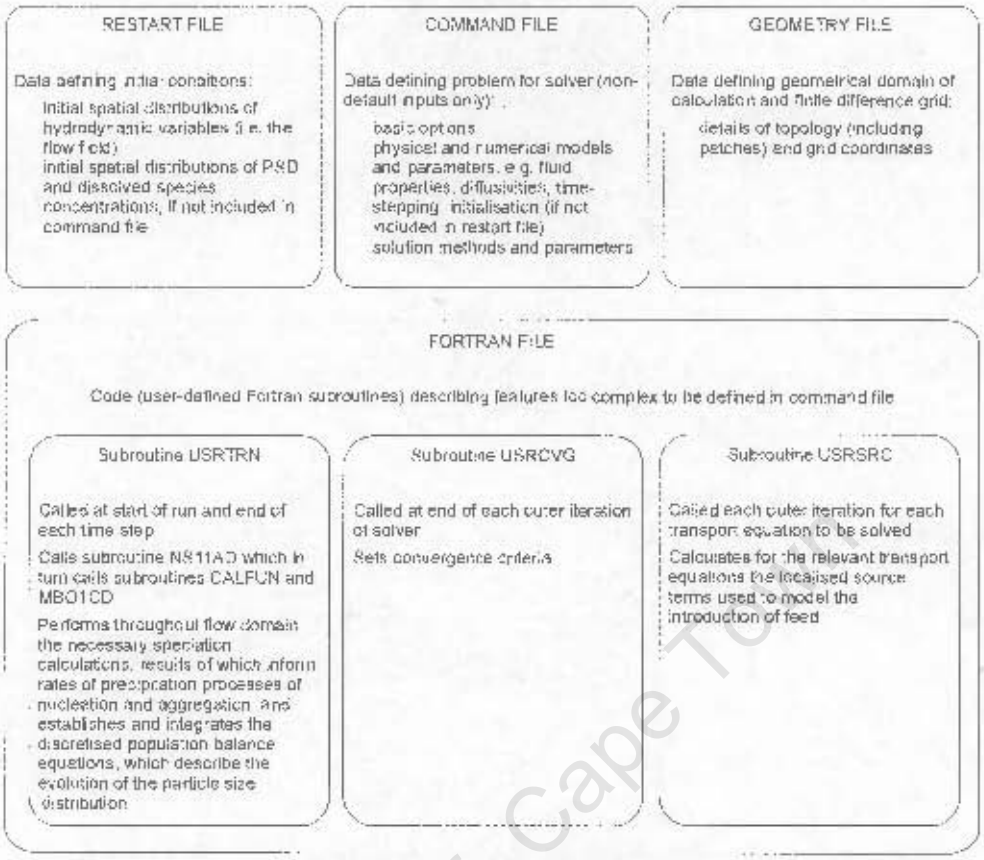

Figure 5.1 Summary of the contents and function of the various data and code files submitted to the double-precision CFX-4.3™ Solver for the simulation of the precipitation process.

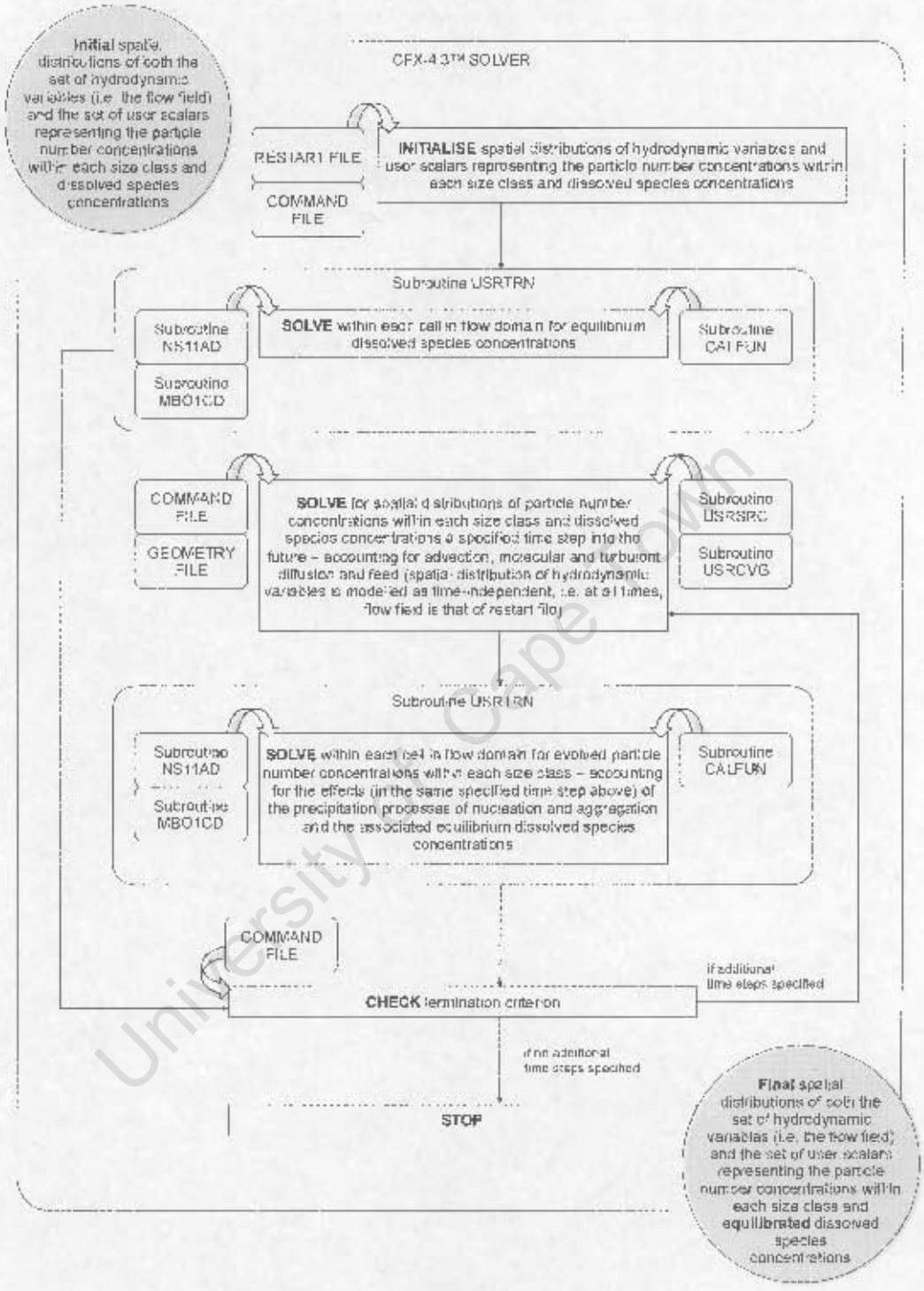

Figure 5.2 Schematic representation of the core structure of the developed modelling approach for the CFD simulation of the precipitation process.

It should perhaps be stated here that opportunities do exist for exploring possible devices aimed at transference of the defining precipitation calculations to the solver, under which circumstances they would then be performed in tandem with those of particle and dissolved species transport as previously described. In so doing, there exist potential benefits to be gained. However, as also suggested above,

there is no assurance that such methods will even be practicable without significant further approximation. The approach selected boasts guaranteed functionality and affords increased model control, flexibility and simplicity of definition. The in-built multiphase, population balance and micromixing modelling capabilities of the CFX® software are not invoked; besides the fact that they each require greater assimilation of calculations into the solver, this is due to issues surrounding flexibility, ease of integration with other model features, and, in the case of multiphase considerations, the attendant computational burden. Important associated assumptions underpinning the model, and which appear reasonable for the low concentrations, low solids volume fractions and small particle sizes encountered (Madec et al. 2001), are that the presence of the dissolved species and solids volume have no effect on the fluid properties or the flow and that the behaviour of particles is akin to that of dissolved species (i.e. no gravitational, buoyancy or drag effects are taken into account).

Figs. 5.3 – 5.6 each provide greater detail of the underlying algorithms which are only broadly outlined in Fig. 5.2. Of these diagrams, it is the information communicated in Fig. 5.4 which is of greatest interest, since this represents the majority of the user-defined modelling. Significantly, although the models presented in this section are designed for use with the coarsely-meshed, simplified geometry and the associated approximate flow field described in Sec. 5.3, the Fortran file could essentially be run unaltered in combination with a detailed geometry and high resolution flow calculation (either a quasi steady-state representation such as that outlined in Sec. 5.3.2.2 or the transient field itself). Both proposals would clearly substantially inflate simulation times and would simply necessitate various minor adjustments primarily to the relevant command and geometry files. Similarly, slight alterations principally to the contents of subroutine USRTRN could also easily realise, amongst other upgrades, the modelling of a smaller nucleus size or an alternative aggregation kernel or the inclusion of growth or other particulate processes.

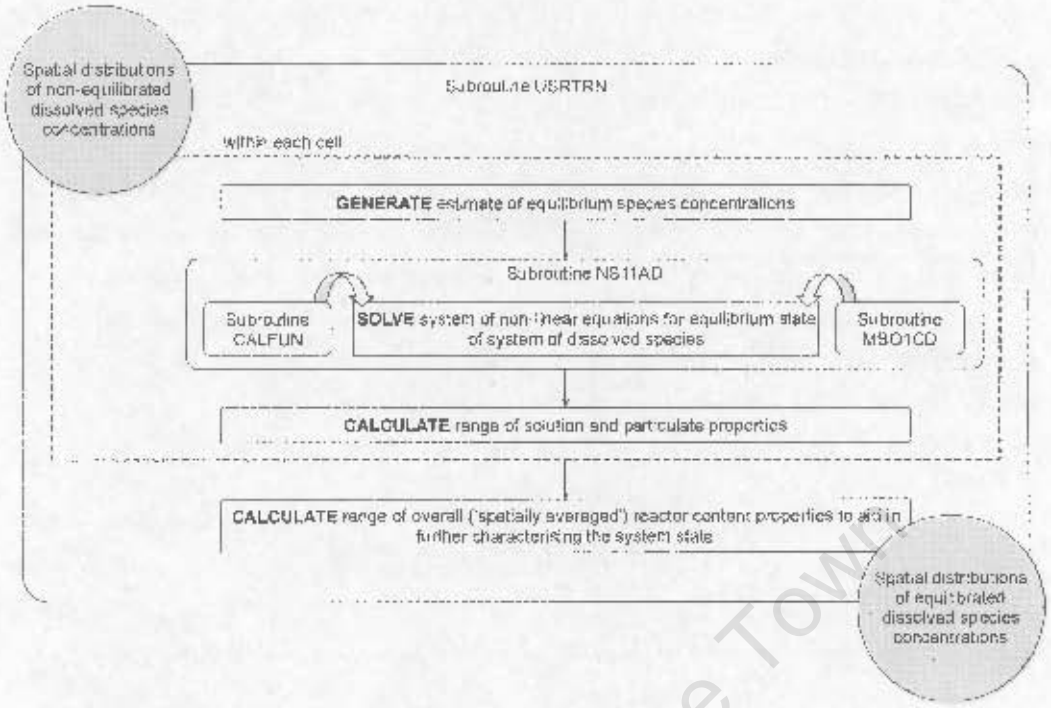

Figure 5.3 Focus on the flow and development of information in the first call to subroutine USRTRN depicted in the basic cycle of Fig. 5.2.

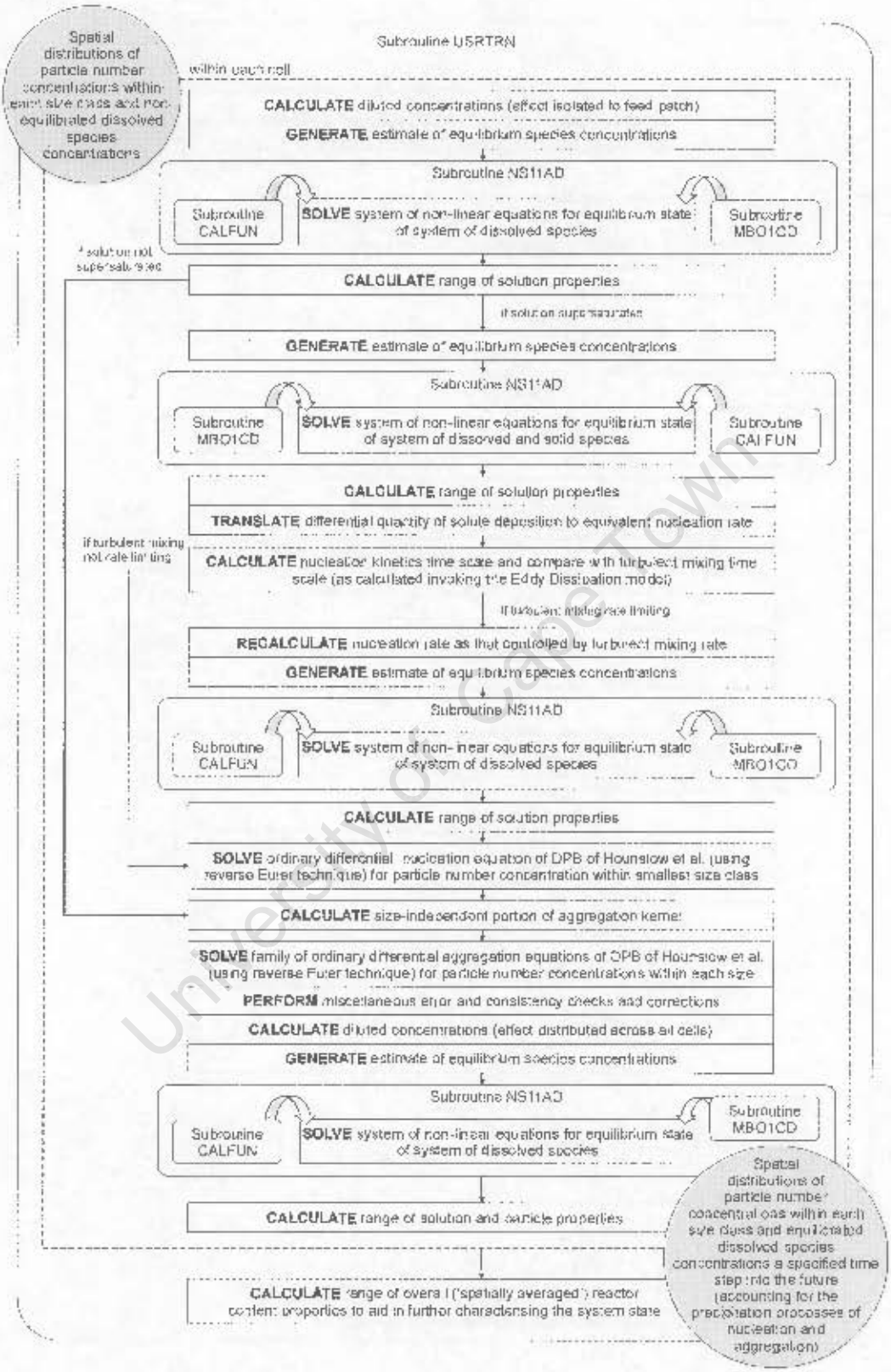

Figure 5.4 Focus on the flow and development of information in the subsequent calls to subroutine USRTRN depicted in the basic cycle of Fig. 5.2 (note certain licence taken in simplifying this schematic representation).

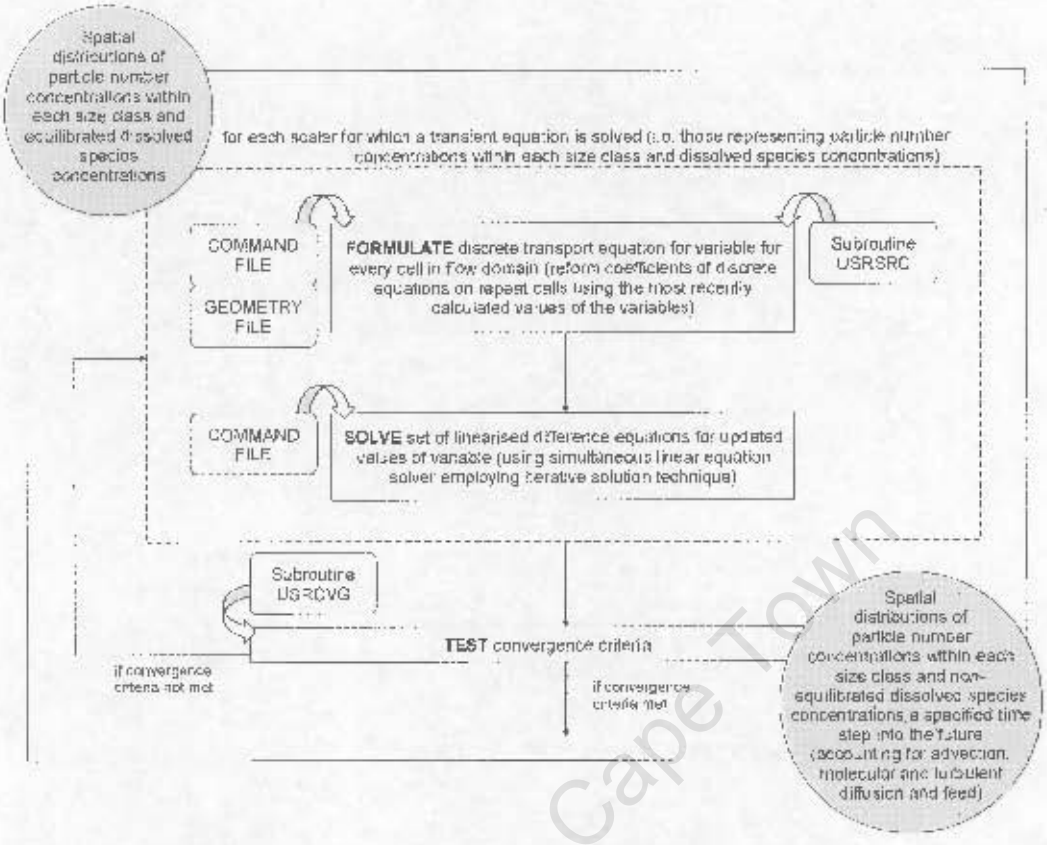

Figure 5.5 Focus on the underlying algorithm of the central solver block depicted in the basic cycle of Fig. 5.2.

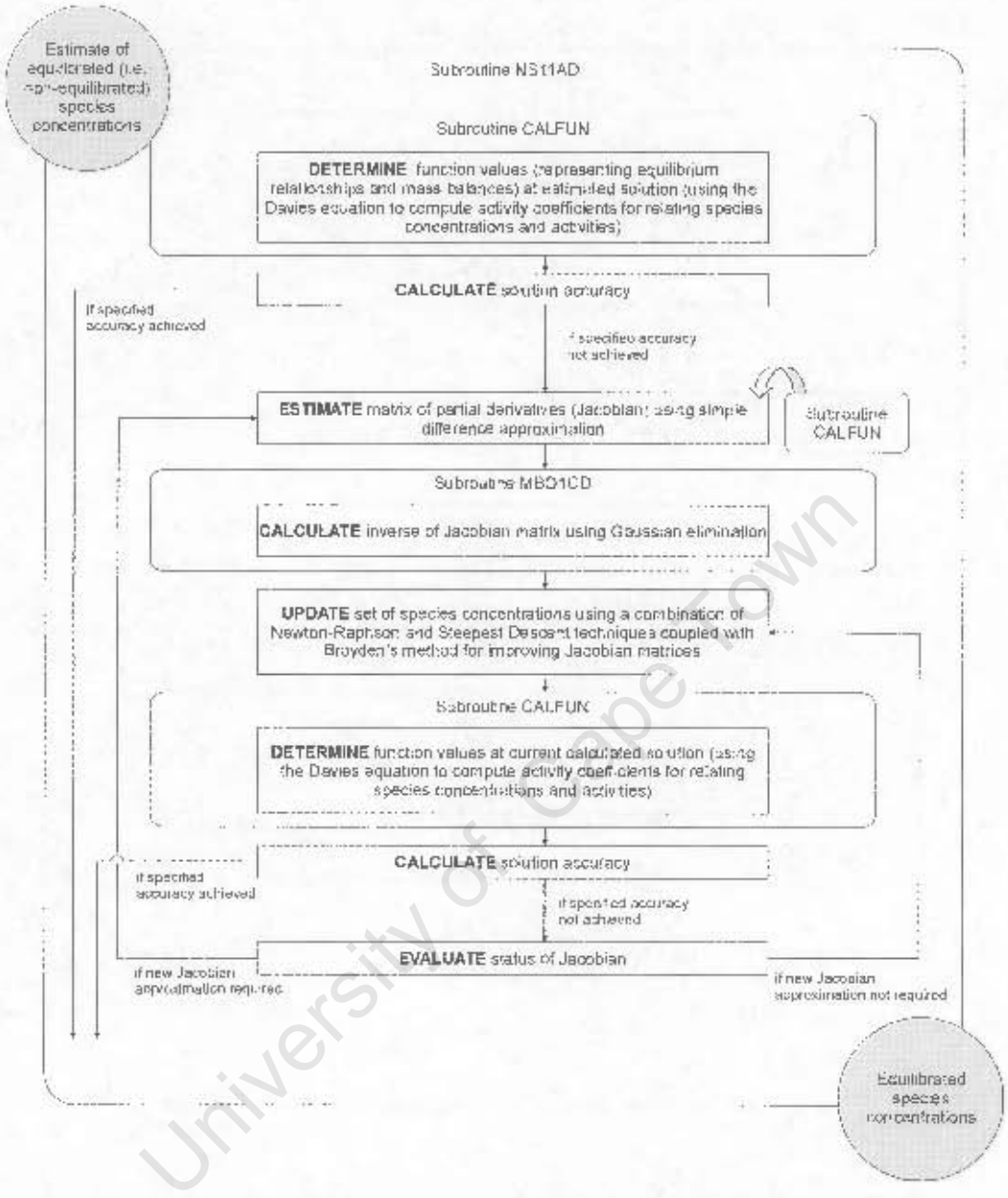

Figure 5.6 Focus on the underlying algorithm of subroutine NS11AD depicted in the subroutine structures of Figs. 5.3 and 5.4.

It should be noted that Fig. 5.4 portrays the sequence of calculations executed for an equilibrium representation of nucleation and for a specific preferred 'dilution model'. If interested in alternative descriptions, please refer to the USRTRN subroutine in Sec. A.2.2, within which all options are encoded (such options, or 'general settings', are selected up-front on pp. 116 and 117). Indeed, if seeking code summaries or greater detail in general, the reader is directed to the code, the structure of which deliberately favours linearity and ease of comprehension over elegance or concision. The diagrams presented in Figs. 5.1 – 5.6 are also regarded as being self-explanatory, and although certain licence to simplify has necessarily been taken in these schematic depictions, they are considered

sufficiently detailed that anything but a very lengthy accompanying textual overview would prove redundant.

5.5 Concluding remarks

The specifics of the CFD modelling component of this study have been documented in this chapter. This includes an outline of the CFX® file structures and general solution methods, details of the simulation of the flow field established in the stirred tank and a description of the integrated model structure aimed at being of ultimate value in endeavours to simulate precipitation systems similar to the one investigated in this work. In so doing, it has been demonstrated that, approximations and inaccuracies notwithstanding, a model core of significant generic value has been developed. As such, the framework is simple to refine, easily accommodating upgrades as greater fundamental understanding or processing power becomes available or as increased experimental resolution allows improved system characterisation.

Chapter

6

Precipitation simulation results and discussion

6.1 Introduction

This chapter demonstrates the application of the models developed in the preceding chapters to the system under investigation. Although a great number of simulations were performed (generating a vast array of output data), only those results sufficient to display and support selected, noteworthy findings are presented. Whilst this does not constitute an in-depth analysis of the data[†], it is regarded as adequate to display the value and integrity of the developed model structures.

In order to simplify references, from hereon in simulations performed under the assumption of uniformly distributed reactor contents are simply known as "homogeneous simulations" and those resolving the spatial dynamics of the system (i.e. those performed using the CFX® software) are referred to as "CFD simulations". It should also be noted that the various simulation data points displayed in the remainder of the chapter are not as 'discrete' as might be inferred from their presentation. In actual fact, results are on hand at the end of each simulation time-step and the discrete markers are employed merely to aid distinction between the different series.

[†] It could be argued that the full range of available data, even those displayed here, warrant further, more in-depth analysis. Such an undertaking is, however, only truly merited and of obvious value when the models have a predictive capacity, a lack of which will become apparent in Sec. 6.2.

6.2 Comparison of simulation results with experimental data

The effective decoupling of the analyses of yield and average particle diameter results is enabled by the absence of particle growth as a mechanism of both solute deposition and size enlargement in the representation of the system. Nucleation is defined by yield alone, whilst average particle diameter, though affected by the rate of nucleation, is primarily a measure of aggregation. (Coefficient of variation data are deemed surplus to the specific requirements here and are therefore not presented.) Note that experimental results supplied in this section reflect that data manipulated as outlined in Sec. 3.4.1.

Prior to the comparison of nucleation and aggregation behaviours in the sections to follow, it is meaningful to contrast the experimental and simulation cumulative total volume profiles presented below.

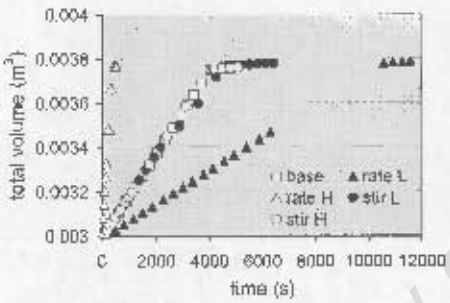

Figure 6.1 Profiles of experimentally recorded total volume for each of the scenarios investigated.

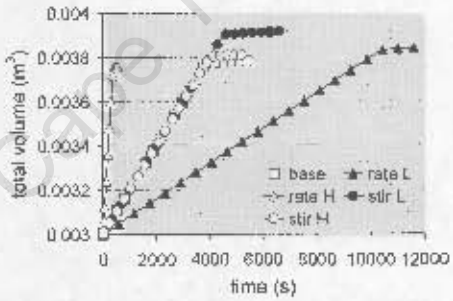

Figure 6.2 Profiles of modelled total volume for each of the scenarios investigated.

In laboratory practice, difficulties were encountered in maintaining the constant target NaOH feed rates listed in Table 3.1. Simulation codes, however, employ these precise target values throughout the period of constant NaOH addition and only make use of experimentally recorded rates during the pH control phase. These factors account for the obscure differences evident within and between Figs. 6.1 and 6.2. In hindsight, the inaccuracies in simulated feed rates amount to modelling errors given the extreme sensitivity of the results to net hydroxide input.

6.2.1 Nucleation

Experimental yield profiles for each of the experimental runs were previously presented in Fig. 4.1. For convenience, this plot is reproduced below (as Fig. 6.3) together with the profiles as predicted by homogeneous simulations employing nucleation kinetics extracted from manipulated experimental data (i.e. Eq. (4.36)) (Fig. 6.4) and employing equilibrium nucleation (Fig. 6.5).

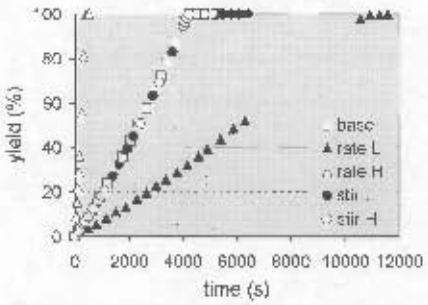

Figure 6.3 Profiles of experimentally measured precipitate yield under each of the operating conditions investigated.

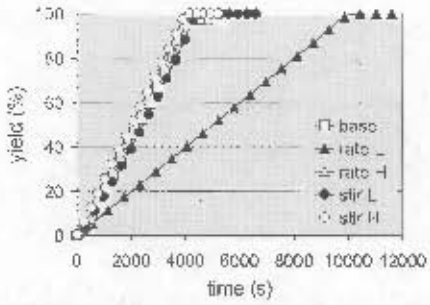

Figure 6.4 Profiles of precipitate yield predicted by homogeneous simulations employing nucleation kinetics extracted from manipulated experimental data.

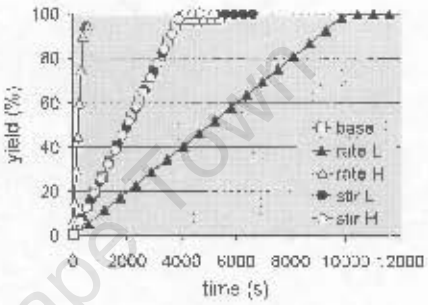

Figure 6.5 Profiles of precipitate yield predicted by homogeneous simulations employing an equilibrium description of nucleation.

The most important observation to be made is that the kinetic nucleation fails to capture the experimental behaviour, with particular reference to the 'rate H' scenario (Fig. 6.4). This is not at all unexpected since, as discussed in greater detail in Sec. 4.4.2.1, the nucleation rate equation predicts an essentially constant number-based rate of nucleation irrespective of supersaturation. Due to the low 'rate H' density reported in Table 3.2, this results in a rate of solute deposition even less than that for the 'base' scenario. 'rate L' behaviour is only successfully simulated because of the facts that:

the overall rate of deposition is automatically limited by the requirement that the solution be supersaturated for nucleation to occur;

the extent of deposition within any single time-step is limited by the small size of the simulation time-step employed; and,

assuming spatial uniformity, experimentally observed solids formation closely approaches that prescribed by equilibrium.

Indeed, barring extremes in modelled density, the nature of the nucleation kinetics is such that they may be expected to reproduce experimental data for addition rates equal to or less than the 'base' rate, under which conditions the nucleation rate equation was extracted. Correspondingly, a larger (within reason) constant rate, such as that which may be elicited from the 'rate H' experimental data, may be expected to satisfactorily reproduce the yield response of each of the experimental scenarios. However, such a model is clearly not an accurate representation of reality and, for obvious reasons, it is not a viable

description of the process. Not least of these reasons, which may be imagined being amplified when conducting CFD simulations (due to their highly interdependent nature), is the effect on the solution conditions controlling aggregation behaviour of the recurrent, systematic overprediction, and subsequent absence of nucleation, characterising the simulation of lower feed rates.

Examination of Fig. 6.5 reveals that equilibrium precipitation, unlike the kinetic nucleation as described by Eq. (4.36), is very successful in accurately predicting all the various experimentally observed yield profiles. This is again somewhat unsurprising given that these homogeneous simulations should effectively replicate the 'experimental' data of Fig. 4.3, which is, in turn, in close agreement with Fig. 6.3. It should be recognised, however, that the results are achieved without the incidence of overprediction mentioned above.

It is conceivable that, under assumptions of homogeneity, the results of 'base', 'stir L' and 'stir H' simulations might be presumed to overlap. Disparity between these profiles in Fig. 6.4 is the result of differences in density (Table 3.2) and cumulative total volume (Fig. 6.2). In Fig. 6.5, the corresponding differences can be attributed to added volume alone, since equilibrium nucleation representation overcomes the complications associated with the variability of the precipitate molar density.

Presented below are comparisons between simulation and experiment of yield and pH responses for the 'base' scenario. The 'CFD – overall' data series reflect those inclusive overall values 'averaged' for the entire volume, whilst the 'CFD – bulk' series represent the results as calculated from the data related to a selected single cell in the bulk zone (level with the turbine disc and diametrically opposite the feed location), which may be considered to correspond to the experimental sampling point. It should be noted that 'CFD – overall' values are undefined for variables such as pH and supersaturation. Since an extended CFD simulation employing equilibrium nucleation is not available, due to the substantial associated computational burden, both the CFD and homogeneous simulations for which results are displayed utilise the 'equilibrium mimic' nucleation kinetics developed in Sec. 4.4.2.1 (Eq. (4.37)). The success of this 'mimicry' is addressed in Sec. 6.3.1.

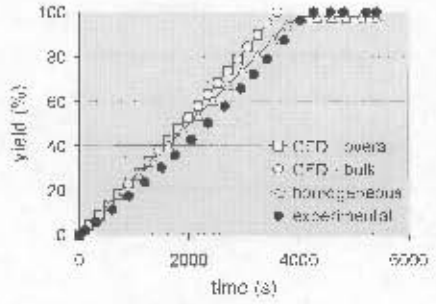

Figure 6.6 Comparison of experimentally measured and various simulated precipitate yield profiles for the 'base' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

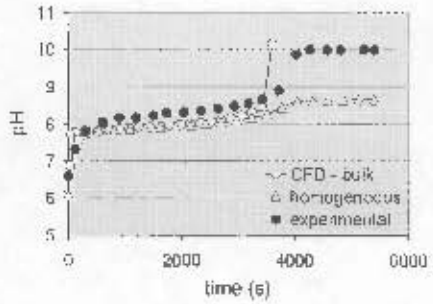

Figure 6.7 Comparison of experimentally measured and simulated pH profiles for the 'base' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

The slight overprediction of yield by the homogeneous model in Fig. 6.6 is not surprising considering the material presented in Sec. 4.4.2.1, where the experimental yield was simply noted as approaching equilibrium solute deposition. The CFD simulation calculates a yield only a little greater than the homogeneous simulation, a result which was to be anticipated due to the expected development of local regions of high supersaturation and the fact that dissolution of precipitate is not modelled. In so doing, however, it still displays fairly good agreement with experiment. Another interesting feature of this plot is the excellent correspondence between 'CFD – overall' and 'CFD – bulk' data. Considered in tandem with the similarity between CFD and homogeneous simulation results, this could be interpreted as supporting the notion of a spatially uniform system. More on this topic is offered later in the chapter (Sec. 6.3.2).

Bearing in mind the sensitive nature of the pH response, the level of correspondence between experimental and CFD simulation data in Fig. 6.7 is not poor. Recognition that the greater yield predicted by the CFD simulations might have been expected to manifest in related pH values lower than those calculated by the homogeneous code, underscores the fact that analyses of CFD results are not straightforward. Certainly, study of the data history of a single cell in isolation is insufficient for the purposes of full reconciliation, since CFD results, in general, understandably expose the spatially interdependent nature of the modelling approach.

The homogeneous simulation profiles of the two preceding plots correspond in essence to the 'base' experimental results of Figs. 4.3 and 4.6. The tangled issues (previously raised in Sec. 4.4.2.1) of possible minor adjustments to K_{sp} and Ni:OH values and potential inaccuracies in experimental yield as calculated from total dissolved nickel measurements are again brought to the fore by the full spectrum of information presented in both Figs. 6.6 and 6.7.

Abbreviated yield profiles appear in App. B for each of the five operating points investigated experimentally (Figs. B.1 – B.5). For the 'base' scenario, this amounts to a focus on the initial 180 s of Fig. 6.6. All these plots exhibit similar trends to Fig. 6.6, i.e. CFD simulations predict a slightly greater rate of deposition than homogeneous simulations, which, in turn, marginally overestimate experiment.

On the whole, simulations are relatively successful in approximating experimentally determined yield. Conversely, the author is confident that CFD simulations employing nucleation kinetics extracted from manipulated experimental data would, as the equivalent homogeneous simulations did (illustrated in Figs. 6.3 and 6.4) and for the same basic reasons, fail to capture experimentally observed behaviour, specifically that of the 'rate H' scenario. Furthermore, the continued agreement between 'CFD – overall' and 'CFD – bulk' data evinced in Figs. B.1 – B.5 suggests that the bulk experimental sampling of the system may well provide an accurate measure of the overall system.

For the sake of interest, also included (in Fig. B.6) are the simulation results for the theoretical case of a practically zero impeller speed ('stir 0'), otherwise defined as for the 'base' scenario. For demonstration purposes, the 'base' precipitate molar density is employed in these simulation codes. Note that the selection here of an alternative value should, however, not have any effect on yield results (unlike those of, for example, average particle diameter); this is due to the formulation of the utilised nucleation rate equation (Eq. (4.37)), into which the density parameter is absorbed. Applying similar reasoning, homogeneous simulation results for 'base', 'stir L', 'stir H' and 'stir 0' scenarios are duplicate (a fact alluded to previously in this section), since, for want of a stirrer speed input, the only differences in the relevant process model descriptions is the precipitate molar density. In line with the anticipated behaviour in a system with such limited agitation, the low CFD simulation predictions presented in Fig. B.6 most likely constitute evidence of localised nickel depletion in the feed region. Notwithstanding the increased demands of such a particular scenario on the accurate representation of dilution and micromixing effects, all in all, Figs. B.1 – B.6 display sensible, intuitive responses to changes in operating conditions. With particular reference to the various corresponding differences between 'CFD – overall', 'CFD – bulk' and homogeneous simulation output, this serves as a testament to the mathematical and structural integrity of the models.

Viewed in combination, the results presented in this section and in App. B seem to indicate that an equilibrium description of nucleation is appropriate, subject to prospective tweaking of K_v and Ni:OH values. In so proclaiming, the low order of the 'equilibrium mimic' nucleation rate equation is of some interest, as mentioned earlier in Sec. 4.4.2.1. Motivated by this finding, all further simulation output displayed represents the results of code making use of equilibrium nucleation or 'equilibrium mimic' nucleation kinetics. As such, for all practical purposes (assuming the rate of dissolution is not limiting and given reasonable agitation), a single equilibrium calculation is in fact sufficient to determine the final product yield. However, it must be stated that this fact would only render the CFD model of no obvious value if nucleation were the only particulate process (aggregation has been identified as the prime determiner of product quality (Dustan et al. 2005a, Dustan et al. 2005b)) or if, for example, the mode of operation were switched to batch. Whilst it is conceded that, for a single batch addition, a spatially uniform distribution of nuclei and solution properties would soon develop, the semi-batch nature of the process (allied to practical mixing limitations) dictates that non-homogeneity persists in the vessel for sufficient duration for meaningful, kinetically-controlled aggregation phenomena to proceed in just such a spatially distributed environment. Thus, due to the system's innate inter-dependencies, both nucleation and aggregation require spatially non-uniform representation and,

notwithstanding the rapid nucleation, there is hence inherent value in the CFD modelling approach[†]. Indeed, the specifics of the manner of contacting has a significant effect on the final precipitate product, a fact manifested experimentally in the different particle size characteristics produced under the various operating conditions.

6.2.2 Aggregation

The experimental and homogeneous simulation profiles of average particle diameter presented below in Figs. 6.8 and 6.9 are chiefly intended to display the simple reality that, with the exception of the 'base' scenario, the homogeneous model clearly falls short of accurately simulating the experimentally measured response. This phenomenon is not in the least surprising for reasons to be discussed shortly. Although by no means an established fact, the unusual experimental behaviour exhibited by the 'rate L' scenario does foster speculation as to the accuracy of the laboratory data, especially given the reservations concerning the measurement techniques etc. expressed in Chap. 3. Additionally, it should be borne in mind when considering the following figures that the maximum attainable average particle diameter is limited by the finite range of particle size classes modelled.

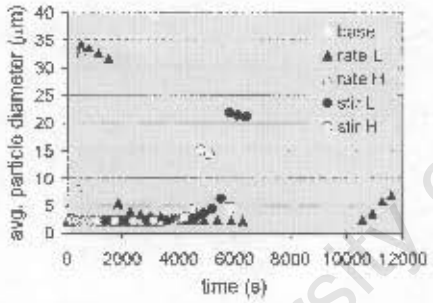

Figure 6.8 Profiles of experimentally measured average particle diameter under each of the operating conditions investigated.

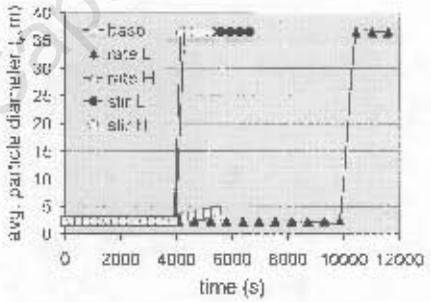

Figure 6.9 Profiles of average particle diameter predicted by homogeneous simulations employing an equilibrium description of nucleation.

It has been previously mentioned with respect to yield results that the only effective differences in the mathematical descriptions of the 'base', 'stir L' and 'stir H' scenarios are the modelled total volume profiles as illustrated in Fig. 6.2. An additional difference which affects calculated average particle diameter is the value of the precipitate molar density, which dictates the number of 'nuclei' produced given a certain molar extent of deposition. However, it is still the former difference which is almost solely responsible for the disparate profiles evidenced for these three scenarios in Fig. 6.9 – this through the effect of NaOH addition on pH profiles (see Fig. 6.10) and the formulation and resultant sensitivity of the aggregation expression.

[†] If the rate of nucleation is lower, the value in the approach is readily apparent.

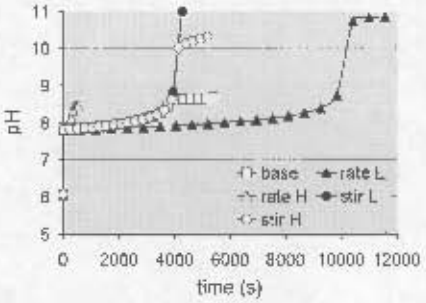

Figure 6.10 Profiles of pH predicted by homogeneous simulations employing an equilibrium description of nucleation.

The above plot also serves to reiterate the extreme sensitivity of pH to hydroxide addition and removal through reactor feed and solids formation respectively.

The profiles of average particle diameter associated with Figs. 6.6 and 6.7 appear in Fig. 6.11 (below), the most important features of which are the respective success and failure of the homogeneous and CFD simulations in satisfactorily predicting experiment. Sudden and rapid increase in the average particle diameter forecast by the homogeneous simulation is most likely the result of the favourable solution conditions for aggregation (in particular pH) accompanying nickel depletion. Similarly, the earlier rapid onset of particle enlargement observed in the CFD model data can be attributed to the pH response in the bulk (refer to Fig. 6.7). In general, CFD results are indicative of fairly complex interplay between solution conditions, particulate properties and aggregation rate on the local level. A further feature of the following plot is again the excellent agreement between 'CFD – overall' and 'CFD – bulk' series.

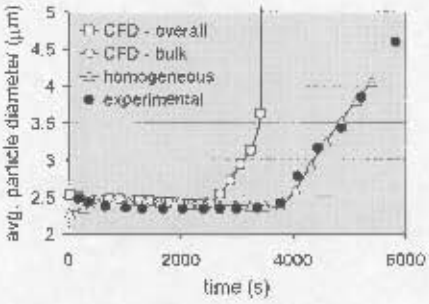

Figure 6.11 Comparison of experimentally measured and various simulated average particle diameter profiles for the 'base' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

The superior fit of homogeneous model data to experiment is entirely to be expected as a direct result of the assumption of spatial uniformity underpinning the aggregation rate extraction exercise. This raises the obvious question about how one goes about validating the CFD modelling work. Indeed, the

only model for which acceptable agreement with experiment could reasonably have been anticipated is the homogeneous 'base' code, for these are the conditions under which the aggregation behaviour of the system was characterised. There is thus clearly no real predictive capability in either CFD or homogeneous models with respect to particle enlargement, a result† which was not unforeseeable in light of the information presented in Chaps. 3 and 4. Although acknowledged as a significant shortcoming considering that it is the aggregation behaviour which is central in determining precipitate product character and the resultant dewatering performance, the fact that the experimentally derived expression for aggregation has little practical value does not compromise the inherent value in the modelling framework.

In addition, regardless of the above limitations, the structural soundness and self-consistency of the models developed in Chaps. 4 and 5 is not in doubt. This is further illustrated by the abbreviated profiles of average particle diameter (corresponding to those of yield discussed briefly in Sec. 6.2.1) presented in App. B. (Figs. B.7 – B.12), which again display smooth, readily explainable basic trends.

6.3 Further observations of interest

Additional findings deemed significant in the context of this chapter's objectives are selected below for exposition and discussion.

6.3.1 Success of 'equilibrium mimic' kinetics

Figs. 6.12 – 6.15 (overleaf) show the near-perfect agreement between the results of 'base' homogeneous simulations employing equilibrium nucleation and those encoded with the nucleation rate developed to mimic it. Similarly, comparison of Figs. 6.5, 6.9 and 6.10 with Figs. B.13 – B.15 respectively, indicate that this accord applies equally well to each of the operating scenarios.

† This result also constitutes the underlying motivation for the greater focus, throughout the dissertation, on yield data than on particle size data, despite recognising the latter as the area in which the modelling is of most value – the relative success of the nucleation representation permits potentially valuable insights into the nature of the test system and the analysis has consequently produced some observations and discussions of interest.

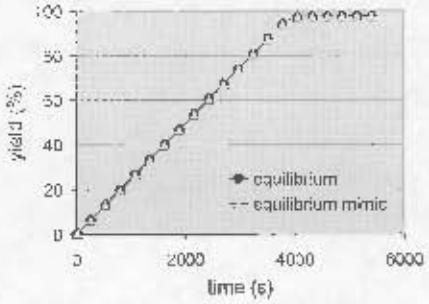

Figure 6.12 Comparison of 'base' precipitate yield profiles predicted by homogeneous simulations employing equilibrium nucleation and kinetic 'equilibrium mimic' nucleation.

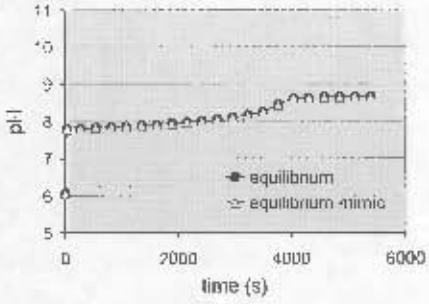

Figure 6.13 Comparison of 'base' pH profiles predicted by homogeneous simulations employing equilibrium nucleation and kinetic 'equilibrium mimic' nucleation.

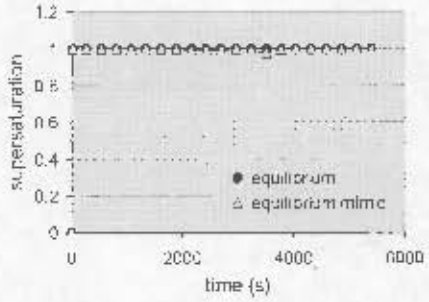

Figure 6.14 Comparison of 'base' supersaturation profiles predicted by homogeneous simulations employing equilibrium nucleation and kinetic 'equilibrium mimic' nucleation.

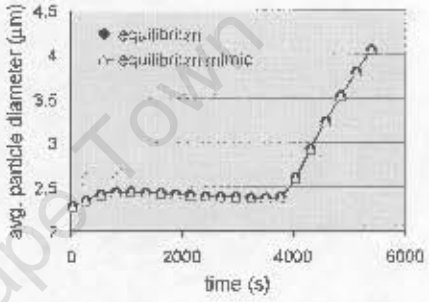

Figure 6.15 Comparison of 'base' average particle diameter profiles predicted by homogeneous simulations employing equilibrium nucleation and kinetic 'equilibrium mimic' nucleation.

Conducting the equivalent investigation for 'base' CFD simulations, Figs. 6.16 – 6.18 also reveal a great degree of overlap, suggesting that the formulation of the nucleation rate equation is indeed successful in mimicking equilibrium precipitation. Fig. 6.19, however, again highlights the highly sensitive nature of aggregation calculations, such that what must amount to only very small discrepancies in predicted solids deposition result in sufficient variations in local solution conditions to have a significant impact on the rate of aggregation. This is envisaged to be primarily applicable to the feed region. The more extreme solution conditions developed locally here do not occur under the assumption of spatial uniformity. This fact, together with the vastly superior number of calculations of extent of nucleation required by the CFD modelling approach, accounts for the comparative match and mismatch displayed in Figs. 6.15 and 6.19 respectively.

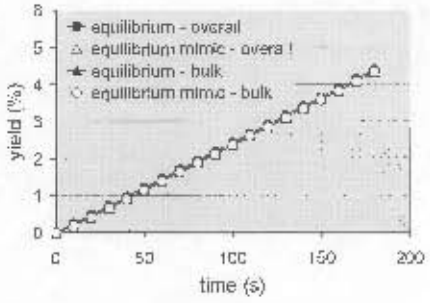

Figure 6.16 Comparison of various 'base' precipitate yield profiles predicted by CFD simulations employing equilibrium nucleation and kinetic 'equilibrium mimic' nucleation.

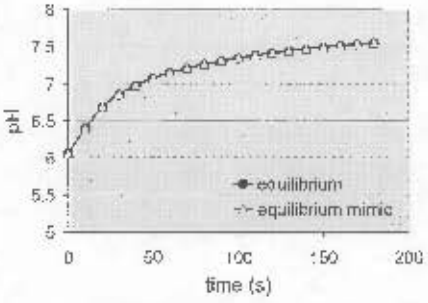

Figure 6.17 Comparison of 'base' pH profiles predicted by CFD simulations employing equilibrium nucleation and kinetic 'equilibrium mimic' nucleation.

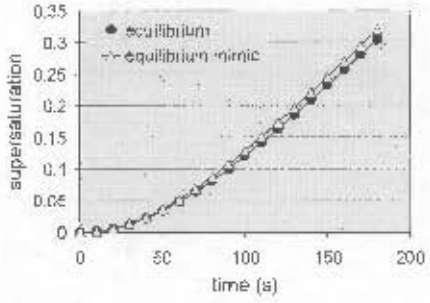

Figure 6.18 Comparison of 'base' supersaturation profiles predicted by CFD simulations employing equilibrium nucleation and kinetic 'equilibrium mimic' nucleation.

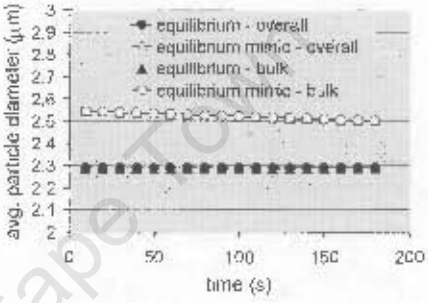

Figure 6.19 Comparison of various 'base' average particle diameter profiles predicted by CFD simulations employing equilibrium nucleation and kinetic 'equilibrium mimic' nucleation.

Despite the relatively poor fit exhibited in Fig. 6.19, it should be pointed out that the illustrated range in the values of average particle diameter is very limited and that the output of the two models may be expected to show analogous trends and even to converge somewhat if a comparison over a longer time period were available.

All things considered, the 'equilibrium mimic' nucleation kinetics may be considered very successful. Recalling that the motivation for the development of this expression was the considerable computational expense incurred by the explicit inclusion of the solid species in equilibrium calculations (see Sec. 4.4.2.1), the equation and general approach constitute a valuable tool in instances of limited accessible computing power.

6.3.2 Spatial distribution of reactor contents

It was mentioned in Sec. 6.2.1 that the level of correspondence between the CFD simulation, homogeneous simulation and experimental profiles presented in Fig. 6.6, and the overlap of calculated 'CFD – overall' and 'CFD – bulk' values, suggested that the system might indeed be spatially homogeneous. Although the agreement between 'CFD – overall' and CFD – bulk' data persists, the disparity between CFD and homogeneous simulation results revealed in plots displayed subsequently

(e.g. Figs. 6.7 and 6.11) dismisses this idea, indicating that the reactor contents are non-uniformly distributed.

Additional unmistakable evidence supporting the hypothesis of a spatially diverse system is offered in Figs. 6.20 and 6.21 (below). The most pertinent responses exhibited in Fig. 6.20 are the simulated profiles of the maximum and minimum values of pH throughout the stirred tank volume, as calculated by the CFD code.

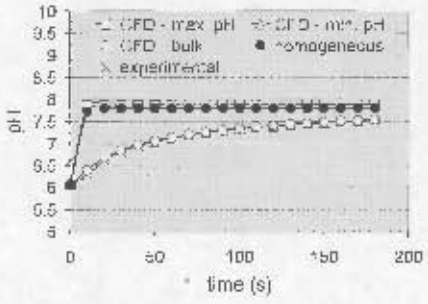

Figure 6.20 Comparison of experimentally measured and various simulated pH profiles for the 'base' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

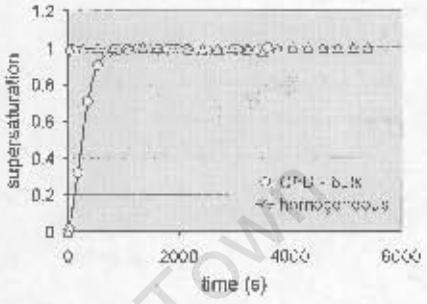

Figure 6.21 Comparison of 'base' supersaturation profiles predicted by CFD and homogeneous simulations employing kinetic 'equilibrium mimic' nucleation.

Bearing in mind that a value of unity is indicative of interphase equilibrium, the supersaturation profile predicted by the CFD simulation (Fig. 6.21) implies that solute deposition most likely occurs exclusively in a relatively small portion of the reactor volume for a significant period of time, and that only limited nucleation occurs in the bulk. This echoes the findings of Seckler et al. (1995) (Sec. 2.4).

Results of specific investigations into the various macromixing time scales also provide a measure of the spatial non-uniformity. Tabulated below, they reflect in each case the time required (as predicted by CFD simulation, and in the absence of precipitation) before the calculated pH extrema throughout the flow domain differ by no more than 0.05 units after a 1 s period of feed addition. Times are inclusive of the 1 s feed period. Maximum and minimum pH data associated with each of the scenarios listed in Table 6.1 are presented graphically in App. B (Figs. B.16 – B.20). Without delving into undue detail (given the limitations of the study), it can be stated simply that the results appear logical and that they suggest a maximum characteristic macromixing time of approximately 15 s for the 'base' level of agitation.

Table 6.1 Calculated characteristic macromixing times for each of the operating conditions investigated.

	Macromixing time scale (s)
base	14.8
rate L	13.925
rate H	14.75
stir L	9.475
stir H	10

Finally, although not supplied here, it may be noted that colour visualisations of the distribution of practically any one of the modelled solution or particulate properties (generated within the CFX® graphical post-processor from dump file data) further and convincingly illustrate the heterogeneity of the system under study.

It is apparent from the preceding analysis that, for the purposes of effective modelling attempts, the notion of a perfectly mixed reactor volume has, in line with rationally founded expectations, been refuted. This is in no small part due to the rapid kinetics which are characteristic of hydroxide precipitation processes. Detailed treatment of the effects of mixing and hydrodynamics, such as that achieved through CFD modelling, is thus of major importance in providing a rigorous, spatially distributed representation of key variable values. However, so as to exploit constructively this knowledge of spatial dynamics, a consistent and comparable level of experimental resolution is required, thereby facilitating improved kinetic description and greater fundamental characterisation of the system as a whole.

6.3.3 Significance of speciation calculations

The significant difference between the total dissolved nickel concentration and Ni^{2+} activity profiles displayed in Fig. 6.22 highlights the importance of the speciation calculations. Having said that, however, it is worth stating that such calculations are only of benefit if, as would be expected from theoretical considerations, kinetic expressions harnessing the resultant output actually capture the processes' true dependencies. As an example, use of a species activity as opposed to a component concentration (employing the vocabulary of Chap. 4) in determining a measure of supersaturation could only be considered superior if the relevant rate is correctly a function of the more accurately established value.

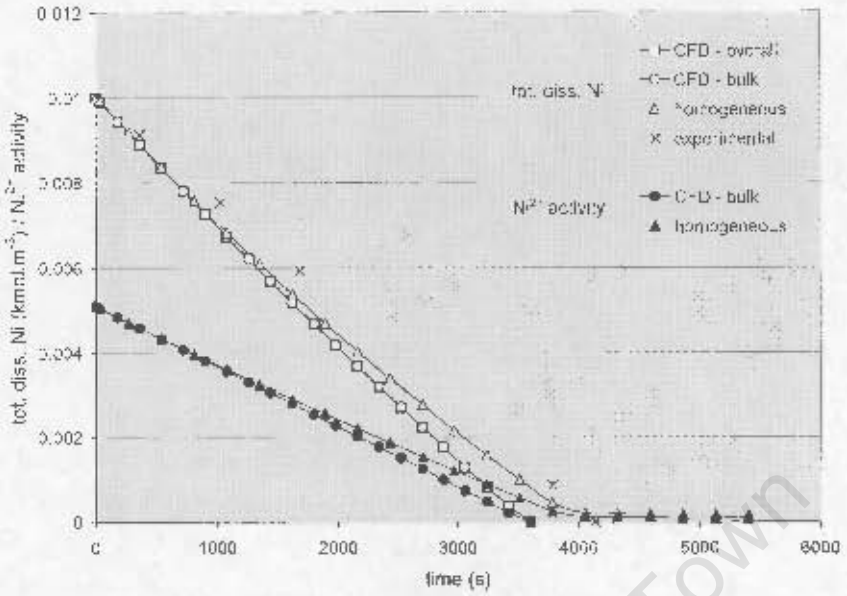

Figure 6.22 Comparison of experimentally measured and various simulated nickel data profiles for the 'base' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

Also evident in the above plot is the fairly close agreement between CFD and homogeneous simulation results, though this is not surprising considering the equivalent similarity of the corresponding yield predictions illustrated in Fig. 6.6. A final interesting feature of Fig. 6.22 is the observation that 'CFD overall' data points are, once again, mirrored by those of the 'CFD – bulk' series, this time with respect to total nickel in solution. The consistency of this relationship for realistic stirrer speeds allows one to conclude that bulk data, though certainly not reflective of the variables' uniform distribution, does give a reliable indication of the overall values in the reaction vessel. As it is for pH and supersaturation, the concept of an overall Ni^{2+} activity in a spatially distributed system is undefined.

6.3.4 CPU time

In order to identify the most computationally intensive calculations within the integrated model, and also give an idea of the overall load, it is valuable to analyse the CPU times expended by different simulation codes. Although the rapid current (and future predicted) advances in computer technology mean that processing times are seldom decisively limiting, the solution of high-definition CFD simulations can be incredibly time-consuming activities and knowledge of the relative burdens of the various mathematical routines can be instructive in directing the focus of future work.

The above-mentioned technological advances make a qualitative discussion of required computational times more meaningful than a record of actual absolute times. However, on a current-generation, mid-range desktop machine, an approximate characteristic figure for a homogeneous simulation is 0.0003 s per time-step, whilst for the comparatively crude CFD codes developed in this work it is 10.5 s per time-step. Omission of different model features in isolation reveals that roughly 80% of the

computational effort in CFD simulation runs is expended on the calculation each time-step of the updated distributions of species and particle number concentrations. Processing of the user-defined Fortran routines, which may be broadly divided into calculations of solution thermodynamics and those aimed at resolving the discretised population balance, then constitutes 20% of the total burden. Of this, 70% goes toward the solution speciation. Homogeneous simulation times also suggest this 70/30 split; indeed the various apportionments ascertained are remarkably consistent. The homogeneous modelling approach may be viewed conceptually as a single-celled CFD simulation, with thus no cause to solve the partial differential equations describing the dynamic distributions of species and particles. Despite this, it is still reassuring that multiplication of the characteristic 'CPU time per time-step' for the homogeneous code by the number of cells in the CFD volume mesh should then yield a value approximately 20% of the measured 'time per time-step' for the CFD simulations.

The stated CPU times are for simulations employing kinetic representation of nucleation. Incorporation of the solid species into the equilibrium calculations can increase the time required to speciate the system by up to 3500%. Increasing the desired accuracy of the speciation calculations does not, however, appear to come at significant computational expense. Selection of the convergence criteria for the solution of the underpinning discretised partial differential equations obviously has the potential to significantly affect the normalised processing time, whilst selection of the time-step size clearly impacts the total actual time. Specification then of these two parameter values is somewhat of a compromise between acceptable computational times and satisfactory accuracy.

6.4 Concluding remarks

This chapter has demonstrated clearly that the developed models fail to capture the experimentally observed behaviour. Disregarding even the geometrical description error outlined in Chap. 5, this is not unexpected, due ultimately to the resolution of the experimental data used in characterising the system. Consequently, whilst this deficiency does not diminish the inherent structural value of the developed modelling approach, the results provide only limited insight into the system of interest, and the chapter has thus served primarily to showcase the application and internal consistency of the models. Were it not for these facts, more comprehensive scrutiny and discussion of the data, both of additional simulation results and those presented here, would be justified in the interests of greater explication. This might extend to the analysis of the graphic colour representations alluded to in Sec. 6.3.2.

Such as it is, displayed here have been only those results required to demonstrate and corroborate the principal findings, which are summarised in Chap. 7. The useful knowledge discerned in the development and assessment of the models, despite their inherent limitations, also informs the recommendations for future work.

Chapter 7

Conclusions

7.1 Summary of major findings

The various principal findings, arrived at throughout the course of this dissertation, are reviewed here. Summarised, they include the following general conclusions:

CFD is a powerful modelling tool of particular value in the study of systems displaying elevated reaction rates. Precipitation processes frequently exhibit very fast reaction kinetics, making mixing of great importance in determining product quality.

The application of CFD to precipitation research remains relatively limited. Of special relevance are the facts that systems in which aggregation phenomena are recognised are seldom simulated and that the particulate phase is routinely described using a low-resolution moments transformation of the population balance. Perhaps most importantly, however, no attempts to incorporate detailed modelling of the solution chemistry (with its potential impact on rate-governing properties) are reported elsewhere.

The model architecture and resultant CFD modelling capability evolved in this work successfully fill the above-mentioned gap in the field. Solutions of a full discretised population balance (with the capacity to account for aggregation) and an inclusive set of intraphase equilibrium calculations are integrated into the CFD code.

The computational models developed are internally consistent and robust and the generic model framework easily accommodates upgrades, features which are discussed in greater detail in the remainder of this chapter.

Significant findings specific to the investigation of the nickel hydroxide test system include:

The low resolution of the available experimental data (both its extent and precision) is such that it is insufficient to characterise the precipitation system in great detail. This failure relates in particular to the process kinetics – the ideal of extracting rate equations capturing the true dependencies, and which can then be broadly applied with confidence, proved unattainable. The poor model calibration results obtained restrict the predictive capability of the model, thus limiting insight into the system. Nevertheless, the development and application of the models and attempts to reconcile simulation results and experimental observations do enable meaningful conclusions to be drawn, and, at the same time, reinforce the value of the model architecture and its components (which represent the inclusion of key physico-chemical phenomena.

Semi-batch operation of the nickel hydroxide precipitation system, which exhibits rapid reaction kinetics, leads to the development of significant spatial variation in the reactor contents. As a result, solute deposition, for example, is restricted to a small sub-volume around the feed location, an observation in line with those of similar, previous studies (e.g. Seckler et al. 1995). Accordingly, for practical modelling efforts it is not reasonable to assume spatial uniformity.

CFD is able to provide time-dependent spatial distributions of supersaturation and other key governing properties, and is thus potentially of great value in the modelling and analysis of such systems. However, for the system under consideration, in order to successfully exploit the detailed knowledge of spatial dynamics afforded by CFD modelling, an improvement is required in, above all, the experimental characterisation and resultant kinetic description. This applies especially to the representation of aggregation, which plays a dominant role in defining the quality of precipitate product here. More on this issue is offered in Sec. 7.2.

The rate of solute deposition is certainly very close to that prescribed by equilibrium considerations. An equilibrium description of nucleation therefore appears appropriate. It should be noted though that this does not necessarily suggest that a kinetic nucleation expression is inappropriate, but merely that the rate extraction exercise (and the experimental data used to support it) fail to yield an apposite rate function.

The concept of a size-dependent aggregate density may well be a suitable reflection of reality. If implemented, such an approach effectively overcomes the lack of predictive capability stemming from the current reliance upon an experimentally-predetermined precipitate density which is constant with respect to particle size but is an unestablished function of operating point. Further discussion is presented in Sec. 7.2.

Consideration of the full range of aqueous species and the associated intraphase reactions has a significant effect on the calculated solution properties upon which the various precipitation sub-processes are dependent. This fact underlines the importance of rigorous thermodynamic modelling.

The CFD simulation results indicate very close agreement between 'bulk' data (related to a selected single cell in the bulk zone, which corresponds to the experimental sampling point) and 'overall' data (appropriately averaged for the entire volume). This suggests that, although the environment has been shown to be clearly heterogeneous, its nature is such that bulk zone laboratory sampling possibly provides an accurate measure of the overall system, an observation which may be of some interest with respect to experimental design, monitoring and validation efforts.

The 'equilibrium mimic' nucleation rate equation formulation is reasonably successful in replicating the behaviour predicted by a detailed equilibrium representation of solids formation. Considering the substantial additional computational burden experienced through the inclusion of the solid phase in the speciation calculations, such a development is of great potential value in circumstances where available processing time or power is restrictive.

7.2 Recommendations for future work

As alluded to above, attaining a superior fundamental representation of the kinetics of nucleation and aggregation is a priority if the developed models are to be applied with confidence to the system under investigation here. Indeed, without it, the employment of CFD itself amounts to over-resolution, i.e. the potential benefits of the approach are not realised if an accompanying improvement in the extracted rate functions (thereby providing the degree of equivalence required in the level of refinement of the various model features) is not effected. The other deficiency considered to be in most urgent need of address may be overcome through the modelling of a more realistic nucleus size coupled with a consistent, size-dependent aggregate density.

The substantial demands thus placed on system characterisation necessitate extensive laboratory work. Preferably, such an exercise would be conducted in a methodical and systematic manner, investigating the different processes and various influencing factors, insofar as is possible, in isolation. As an example, Manth et al. (1996) have specifically designed a piece of apparatus configured such that precipitation may be studied under homogeneous mixing conditions. Manth and co-workers claim that the highly-turbulent reactant premixing permits the derivation of true, 'thermodynamically correct' kinetic equations. Spatially distributed sampling of the current experimental set-up would also be of value, both in the provision of supporting data for the undertaking just outlined and ultimately in the validation of the CFD model results. As a last resort, such spatially distributed experimental data could also be used in conjunction with a sophisticated CFD model of the system to extract unknown model parameters, in what would be a tedious and highly intensive, iterative procedure. Whatever the methods employed, however, it is most important that they not be predicated on an erroneous assumption such as the one of homogeneity underpinning the present rate extraction technique. Additionally, improved system characterisation would be further aided by higher resolution particle

number data (i.e. the accurate measurement of the numbers of smaller particles), a realisable scenario if the latest particle sizing technologies are utilised. Such information is useful, for example, both in better discerning aggregation phenomena and in helping to establish the mathematical specifics of the desired relationship between aggregate size and density (the latter of which may, to some extent, be guided by theoretical fractal considerations and the assumption that the density of nuclei is that of pure nickel hydroxide).

The computational implementation of the size-dependent aggregate density is unfortunately not as straightforward as that of the upgraded kinetic descriptions. It was mentioned in Sec. 3.4.2 that attempts to model fractal aggregate formation are often frustrated by the difficulties inherent in accounting for non-volume-conserving aggregation in the population balance. For use here with the existing discretised population balance, an extension and re-discretisation of the size domain is suggested to surmount these problems. Accordingly, the re-discretisation is informed by the translation of a plot of 'precipitate molar density' vs. 'particle size' to one of 'quantity of nickel per particle' vs. 'particle size' and executed such that the nickel content of particles of the diameter of the upper and lower bounds of each size interval are in a ratio of 2:1. Such a strategy is considered to be consistent with the theory employed by Hounslow et al. (1988) in deriving their discretised aggregation equations, which are based upon a size domain discretised such that the volume of particles of the diameter of the upper and lower bounds of each size interval are in a ratio of 2:1. In both instances, mass remains the conserved quantity.

Only once the above-mentioned inaccuracies in the process description have been rectified would it be appropriate to begin refining other aspects of the model. Foremost among these is probably the employment of a more detailed representation of the flow – this might entail correction of the previously highlighted geometric error and the use of the sliding grid solution technique in conjunction with a finely meshed geometry in which all features are modelled explicitly. LDV measurements could also be exploited to validate the flow field generated and assist in assessing the performance of alternative turbulence models, such as the differential stress model. Viewed in the context of the ultimate objectives of the research, additional variables, such as surface potential, which are wanted for downstream (i.e. dewatering) model input, require quantification. Further avenues for increasing the complexity of the developed models include the incorporation of multiphase modelling, consideration of other particulate processes such as breakage (given, for example, the detailed shear distribution throughout the vessel) and ageing effects, more rigorous treatment of micromixing and investigation into the applicability of size-dependent aggregation kernels – each supported by appropriate experimental data. There also exists the option of interfacing with proprietary speciation software.

In pursuing this incremental model sophistication it is recommended that CFD software be maintained as the primary modelling platform; this since most commercial codes possess in-built micromixing and multiphase modelling capabilities and thus represent convenient media for all the most computationally demanding increases in model resolution. The model framework presented in this study is such that most of the above upgrades may be integrated with minimal difficulty and continuing advances in

computing hardware dictate that these refinements are increasingly practicable. Once the resultant models display a capacity to successfully simulate the experimental system, in-depth analysis of the full available PSD and the dump file data visualisations (currently not deemed worthwhile) would be warranted. Such analysis would result in a more detailed explication of results and yield valuable additional insight into the system behaviour.

7.3 Contribution and significance

Within the context of the still comparatively narrow field of the application of CFD to precipitation, the following novelties or rarities may be claimed for this work:

use of a discretised population balance in preference to a moment transformation, thereby facilitating both greater resolution of particulate information and easy exploration of different aggregation kernels;
modelling of aggregation itself;
modelling of detailed speciation calculations; and,
application to a 'real' system of interest, as opposed to an 'ideal' and convenient model system (a fact from which the two preceding points largely stem).

In line with the stated aims (Chap. 1), a tool has thus been developed which is of particular value for multi-component precipitation systems requiring rigorous aqueous thermodynamic modelling. Since the model development demands the non-trivial integration of various model features and since the resultant model structure has strong generic qualities, the contribution is primarily one of model architecture. Regarding the demonstration of the tool, it could be argued that this would have been more elegantly achieved by application to a well-characterised, model system for which the present state of the model, largely unaltered, may be appropriate. However, such a scenario would be unrealistic, and it is argued that the attempted application to a 'difficult' system is, in itself, of value – correspondingly, it was previously articulated that the intention here is not to produce definitive model resolution.

Other contributions of significance in this dissertation include the comprehensive survey of the pertinent theory and literature, the development of code describing the various 'dilution options' (necessitated by the current inability within CFD to model the distinctive transient topology of semi-batch processes), the modifications made to the aggregation equations of the DPB and the conception of the 'equilibrium mimic' nucleation rate expression. Finally, the awareness and experience gained throughout the course of the research allow valuable suggestions for the direction of future work to be formulated. These are geared towards the ultimate objective of this line of research, viz. the development of a high-fidelity model which may find application, as an optimisation tool, to commercial reactor set-ups. It is envisaged that such a tool could be exploited in enhancing both the design of new precipitation-dewatering installations and the control and operability of existing units.

With respect to this ambitious goal, a small but nevertheless important step in the right direction is considered to have been taken in this study.

References

Allison, J.D., Brown, D.S. & Novo-Gradac, K.J. 1991, *MINTEQA2/PRODEFA2, A Geochemical Assessment Model for Environmental Systems: Version 3.0 User's Manual*, EPA/600/3-91/021, U.S. Environmental Protection Agency, Athens, Georgia.

Al-Rashed, M.H. & Jones, A.G. 1999, 'CFD modelling of gas-liquid reactive precipitation', *Chemical Engineering Science*, vol. 54, pp. 4779 – 4784.

Ang, H.-M. & Mullin, J.W. 1979, 'Crystal growth rate determinations from desupersaturation measurements: nickel ammonium sulphate hexahydrate', *Transactions I.Chem.E.*, vol. 57, pp. 237 – 243.

Baes, C.F.J. & Mesmer, R.E. 1976, *The Hydrolysis of Cations*, John Wiley & Sons, New York.

Baldyga, J. & Bourne, J.R. 1984, 'A fluid mechanical approach to turbulent mixing and chemical reaction', *Chemical Engineering Communications*, vol. 28, pp. 231 – 281.

Baldyga, J. & Bourne, J.R. 1989a, 'Simplification of micromixing calculations I. Derivation and application of new model', *Chemical Engineering Journal*, vol. 42, pp. 83 – 92.

Baldyga, J. & Bourne, J.R. 1989b, 'Simplification of micromixing calculations II. New applications', *Chemical Engineering Journal*, vol. 42, pp. 93 – 101.

Baldyga, J. & Bourne, J.R. 1992, 'Interaction between mixing on various scales in stirred tank reactors', *Chemical Engineering Science*, vol. 47, pp. 1839 – 1848.

Baldyga, J., Bourne, J.R. & Hearn, S.J. 1997, 'Interaction between chemical reactions and mixing on various scales', *Chemical Engineering Science*, vol. 52, pp. 457 – 466.

Baldyga, J. & Orciuch, W. 1997, 'Closure problem for precipitation', *Transactions I.Chem.E.*, vol. 75A, pp. 160 – 170.

Baldyga, J. & Orciuch, W. 2001, 'Some hydrodynamic aspects of precipitation', *Powder Technology*, vol. 121, pp. 9 – 19.

Baldyga, J., Podgorska, W. & Pohorecki, R. 1995, 'Mixing-precipitation model with application to double feed semibatch precipitation', *Chemical Engineering* Science, vol. 50, pp. 1281 – 1300.

Bramley, A.S., Hounslow, M.J. & Ryall, R.L. 1996, 'Aggregation during precipitation from solution: a method for extracting rates from experimental data', *Journal of Colloid and Interface Science*, vol. 183, pp. 155 –165.

Brucato, A., Ciofalo, M., Grisafi, F. & Micale, G. 1998, 'Numerical prediction of flow fields in baffled stirred vessels: A comparison of alternative modelling approaches', *Chemical Engineering Science*, vol. 53, pp. 3653 – 3684.

Cairncross, L.R.C. 1998, *Simulation of the precipitation of sparingly soluble compounds*, MSc dissertation, University of Cape Town.

Debye, P. & Hückel, E. 1923, 'Zur theorie der elektrolyte. I. Gefrierpunktserniedrigung und verwandte erscheinungen', *Physikalische Zeitschrift*, vol. 24, pp. 185 – 206.

Demopoulos, G.P. 1993, 'Precipitation in aqueous processing of inorganic materials', in *Proceedings of the First International Conference on Processing Materials for Properties*, eds. H. Henein & T. Oki, Honolulu, pp. 537 – 540.

Demopoulos, G.P., Zinck, J. & Kondos, P.D. 1995, 'Production of super dense sludges with a novel neutralization process', in *International Symposium on Waste Processing and Recycling in Mineral and Metallurgical Industries II: Proceedings of the 34th Annual Conference of Metallurgists of CIM*, eds. S.R. Rao, L.M. Amaratunga, G.G. Richards & P.D. Kondos, Vancouver, pp. 401 – 411.

Dirksen, J.A. & Ring, T.A. 1991, 'Fundamentals of crystallization: kinetic effects on particle size distributions and morphology', *Chemical Engineering Science*, vol. 46, pp. 2389 – 2427.

Dustan, A.C. 2001, *The effect of precipitation conditions on product character and dewatering: an integrated study for base-metal hydroxides*, PhD thesis, University of Cape Town.

Dustan, A.C., Cohen, B. & Petrie, J.G. 2005a, 'Modelling dewatering behaviour through an understanding of solids formation processes. Part I – solids formation considerations', *Advances in Colloid and Interface Science*, vol. 13, pp. 99 – 110.

Dustan, A.C., Cohen, B. & Petrie, J.G. 2005b, 'Modelling dewatering behaviour through an understanding of solids formation processes. Part II – solids separation considerations', *Advances in Colloid and Interface Science*, vol. 13, pp. 85 – 97.

Falk, L. & Schaer, E. 2001, 'A PDF modelling of precipitation reactors', *Chemical Engineering Science*, vol. 56, pp. 2445 – 2457.

Fletcher, C.A.J., Wood, M.G. & Matthews, B.W. 1998, 'CFD: A practical analysis tool for chemical engineering and multiphase flows', in *Creating Competitive Resources: Proceedings of the 26th Australasian Chemical Engineering Conference (Chemeca98)*, Port Douglas.

Fletcher, D.F. 1997, *Lecture notes on computational fluid dynamics for advanced fluid dynamics modelling U4.655*, Department of Chemical Engineering, University of Sydney.

Fox, R.O. 1998, 'On the relationship between Lagrangian micromixing models and computational fluid dynamics', *Chemical Engineering and Processing*, vol. 37, pp. 521 – 535.

Franke, J. & Mersmann, A. 1995, 'The influence of the operational conditions on the precipitation process', *Chemical Engineering Science*, vol. 50, pp. 1737 – 1753.

Gardner, K.H. & Theis, T.L. 1996, 'A unified kinetic model for particle aggregation', *Journal of Colloid and Interface Science*, vol. 180, pp. 162 – 173.

Ginter, D.M. & Loyalka, S.K. 1996, 'Apparent size-dependent growth in aggregating crystallizers', *Chemical Engineering Science*, vol. 51, pp. 3685 – 3695.

Harada, M., Arima, K., Eguchi, W. & Nagata, S. 1962, 'Micromixing in a continuous flow reactor', *Memoirs of the Faculty of Engineering, Kyoto University Japan*, vol. 24, p. 431.

Hill, P.J. & Ng, K.M. 1997, 'Simulation of solids processes accounting for particle-size distribution', *A.I.Ch.E. Journal*, vol. 43, pp. 715 – 726.

Hjertager, L.K., Hjertager, B.H. & Solberg, T. 2002, 'CFD modelling of fast chemical reactions in turbulent liquid flows', *Computers and Chemical Engineering*, vol. 26, pp. 507 – 515.

Hogg, R. 2000, 'Flocculation and dewatering', *International Journal of Mineral Processing*, vol. 58, pp. 223 – 236.

Hounslow, M.J. 1990, *A discretised population balance for simultaneous nucleation, growth and aggregation*, PhD thesis, University of Adelaide.

Hounslow, M.J., Ryall, R.L. & Marshall, V.R. 1988, 'A discretised population balance for nucleation, growth, and aggregation', *A.I.Ch.E. Journal*, vol. 34, pp. 1821 – 1832.

Hounslow, M.J., Lewis, A.E., Sanders, S.J. & Bondy, R. 2005, 'Generic crystallizer model: I. A model framework for a well-mixed compartment', *A.I.Ch.E. Journal*, vol. 51, pp. 2942 – 2955.

Hulbert, H.M. & Katz, S. 1964, 'Some problems in particle technology: a statistical mechanical formulation', *Chemical Engineering Science*, vol. 19, pp. 555 – 574.

Ilievski, D. & White, E.T. 1994, 'Agglomeration during precipitation: agglomeration mechanism identification for $Al(OH)_3$ crystals in stirred caustic aluminate solutions', *Chemical Engineering Science*, vol. 49, pp. 3227 – 3239.

Jaworski, Z. & Nienow, A.W. 2003, 'CFD modelling of continuous precipitation of barium sulphate in a stirred tank', *Chemical Engineering Journal*, vol. 91, pp. 167 – 174.

Jenne, M. & Reuss, M. 1999, 'A critical assessment on the use of k-ε turbulence models for simulation of the turbulent liquid flow induced by a Rushton-turbine in baffled stirred-tank reactors', *Chemical Engineering Science*, vol. 54, pp. 3921 – 3941.

Koh, P.T.L. & Wu, J. 1998, 'CFD simulation of flow in mechanically stirred tanks with fluidfoil impellers', in *Creating Competitive Resources: Proceedings of the 26th Australasian Chemical Engineering Conference (Chemeca98)*, Port Douglas.

Lane, G.L. & Koh, P.T.L. 1997, 'CFD simulation of a Rushton turbine in a baffled tank', in *Proceedings of the International Conference on CFD in Mineral and Metal Processing and Power Generation*, eds. M.P. Schwarz, M.R. Davidson & A.K. Easton, CSIRO, Clayton, pp. 377 – 385.

Linde, D.R. (ed.) 1996, *CRC Handbook of Chemistry and Physics*, 77th edn, CRC Press, Boca Raton.

Madec, L., Falk, L. & Plasari, E. 2001, 'Simulation of agglomeration reactors via a coupled CFD/direct Monte-Carlo method', *Chemical Engineering Science*, vol. 56, pp. 1731 – 1736.

Magnussen, B.F. & Hjertager, B.H. 1976, 'On mathematical modelling of turbulent combustion with special emphasis on soot formation and combustion', in *Proceedings of the 16th International Symposium on Combustion*, The Combustion Institute, p. 719.

Manth, T., Mignon, D. & Offermann, H. 1996, 'Experimental investigation of precipitation reactions under homogeneous mixing conditions', *Chemical Engineering Science*, vol. 51, pp. 2571 – 2576.

Marchisio, D.L. & Barresi, A.A. 2003, 'CFD simulation of mixing and reaction: the relevance of the micro-mixing model', *Chemical Engineering Science*, vol. 58, pp. 3579 – 3587.

McCabe, W.L. 1929, 'Crystal growth in aqueous systems', *Industrial and Engineering Chemistry*, vol. 21, pp. 30 – 33 and 112 – 119.

Mehta, R.V. & Tarbell, J.M. 1983, 'A four environment model of mixing and chemical reaction. Part I – model development', *A.I.Ch.E. Journal*, vol. 29, p. 320.

Mersmann, A. (ed.) 1995, *Crystallisation Technology Handbook*, Marcel Dekker, New York.

Mersmann, A. 1999, 'Crystallization and precipitation', *Chemical Engineering and Processing*, vol. 38, pp. 345 – 353.

Middleton, J.C., Pierce, F. & Lynch, P.M. 1986, 'Computations of flow fields and complex reaction yield in turbulent stirred reactors and comparison with experimental data', *Chemical Engineering Research & Design*, vol. 64, p. 18.

Mullin, J.W. 2001, *Crystallisation*, 4th edn, Butterworth-Heinemann, Oxford.

Patterson, J.W. 1987, 'Metals separation and recovery', in *Metals Speciation, Separation, and Recovery, Volume I*, eds. J.W. Patterson & R. Passino, Lewis, Chelsea, Michigan, pp. 27 – 42.

Patterson, J.W., Luo, B., Marani, D. & Passino, R. 1990, 'Nucleation and crystal growth studies on precipitation of cadmium hydroxide from aqueous solutions', in *Metals Speciation, Separation, and Recovery, Volume II*, eds. J.W. Patterson & R. Passino, Lewis, Chelsea, Michigan, pp.91 – 113.

Pavlides, A.G. 1995, *The precipitation of iron oxy-hydroxides from ferrous sulphate solution*, MSc dissertation, University of the Witwatersrand.

Petrie, J., Krige, A., Dustan, A. & Fletcher, D. 2002, 'A CFD analysis of reactive crystallisation in stirred tanks', in *Proceedings of the 4th World Congress on Particle Technology (WCPT4)*, Sydney.

Rafal, M., Berthold, J.W., Scrivner, N.C. & Grise, S.L. 1994, 'Models for electrolyte solutions', in *Models for Thermodynamic and Phase Equilibria Calculations*, ed. S.I. Sandler, Marcel Dekker, New York, pp. 601 – 670.

Ranade, V.V. 1997, 'An efficient computational model for simulating flow in stirred vessels: a case of Rushton turbine', *Chemical Engineering Science*, vol. 52, pp. 4473 – 4484.

Randolph, A.D. & Larson, M.A. 1971, *Theory of Particulate Processes: Analysis and Techniques of Continuous Crystallization*, 2nd edn, Academic Press, New York.

Rawle, A. (2001), *Basic Principles of Particle Size Analysis*, [Online], Malvern Instruments Limited, Available from: <http://www.malvern.co.uk/malvern/kbase.nsf/allbyno/KB000021?opendocument> [25 February 2004].

Rigby, G.D., Rielly, C.D., Lane, G. & Evans, G.M. 1998, 'Numerical modelling of trailing vortices in the wake of Rushton impeller blades', in *Creating Competitive Resources: Proceedings of the 26th Australasian Chemical Engineering Conference (Chemeca98)*, Port Douglas.

Ritchie, B.W. & Togby, A.H. 1979, 'A three-environment micromixing model for chemical reactors with arbitrary separate feedstreams', *Chemical Engineering Journal*, vol. 17, pp. 173.

Rousseaux, J.-M., Vial, C., Muhr, H. & Plasari, E. 2001, 'CFD simulation of precipitation in the sliding-surface mixing device', *Chemical Engineering Science*, vol. 56, pp. 1677 – 1685.

Sandler, S.I. 1989, *Chemical and Engineering Thermodynamics*, John Wiley & Sons, New York.

Sastry, K.V.S. 1975, 'Similarity size distribution of agglomerates during their growth by coalescence in granulation or green pelletization', *International Journal of Mineral Processing*, vol. 2, pp. 187 – 203.

Seckler, M.M., Bruinsma, O.S.L. & Van Rosmalen, G.M. 1995, 'Influence of hydrodynamics on precipitation: a computational study', *Chemical Engineering Communications*, vol. 135, pp. 113 – 131.

Seyssiecq, I., Veesler, S., Mangin, D., Klein, J.P. & Boistelle, R. 2000, 'Modelling gibbsite agglomeration in a constant supersaturation crystallizer', *Chemical Engineering Science*, vol. 55, pp. 5565 – 5578.

Smit, D.J., Hounslow, M.J. & Paterson, W.R. 1994, 'Aggregation and gelation – I. Analytical solutions for CST and batch operation', *Chemical Engineering Science*, vol. 49, pp. 1025 – 1035.

Smit, D.J., Hounslow, M.J. & Paterson, W.R. 1995, 'Aggregation and gelation: III. Numerical classification of kernels and case studies of aggregation and growth', *Chemical Engineering Science*, vol. 50, pp. 849 – 862.

Söhnel, O. & Garside, J. 1992, *Precipitation – Basic Principles and Industrial Applications*, Butterworth-Heinemann, Oxford.

Spanhel, L. & Anderson, M.A. 1991, 'Semiconductor clusters in the sol-gel process: quantized aggregation, gelation, and crystal growth in concentrated ZnO colloids', *Journal of the American Chemical Society*, vol. 113, pp. 2826 – 2833.

Stumm, W. & Morgan, J.J. 1981, *Aquatic Chemistry*, John Wiley & Sons, New York.

Suoninen, E., Juntunen, T., Juslen, H. & Pessa, M. 1973, 'Structure and ageing of $Ni(OH)_2$ precipitated from sulfate and chloride solutions', *Acta Chemica Scandinavica*, vol. 27, pp. 2013 – 2019.

Tavare, N.S. 1987, 'Batch crystallizers: a review', *Chemical Engineering Communications*, vol. 61, pp. 259 – 318.

Urban, Z. & Liberis, L. 1999, 'Hybrid gPROMS-CFD modelling of an industrial scale crystalliser with rigorous crystal nucleation and growth kinetics and a full population balance', in *Process Simulation: On the Cutting Edge: Proceedings of Chemputers 1999*, Dusseldorf.

van Leeuwen, M.L.J. 1998, *Precipitation and mixing*, PhD thesis, Delft University of Technology.

van Leeuwen, M.L.J., Bruinsma, O.S.L. & van Rosmalen, G.M. 1996, 'Influence of mixing on the product quality in precipitation', *Chemical Engineering Science*, vol. 51, pp. 2595 – 2600.

Vicum, L., Ottiger, S., Mazzotti, M., Makowski, L. & Baldyga, J. 2004, 'Multi-scale modelling of a reactive mixing process in a semi-batch stirred tank', *Chemical Engineering Science*, vol. 59, pp. 1767 – 1783.

Villermaux, J. & Falk, L. 1994, 'A generalised mixing model for initial contacting of reactive fluids', *Chemical Engineering Science*, vol. 49, pp. 5127 – 5140.

von Smoluchowski, M. 1916, 'Versuch einer mathematischen theorie der koagulations kinetik kolloider lösungen', *Zeitschrift für Physikalische Chemie*, vol. 92, pp. 129 – 168.

Wei, H. & Garside, J. 1997, 'Application of CFD modelling to precipitation systems', *Transactions I.Chem.E.*, vol. 75A, pp. 219 – 227.

Zauner, R. & Jones, A.G. 2002, 'On the influence of mixing on crystal precipitation processes – application of the segregated feed model', *Chemical Engineering Science*, vol. 57, pp. 821 – 831.

Zemaitis, J.F., Clark, D.M., Rafal, M. & Scrivner, N.C. 1986, *Handbook of Aqueous Electrolyte Thermodynamics*, A.I.Ch.E., New York.

Appendix A

Selected annotated CFX® input files

A.1 Simplified flow simulation

A.1.1 Command file

```
/*This command file is intended to be run in double precision and in conjunction
with the Fortran file approximate_flow.f and the geometry file approximate.geo.
The simulation employs a relatively coarse grid and a much simplified approach
in an attempt to generate a steady state flow field, which, ignoring the
characteristic unsteady interaction between baffles and impeller blades,
reproduces the final quasi steady state representation of the initial
experimental field predicted by the high fidelity, sliding mesh simulation
method.  Efforts to model the effects of the impeller are made through the
introduction of source terms in the velocity equations (better known as momentum
sources), one of a number of similar non-explicit techniques.  No extra source
terms are included in the modelled turbulence equations.  When investigating
alternative impeller speeds, the momentum source values are simply scaled
accordingly.  The widely applied k-epsilon turbulence model is used.  Only non-
default inputs are included in this command file.*/
>>CFX4
  >>OPTIONS
    RECTANGULAR GRID
    CYLINDRICAL COORDINATES
    AXIS INCLUDED
    TURBULENT FLOW
    USER SCALAR EQUATIONS 6
  >>USER FORTRAN
    USRTRN
  >>VARIABLE NAMES
    /*all names below are of special class recognised by software; variables
    treated as places to store information rather than for solution in scalar
    transport equations; included as user scalars to make variable values
    available to output facilities, e.g. allows writing to dump file for
    graphical display*/
    /*x-y component of shear stress throughout domain [N.m-2]*/
    USER SCALAR1 'XY NODAL SHEAR STRESS'
    /*y-z component of shear stress throughout domain [N.m-2]*/
    USER SCALAR2 'YZ NODAL SHEAR STRESS'
    /*z-x component of shear stress throughout domain [N.m-2]*/
```

```
    USER SCALAR3 'ZX NODAL SHEAR STRESS'
    /*turbulent mixing time scale (user-defined at cell centres) [s]*/
    USER SCALAR4 'USRDCC TMIX'
    /*radial velocity (user-defined at all nodes) [m.s-1]*/
    USER SCALAR5 'USRD VRAD'
    /*rotational velocity (user-defined at all nodes) [m.s-1]*/
    USER SCALAR6 'USRD VROT'
>>MODEL DATA
  /*simulation of impeller effect*/
  >>SOURCES
    /*volume swept by impeller blades*/
    PATCH NAME 'USER3D_IMPBLADE'
    PER UNIT VOLUME
    /*values 'tuned' such that results approximate flow field predicted by
    rigorous, high resolution, sliding mesh simulation (in this case, impeller
    speed of 150 r.p.m.)*/
    /*axial component*/
    U VELOCITY -4.500000E+03 /*[kg.m-2.s-2]*/ 0.000000E+00
    /*radial component*/
    V VELOCITY 6.000000E+03 /*[kg.m-2.s-2]*/ 0.000000E+00
    /*rotational component*/
    W VELOCITY -5.250000E+03 /*[kg.m-2.s-2]*/ 0.000000E+00
  >>PHYSICAL PROPERTIES
    >>STANDARD FLUID
      FLUID 'WATER'
      STANDARD FLUID REFERENCE TEMPERATURE 2.9815E+02 /*[K]*/
>>SOLVER DATA
  >>PROGRAM CONTROL
    MINIMUM NUMBER OF ITERATIONS 10
    MAXIMUM NUMBER OF ITERATIONS 5000
  >>UNDER RELAXATION FACTORS
    U VELOCITY 5.0000E-01
    V VELOCITY 5.0000E-01
    W VELOCITY 5.0000E-01
>>MODEL BOUNDARY CONDITIONS
  >>WALL BOUNDARIES
    /*free liquid surface*/
    PATCH NAME 'WALL_SURFACE'
    /*free slip condition - zero wall shear stress*/
    TAUX 0.0 /*[N.m-2]*/
    TAUY 0.0 /*[N.m-2]*/
    TAUZ 0.0 /*[N.m-2]*/
  >>WALL BOUNDARIES
    /*impeller disc*/
    PATCH NAME 'WALL_IMPDISC'
    /*no slip condition on rotating wall - angular velocity that of impeller*/
    ANGULAR VELOCITY -15.7079632679 0.0 0.0 /*[(rad).s-1]*/
>>STOP
```

A.1.2 Fortran file

```
C This Fortran file (subroutine USRTRN) calculates the steady state values of
C turbulent mixing time scale, radial velocity and rotational velocity within each
C interior cell.
C
      SUBROUTINE USRTRN(U,V,W,P,VFRAC,DEN,VIS,TE,ED,RS,T,H,RF,SCAL,XP,
     +                  YP,ZP,VOL,AREA,VPOR,ARPOR,WFACT,CONV,IPT,IBLK,
     +                  IPVERT,IPNODN,IPFACN,IPNODF,IPNODB,IPFACB,WORK,
     +                  IWORK,CWORK)
C
C***********************************************************************
C
C   USER SUBROUTINE TO ALLOW USERS TO MODIFY OR MONITOR THE SOLUTION AT
C   THE END OF EACH TIME STEP
C   THIS SUBROUTINE IS CALLED BEFORE THE START OF THE RUN AS WELL AS AT
C   THE END OF EACH TIME STEP
C
C   >>> IMPORTANT                                                    <<<
C   >>>                                                            <<<
C   >>> USERS MAY ONLY ADD OR ALTER PARTS OF THE SUBROUTINE WITHIN  <<<
C   >>> THE DESIGNATED USER AREAS                                   <<<
C
C***********************************************************************
```

```
C
C   THIS SUBROUTINE IS CALLED BY THE FOLLOWING SUBROUTINES
C      CUSR  TRNMOD
C
C***********************************************************************
C
C   CREATED
C      27/04/90  ADB
C   MODIFIED
C      05/08/91  IRH  NEW STRUCTURE
C      01/10/91  DSC  REDUCE COMMENT LINE GOING OVER COLUMN 72.
C      29/11/91  PHA  UPDATE CALLED BY COMMENT, ADD RF ARGUMENT,
C                     CHANGE LAST DIMENSION OF RS TO 6 AND IVERS TO 2
C      05/06/92  PHA  ADD PRECISION FLAG AND CHANGE IVERS TO 3
C      03/07/92  DSC  CORRECT COMMON MLTGRD.
C      23/11/93  CSH  EXPLICITLY DIMENSION IPVERT ETC.
C      03/02/94  PHA  CHANGE FLOW3D TO CFDS-FLOW3D
C      22/08/94  NSW  MOVE 'IF(IUSED.EQ.0) RETURN' OUT OF USER AREA
C      19/12/94  NSW  CHANGE FOR CFX-F3D
C      02/07/97  NSW  UPDATE FOR CFX-4
C      02/07/99  NSW  INCLUDE NEW EXAMPLE FOR CALCULATING FLUX OF A
C                     SCALAR AT A PRESSURE BOUNDARY
C
C***********************************************************************
C
C   SUBROUTINE ARGUMENTS
C
C     U      - U COMPONENT OF VELOCITY
C     V      - V COMPONENT OF VELOCITY
C     W      - W COMPONENT OF VELOCITY
C     P      - PRESSURE
C     VFRAC  - VOLUME FRACTION
C     DEN    - DENSITY OF FLUID
C     VIS    - VISCOSITY OF FLUID
C     TE     - TURBULENT KINETIC ENERGY
C     ED     - EPSILON
C     RS     - REYNOLD STRESSES
C     T      - TEMPERATURE
C     H      - ENTHALPY
C     RF     - REYNOLD FLUXES
C     SCAL   - SCALARS (THE FIRST 'NCONC' OF THESE ARE MASS FRACTIONS)
C     XP     - X COORDINATES OF CELL CENTRES
C     YP     - Y COORDINATES OF CELL CENTRES
C     ZP     - Z COORDINATES OF CELL CENTRES
C     VOL    - VOLUME OF CELLS
C     AREA   - AREA OF CELLS
C     VPOR   - POROUS VOLUME
C     ARPOR  - POROUS AREA
C     WFACT  - WEIGHT FACTORS
C     CONV   - CONVECTION COEFFICIENTS
C
C     IPT    - 1D POINTER ARRAY
C     IBLK   - BLOCK SIZE INFORMATION
C     IPVERT - POINTER FROM CELL CENTRES TO 8 NEIGHBOURING VERTICES
C     IPNODN - POINTER FROM CELL CENTRES TO 6 NEIGHBOURING CELLS
C     IPFACN - POINTER FROM CELL CENTRES TO 6 NEIGHBOURING FACES
C     IPNODF - POINTER FROM CELL FACES TO 2 NEIGHBOURING CELL CENTRES
C     IPNODB - POINTER FROM BOUNDARY CENTRES TO CELL CENTRES
C     IPFACB - POINTER FROM BOUNDARY CENTRES TO BOUNDARY FACESS
C
C     WORK   - REAL WORKSPACE ARRAY
C     IWORK  - INTEGER WORKSPACE ARRAY
C     CWORK  - CHARACTER WORKSPACE ARRAY
C
C   SUBROUTINE ARGUMENTS PRECEDED WITH A '*' ARE ARGUMENTS THAT MUST
C   BE SET BY THE USER IN THIS ROUTINE.
C
C   NOTE THAT OTHER DATA MAY BE OBTAINED FROM CFX-4 USING THE
C   ROUTINE GETADD, FOR FURTHER DETAILS SEE THE VERSION 4
C   USER MANUAL.
C
C***********************************************************************
C
      DOUBLE PRECISION U
      DOUBLE PRECISION V
      DOUBLE PRECISION W
      DOUBLE PRECISION P
```

```
      DOUBLE PRECISION VFRAC
      DOUBLE PRECISION DEN
      DOUBLE PRECISION VIS
      DOUBLE PRECISION TE
      DOUBLE PRECISION ED
      DOUBLE PRECISION RS
      DOUBLE PRECISION T
      DOUBLE PRECISION H
      DOUBLE PRECISION RF
      DOUBLE PRECISION SCAL
      DOUBLE PRECISION XP
      DOUBLE PRECISION YP
      DOUBLE PRECISION ZP
      DOUBLE PRECISION VOL
      DOUBLE PRECISION AREA
      DOUBLE PRECISION VPOR
      DOUBLE PRECISION ARPOR
      DOUBLE PRECISION WFACT
      DOUBLE PRECISION CONV
      DOUBLE PRECISION WORK
      DOUBLE PRECISION SMALL
      DOUBLE PRECISION SORMAX
      DOUBLE PRECISION DTUSR
      DOUBLE PRECISION TIME
      DOUBLE PRECISION DT
      DOUBLE PRECISION DTINVF
      DOUBLE PRECISION TPARM
      DOUBLE PRECISION SGNWL
      LOGICAL LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP,LRECT,LCYN,LAXIS,
     +        LPOROS,LTRANS
C
      CHARACTER*(*) CWORK
C
C+++++++++++++++++ USER AREA 1 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS EXPLICITLY DECLARED VARIABLES
C
      INTEGER I,INODE
C
C+++++++++++++++++ END OF USER AREA 1 +++++++++++++++++++++++++++++++++++
C
      COMMON /ALL/NBLOCK,NCELL,NBDRY,NNODE,NFACE,NVERT,NDIM,
     +       /ALLWRK/NRWS,NIWS,NCWS,IWRFRE,IWIFRE,IWCFRE,/ADDIMS/NPHASE,
     +       NSCAL,NVAR,NPROP,NDVAR,NDPROP,NDXNN,NDGEOM,NDCOEF,NILIST,
     +       NRLIST,NTOPOL,/CHKUSR/IVERS,IUCALL,IUSED,/CONC/NCONC,
     +       /DEVICE/NREAD,NWRITE,NRDISK,NWDISK,/IDUM/ILEN,JLEN,
     +       /LOGIC/LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP,LRECT,LCYN,
     +       LAXIS,LPOROS,LTRANS,/MLTGRD/MLEVEL,NLEVEL,ILEVEL,
     +       /SGLDBL/IFLGPR,ICHKPR,/SPARM/SMALL,SORMAX,NITER,INDPRI,
     +       MAXIT,NODREF,NODMON,/TIMUSR/DTUSR,/TRANSI/NSTEP,KSTEP,MF,
     +       INCORE,/TRANSR/TIME,DT,DTINVF,TPARM
C
C+++++++++++++++++ USER AREA 2 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DECLARE THEIR OWN COMMON BLOCKS
C     THESE SHOULD START WITH THE CHARACTERS 'UC' TO ENSURE
C     NO CONFLICT WITH NON-USER COMMON BLOCKS
C
C+++++++++++++++++ END OF USER AREA 2 +++++++++++++++++++++++++++++++++++
C
      DIMENSION U(NNODE,NPHASE),V(NNODE,NPHASE),W(NNODE,NPHASE),
     +          P(NNODE,NPHASE),VFRAC(NNODE,NPHASE),DEN(NNODE,NPHASE),
     +          VIS(NNODE,NPHASE),TE(NNODE,NPHASE),ED(NNODE,NPHASE),
     +          RS(NNODE,NPHASE,6),T(NNODE,NPHASE),H(NNODE,NPHASE),
     +          RF(NNODE,NPHASE,4),SCAL(NNODE,NPHASE,NSCAL)
      DIMENSION XP(NNODE),YP(NNODE),ZP(NNODE),VOL(NCELL),AREA(NFACE,3),
     +          VPOR(NCELL),ARPOR(NFACE,3),WFACT(NFACE),
     +          CONV(NFACE,NPHASE),IPT(*),IBLK(5,NBLOCK),
     +          IPVERT(NCELL,8),IPNODN(NCELL,6),IPFACN(NCELL,6),
     +          IPNODF(NFACE,4),IPNODB(NBDRY,4),IPFACB(NBDRY),IWORK(*),
     +          WORK(*),CWORK(*)
      DIMENSION SGNWL(6)
C
C+++++++++++++++++ USER AREA 3 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DIMENSION THEIR ARRAYS
C
C---- AREA FOR USERS TO DEFINE DATA STATEMENTS
C
C+++++++++++++++++ END OF USER AREA 3 +++++++++++++++++++++++++++++++++++
```

```
C
      DATA SGNWL/1.0D0,1.0D0,1.0D0,-1.0D0,-1.0D0,-1.0D0/
C
C---- STATEMENT FUNCTION FOR ADDRESSING
      IP(I,J,K) = IPT((K-1)*ILEN*JLEN+ (J-1)*ILEN+I)
C
C----VERSION NUMBER OF USER ROUTINE AND PRECISION FLAG
C
      IVERS = 3
      ICHKPR = 2
C
C+++++++++++++++++ USER AREA 4 ++++++++++++++++++++++++++++++++++++++++++
C---- TO USE THIS USER ROUTINE FIRST SET IUSED=1
C
      IUSED = 1
C
C+++++++++++++++++ END OF USER AREA 4 ++++++++++++++++++++++++++++++++++
C
      IF (IUSED.EQ.0) RETURN
C
C---- FRONTEND CHECKING OF USER ROUTINE
      IF (IUCALL.EQ.0) RETURN
C
C+++++++++++++++++ USER AREA 5 ++++++++++++++++++++++++++++++++++++++++++
C
C---- EXAMPLE (SET TIME INCREMENT FOR NEXT TIME STEP)
C
C     DTUSR = 0.1
C
C---- END OF EXAMPLE
C
C---- EXAMPLE (CALCULATE FLUX OF FIRST SCALAR AT A PRESSURE BOUNDARY)
C
C     IPHASE = 1
C     FLUX = 0.0
C  USE IPALL TO FIND ADDRESSES OF BOUNDARY NODES ON PATCH PRESS1
C     CALL IPALL('PRESS1','PRESS','PATCH','CENTRES'
C    +          ,IPT,NPT,CWORK,IWORK)
C  LOOP OVER ALL BOUNDARY NODES
C     DO 300 I=1,NPT
C  USE ARRAY IPT TO GET ADDRESS
C       INODE = IPT(I)
C       IBDRY = INODE - NCELL
C       IFACE = IPFACB(IBDRY)
C       INDUM = IPNODB(IBDRY,2)
C       NWL   = IPNODB(IBDRY,4)
C       FLUX  = FLUX
C    +        + SGNWL(NWL)*CONV(IFACE,IPHASE)*SCAL(INDUM,IPHASE,1)
C 300 CONTINUE
C
C---- END OF EXAMPLE
C
C     call utility routine GETSCA to find number of scalar variable within SCAL array
C     turbulent mixing time scale [s]
      CALL GETSCA('USRDCC TMIX',ITM,CWORK)
C     radial velocity [m.s-1]
      CALL GETSCA('USRD VRAD',IRAD,CWORK)
C     rotational velocity [m.s-1]
      CALL GETSCA('USRD VROT',IROT,CWORK)
C
C     call utility routine IPALL to return cell-centre addresses of those cells
C     occupied by tank contents (i.e. all interior cells)
      CALL IPALL('*','*','BLOCK','CENTRES',IPT,NPT,CWORK,IWORK)
C
C     within each cell in flow domain
      DO 10 I = 1,NPT
C
        INODE = IPT(I)
C
C       calculate turbulent mixing time scale (corresponds to viscous mixing variant
C       of eddy break-up combustion model) [s]
        SCAL(INODE,1,ITM) = 0.04237*((((DEN(INODE,1))*((TE(INODE,1))**
     &                      2.0))/((VIS(INODE,1))*(ED(INODE,1))))**0.25
     &                      )*((TE(INODE,1))/(ED(INODE,1)))
C
 10   CONTINUE
C
```

```
C     at each cell and boundary centre
      DO 20 INODE = 1,NNODE
C       calculate radial velocity [m.s-1]
        SCAL(INODE,1,IRAD) = V(INODE,1)
C       calculate rotational velocity [m.s-1]
        SCAL(INODE,1,IROT) = W(INODE,1)
 20   CONTINUE
C
C+++++++++++++++++ END OF USER AREA 5 +++++++++++++++++++++++++++++++++++
C
      RETURN
C
      END
```

A.2 Precipitation process simulation

A.2.1 Command file

```
/*This command file is intended to be run in double precision and in conjunction
with the Fortran file precipitation.f, the geometry file approximate.geo and a
restart file generated by the approximate steady state flow calculation or this
code itself.  Either restart option provides the fixed flow field to be used in
the various relevant calculations; the latter also contains the initial
distributions of species and particle number concentrations.  The transient
simulation essentially models the course of the experimental semi-batch
precipitation process (NiCl2/NaOH system), charting the changing solution
conditions and the ultimately resultant evolution of the particle size
distribution on a local scale.  The multiphase, population balance and
micromixing modelling capabilities of the software are not invoked, due
variously to issues largely surrounding simplicity (and related computational
demands), flexibility and ease of integration with other model features.  Inputs
in the main comprise values pertaining to the solution of particle and dissolved
species transport equations.  Only non-default inputs are included in this
command file.*/
>>CFX4
  >>SET LIMITS
    TOTAL REAL WORK SPACE 5000000
    TOTAL CHARACTER WORK SPACE 2000
  >>OPTIONS
    RECTANGULAR GRID
    CYLINDRICAL COORDINATES
    AXIS INCLUDED
    TURBULENT FLOW
    TRANSIENT FLOW
    USER SCALAR EQUATIONS 55
  >>USER FORTRAN
    USRTRN
    USRCVG
    USRSRC
  >>VARIABLE NAMES
    /*dissolved species concentrations (scalar transport equations solved)
    [kmol.m-3]*/
    /*H.+*/
    USER SCALAR1 'HION'
    /*Ni.2+*/
    USER SCALAR2 'NIION'
    /*Na.+*/
    USER SCALAR3 'NAION'
    /*Cl.-*/
    USER SCALAR4 'CLION'
    /*OH.-*/
    USER SCALAR5 'OHION'
    /*Ni(OH)2 aqueous*/
    USER SCALAR6 'NIOH2'
    /*NiCl2 aqueous*/
    USER SCALAR7 'NICL2'
    /*NiCl.+*/
    USER SCALAR8 'NICLION'
```

```
/*NiOH.+*/
USER SCALAR9 'NIOHION'
/*Ni2OH.3+*/
USER SCALAR10 'NI2OHION'
/*Ni(OH)3.-*/
USER SCALAR11 'NIOH3ION'
/*particle number concentrations (scalar transport equations solved);
behaviour that of dissolved species (advection, diffusion, no gravitational/
buoyancy or drag effects) [m-3]*/
/*1st size interval*/
USER SCALAR12 'PN1'
/*2nd size interval*/
USER SCALAR13 'PN2'
/*3rd size interval*/
USER SCALAR14 'PN3'
/*4th size interval*/
USER SCALAR15 'PN4'
/*5th size interval*/
USER SCALAR16 'PN5'
/*6th size interval*/
USER SCALAR17 'PN6'
/*7th size interval*/
USER SCALAR18 'PN7'
/*8th size interval*/
USER SCALAR19 'PN8'
/*9th size interval*/
USER SCALAR20 'PN9'
/*10th size interval*/
USER SCALAR21 'PN10'
/*11th size interval*/
USER SCALAR22 'PN11'
/*12th size interval*/
USER SCALAR23 'PN12'
/*13th size interval*/
USER SCALAR24 'PN13'
/*all names below are of special class recognised by software; variables
treated as places to store information rather than for solution in scalar
transport equations; included as user scalars to make variable values
available to output facilities, e.g. allows writing to dump file for
graphical display*/
/*supersaturation (user-defined at cell centres)*/
USER SCALAR25 'USRDCC S'
/*pH (user-defined at cell centres)*/
USER SCALAR26 'USRDCC PH'
/*ionic strength (user-defined at cell centres) [kmol.m-3]*/
USER SCALAR27 'USRDCC I'
/*Ni.2+ activity (user-defined at cell centres)*/
USER SCALAR28 'USRDCC ACTNIION'
/*total dissolved nickel concentration (user-defined at cell centres)
[kmol.m-3]*/
USER SCALAR29 'USRDCC TOT DISS NI'
/*0th moment (user-defined at cell centres) [m-3]*/
USER SCALAR30 'USRDCC MOM0'
/*1st moment (user-defined at cell centres) [micron.m-3]*/
USER SCALAR31 'USRDCC MOM1'
/*2nd moment (user-defined at cell centres) [micron2.m-3]*/
USER SCALAR32 'USRDCC MOM2'
/*3rd moment (user-defined at cell centres) [micron3.m-3]*/
USER SCALAR33 'USRDCC MOM3'
/*solids volume fraction (user-defined at cell centres)*/
USER SCALAR34 'USRDCC SOL VOL FRAC'
/*precipitate yield (user-defined at cell centres) [kmol.m-3]*/
USER SCALAR35 'USRDCC PPT'
/*average particle diameter (user-defined at cell centres) [micron]*/
USER SCALAR36 'USRDCC PAVE'
/*coefficient of variation (user-defined at cell centres)*/
USER SCALAR37 'USRDCC COVAR'
/*x-y component of shear stress throughout domain [N.m-2]*/
USER SCALAR38 'XY NODAL SHEAR STRESS'
/*y-z component of shear stress throughout domain [N.m-2]*/
USER SCALAR39 'YZ NODAL SHEAR STRESS'
/*z-x component of shear stress throughout domain [N.m-2]*/
USER SCALAR40 'ZX NODAL SHEAR STRESS'
/*turbulent mixing time scale (user-defined at cell centres) [s]*/
USER SCALAR41 'USRDCC TMIX'
/*kinetics time scale (user-defined at cell centres) [s]*/
USER SCALAR42 'USRDCC TKIN'
```

```
    /*radial velocity (user-defined at all nodes) [m.s-1]*/
    USER SCALAR43 'USRD VRAD'
    /*rotational velocity (user-defined at all nodes) [m.s-1]*/
    USER SCALAR44 'USRD VROT'
    /*following variables included as user scalars solely to store equilibrium
    species values from one time step to next for use as initial speciation
    estimates where necessary (user-defined at cell centres) [log(kmol.m-3)]*/
    /*H.+*/
    USER SCALAR45 'USRDCC STORE1'
    /*Ni.2+*/
    USER SCALAR46 'USRDCC STORE2'
    /*Na.+*/
    USER SCALAR47 'USRDCC STORE3'
    /*Cl.-*/
    USER SCALAR48 'USRDCC STORE4'
    /*OH.-*/
    USER SCALAR49 'USRDCC STORE5'
    /*Ni(OH)2 aqueous*/
    USER SCALAR50 'USRDCC STORE6'
    /*NiCl2 aqueous*/
    USER SCALAR51 'USRDCC STORE7'
    /*NiCl.+*/
    USER SCALAR52 'USRDCC STORE8'
    /*NiOH.+*/
    USER SCALAR53 'USRDCC STORE9'
    /*Ni2OH.3+*/
    USER SCALAR54 'USRDCC STORE10'
    /*Ni(OH)3.-*/
    USER SCALAR55 'USRDCC STORE11'
>>MODEL DATA
  /*SET INITIAL GUESS subcommand not required if restart file contains
  appropriate information related to initial distribution of species and
  particle number concentrations*/
  >>SET INITIAL GUESS
    /*uniform initialisation*/
    >>SET CONSTANT GUESS
      /*in this case, 0.01 kmol.m-3 solution of NiCl2 [kmol.m-3]*/
      HION 0.0000E+00
      NIION 1.0000E-02
      NAION 0.0000E+00
      CLION 2.0000E-02
      OHION 0.0000E+00
      NIOH2 0.0000E+00
      NICL2 0.0000E+00
      NICLION 0.0000E+00
      NIOHION 0.0000E+00
      NI2OHION 0.0000E+00
      NIOH3ION 0.0000E+00
      /*in this case, no solids present [m-3]*/
      PN1 0.0
      PN2 0.0
      PN3 0.0
      PN4 0.0
      PN5 0.0
      PN6 0.0
      PN7 0.0
      PN8 0.0
      PN9 0.0
      PN10 0.0
      PN11 0.0
      PN12 0.0
      PN13 0.0
  >>PHYSICAL PROPERTIES
    >>STANDARD FLUID
      /*assumed that presence of dissolved species and solids volume has no
      effect on fluid/suspension properties*/
      FLUID 'WATER'
      STANDARD FLUID REFERENCE TEMPERATURE 2.9815E+02 /*[K]*/
    >>SCALAR PARAMETERS
      >>DIFFUSIVITIES
        /*approximate values for dissolved aqueous species [(kg.m-3).(m2.s-1)]*/
        HION 1.0000E-06
        NIION 1.0000E-06
        NAION 1.0000E-06
        CLION 1.0000E-06
        OHION 1.0000E-06
        NIOH2 1.0000E-06
```

```
    NICL2 1.0000E-06
    NICLION 1.0000E-06
    NIOHION 1.0000E-06
    NI2OHION 1.0000E-06
    NIOH3ION 1.0000E-06
    /*approximate values predicted by Stokes-Einstein equation for colloidal
    particles of representative size [(kg.m-3).(m2.s-1)]*/
    PN1 1.0E-11
    PN2 1.0E-11
    PN3 1.0E-11
    PN4 1.0E-11
    PN5 1.0E-11
    PN6 1.0E-11
    PN7 1.0E-11
    PN8 1.0E-11
    PN9 1.0E-11
    PN10 1.0E-11
    PN11 1.0E-11
    PN12 1.0E-11
    PN13 1.0E-11
>>TRANSIENT PARAMETERS
  >>FIXED TIME STEPPING
    /*selected time step size somewhat of a compromise between solution
    accuracy and computational time efficiency*/
    TIME STEPS 7200*2.5000E-2 /*[s]*/
    /*INITIAL TIME keyword required if default initial time (i.e. zero or,
    when run is a restart, that taken from restart file) to be overwritten*/
>>TURBULENCE PARAMETERS
  /*arbitrary specification of below parameters (specification required
  since default formula inappropriate due to above diffusivities and
  resultant high molecular Schmidt numbers; arbitrary since solution
  independent of parameter values as a result of zero flux of scalars at all
  walls)*/
  >>LOGLAYER CONSTANT
    HION 10.0
    NIION 10.0
    NAION 10.0
    CLION 10.0
    OHION 10.0
    NIOH2 10.0
    NICL2 10.0
    NICLION 10.0
    NIOHION 10.0
    NI2OHION 10.0
    NIOH3ION 10.0
    PN1 10.0
    PN2 10.0
    PN3 10.0
    PN4 10.0
    PN5 10.0
    PN6 10.0
    PN7 10.0
    PN8 10.0
    PN9 10.0
    PN10 10.0
    PN11 10.0
    PN12 10.0
    PN13 10.0
  >>SUBLAYER THICKNESS
    HION 10.0
    NIION 10.0
    NAION 10.0
    CLION 10.0
    OHION 10.0
    NIOH2 10.0
    NICL2 10.0
    NICLION 10.0
    NIOHION 10.0
    NI2OHION 10.0
    NIOH3ION 10.0
    PN1 10.0
    PN2 10.0
    PN3 10.0
    PN4 10.0
    PN5 10.0
    PN6 10.0
    PN7 10.0
```

```
        PN8 10.0
        PN9 10.0
        PN10 10.0
        PN11 10.0
        PN12 10.0
        PN13 10.0
>>SOLVER DATA
  >>PROGRAM CONTROL
    /*species and particle number concentration equations solved on their own
    with fixed velocity field of restart file; assumed that volume addition and
    presence of dissolved species and solid volume have no effect on flow;
    characteristic unsteady (quasi steady state) interaction between baffles and
    impeller blades neglected*/
    ITERATIONS OF HYDRODYNAMIC EQUATIONS 0
  >>UNDER RELAXATION FACTORS
    HION 8.0000E-01
    NIION 8.0000E-01
    NAION 8.0000E-01
    CLION 8.0000E-01
    OHION 8.0000E-01
    NIOH2 8.0000E-01
    NICL2 8.0000E-01
    NICLION 8.0000E-01
    NIOHION 8.0000E-01
    NI2OHION 8.0000E-01
    NIOH3ION 8.0000E-01
    PN1 8.0000E-01
    PN2 8.0000E-01
    PN3 8.0000E-01
    PN4 8.0000E-01
    PN5 8.0000E-01
    PN6 8.0000E-01
    PN7 8.0000E-01
    PN8 8.0000E-01
    PN9 8.0000E-01
    PN10 8.0000E-01
    PN11 8.0000E-01
    PN12 8.0000E-01
    PN13 8.0000E-01
>>MODEL BOUNDARY CONDITIONS
>>OUTPUT OPTIONS
  >>DUMP FILE FORMAT
    /*double precision specified to prevent floating error during write*/
    DOUBLE PRECISION
>>STOP
```

A.2.2 Fortran file

```
C Throughout the flow domain this Fortran file performs the necessary speciation
C calculations, the results of which inform the rates of the precipitation processes
C of nucleation and aggregation, and establishes and integrates the discretised
C population balance equations, which describe the evolution of the particle size
C distribution.  In addition, CFX-4.3 Solver convergence criteria are set and the
C localised source terms used to model the introduction of feed are calculated for the
C relevant transport equations.  Indeed, if employing a fixed pre-calculated flow
C field, it is only the transport (advection and diffusion) of particles and dissolved
C species which is modelled exclusively external to a user-defined subroutine.  This
C file could be run in tandem with a fixed flow field generated by the high resolution
C flow calculation (a selected quasi steady state representation) or even the
C transient field itself.  Both proposals would substantially inflate simulation times
C and necessitate various but relatively minor alterations to the relevant command and
C geometry files.  The latter may require revision of subroutine USRCVG within this
C file.
C
C*********************************************************************
C
C This subprogram is a long one and, as such, only a relatively brief summary is given
C here.
C
C Subroutine USRTRN is called by CFX-4.3 Solver before the start of the run and at the
C end of each time step.  At the first call, initial conditions are read and the
C system simply equilibrated within each interior cell using subroutine NS11AD.
C Species and particle number concentrations are then passed back to CFX-4.3 Solver.
```

```
C At subsequent calls, many more calculations are performed, again within each
C interior cell.  Cell contents are first diluted as preferred and the system is
C equilibrated through a call to subroutine NS11AD.  These calculations allow the
C various solution conditions governing the rates of nucleation and aggregation to be
C established.  Other precipitation processes affecting the properties of the solid
C phase are deemed insignificant and neglected.  If limited by the turbulent mixing
C rate, the nucleation rate is adjusted before the effects of the formation of new
C nuclei and aggregates are taken into account and the particle size distribution
C updated using Euler's method of numerical integration.  The discretised population
C balance of Hounslow et al. is amended and employed here.  Observing the impact of
C solute deposition, the equilibrium concentrations of the dissolved species are then
C determined (subroutine NS11AD again invoked) prior to their return, together with
C the associated particle number concentrations, to CFX-4.3 Solver.  The various other
C user scalars are evaluated as necessary.  Assorted data of interest are also written
C to user files at selected times.  Required parameters and option settings are
C inputted, and a number of checks are included to prevent the generation and
C propagation of errors.  Particles are assumed to be spherical.  Over reasonable
C periods of feed addition the various dilution options (both those encoded and the
C many others possible) are, as a result of the small time step size, ultimately not
C too dissimilar in effect.  Finally, reservations over the different available
C formulations of the equations for the nucleation rate and the size-independent
C portion of the aggregation kernel (extracted from bulk experimental measurements)
C must be expressed.  In order to exploit adequately the detailed spatial dynamics
C afforded, inspire confidence in results and gain real insight into the system of
C interest, a more fundamental description of the kinetics, obtained by a more
C appropriate method and capturing true dependencies, is desirable.  Also ideally
C preferred is the modelling of a smaller, more realistic nucleus size in conjunction
C with a consistent, size-dependent aggregate density.
C
      SUBROUTINE USRTRN(U,V,W,P,VFRAC,DEN,VIS,TE,ED,RS,T,H,RF,SCAL,XP,
     +                  YP,ZP,VOL,AREA,VPOR,ARPOR,WFACT,CONV,IPT,IBLK,
     +                  IPVERT,IPNODN,IPFACN,IPNODF,IPNODB,IPFACB,WORK,
     +                  IWORK,CWORK)
C
C***********************************************************************
C
C   USER SUBROUTINE TO ALLOW USERS TO MODIFY OR MONITOR THE SOLUTION AT
C   THE END OF EACH TIME STEP
C   THIS SUBROUTINE IS CALLED BEFORE THE START OF THE RUN AS WELL AS AT
C   THE END OF EACH TIME STEP
C
C   >>> IMPORTANT                                                    <<<
C   >>>                                                            <<<
C   >>> USERS MAY ONLY ADD OR ALTER PARTS OF THE SUBROUTINE WITHIN  <<<
C   >>> THE DESIGNATED USER AREAS                                  <<<
C
C***********************************************************************
C
C   THIS SUBROUTINE IS CALLED BY THE FOLLOWING SUBROUTINES
C      CUSR  TRNMOD
C
C***********************************************************************
C
C   CREATED
C      27/04/90  ADB
C   MODIFIED
C      05/08/91  IRH  NEW STRUCTURE
C      01/10/91  DSC  REDUCE COMMENT LINE GOING OVER COLUMN 72.
C      29/11/91  PHA  UPDATE CALLED BY COMMENT, ADD RF ARGUMENT,
C                     CHANGE LAST DIMENSION OF RS TO 6 AND IVERS TO 2
C      05/06/92  PHA  ADD PRECISION FLAG AND CHANGE IVERS TO 3
C      03/07/92  DSC  CORRECT COMMON MLTGRD.
C      23/11/93  CSH  EXPLICITLY DIMENSION IPVERT ETC.
C      03/02/94  PHA  CHANGE FLOW3D TO CFDS-FLOW3D
C      22/08/94  NSW  MOVE 'IF(IUSED.EQ.0) RETURN' OUT OF USER AREA
C      19/12/94  NSW  CHANGE FOR CFX-F3D
C      02/07/97  NSW  UPDATE FOR CFX-4
C      02/07/99  NSW  INCLUDE NEW EXAMPLE FOR CALCULATING FLUX OF A
C                     SCALAR AT A PRESSURE BOUNDARY
C
C***********************************************************************
C
C   SUBROUTINE ARGUMENTS
C
C     U      - U COMPONENT OF VELOCITY
C     V      - V COMPONENT OF VELOCITY
C     W      - W COMPONENT OF VELOCITY
```

```
C     P      - PRESSURE
C     VFRAC  - VOLUME FRACTION
C     DEN    - DENSforTY OF FLUID
C     VIS    - VISCOSITY OF FLUID
C     TE     - TURBULENT KINETIC ENERGY
C     ED     - EPSILON
C     RS     - REYNOLD STRESSES
C     T      - TEMPERATURE
C     H      - ENTHALPY
C     RF     - REYNOLD FLUXES
C     SCAL   - SCALARS (THE FIRST 'NCONC' OF THESE ARE MASS FRACTIONS)
C     XP     - X COORDINATES OF CELL CENTRES
C     YP     - Y COORDINATES OF CELL CENTRES
C     ZP     - Z COORDINATES OF CELL CENTRES
C     VOL    - VOLUME OF CELLS
C     AREA   - AREA OF CELLS
C     VPOR   - POROUS VOLUME
C     ARPOR  - POROUS AREA
C     WFACT  - WEIGHT FACTORS
C     CONV   - CONVECTION COEFFICIENTS
C
C     IPT    - 1D POINTER ARRAY
C     IBLK   - BLOCK SIZE INFORMATION
C     IPVERT - POINTER FROM CELL CENTRES TO 8 NEIGHBOURING VERTICES
C     IPNODN - POINTER FROM CELL CENTRES TO 6 NEIGHBOURING CELLS
C     IPFACN - POINTER FROM CELL CENTRES TO 6 NEIGHBOURING FACES
C     IPNODF - POINTER FROM CELL FACES TO 2 NEIGHBOURING CELL CENTRES
C     IPNODB - POINTER FROM BOUNDARY CENTRES TO CELL CENTRES
C     IPFACB - POINTER FROM BOUNDARY CENTRES TO BOUNDARY FACES
C
C     WORK   - REAL WORKSPACE ARRAY
C     IWORK  - INTEGER WORKSPACE ARRAY
C     CWORK  - CHARACTER WORKSPACE ARRAY
C
C   SUBROUTINE ARGUMENTS PRECEDED WITH A '*' ARE ARGUMENTS THAT MUST
C   BE SET BY THE USER IN THIS ROUTINE.
C
C   NOTE THAT OTHER DATA MAY BE OBTAINED FROM CFX-4 USING THE
C   ROUTINE GETADD, FOR FURTHER DETAILS SEE THE VERSION 4
C   USER MANUAL.
C
C*************************************************************************
C
      DOUBLE PRECISION U
      DOUBLE PRECISION V
      DOUBLE PRECISION W
      DOUBLE PRECISION P
      DOUBLE PRECISION VFRAC
      DOUBLE PRECISION DEN
      DOUBLE PRECISION VIS
      DOUBLE PRECISION TE
      DOUBLE PRECISION ED
      DOUBLE PRECISION RS
      DOUBLE PRECISION T
      DOUBLE PRECISION H
      DOUBLE PRECISION RF
      DOUBLE PRECISION SCAL
      DOUBLE PRECISION XP
      DOUBLE PRECISION YP
      DOUBLE PRECISION ZP
      DOUBLE PRECISION VOL
      DOUBLE PRECISION AREA
      DOUBLE PRECISION VPOR
      DOUBLE PRECISION ARPOR
      DOUBLE PRECISION WFACT
      DOUBLE PRECISION CONV
      DOUBLE PRECISION WORK
      DOUBLE PRECISION SMALL
      DOUBLE PRECISION SORMAX
      DOUBLE PRECISION DTUSR
      DOUBLE PRECISION TIME
      DOUBLE PRECISION DT
      DOUBLE PRECISION DTINVF
      DOUBLE PRECISION TPARM
      DOUBLE PRECISION SGNWL
      LOGICAL LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP,LRECT,LCYN,LAXIS,
     +        LPOROS,LTRANS
```

```
C
      CHARACTER*(*) CWORK
C
C+++++++++++++++++ USER AREA 1 ++++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS EXPLICITLY DECLARED VARIABLES
C
      DOUBLE PRECISION KSP,PI,PSIZE,R,RHO
      DOUBLE PRECISION FEEDCONC,FEEDVOL
      DOUBLE PRECISION INITTIME,INITVOL
      DOUBLE PRECISION FEEDPVOL,GEOMVOL,TOTALVOL
      DOUBLE PRECISION PAVE1,PAVE2,PAVE3
      DOUBLE PRECISION AGGREG,BIRTH,DPPT,MOM3I,PN
      DOUBLE PRECISION AR,AR1,AR2,AR3,AR4,EXP1,EXP2,SUM1,SUM3,SUM4,
     &                 TERM1,TERM2,TERM3,TERM4
      DOUBLE PRECISION PHMAX,PHMIN
      DOUBLE PRECISION OVERCV,OVERMOM0,OVERMOM1,OVERMOM2,OVERMOM3,OVERN
     &                 ,OVERPAVE,OVERPPT,OVERSVF,OVERTDNC
      DOUBLE PRECISION ACTIV
      DOUBLE PRECISION CC
      DOUBLE PRECISION LAS,STION
      DOUBLE PRECISION A,ACC,F,STEP,STPMAX,WO,X
C
      INTEGER GENSET1,GENSET2,GENSET3,GENSET4,GENSET5
      INTEGER NOPSI,FEEDSTEP
      INTEGER I,INODE,J,K
      INTEGER FEED
      INTEGER IPRINT,IW,MAXFUN,N
      INTEGER SPECSET1,SPECSET2
C
C+++++++++++++++++ END OF USER AREA 1 +++++++++++++++++++++++++++++++++++
C
      COMMON /ALL/NBLOCK,NCELL,NBDRY,NNODE,NFACE,NVERT,NDIM,
     +       /ALLWRK/NRWS,NIWS,NCWS,IWRFRE,IWIFRE,IWCFRE,/ADDIMS/NPHASE,
     +       NSCAL,NVAR,NPROP,NDVAR,NDPROP,NDXNN,NDGEOM,NDCOEF,NILIST,
     +       NRLIST,NTOPOL,/CHKUSR/IVERS,IUCALL,IUSED,/CONC/NCONC,
     +       /DEVICE/NREAD,NWRITE,NRDISK,NWDISK,/IDUM/ILEN,JLEN,
     +       /LOGIC/LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP,LRECT,LCYN,
     +       LAXIS,LPOROS,LTRANS,/MLTGRD/MLEVEL,NLEVEL,ILEVEL,
     +       /SGLDBL/IFLGPR,ICHKPR,/SPARM/SMALL,SORMAX,NITER,INDPRI,
     +       MAXIT,NODREF,NODMON,/TIMUSR/DTUSR,/TRANSI/NSTEP,KSTEP,MF,
     +       INCORE,/TRANSR/TIME,DT,DTINVF,TPARM
C
C+++++++++++++++++ USER AREA 2 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DECLARE THEIR OWN COMMON BLOCKS
C     THESE SHOULD START WITH THE CHARACTERS 'UC' TO ENSURE
C     NO CONFLICT WITH NON-USER COMMON BLOCKS
C
      COMMON /UCSPEC1/KSP
      COMMON /UCSPEC2/CC
      COMMON /UCSPEC3/LAS,STION
      COMMON /UCSPEC4/SPECSET1,SPECSET2
      COMMON /UCFEED1/FEEDCONC,FEEDVOL
      COMMON /UCFEED2/FEEDPVOL,GEOMVOL,TOTALVOL
      COMMON /UCFEED3/GENSET3
C
C+++++++++++++++++ END OF USER AREA 2 +++++++++++++++++++++++++++++++++++
C
      DIMENSION U(NNODE,NPHASE),V(NNODE,NPHASE),W(NNODE,NPHASE),
     +          P(NNODE,NPHASE),VFRAC(NNODE,NPHASE),DEN(NNODE,NPHASE),
     +          VIS(NNODE,NPHASE),TE(NNODE,NPHASE),ED(NNODE,NPHASE),
     +          RS(NNODE,NPHASE,6),T(NNODE,NPHASE),H(NNODE,NPHASE),
     +          RF(NNODE,NPHASE,4),SCAL(NNODE,NPHASE,NSCAL)
      DIMENSION XP(NNODE),YP(NNODE),ZP(NNODE),VOL(NCELL),AREA(NFACE,3),
     +          VPOR(NCELL),ARPOR(NFACE,3),WFACT(NFACE),
     +          CONV(NFACE,NPHASE),IPT(*),IBLK(5,NBLOCK),
     +          IPVERT(NCELL,8),IPNODN(NCELL,6),IPFACN(NCELL,6),
     +          IPNODF(NFACE,4),IPNODB(NBDRY,4),IPFACB(NBDRY),IWORK(*),
     +          WORK(*),CWORK(*)
      DIMENSION SGNWL(6)
C
C+++++++++++++++++ USER AREA 3 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DIMENSION THEIR ARRAYS
C
C     specify bounds of explicit-shape arrays as constant expressions to be adjusted
C     as and when necessary; large automatic arrays, e.g. those using NCELL as array
C     specification, can result in stack overflow; alternative is to increase stack
C     size
```

```
C
      DIMENSION PSIZE(13)
      DIMENSION PAVE1(13),PAVE2(13),PAVE3(13)
      DIMENSION AGGREG(13,10064),BIRTH(10064),DPPT(10064),MOM3I(10064),
     &          PN(13,10064)
      DIMENSION AR(10064)
      DIMENSION OVERN(13)
      DIMENSION ACTIV(12,10064)
      DIMENSION CC(4)
      DIMENSION LAS(12)
      DIMENSION A(12,12),F(12),WO(400),X(12)
C
      DIMENSION FEED(10064)
      DIMENSION IW(12)
C
C---- AREA FOR USERS TO DEFINE DATA STATEMENTS
C
C+++++++++++++++++ END OF USER AREA 3 +++++++++++++++++++++++++++++++++++
C
      DATA SGNWL/1.0D0,1.0D0,1.0D0,-1.0D0,-1.0D0,-1.0D0/
C
C---- STATEMENT FUNCTION FOR ADDRESSING
      IP(I,J,K) = IPT((K-1)*ILEN*JLEN+ (J-1)*ILEN+I)
C
C---- VERSION NUMBER OF USER ROUTINE AND PRECISION FLAG
C
      IVERS = 3
      ICHKPR = 2
C
C+++++++++++++++++ USER AREA 4 +++++++++++++++++++++++++++++++++++++++++
C---- TO USE THIS USER ROUTINE FIRST SET IUSED=1
C
      IUSED = 1
C
C+++++++++++++++++ END OF USER AREA 4 +++++++++++++++++++++++++++++++++++
C
      IF (IUSED.EQ.0) RETURN
C
C---- FRONTEND CHECKING OF USER ROUTINE
      IF (IUCALL.EQ.0) RETURN
C
C+++++++++++++++++ USER AREA 5 +++++++++++++++++++++++++++++++++++++++++
C
C---- EXAMPLE (SET TIME INCREMENT FOR NEXT TIME STEP)
C
C     DTUSR = 0.1
C
C---- END OF EXAMPLE
C
C---- EXAMPLE (CALCULATE FLUX OF FIRST SCALAR AT A PRESSURE BOUNDARY)
C
C     IPHASE = 1
C     FLUX = 0.0
C  USE IPALL TO FIND ADDRESSES OF BOUNDARY NODES ON PATCH PRESS1
C     CALL IPALL('PRESS1','PRESS','PATCH','CENTRES'
C    +          ,IPT,NPT,CWORK,IWORK)
C  LOOP OVER ALL BOUNDARY NODES
C     DO 300 I=1,NPT
C  USE ARRAY IPT TO GET ADDRESS
C       INODE = IPT(I)
C       IBDRY = INODE - NCELL
C       IFACE = IPFACB(IBDRY)
C       INDUM = IPNODB(IBDRY,2)
C       NWL   = IPNODB(IBDRY,4)
C       FLUX  = FLUX
C    +        + SGNWL(NWL)*CONV(IFACE,IPHASE)*SCAL(INDUM,IPHASE,1)
C 300 CONTINUE
C
C---- END OF EXAMPLE
C
C     select various general settings
C     1 - employ equilibrium description of nucleation; 2 - employ kinetic description
C     of nucleation
      GENSET1 = 1
C     1 - perform aggregation-related calculations after effects of nucleation
C     established; 2 - perform aggregation-related calculations before effects of
C     nucleation established
```

```
      GENSET2 = 1
C     1 - apportion fluid volume associated with feed among all cells (i.e.
C     distribute dilution effect); 2 - confine fluid volume associated with feed to
C     feed patch (i.e. isolate dilution effect); 3 - confine fluid volume associated
C     with feed to feed patch, apportion among all cells prior to return to CFX-4.3
C     Solver, team with time-dependent feed source coding in subroutine USRSRC
C     (strategy ensures mass conservation)
      GENSET3 = 3
C     1 - do not stop if contents of stirred tank deemed to be uniformly distributed;
C     2 - stop if contents of stirred tank deemed to be uniformly distributed
      GENSET4 = 1
C     1 - read uniform initial conditions from command file; 2 - read initial
C     conditions from restart file
      GENSET5 = 1
C
C     write output to file fort.99
      WRITE(99,*) 'TIME',TIME
C
C     calculate pi
      PI = ACOS(-1.0)
C
C     specify various model parameters
C     lower bound of first size interval [micron]
      PSIZE(1) = 2.016
C     ratio of upper and lower bounds for any size interval (non-adjustable
C     for DPB of Hounslow et al.)
      R = 2.0**(1.0/3.0)
C     number of particle size intervals
      NOPSI = 13
C     Ni(OH)2 solubility product at temperature of interest (25 deg C)
      KSP = 1.9953E-15
C     experimentally determined precipitate molar density [kmol.m-3]
      RHO = 1.26
C     volumetric feed rate [m3.s-1]
      FEEDVOL = 2.0E-7
C     feed NaOH concentration [kmol.m-3]
      FEEDCONC = 0.075
C
C     if uniform initial conditions read from command file
      IF (GENSET5.EQ.1) THEN
C       specify initial volume [m3]
        INITVOL = 0.003
C       calculate initial time [s]
        INITTIME = TIME-(KSTEP*DT)
C     if initial conditions read from restart file
      ELSE IF (GENSET5.EQ.2) THEN
C       specify initial volume associated with relevant uniform initialisation [m3]
        INITVOL = 0.003
C       specify initial time associated with relevant uniform initialisation [s]
        INITTIME = 0.0
      END IF
C
C     specify duration of feed addition in terms of number of time steps (equate to
C     NSTEP if feed uninterrupted for duration of simulation)
      FEEDSTEP = NSTEP
C
C     if feed addition ongoing
      IF (KSTEP.LE.FEEDSTEP) THEN
C       calculate total volume [m3]
        TOTALVOL = INITVOL+(FEEDVOL*(TIME-INITTIME))
C     if feed addition halted
      ELSE
C       calculate total volume [m3]
        TOTALVOL = INITVOL+(FEEDVOL*(TIME-INITTIME-(KSTEP-FEEDSTEP)*DT)
     &             )
C       specify zero volumetric feed rate [m3.s-1]
        FEEDVOL = 0.0
      END IF
C
C     call utility routine GETSCA to find number of scalar variable within SCAL array
C     dissolved species concentrations [kmol.m-3]
C     H.+
      CALL GETSCA('HION',IH,CWORK)
C     Ni.2+
      CALL GETSCA('NIION',INI,CWORK)
C     Na.+
      CALL GETSCA('NAION',INA,CWORK)
```

```
C     Cl.-
      CALL GETSCA('CLION',ICL,CWORK)
C     OH.-
      CALL GETSCA('OHION',IOH,CWORK)
C     Ni(OH)2 aqueous
      CALL GETSCA('NIOH2',INIOH2,CWORK)
C     NiCl2 aqueous
      CALL GETSCA('NICL2',INICL2,CWORK)
C     NiCl.+
      CALL GETSCA('NICLION',INICL,CWORK)
C     NiOH.+
      CALL GETSCA('NIOHION',INIOH,CWORK)
C     Ni2OH.3+
      CALL GETSCA('NI2OHION',INI2OH,CWORK)
C     Ni(OH)3.-
      CALL GETSCA('NIOH3ION',INIOH3,CWORK)
C     particle number concentrations [m-3]
C     1st size interval
      CALL GETSCA('PN1',I1,CWORK)
C     2nd size interval
      CALL GETSCA('PN2',I2,CWORK)
C     3rd size interval
      CALL GETSCA('PN3',I3,CWORK)
C     4th size interval
      CALL GETSCA('PN4',I4,CWORK)
C     5th size interval
      CALL GETSCA('PN5',I5,CWORK)
C     6th size interval
      CALL GETSCA('PN6',I6,CWORK)
C     7th size interval
      CALL GETSCA('PN7',I7,CWORK)
C     8th size interval
      CALL GETSCA('PN8',I8,CWORK)
C     9th size interval
      CALL GETSCA('PN9',I9,CWORK)
C     10th size interval
      CALL GETSCA('PN10',I10,CWORK)
C     11th size interval
      CALL GETSCA('PN11',I11,CWORK)
C     12th size interval
      CALL GETSCA('PN12',I12,CWORK)
C     13th size interval
      CALL GETSCA('PN13',I13,CWORK)
C     supersaturation
      CALL GETSCA('USRDCC S',ISUPER,CWORK)
C     pH
      CALL GETSCA('USRDCC PH',IPH,CWORK)
C     ionic strength [kmol.m-3]
      CALL GETSCA('USRDCC I',ISTION,CWORK)
C     Ni.2+ activity
      CALL GETSCA('USRDCC ACTNIION',IANI,CWORK)
C     total dissolved nickel concentration [kmol.m-3]
      CALL GETSCA('USRDCC TOT DISS NI',ITDNC,CWORK)
C     0th moment [m-3]
      CALL GETSCA('USRDCC MOM0',IM0,CWORK)
C     1st moment [micron.m-3]
      CALL GETSCA('USRDCC MOM1',IM1,CWORK)
C     2nd moment [micron2.m-3]
      CALL GETSCA('USRDCC MOM2',IM2,CWORK)
C     3rd moment [micron3.m-3]
      CALL GETSCA('USRDCC MOM3',IM3,CWORK)
C     solids volume fraction
      CALL GETSCA('USRDCC SOL VOL FRAC',ISVF,CWORK)
C     precipitate yield [kmol.m-3]
      CALL GETSCA('USRDCC PPT',IPPT,CWORK)
C     average particle diameter [micron]
      CALL GETSCA('USRDCC PAVE',IPA,CWORK)
C     coefficient of variation
      CALL GETSCA('USRDCC COVAR',ICV,CWORK)
C     turbulent mixing time scale [s]
      CALL GETSCA('USRDCC TMIX',ITM,CWORK)
C     kinetics time scale [s]
      CALL GETSCA('USRDCC TKIN',ITK,CWORK)
C     logarithm of dissolved species concentrations (stored from previous time step;
C     stored for use in following time step) [log(kmol.m-3)]
C     H.+
      CALL GETSCA('USRDCC STORE1',IS1,CWORK)
```

```
C     Ni.2+
      CALL GETSCA('USRDCC STORE2',IS2,CWORK)
C     Na.+
      CALL GETSCA('USRDCC STORE3',IS3,CWORK)
C     Cl.-
      CALL GETSCA('USRDCC STORE4',IS4,CWORK)
C     OH.-
      CALL GETSCA('USRDCC STORE5',IS5,CWORK)
C     Ni(OH)2 aqueous
      CALL GETSCA('USRDCC STORE6',IS6,CWORK)
C     NiCl2 aqueous
      CALL GETSCA('USRDCC STORE7',IS7,CWORK)
C     NiCl.+
      CALL GETSCA('USRDCC STORE8',IS8,CWORK)
C     NiOH.+
      CALL GETSCA('USRDCC STORE9',IS9,CWORK)
C     Ni2OH.3+
      CALL GETSCA('USRDCC STORE10',IS10,CWORK)
C     Ni(OH)3.-
      CALL GETSCA('USRDCC STORE11',IS11,CWORK)
C
C     calculate lower bounds of size intervals [micron]
      DO 10 I = 2,NOPSI
        PSIZE(I) = PSIZE(I-1)*R
 10   CONTINUE
C
C     calculate appropriate mean size in each size interval for calculating 1st, 2nd
C     and 3rd moments [micron]
      DO 20 I = 1,NOPSI
        PAVE1(I) = (1.0/2.0)*(((R**2.0)-1.0)/(R-1.0))*PSIZE(I)
        PAVE2(I) = ((1.0/3.0)*(((R**3.0)-1.0)/(R-1.0))*(PSIZE(I)**2.0))
     &             **(1.0/2.0)
        PAVE3(I) = ((1.0/4.0)*(((R**4.0)-1.0)/(R-1.0))*(PSIZE(I)**3.0))
     &             **(1.0/3.0)
 20   CONTINUE
C
C     initialise tank geometry volume (mesh volume) [m3]
      GEOMVOL = 0.0
C
C     call utility routine IPALL to return cell-centre addresses of those cells
C     occupied by tank contents (i.e. all interior cells)
      CALL IPALL('*','*','BLOCK','CENTRES',IPT,NPT,CWORK,IWORK)
C
C     within each cell in flow domain
      DO 30 I = 1,NPT
        INODE = IPT(I)
C       calculate tank geometry volume (mesh volume) [m3]
        GEOMVOL = GEOMVOL+VOL(INODE)
C       flag as non-feed cell
        FEED(INODE) = 1
 30   CONTINUE
C
C     initialise feed patch volume [m3]
      FEEDPVOL = 0.0
C
C     call utility routine IPALL to return cell-centre addresses of those cells
C     occupying feed patch
      CALL IPALL('USER3D_FEED','USER3D','PATCH','CENTRES',IPT,NPT,CWORK
     &     ,IWORK)
C
C     within each cell in feed patch
      DO 40 I = 1,NPT
        INODE = IPT(I)
C       calculate feed patch volume [m3]
        FEEDPVOL = FEEDPVOL+VOL(INODE)
C       flag as feed cell
        FEED(INODE) = 2
 40   CONTINUE
C
C     initialise maximum pH with arbitrarily small value
      PHMAX = 0.0
C     initialise minimum pH with arbitrarily large value
      PHMIN = 100.0
C
C     call utility routine IPALL to return cell-centre addresses of those cells
C     occupied by tank contents (i.e. all interior cells)
      CALL IPALL('*','*','BLOCK','CENTRES',IPT,NPT,CWORK,IWORK)
```

```
C
C     within each cell in flow domain
      DO 50 K = 1,NPT
C
        INODE = IPT(K)
C
C       stop if negative species concentration(s) outputted by CFX-4.3 Solver
        IF (SCAL(INODE,1,IH).LT.0.0.OR.SCAL(INODE,1,INI).LT.0.0.OR.
     &     SCAL(INODE,1,INA).LT.0.0.OR.SCAL(INODE,1,ICL).LT.0.0.OR.
     &     SCAL(INODE,1,IOH).LT.0.0.OR.SCAL(INODE,1,INIOH2).LT.0.0.OR.
     &     SCAL(INODE,1,INICL2).LT.0.0.OR.SCAL(INODE,1,INICL).LT.0.0.OR
     &     .SCAL(INODE,1,INIOH).LT.0.0.OR.SCAL(INODE,1,INI2OH).LT.0.0.
     &     OR.SCAL(INODE,1,INIOH3).LT.0.0) THEN
C         write output to file fort.99
          WRITE(99,*) 'ERROR: NEGATIVE SPECIES CONCENTRATION(S) '//
     &                'OUTPUTTED BY CFX-4.3 SOLVER'
          WRITE(99,*) 'CELL',INODE
C         stop with message output
          STOP 'NEGATIVE SPECIES CONCENTRATION(S) OUTPUTTED BY CFX-4.3 S
     &OLVER'
        END IF
C
C       stop if negative particle number concentration(s) outputted by CFX-4.3 Solver
        IF (SCAL(INODE,1,I1).LT.0.0.OR.SCAL(INODE,1,I2).LT.0.0.OR.
     &     SCAL(INODE,1,I3).LT.0.0.OR.SCAL(INODE,1,I4).LT.0.0.OR.
     &     SCAL(INODE,1,I5).LT.0.0.OR.SCAL(INODE,1,I6).LT.0.0.OR.
     &     SCAL(INODE,1,I7).LT.0.0.OR.SCAL(INODE,1,I8).LT.0.0.OR.
     &     SCAL(INODE,1,I9).LT.0.0.OR.SCAL(INODE,1,I10).LT.0.0.OR.
     &     SCAL(INODE,1,I11).LT.0.0.OR.SCAL(INODE,1,I12).LT.0.0.OR.
     &     SCAL(INODE,1,I13).LT.0.0) THEN
C         write output to file fort.99
          WRITE(99,*) 'ERROR: NEGATIVE PARTICLE NUMBER '//
     &                'CONCENTRATION(S) OUTPUTTED BY CFX-4.3 SOLVER'
          WRITE(99,*) 'CELL',INODE
C         stop with message output
          STOP 'NEGATIVE PARTICLE NUMBER CONCENTRATION(S) OUTPUTTED BY C
     &FX-4.3 SOLVER'
        END IF
C
C       temporarily rename particle number concentration variables for ease of
C       referencing [m-3]
C       1st size interval
        PN(1,INODE) = SCAL(INODE,1,I1)
C       2nd size interval
        PN(2,INODE) = SCAL(INODE,1,I2)
C       3rd size interval
        PN(3,INODE) = SCAL(INODE,1,I3)
C       4th size interval
        PN(4,INODE) = SCAL(INODE,1,I4)
C       5th size interval
        PN(5,INODE) = SCAL(INODE,1,I5)
C       6th size interval
        PN(6,INODE) = SCAL(INODE,1,I6)
C       7th size interval
        PN(7,INODE) = SCAL(INODE,1,I7)
C       8th size interval
        PN(8,INODE) = SCAL(INODE,1,I8)
C       9th size interval
        PN(9,INODE) = SCAL(INODE,1,I9)
C       10th size interval
        PN(10,INODE) = SCAL(INODE,1,I10)
C       11th size interval
        PN(11,INODE) = SCAL(INODE,1,I11)
C       12th size interval
        PN(12,INODE) = SCAL(INODE,1,I12)
C       13th size interval
        PN(13,INODE) = SCAL(INODE,1,I13)
C
C       at beginning of first time step
        IF (KSTEP.EQ.0) THEN
C
C         calculate initial 3rd moment [micron3.m-3]
          MOM3I(INODE) = 0.0
C
          DO 60 I = 1,NOPSI
            MOM3I(INODE) = MOM3I(INODE)+((PAVE3(I)**3.0)*PN(I,INODE))
 60       CONTINUE
```

```
C
C         calculate precipitate yield [kmol.m-3]
          SCAL(INODE,1,IPPT) = (PI*MOM3I(INODE)/6.0)*RHO*1.0E-18
C
C         if uniform initial conditions read from command file
          IF (GENSET5.EQ.1) THEN
C
C           estimate logarithm of equilibrium species concentrations (supplied as
C           initial estimate to subroutine NS11AD; required degree of accuracy
C           provided by stand-alone equilibrium calculation using powerful and robust
C           solver such as that offered by gPROMS modelling platform) [log(kmol.m-3)]
C           H.+
            X(1) = -6.0
C           Ni.2+
            X(2) = -2.0
C           Na.+
            X(3) = -1000.0
C           Cl.-
            X(4) = -1.7
C           OH.-
            X(5) = -7.9
C           Ni(OH)2 aqueous
            X(6) = -9.0
C           NiC12 aqueous
            X(7) = -4.7
C           NiCl.+
            X(8) = -3.6
C           NiOH.+
            X(9) = -6.0
C           Ni2OH.3+
            X(10) = -8.6
C           Ni(OH)3.-
            X(11) = -14.0
C
C           calculate total component concentrations (excluding solid species; used in
C           subroutine CALFUN to close mass balance) [kmol.m-3]
C           H.+
            CC(1) = SCAL(INODE,1,IH)-SCAL(INODE,1,IOH)-(2.0*
     &              SCAL(INODE,1,INIOH2))-SCAL(INODE,1,INIOH)-
     &              SCAL(INODE,1,INI2OH)-(3.0*SCAL(INODE,1,INIOH3))
C           Ni.2+
            CC(2) = SCAL(INODE,1,INI)+SCAL(INODE,1,INIOH2)+
     &              SCAL(INODE,1,INICL2)+SCAL(INODE,1,INICL)+
     &              SCAL(INODE,1,INIOH)+(2.0*SCAL(INODE,1,INI2OH))+
     &              SCAL(INODE,1,INIOH3)
C           Na.+
            CC(3) = SCAL(INODE,1,INA)
C           Cl.-
            CC(4) = SCAL(INODE,1,ICL)+(2.0*SCAL(INODE,1,INICL2))+
     &              SCAL(INODE,1,INICL)
C
C         if initial conditions read from restart file
          ELSE IF (GENSET5.EQ.2) THEN
C
C           estimate logarithm of equilibrium species concentrations as logarithm of
C           current species concentrations (supplied as initial estimate to subroutine
C           NS11AD) [log(kmol.m-3)]
C           H.+
            X(1) = LOG10(SCAL(INODE,1,IH))
C
C           Ni.2+
            IF (SCAL(INODE,1,INI).EQ.0.0) THEN
              X(2) = -1000.0
            ELSE
              X(2) = LOG10(SCAL(INODE,1,INI))
            END IF
C
C           Na.+
            IF (SCAL(INODE,1,INA).EQ.0.0) THEN
              X(3) = -1000.0
            ELSE
              X(3) = LOG10(SCAL(INODE,1,INA))
            END IF
C
C           Cl.-
            X(4) = LOG10(SCAL(INODE,1,ICL))
C           OH.-
```

```
            X(5) = LOG10(SCAL(INODE,1,IOH))
C
C           Ni(OH)2 aqueous
            IF (SCAL(INODE,1,INIOH2).EQ.0.0) THEN
              X(6) = -1000.0
            ELSE
              X(6) = LOG10(SCAL(INODE,1,INIOH2))
            END IF
C
C           NiCl2 aqueous
            IF (SCAL(INODE,1,INICL2).EQ.0.0) THEN
              X(7) = -1000.0
            ELSE
              X(7) = LOG10(SCAL(INODE,1,INICL2))
            END IF
C
C           NiCl.+
            IF (SCAL(INODE,1,INICL).EQ.0.0) THEN
              X(8) = -1000.0
            ELSE
              X(8) = LOG10(SCAL(INODE,1,INICL))
            END IF
C
C           NiOH.+
            IF (SCAL(INODE,1,INIOH).EQ.0.0) THEN
              X(9) = -1000.0
            ELSE
              X(9) = LOG10(SCAL(INODE,1,INIOH))
            END IF
C
C           Ni2OH.3+
            IF (SCAL(INODE,1,INI2OH).EQ.0.0) THEN
              X(10) = -1000.0
            ELSE
              X(10) = LOG10(SCAL(INODE,1,INI2OH))
            END IF
C
C           Ni(OH)3.-
            IF (SCAL(INODE,1,INIOH3).EQ.0.0) THEN
              X(11) = -1000.0
            ELSE
              X(11) = LOG10(SCAL(INODE,1,INIOH3))
            END IF
C
C           calculate total component concentrations (excluding solid species; used in
C           subroutine CALFUN to close mass balance) [kmol.m-3]
C           H.+
            CC(1) = (10.0**X(1))-(10.0**X(5))-(2.0*(10.0**X(6)))-(10.0
     &              **X(9))-(10.0**X(10))-(3.0*(10.0**X(11)))
C           Ni.2+
            CC(2) = (10.0**X(2))+(10.0**X(6))+(10.0**X(7))+(10.0**X(8))
     &              +(10.0**X(9))+(2.0*(10.0**X(10)))+(10.0**X(11))
C           Na.+
            CC(3) = 10.0**X(3)
C           Cl.-
            CC(4) = (10.0**X(4))+(2.0*(10.0**X(7)))+(10.0**X(8))
C
          END IF
C
C         calculate logarithm of current Ni(OH)2 solid concentration (supplied to
C         subroutine NS11AD and returned unchanged) [log(kmol.m-3)]
          IF (SCAL(INODE,1,IPPT).GT.0.0) THEN
            X(12) = LOG10(SCAL(INODE,1,IPPT))
          ELSE
            X(12) = -1000.0
          END IF
C
C         specify input parameters required by subroutine NS11AD
C         number of equations; number of unknowns (species concentrations)
          N = 12
C         X-step used to calculate estimates of Jacobian (partial derivatives) matrix
C         numerically
          STEP = 1.0E-12
C         generous estimate of 'distance' between initial approximation and required
C         solution
          STPMAX = 5.0
C         maximum number of calls of subroutine CALFUN
```

```
      MAXFUN = 500000000
C     no printing
      IPRINT = 0
C     required accuracy (maximum sum of square errors)
      ACC = 1.0E-20
C
C     exclude solid species from equilibrium calculation
      SPECSET1 = 1
C
C     if nickel present in solution
      IF (CC(2).GT.0.0) THEN
C       include nickel species in equilibrium calculation
        SPECSET2 = 1
C     if no nickel present in solution
      ELSE
C       exclude nickel species from equilibrium calculation
        SPECSET2 = 2
      END IF
C
C     call subroutine NS11AD to resolve equilibrium state of system
      CALL NS11AD(N,X,F,A,STEP,STPMAX,ACC,MAXFUN,IPRINT,WO,IW)
C
C     calculate species activities
      DO 70 I = 1,11
        ACTIV(I,INODE) = 10.0**LAS(I)
 70   CONTINUE
C
C     calculate supersaturation
      SCAL(INODE,1,ISUPER) = ((ACTIV(2,INODE))*((ACTIV(5,INODE))**
     &                        2.0))/KSP
C     calculate pH
      SCAL(INODE,1,IPH) = -LAS(1)
C     calculate ionic strength [kmol.m-3]
      SCAL(INODE,1,ISTION) = STION
C     calculate Ni.2+ activity
      SCAL(INODE,1,IANI) = ACTIV(2,INODE)
C     calculate total dissolved nickel concentration [kmol.m-3]
      SCAL(INODE,1,ITDNC) = (10.0**X(2))+(10.0**X(6))+(10.0**X(7))+
     &                      (10.0**X(8))+(10.0**X(9))+(2.0*(10.0**
     &                      X(10)))+(10.0**X(11))
C
C   at end of each time step
    ELSE
C
C     if dilution effect distributed
      IF (GENSET3.EQ.1) THEN
C
C       calculate diluted particle number concentrations [m-3]
        DO 80 I = 1,NOPSI
          PN(I,INODE) = PN(I,INODE)*(TOTALVOL-(FEEDVOL*DT))/
     &                  TOTALVOL
 80     CONTINUE
C
C       estimate logarithm of equilibrium species concentrations as logarithm of
C       current diluted species concentrations (supplied as initial estimate to
C       subroutine NS11AD where sufficient) [log(kmol.m-3)]
C       H.+
        X(1) = LOG10(SCAL(INODE,1,IH)*(TOTALVOL-(FEEDVOL*DT))/
     &         TOTALVOL)
C
C       Ni.2+
        IF (SCAL(INODE,1,INI).EQ.0.0) THEN
          X(2) = -1000.0
        ELSE
          X(2) = LOG10(SCAL(INODE,1,INI)*(TOTALVOL-(FEEDVOL*DT))/
     &           TOTALVOL)
        END IF
C
C       Na.+
        IF (SCAL(INODE,1,INA).EQ.0.0) THEN
          X(3) = -1000.0
        ELSE
          X(3) = LOG10(SCAL(INODE,1,INA)*(TOTALVOL-(FEEDVOL*DT))/
     &           TOTALVOL)
        END IF
C
C       Cl.-
```

```
            X(4) = LOG10(SCAL(INODE,1,ICL)*(TOTALVOL-(FEEDVOL*DT))/
     &             TOTALVOL)
C           OH.-
            X(5) = LOG10(SCAL(INODE,1,IOH)*(TOTALVOL-(FEEDVOL*DT))/
     &             TOTALVOL)
C
C           Ni(OH)2 aqueous
            IF (SCAL(INODE,1,INIOH2).EQ.0.0) THEN
              X(6) = -1000.0
            ELSE
              X(6) = LOG10(SCAL(INODE,1,INIOH2)*(TOTALVOL-(FEEDVOL*DT))
     &               /TOTALVOL)
            END IF
C
C           NiCl2 aqueous
            IF (SCAL(INODE,1,INICL2).EQ.0.0) THEN
              X(7) = -1000.0
            ELSE
              X(7) = LOG10(SCAL(INODE,1,INICL2)*(TOTALVOL-(FEEDVOL*DT))
     &               /TOTALVOL)
            END IF
C
C           NiCl.+
            IF (SCAL(INODE,1,INICL).EQ.0.0) THEN
              X(8) = -1000.0
            ELSE
              X(8) = LOG10(SCAL(INODE,1,INICL)*(TOTALVOL-(FEEDVOL*DT))/
     &               TOTALVOL)
            END IF
C
C           NiOH.+
            IF (SCAL(INODE,1,INIOH).EQ.0.0) THEN
              X(9) = -1000.0
            ELSE
              X(9) = LOG10(SCAL(INODE,1,INIOH)*(TOTALVOL-(FEEDVOL*DT))/
     &               TOTALVOL)
            END IF
C
C           Ni2OH.3+
            IF (SCAL(INODE,1,INI2OH).EQ.0.0) THEN
              X(10) = -1000.0
            ELSE
              X(10) = LOG10(SCAL(INODE,1,INI2OH)*(TOTALVOL-(FEEDVOL*DT)
     &                )/TOTALVOL)
            END IF
C
C           Ni(OH)3.-
            IF (SCAL(INODE,1,INIOH3).EQ.0.0) THEN
              X(11) = -1000.0
            ELSE
              X(11) = LOG10(SCAL(INODE,1,INIOH3)*(TOTALVOL-(FEEDVOL*DT)
     &                )/TOTALVOL)
            END IF
C
C         if dilution effect isolated
          ELSE IF (GENSET3.EQ.2) THEN
C
C           within feed cell(s)
            IF (FEED(INODE).EQ.2) THEN
C
C             calculate diluted particle number concentrations [m-3]
              DO 90 I = 1,NOPSI
                PN(I,INODE) = PN(I,INODE)*((VOL(INODE)*INITVOL/GEOMVOL)
     &                        +(VOL(INODE)/FEEDPVOL)*((TOTALVOL-INITVOL
     &                        )-(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/
     &                        GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL-
     &                        INITVOL))
 90           CONTINUE
C
C             estimate logarithm of equilibrium species concentrations as logarithm of
C             current diluted species concentrations (supplied as initial estimate to
C             subroutine NS11AD where sufficient) [log(kmol.m-3)]
C             H.+
              X(1) = LOG10(SCAL(INODE,1,IH)*((VOL(INODE)*INITVOL/
     &               GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-INITVOL)
     &               -(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/GEOMVOL)+(
     &               VOL(INODE)/FEEDPVOL)*(TOTALVOL-INITVOL)))
```

```
C
C             Ni.2+
              IF (SCAL(INODE,1,INI).EQ.0.0) THEN
                X(2) = -1000.0
              ELSE
                X(2) = LOG10(SCAL(INODE,1,INI)*((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-
     &                 INITVOL)-(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL-INITVOL
     &                 )))
              END IF
C
C             Na.+
              IF (SCAL(INODE,1,INA).EQ.0.0) THEN
                X(3) = -1000.0
              ELSE
                X(3) = LOG10(SCAL(INODE,1,INA)*((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-
     &                 INITVOL)-(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL-INITVOL
     &                 )))
              END IF
C
C             Cl.-
              X(4) = LOG10(SCAL(INODE,1,ICL)*((VOL(INODE)*INITVOL/
     &               GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-INITVOL)
     &               -(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/GEOMVOL)+(
     &               VOL(INODE)/FEEDPVOL)*(TOTALVOL-INITVOL)))
C             OH.-
              X(5) = LOG10(SCAL(INODE,1,IOH)*((VOL(INODE)*INITVOL/
     &               GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-INITVOL)
     &               -(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/GEOMVOL)+(
     &               VOL(INODE)/FEEDPVOL)*(TOTALVOL-INITVOL)))
C
C             Ni(OH)2 aqueous
              IF (SCAL(INODE,1,INIOH2).EQ.0.0) THEN
                X(6) = -1000.0
              ELSE
                X(6) = LOG10(SCAL(INODE,1,INIOH2)*((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-
     &                 INITVOL)-(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL-INITVOL
     &                 )))
              END IF
C
C             NiCl2 aqueous
              IF (SCAL(INODE,1,INICL2).EQ.0.0) THEN
                X(7) = -1000.0
              ELSE
                X(7) = LOG10(SCAL(INODE,1,INICL2)*((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-
     &                 INITVOL)-(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL-INITVOL
     &                 )))
              END IF
C
C             NiCl.+
              IF (SCAL(INODE,1,INICL).EQ.0.0) THEN
                X(8) = -1000.0
              ELSE
                X(8) = LOG10(SCAL(INODE,1,INICL)*((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-
     &                 INITVOL)-(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL-INITVOL
     &                 )))
              END IF
C
C             NiOH.+
              IF (SCAL(INODE,1,INIOH).EQ.0.0) THEN
                X(9) = -1000.0
              ELSE
                X(9) = LOG10(SCAL(INODE,1,INIOH)*((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-
     &                 INITVOL)-(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/
     &                 GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL-INITVOL
     &                 )))
              END IF
```

```
C
C             Ni2OH.3+
              IF (SCAL(INODE,1,INI2OH).EQ.0.0) THEN
                X(10) = -1000.0
              ELSE
                X(10) = LOG10(SCAL(INODE,1,INI2OH)*((VOL(INODE)*INITVOL
     &                  /GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-
     &                  INITVOL)-(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/
     &                  GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL-
     &                  INITVOL)))
              END IF
C
C             Ni(OH)3.-
              IF (SCAL(INODE,1,INIOH3).EQ.0.0) THEN
                X(11) = -1000.0
              ELSE
                X(11) = LOG10(SCAL(INODE,1,INIOH3)*((VOL(INODE)*INITVOL
     &                  /GEOMVOL)+(VOL(INODE)/FEEDPVOL)*((TOTALVOL-
     &                  INITVOL)-(FEEDVOL*DT)))/((VOL(INODE)*INITVOL/
     &                  GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL-
     &                  INITVOL)))
              END IF
C
C           within each remaining cell in flow domain
            ELSE
C
C             estimate logarithm of equilibrium species concentrations as logarithm of
C             current species concentrations (supplied as initial estimate to
C             subroutine NS11AD where sufficient) [log(kmol.m-3)]
C             H.+
              X(1) = LOG10(SCAL(INODE,1,IH))
C
C             Ni.2+
              IF (SCAL(INODE,1,INI).EQ.0.0) THEN
                X(2) = -1000.0
              ELSE
                X(2) = LOG10(SCAL(INODE,1,INI))
              END IF
C
C             Na.+
              IF (SCAL(INODE,1,INA).EQ.0.0) THEN
                X(3) = -1000.0
              ELSE
                X(3) = LOG10(SCAL(INODE,1,INA))
              END IF
C
C             Cl.-
              X(4) = LOG10(SCAL(INODE,1,ICL))
C             OH.-
              X(5) = LOG10(SCAL(INODE,1,IOH))
C
C             Ni(OH)2 aqueous
              IF (SCAL(INODE,1,INIOH2).EQ.0.0) THEN
                X(6) = -1000.0
              ELSE
                X(6) = LOG10(SCAL(INODE,1,INIOH2))
              END IF
C
C             NiCl2 aqueous
              IF (SCAL(INODE,1,INICL2).EQ.0.0) THEN
                X(7) = -1000.0
              ELSE
                X(7) = LOG10(SCAL(INODE,1,INICL2))
              END IF
C
C             NiCl.+
              IF (SCAL(INODE,1,INICL).EQ.0.0) THEN
                X(8) = -1000.0
              ELSE
                X(8) = LOG10(SCAL(INODE,1,INICL))
              END IF
C
C             NiOH.+
              IF (SCAL(INODE,1,INIOH).EQ.0.0) THEN
                X(9) = -1000.0
              ELSE
                X(9) = LOG10(SCAL(INODE,1,INIOH))
```

```
              END IF
C
C             Ni2OH.3+
              IF (SCAL(INODE,1,INI2OH).EQ.0.0) THEN
                X(10) = -1000.0
              ELSE
                X(10) = LOG10(SCAL(INODE,1,INI2OH))
              END IF
C
C             Ni(OH)3.-
              IF (SCAL(INODE,1,INIOH3).EQ.0.0) THEN
                X(11) = -1000.0
              ELSE
                X(11) = LOG10(SCAL(INODE,1,INIOH3))
              END IF
C
            END IF
C
C           if dilution effect isolated, then distributed and teamed with time-dependent
C           feed source term
            ELSE IF (GENSET3.EQ.3) THEN
C
C             within feed cell(s)
              IF (FEED(INODE).EQ.2) THEN
C
C               calculate diluted particle number concentrations [m-3]
                DO 100 I = 1,NOPSI
                  PN(I,INODE) = PN(I,INODE)*((VOL(INODE)/GEOMVOL)*(
     &                        TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/
     &                        GEOMVOL)*(TOTALVOL-FEEDVOL*DT)+(
     &                        VOL(INODE)/FEEDPVOL)*(FEEDVOL*DT))
 100            CONTINUE
C
C               estimate logarithm of equilibrium species concentrations as logarithm of
C               current diluted species concentrations (supplied as initial estimate to
C               subroutine NS11AD where sufficient) [log(kmol.m-3)]
C               H.+
                X(1) = LOG10(SCAL(INODE,1,IH)*((VOL(INODE)/GEOMVOL)*(
     &                 TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*(
     &                 TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                 FEEDVOL*DT)))
C
C               Ni.2+
                IF (SCAL(INODE,1,INI).EQ.0.0) THEN
                  X(2) = -1000.0
                ELSE
                  X(2) = LOG10(SCAL(INODE,1,INI)*((VOL(INODE)/GEOMVOL)*(
     &                   TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*(
     &                   TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                   FEEDVOL*DT)))
                END IF
C
C               Na.+
                IF (SCAL(INODE,1,INA).EQ.0.0) THEN
                  X(3) = -1000.0
                ELSE
                  X(3) = LOG10(SCAL(INODE,1,INA)*((VOL(INODE)/GEOMVOL)*(
     &                   TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*(
     &                   TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                   FEEDVOL*DT)))
                END IF
C
C               Cl.-
                X(4) = LOG10(SCAL(INODE,1,ICL)*((VOL(INODE)/GEOMVOL)*(
     &                 TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*(
     &                 TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                 FEEDVOL*DT)))
C               OH.-
                X(5) = LOG10(SCAL(INODE,1,IOH)*((VOL(INODE)/GEOMVOL)*(
     &                 TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*(
     &                 TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                 FEEDVOL*DT)))
C
C               Ni(OH)2 aqueous
                IF (SCAL(INODE,1,INIOH2).EQ.0.0) THEN
                  X(6) = -1000.0
                ELSE
```

```
                X(6) = LOG10(SCAL(INODE,1,INIOH2)*((VOL(INODE)/GEOMVOL)
     &                 *(TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*(
     &                 TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                 FEEDVOL*DT)))
              END IF
C
C             NiCl2 aqueous
              IF (SCAL(INODE,1,INICL2).EQ.0.0) THEN
                X(7) = -1000.0
              ELSE
                X(7) = LOG10(SCAL(INODE,1,INICL2)*((VOL(INODE)/GEOMVOL)
     &                 *(TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*(
     &                 TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                 FEEDVOL*DT)))
              END IF
C
C             NiCl.+
              IF (SCAL(INODE,1,INICL).EQ.0.0) THEN
                X(8) = -1000.0
              ELSE
                X(8) = LOG10(SCAL(INODE,1,INICL)*((VOL(INODE)/GEOMVOL)*
     &                 (TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*(
     &                 TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                 FEEDVOL*DT)))
              END IF
C
C             NiOH.+
              IF (SCAL(INODE,1,INIOH).EQ.0.0) THEN
                X(9) = -1000.0
              ELSE
                X(9) = LOG10(SCAL(INODE,1,INIOH)*((VOL(INODE)/GEOMVOL)*
     &                 (TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*(
     &                 TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                 FEEDVOL*DT)))
              END IF
C
C             Ni2OH.3+
              IF (SCAL(INODE,1,INI2OH).EQ.0.0) THEN
                X(10) = -1000.0
              ELSE
                X(10) = LOG10(SCAL(INODE,1,INI2OH)*((VOL(INODE)/GEOMVOL
     &                  )*(TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*
     &                  (TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                  FEEDVOL*DT)))
              END IF
C
C             Ni(OH)3.-
              IF (SCAL(INODE,1,INIOH3).EQ.0.0) THEN
                X(11) = -1000.0
              ELSE
                X(11) = LOG10(SCAL(INODE,1,INIOH3)*((VOL(INODE)/GEOMVOL
     &                  )*(TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*
     &                  (TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                  FEEDVOL*DT)))
              END IF
C
C           within each remaining cell in flow domain
            ELSE
C
C             estimate logarithm of equilibrium species concentrations as logarithm of
C             current species concentrations (supplied as initial estimate to
C             subroutine NS11AD where sufficient) [log(kmol.m-3)]
C             H.+
              X(1) = LOG10(SCAL(INODE,1,IH))
C
C             Ni.2+
              IF (SCAL(INODE,1,INI).EQ.0.0) THEN
                X(2) = -1000.0
              ELSE
                X(2) = LOG10(SCAL(INODE,1,INI))
              END IF
C
C             Na.+
              IF (SCAL(INODE,1,INA).EQ.0.0) THEN
                X(3) = -1000.0
              ELSE
                X(3) = LOG10(SCAL(INODE,1,INA))
```

```
              END IF
C
C             Cl.-
              X(4) = LOG10(SCAL(INODE,1,ICL))
C             OH.-
              X(5) = LOG10(SCAL(INODE,1,IOH))
C
C             Ni(OH)2 aqueous
              IF (SCAL(INODE,1,INIOH2).EQ.0.0) THEN
                X(6) = -1000.0
              ELSE
                X(6) = LOG10(SCAL(INODE,1,INIOH2))
              END IF
C
C             NiCl2 aqueous
              IF (SCAL(INODE,1,INICL2).EQ.0.0) THEN
                X(7) = -1000.0
              ELSE
                X(7) = LOG10(SCAL(INODE,1,INICL2))
              END IF
C
C             NiCl.+
              IF (SCAL(INODE,1,INICL).EQ.0.0) THEN
                X(8) = -1000.0
              ELSE
                X(8) = LOG10(SCAL(INODE,1,INICL))
              END IF
C
C             NiOH.+
              IF (SCAL(INODE,1,INIOH).EQ.0.0) THEN
                X(9) = -1000.0
              ELSE
                X(9) = LOG10(SCAL(INODE,1,INIOH))
              END IF
C
C             Ni2OH.3+
              IF (SCAL(INODE,1,INI2OH).EQ.0.0) THEN
                X(10) = -1000.0
              ELSE
                X(10) = LOG10(SCAL(INODE,1,INI2OH))
              END IF
C
C             Ni(OH)3.-
              IF (SCAL(INODE,1,INIOH3).EQ.0.0) THEN
                X(11) = -1000.0
              ELSE
                X(11) = LOG10(SCAL(INODE,1,INIOH3))
              END IF
C
            END IF
C
          END IF
C
C         calculate initial 3rd moment [micron3.m-3]
          MOM3I(INODE) = 0.0
C
          DO 110 I = 1,NOPSI
            MOM3I(INODE) = MOM3I(INODE)+((PAVE3(I)**3.0)*PN(I,INODE))
 110      CONTINUE
C
C         calculate precipitate yield [kmol.m-3]
          SCAL(INODE,1,IPPT) = (PI*MOM3I(INODE)/6.0)*RHO*1.0E-18
C
C         calculate total component concentrations (excluding solid species; used in
C         subroutine CALFUN to close mass balance) [kmol.m-3]
C         H.+
          CC(1) = (10.0**X(1))-(10.0**X(5))-(2.0*(10.0**X(6)))-(10.0**
     &              X(9))-(10.0**X(10))-(3.0*(10.0**X(11)))
C         Ni.2+
          CC(2) = (10.0**X(2))+(10.0**X(6))+(10.0**X(7))+(10.0**X(8))+(
     &              10.0**X(9))+(2.0*(10.0**X(10)))+(10.0**X(11))
C         Na.+
          CC(3) = 10.0**X(3)
C         Cl.-
          CC(4) = (10.0**X(4))+(2.0*(10.0**X(7)))+(10.0**X(8))
C
C         calculate logarithm of current Ni(OH)2 solid concentration (supplied to
```

```
C         subroutine NS11AD and returned unchanged) [log(kmol.m-3)]
          IF (SCAL(INODE,1,IPPT).GT.0.0) THEN
            X(12) = LOG10(SCAL(INODE,1,IPPT))
          ELSE
            X(12) = -1000.0
          END IF
C
C         estimate logarithm of equilibrium species concentrations as final
C         equilibrium values at previous time step (supplied as initial estimate to
C         subroutine NS11AD where necessary) [log(kmol.m-3)]
C         H.+
          X(1) = SCAL(INODE,1,IS1)
C         Ni.2+
          X(2) = SCAL(INODE,1,IS2)
C         Na.+
C          X(3) = SCAL(INODE,1,IS3)
C         Cl.-
          X(4) = SCAL(INODE,1,IS4)
C         OH.-
          X(5) = SCAL(INODE,1,IS5)
C         Ni(OH)2 aqueous
          X(6) = SCAL(INODE,1,IS6)
C         NiCl2 aqueous
          X(7) = SCAL(INODE,1,IS7)
C         NiCl.+
          X(8) = SCAL(INODE,1,IS8)
C         NiOH.+
          X(9) = SCAL(INODE,1,IS9)
C         Ni2OH.3+
          X(10) = SCAL(INODE,1,IS10)
C         Ni(OH)3.-
          X(11) = SCAL(INODE,1,IS11)
C
C         specify input parameters required by subroutine NS11AD
C         number of equations; number of unknowns (species concentrations)
          N = 12
C         X-step used to calculate estimates of Jacobian (partial derivatives) matrix
C         numerically
          STEP = 1.0E-12
C         generous estimate of 'distance' between initial approximation and required
C         solution
          STPMAX = 5.0
C         maximum number of calls of subroutine CALFUN
          MAXFUN = 500000000
C         no printing
          IPRINT = 0
C         required accuracy (maximum sum of square errors)
          ACC = 1.0E-20
C
C         exclude solid species from equilibrium calculation
          SPECSET1 = 1
C
C         if nickel present in solution
          IF (CC(2).GT.0.0) THEN
C           include nickel species in equilibrium calculation
            SPECSET2 = 1
C         if no nickel present in solution
          ELSE
C           exclude nickel species from equilibrium calculation
            SPECSET2 = 2
          END IF
C
C         call subroutine NS11AD to resolve equilibrium state of system
          CALL NS11AD(N,X,F,A,STEP,STPMAX,ACC,MAXFUN,IPRINT,WO,IW)
C
C         calculate species activities
          DO 120 I = 1,11
            ACTIV(I,INODE) = 10.0**LAS(I)
 120      CONTINUE
C
C         calculation supersaturation
          SCAL(INODE,1,ISUPER) = ((ACTIV(2,INODE))*((ACTIV(5,INODE))**
     &                            2.0))/KSP
C         calculate pH
          SCAL(INODE,1,IPH) = -LAS(1)
C         calculate ionic strength [kmol.m-3]
          SCAL(INODE,1,ISTION) = STION
```

```
C         calculate Ni.2+ activity
          SCAL(INODE,1,IANI) = ACTIV(2,INODE)
C         calculate total dissolved nickel concentration [kmol.m-3]
          SCAL(INODE,1,ITDNC) = (10.0**X(2))+(10.0**X(6))+(10.0**X(7))+
     &                          (10.0**X(8))+(10.0**X(9))+(2.0*(10.0**
     &                          X(10)))+(10.0**X(11))
C
C         if aggregation-related calculations performed before effects of nucleation
C         established
          IF (GENSET2.EQ.2) THEN
C
C           calculate size-independent portion of aggregation kernel (units dependent
C           upon aggregation kernel)
            AR(INODE) = 8.33E-78*EXP((14.0*SCAL(INODE,1,IPH))+(1050.0*
     &                  SCAL(INODE,1,ISTION)))
C
          END IF
C
C         if solution supersaturated
          IF (SCAL(INODE,1,ISUPER).GT.1.0) THEN
C
C           if equilibrium description of nucleation employed
            IF (GENSET1.EQ.1) THEN
C
C             calculate total component concentrations (including solid species; used
C             in subroutine CALFUN to close mass balance) [kmol.m-3]
C             H.+
              CC(1) = (10.0**X(1))-(10.0**X(5))-(2.0*(10.0**X(6)))-(
     &                10.0**X(9))-(10.0**X(10))-(3.0*(10.0**X(11)))-(
     &                2.0*(10.0**X(12)))
C             Ni.2+
              CC(2) = (10.0**X(2))+(10.0**X(6))+(10.0**X(7))+(10.0**
     &                X(8))+(10.0**X(9))+(2.0*(10.0**X(10)))+(10.0**
     &                X(11))+(10.0**X(12))
C             Na.+
              CC(3) = 10.0**X(3)
C             Cl.-
              CC(4) = (10.0**X(4))+(2.0*(10.0**X(7)))+(10.0**X(8))
C
C             estimate logarithm of equilibrium Ni(OH)2 solid concentration if no
C             solid currently present (supplied as initial estimate to subroutine
C             NS11AD) [log(kmol.m-3)]
              IF (SCAL(INODE,1,IPPT).EQ.0.0) X(12) = -6.0
C
C             specify input parameters required by subroutine NS11AD
C             number of equations; number of unknowns (species concentrations)
              N = 12
C             X-step used to calculate estimates of Jacobian (partial derivatives)
C             matrix numerically
              STEP = 1.0E-12
C             generous estimate of 'distance' between initial approximation and
C             required solution
              STPMAX = 5.0
C             maximum number of calls of subroutine CALFUN
              MAXFUN = 500000000
C             no printing
              IPRINT = 0
C             required accuracy (maximum sum of square errors)
              ACC = 1.0E-20
C
C             include solid species in equilibrium calculation
              SPECSET1 = 2
C
C             if nickel present in solution
              IF (CC(2).GT.0.0) THEN
C               include nickel species in equilibrium calculation
                SPECSET2 = 1
C             if no nickel present in solution
              ELSE
C               exclude nickel species from equilibrium calculation
                SPECSET2 = 2
              END IF
C
C             call subroutine NS11AD to resolve equilibrium state of system
              CALL NS11AD(N,X,F,A,STEP,STPMAX,ACC,MAXFUN,IPRINT,WO,IW)
C
C             calculate species activities
```

```
              DO 130 I = 1,11
                ACTIV(I,INODE) = 10.0**LAS(I)
 130          CONTINUE
C
C             calculate supersaturation
              SCAL(INODE,1,ISUPER) = ((ACTIV(2,INODE))*((ACTIV(5,INODE)
     &                               )**2.0))/KSP
C             calculate pH
              SCAL(INODE,1,IPH) = -LAS(1)
C             calculate ionic strength [kmol.m-3]
              SCAL(INODE,1,ISTION) = STION
C             calculate Ni.2+ activity
              SCAL(INODE,1,IANI) = ACTIV(2,INODE)
C             calculate total dissolved nickel concentration [kmol.m-3]
              SCAL(INODE,1,ITDNC) = (10.0**X(2))+(10.0**X(6))+(10.0**
     &                             X(7))+(10.0**X(8))+(10.0**X(9))+(
     &                             2.0*(10.0**X(10)))+(10.0**X(11))
C
C             calculate nucleation rate corresponding to equilibrium solute deposition
C             [m-3.s-1]
              BIRTH(INODE) = (((10.0**X(12))-SCAL(INODE,1,IPPT))*
     &                       1.0E+18)/(RHO*(PI*(PAVE3(1)**3.0)/6.0)*DT)
C             calculate change in precipitate yield [kmol.m-3]
              DPPT(INODE) = BIRTH(INODE)*DT*(PI*(PAVE3(1)**3.0)/6.0)*
     &                      1.0E-18*RHO
C
C           if kinetic description of nucleation employed
            ELSE IF (GENSET1.EQ.2) THEN
C
C             calculate nucleation rate [m-3.s-1]
              BIRTH(INODE) = ((1.265E-5*((SCAL(INODE,1,ISUPER)-1.0)**
     &                       0.77))*1.0E+18)/(RHO*(PI*(PAVE3(1)**3.0)/
     &                       6.0)*DT)
C             calculate change in precipitate yield [kmol.m-3]
              DPPT(INODE) = BIRTH(INODE)*DT*(PI*(PAVE3(1)**3.0)/6.0)*
     &                      1.0E-18*RHO
C
C             if change in precipitate yield greater than total dissolved nickel
C             concentration (i.e. extent of nucleation exceeds maximum allowable)
              IF (DPPT(INODE).GT.SCAL(INODE,1,ITDNC)) THEN
C               calculate adjusted change in precipitate yield [kmol.m-3]
                DPPT(INODE) = SCAL(INODE,1,ITDNC)
C               calculate adjusted nucleation rate [m-3.s-1]
                BIRTH(INODE) = (DPPT(INODE)*1.0E+18)/(RHO*(PI*(PAVE3(1)
     &                         **3.0)/6.0)*DT)
              END IF
C
            END IF
C
C           calculate kinetics time scale (estimated as inverse of approximate rate of
C           change of dissolved nickel mass fraction; inspired by corresponding use of
C           fuel mass fraction in eddy break-up combustion model) [s]
            SCAL(INODE,1,ITK) = 1.0/(((DPPT(INODE)*58.71)/DEN(INODE,1))
     &                          /DT)
C
C           if turbulent mixing time scale greater than kinetics time scale (i.e.
C           reaction rate controlled by turbulent mixing rate)
            IF (SCAL(INODE,1,ITM).GT.SCAL(INODE,1,ITK)) THEN
C
C             write output to file fort.99
              WRITE(99,*) 'INFO: TURBULENT MIXING IS RATE LIMITING'
              WRITE(99,*) 'CELL',INODE
C
C             calculate adjusted nucleation rate [m-3.s-1]
              BIRTH(INODE) = BIRTH(INODE)*(SCAL(INODE,1,ITK)/
     &                       SCAL(INODE,1,ITM))
C             calculate adjusted change in precipitate yield [kmol.m-3]
              DPPT(INODE) = DPPT(INODE)*(SCAL(INODE,1,ITK)/
     &                      SCAL(INODE,1,ITM))
C
C             if equilibrium description of nucleation employed
              IF (GENSET1.EQ.1) THEN
C
C               calculate adjusted total component concentrations (excluding solid
C               species; used in subroutine CALFUN to close mass balance) [kmol.m-3]
C               H.+
                CC(1) = (10.0**X(1))-(10.0**X(5)+(2.0*DPPT(INODE)*((
```

```
     &                SCAL(INODE,1,ITM)/SCAL(INODE,1,ITK))-1.0)))-(
     &                2.0*(10.0**X(6)))-(10.0**X(9))-(10.0**X(10))-(
     &                3.0*(10.0**X(11)))
C               Ni.2+
                CC(2) = (10.0**X(2)+(DPPT(INODE)*((SCAL(INODE,1,ITM)/
     &                SCAL(INODE,1,ITK))-1.0)))+(10.0**X(6))+(10.0**
     &                X(7))+(10.0**X(8))+(10.0**X(9))+(2.0*(10.0**
     &                X(10)))+(10.0**X(11))
C               Na.+
                CC(3) = 10.0**X(3)
C               Cl.-
                CC(4) = (10.0**X(4))+(2.0*(10.0**X(7)))+(10.0**X(8))
C
C               calculate logarithm of current reduced Ni(OH)2 solid concentration (
C               supplied to subroutine NS11AD and returned unchanged) [log(kmol.m-3)]
                X(12) = LOG10(10.0**X(12)-(DPPT(INODE)*((
     &                SCAL(INODE,1,ITM)/SCAL(INODE,1,ITK))-1.0)))
C
C               estimate logarithm of equilibrium species concentrations as logarithm
C               of current augmented species concentrations (supplied as initial
C               estimate to subroutine NS11AD) [log(kmol.m-3)]
C               Ni.2+
                IF ((10.0**X(2)+(DPPT(INODE)*((SCAL(INODE,1,ITM)/
     &             SCAL(INODE,1,ITK))-1.0))).GT.0.0) X(2) = LOG10(10.0
     &             **X(2)+(DPPT(INODE)*((SCAL(INODE,1,ITM)/
     &             SCAL(INODE,1,ITK))-1.0)))
C               OH.-
                IF ((10.0**X(5)+(2.0*DPPT(INODE)*((SCAL(INODE,1,ITM)/
     &             SCAL(INODE,1,ITK))-1.0))).GT.0.0) X(5) = LOG10(10.0
     &             **X(5)+(2.0*DPPT(INODE)*((SCAL(INODE,1,ITM)/
     &             SCAL(INODE,1,ITK))-1.0)))
C
C               specify input parameters required by subroutine NS11AD
C               number of equations; number of unknowns (species concentrations)
                N = 12
C               X-step used to calculate estimates of Jacobian (partial derivatives)
C               matrix numerically
                STEP = 1.0E-12
C               generous estimate of 'distance' between initial approximation and
C               required solution
                STPMAX = 5.0
C               maximum number of calls of subroutine CALFUN
                MAXFUN = 500000000
C               no printing
                IPRINT = 0
C               required accuracy (maximum sum of square errors)
                ACC = 1.0E-20
C
C               exclude solid species from equilibrium calculation
                SPECSET1 = 1
C
C               if nickel present in solution
                IF (CC(2).GT.0.0) THEN
C                 include nickel species in equilibrium calculation
                  SPECSET2 = 1
C               if no nickel present in solution
                ELSE
C                 exclude nickel species from equilibrium calculation
                  SPECSET2 = 2
                END IF
C
C               call subroutine NS11AD to resolve equilibrium state of system
                CALL NS11AD(N,X,F,A,STEP,STPMAX,ACC,MAXFUN,IPRINT,WO,IW
     &               )
C
C               calculate species activities
                DO 140 I = 1,11
                  ACTIV(I,INODE) = 10.0**LAS(I)
 140            CONTINUE
C
C               calculate supersaturation
                SCAL(INODE,1,ISUPER) = ((ACTIV(2,INODE))*((
     &                                  ACTIV(5,INODE))**2.0))/KSP
C               calculate pH
                SCAL(INODE,1,IPH) = -LAS(1)
C               calculate ionic strength [kmol.m-3]
                SCAL(INODE,1,ISTION) = STION
```

```
C               calculate Ni.2+ activity
                SCAL(INODE,1,IANI) = ACTIV(2,INODE)
C               calculate total dissolved nickel concentration [kmol.m-3]
                SCAL(INODE,1,ITDNC) = (10.0**X(2))+(10.0**X(6))+(10.0**
     &                               X(7))+(10.0**X(8))+(10.0**X(9))+(
     &                               2.0*(10.0**X(10)))+(10.0**X(11))
C
              END IF
C
            END IF
C
C           if kinetic description of nucleation employed
            IF (GENSET1.EQ.2) THEN
C
C             calculate adjusted total component concentrations (excluding solid
C             species; used in subroutine CALFUN to close mass balance) [kmol.m-3]
C             H.+
              CC(1) = (10.0**X(1))-(10.0**X(5)-(2.0*DPPT(INODE)))-(2.0
     &                *(10.0**X(6)))-(10.0**X(9))-(10.0**X(10))-(3.0*(
     &                10.0**X(11)))
C             Ni.2+
              CC(2) = (10.0**X(2)-DPPT(INODE))+(10.0**X(6))+(10.0**X(7)
     &                )+(10.0**X(8))+(10.0**X(9))+(2.0*(10.0**X(10)))+(
     &                10.0**X(11))
C             Na.+
              CC(3) = 10.0**X(3)
C             Cl.-
              CC(4) = (10.0**X(4))+(2.0*(10.0**X(7)))+(10.0**X(8))
C
C             calculate logarithm of current augmented Ni(OH)2 solid concentration
C             (supplied to subroutine NS11AD and returned unchanged) [log(kmol.m-3)]
              X(12) = LOG10(10.0**X(12)+DPPT(INODE))
C
C             estimate logarithm of equilibrium species concentrations as logarithm of
C             current reduced species concentrations (supplied as initial estimate to
C             subroutine NS11AD) [log(kmol.m-3)]
C             Ni.2+
              IF ((10.0**X(2)-DPPT(INODE)).GT.0.0) X(2) = LOG10(10.0**
     &           X(2)-DPPT(INODE))
C             OH.-
              IF ((10.0**X(5)-(2.0*DPPT(INODE))).GT.0.0) X(5) = LOG10(
     &           10.0**X(5)-(2.0*DPPT(INODE)))
C
C             if no nickel present in solution
              IF (CC(2).EQ.0.0) THEN
C               specify logarithm of 'zero' species concentrations (supplied to
C               subroutine NS11AD and returned unchanged) [log(kmol.m-3)]
C               Ni.2+
                X(2) = -1000.0
C               Ni(OH)2 aqueous
                X(6) = -1000.0
C               NiCl2 aqueous
                X(7) = -1000.0
C               NiCl.+
                X(8) = -1000.0
C               NiOH.+
                X(9) = -1000.0
C               Ni2OH.3+
                X(10) = -1000.0
C               Ni(OH)3.-
                X(11) = -1000.0
C               write output to file fort.99
                WRITE(99,*) 'INFO: NO NICKEL REMAINING IN SOLUTION'
                WRITE(99,*) 'CELL',INODE
              END IF
C
C             specify input parameters required by subroutine NS11AD
C             number of equations; number of unknowns (species concentrations)
              N = 12
C             X-step used to calculate estimates of Jacobian (partial derivatives)
C             matrix numerically
              STEP = 1.0E-12
C             generous estimate of 'distance' between initial approximation and
C             required solution
              STPMAX = 5.0
C             maximum number of calls of subroutine CALFUN
              MAXFUN = 500000000
```

```
C             no printing
              IPRINT = 0
C             required accuracy (maximum sum of square errors)
              ACC = 1.0E-20
C
C             exclude solid species from equilibrium calculation
              SPECSET1 = 1
C
C             if nickel present in solution
              IF (CC(2).GT.0.0) THEN
C               include nickel species in equilibrium calculation
                SPECSET2 = 1
C             if no nickel present in solution
              ELSE
C               exclude nickel species from equilibrium calculation
                SPECSET2 = 2
              END IF
C
C             call subroutine NS11AD to resolve equilibrium state of system
              CALL NS11AD(N,X,F,A,STEP,STPMAX,ACC,MAXFUN,IPRINT,WO,IW)
C
C             calculate species activities
              DO 150 I = 1,11
                ACTIV(I,INODE) = 10.0**LAS(I)
 150          CONTINUE
C
C             calculate supersaturation
              SCAL(INODE,1,ISUPER) = ((ACTIV(2,INODE))*((ACTIV(5,INODE)
     &                                )**2.0))/KSP
C             calculate pH
              SCAL(INODE,1,IPH) = -LAS(1)
C             calculate ionic strength [kmol.m-3]
              SCAL(INODE,1,ISTION) = STION
C             calculate Ni.2+ activity
              SCAL(INODE,1,IANI) = ACTIV(2,INODE)
C             calculate total dissolved nickel concentration [kmol.m-3]
              SCAL(INODE,1,ITDNC) = (10.0**X(2))+(10.0**X(6))+(10.0**
     &                              X(7))+(10.0**X(8))+(10.0**X(9))+(
     &                              2.0*(10.0**X(10)))+(10.0**X(11))
C
            END IF
C
C           if solution not supersaturated
            ELSE
C
C             specify zero nucleation rate [m-3.s-1]
              BIRTH(INODE) = 0.0
C             specify zero change in precipitate yield [kmol.m-3]
              DPPT(INODE) = 0.0
C             specify very large kinetics time scale to denote infinite value [s]
              SCAL(INODE,1,ITK) = 1.0E+100
C
            END IF
C
C           if aggregation-related calculations performed after effects of nucleation
C           established
            IF (GENSET2.EQ.1) THEN
C
C             calculate particle number concentration in 1st size interval using simple
C             reverse Euler numerical integration technique [m-3]
              PN(1,INODE) = PN(1,INODE)+(BIRTH(INODE)*DT)
C
C             calculate size-independent portion of aggregation kernel (units dependent
C             upon aggregation kernel)
              AR(INODE) = 8.33E-78*EXP((14.0*SCAL(INODE,1,IPH))+(1050.0*
     &                    SCAL(INODE,1,ISTION)))
C
            END IF
C
C           within each size interval
            DO 160 I = 1,NOPSI
C
C             initialise 1st term in modified aggregation rate equation of DPB of
C             Hounslow et al. [m-3.s-1]
              TERM1 = 0.0
C
C             if appropriate for size interval
```

```
            IF (I.GT.2) THEN
C
C             evaluate summation in 1st term in modified aggregation rate equation of
C             DPB of Hounslow et al. [s-1]
              SUM1 = 0.0
C
              DO 170 J = 1,I-2
                EXP1 = J-I+1
C               size-independent kernel [m3.s-1]
                AR1 = AR(INODE)
C               Brownian kernel [m3.s-1]
C                AR1 = AR(INODE)*((PAVE1(I-1)+PAVE1(J))*((PAVE1(I-1)**(-
C     &                 1.0))+(PAVE1(J)**(-1.0))))
C               gravitational kernel [m3.s-1]
C                AR1 = AR(INODE)*(((PAVE1(I-1)+PAVE1(J))**2.0)*(ABS(
C     &                 PAVE1(I-1)-PAVE1(J))))
C               shear kernel [m3.s-1]
C                AR1 = AR(INODE)*((PAVE1(I-1)+PAVE1(J))**3.0)
C               particle inertia kernel [m3.s-1]
C                AR1 = AR(INODE)*(((PAVE1(I-1)+PAVE1(J))**2.0)*(ABS((
C     &                 PAVE1(I-1)**2.0)-(PAVE1(J)**2.0))))
C               Thompson kernel [m3.s-1]
C                AR1 = AR(INODE)*((((PAVE1(I-1)**3.0)-(PAVE1(J)**3.0))**
C     &                 2.0)/((PAVE1(I-1)**3.0)+(PAVE1(J)**3.0)))
C               linear sum kernel [m3.s-1]
C                AR1 = AR(INODE)*(PAVE1(I-1)+PAVE1(J))
C               quadratic sum kernel [m3.s-1]
C                AR1 = AR(INODE)*((PAVE1(I-1)**2.0)+(PAVE1(J)**2.0))
C               cubic sum kernel [m3.s-1]
C                AR1 = AR(INODE)*((PAVE1(I-1)**3.0)+(PAVE1(J)**3.0))
                SUM1 = SUM1+((2.0**EXP1)*AR1*PN(J,INODE))
 170          CONTINUE
C
C             calculate 1st term in modified aggregation rate equation of DPB of
C             Hounslow et al. [m-3.s-1]
              TERM1 = PN(I-1,INODE)*SUM1
C
            END IF
C
C           initialise 2nd term in modified aggregation rate equation of DPB of
C           Hounslow et al. [m-3.s-1]
            TERM2 = 0.0
C
C           if appropriate for size interval
            IF (I.GT.1) THEN
C
C             size-independent kernel [m3.s-1]
              AR2 = AR(INODE)
C             Brownian kernel [m3.s-1]
C              AR2 = AR(INODE)*((2.0*PAVE1(I-1))*(2.0*(PAVE1(I-1)**(-1.0
C     &               ))))
C             gravitational kernel [m3.s-1]
C              AR2 = 0.0
C             shear kernel [m3.s-1]
C              AR2 = AR(INODE)*((2.0*PAVE1(I-1))**3.0)
C             particle inertia kernel [m3.s-1]
C              AR2 = 0.0
C             Thompson kernel [m3.s-1]
C              AR2 = 0.0
C             linear sum kernel [m3.s-1]
C              AR2 = AR(INODE)*(2.0*PAVE1(I-1))
C             quadratic sum kernel [m3.s-1]
C              AR2 = AR(INODE)*(2.0*(PAVE1(I-1)**2.0))
C             cubic sum kernel [m3.s-1]
C              AR2 = AR(INODE)*(2.0*(PAVE1(I-1)**3.0))
C
C             calculate 2nd term in modified aggregation rate equation of DPB of
C             Hounslow et al. [m-3.s-1]
              TERM2 = 0.5*AR2*(PN(I-1,INODE)**2.0)
C
            END IF
C
C           initialise 3rd term in modified aggregation rate equation of DPB of
C           Hounslow et al. [m-3.s-1]
            TERM3 = 0.0
C
C           if appropriate for size interval
```

```
            IF (I.GT.1.AND.I.LT.NOPSI) THEN
C
C             evaluate summation in 3rd term in modified aggregation rate equation of
C             DPB of Hounslow et al. [s-1]
              SUM3 = 0.0
C
              DO 180 J = 1,I-1
                EXP2 = J-I
C               size-independent kernel [m3.s-1]
                AR3 = AR(INODE)
C               Brownian kernel [m3.s-1]
C                AR3 = AR(INODE)*((PAVE1(I)+PAVE1(J))*((PAVE1(I)**(-1.0)
C     &               )+(PAVE1(J)**(-1.0))))
C               gravitational kernel [m3.s-1]
C                AR3 = AR(INODE)*(((PAVE1(I)+PAVE1(J))**2.0)*(ABS(
C     &               PAVE1(I)-PAVE1(J))))
C               shear kernel [m3.s-1]
C                AR3 = AR(INODE)*((PAVE1(I)+PAVE1(J))**3.0)
C               particle inertia kernel [m3.s-1]
C                AR3 = AR(INODE)*(((PAVE1(I)+PAVE1(J))**2.0)*(ABS((
C     &               PAVE1(I)**2.0)-(PAVE1(J)**2.0))))
C               Thompson kernel [m3.s-1]
C                AR3 = AR(INODE)*((((PAVE1(I)**3.0)-(PAVE1(J)**3.0))**
C     &               2.0)/((PAVE1(I)**3.0)+(PAVE1(J)**3.0)))
C               linear sum kernel [m3.s-1]
C                AR3 = AR(INODE)*(PAVE1(I)+PAVE1(J))
C               quadratic sum kernel [m3.s-1]
C                AR3 = AR(INODE)*((PAVE1(I)**2.0)+(PAVE1(J)**2.0))
C               cubic sum kernel [m3.s-1]
C                AR3 = AR(INODE)*((PAVE1(I)**3.0)+(PAVE1(J)**3.0))
                SUM3 = SUM3+((2.0**EXP2)*AR3*PN(J,INODE))
 180          CONTINUE
C
C             calculate 3rd term in modified aggregation rate equation of DPB of
C             Hounslow et al. [m-3.s-1]
              TERM3 = PN(I,INODE)*SUM3
C
            END IF
C
C           initialise 4th term in modified aggregation rate equation of DPB of
C           Hounslow et al. [m-3.s-1]
            TERM4 = 0.0
C
C           if appropriate for size interval
            IF (I.LT.NOPSI) THEN
C
C             evaluate summation in 4th term in modified aggregation rate equation of
C             DPB of Hounslow et al. [s-1]
              SUM4 = 0.0
C
              DO 190 J = I,NOPSI-1
C               size-independent kernel [m3.s-1]
                AR4 = AR(INODE)
C               Brownian kernel [m3.s-1]
C                AR4 = AR(INODE)*((PAVE1(I)+PAVE1(J))*((PAVE1(I)**(-1.0)
C     &               )+(PAVE1(J)**(-1.0))))
C               gravitational kernel [m3.s-1]
C                AR4 = AR(INODE)*(((PAVE1(I)+PAVE1(J))**2.0)*(ABS(
C     &               PAVE1(I)-PAVE1(J))))
C               shear kernel [m3.s-1]
C                AR4 = AR(INODE)*((PAVE1(I)+PAVE1(J))**3.0)
C               particle inertia kernel [m3.s-1]
C                AR4 = AR(INODE)*(((PAVE1(I)+PAVE1(J))**2.0)*(ABS((
C     &               PAVE1(I)**2.0)-(PAVE1(J)**2.0))))
C               Thompson kernel [m3.s-1]
C                AR4 = AR(INODE)*((((PAVE1(I)**3.0)-(PAVE1(J)**3.0))**
C     &               2.0)/((PAVE1(I)**3.0)+(PAVE1(J)**3.0)))
C               linear sum kernel [m3.s-1]
C                AR4 = AR(INODE)*(PAVE1(I)+PAVE1(J))
C               quadratic sum kernel [m3.s-1]
C                AR4 = AR(INODE)*((PAVE1(I)**2.0)+(PAVE1(J)**2.0))
C               cubic sum kernel [m3.s-1]
C                AR4 = AR(INODE)*((PAVE1(I)**3.0)+(PAVE1(J)**3.0))
                SUM4 = SUM4+(AR4*PN(J,INODE))
 190          CONTINUE
C
C             calculate 4th term in modified aggregation rate equation of DPB of
```

```
c             Hounslow et al. [m-3.s-1]
              TERM4 = PN(I,INODE)*SUM4
C
            END IF
C
C           calculate aggregation rate [m-3.s-1]
            AGGREG(I,INODE) = TERM1+TERM2-TERM3-TERM4
C
 160      CONTINUE
C
C        calculate particle number concentrations using simple reverse Euler
C        numerical integration technique [m-3]
         DO 200 I = 1,NOPSI
           PN(I,INODE) = PN(I,INODE)+(AGGREG(I,INODE)*DT)
 200     CONTINUE
C
C        if aggregation-related calculations performed before effects of nucleation
C        established
         IF (GENSET2.EQ.2) THEN
C
C          calculate particle number concentration in 1st size interval using simple
C          reverse Euler numerical integration technique [m-3]
           PN(1,INODE) = PN(1,INODE)+(BIRTH(INODE)*DT)
C
         END IF
C
C        correct generation of negative particle number concentration(s) (assuming
C        cause to be aggregation of pairs of like particles to produce particles in
C        immediately larger size class)
         DO 210 I = 1,NOPSI-1
C
           IF (PN(I,INODE).LT.0.0) THEN
             PN(I+1,INODE) = PN(I+1,INODE)+(0.5*PN(I,INODE))  ! [m-3]
             PN(I,INODE) = 0.0  ! [m-3]
C            write output to file fort.99
             WRITE(99,*) 'INFO: NEGATIVE NUMBER CONCENTRATION RIGHTED'
             WRITE(99,*) 'SIZE CLASS',I
             WRITE(99,*) 'CELL',INODE
           END IF
C
 210     CONTINUE
C
C        if negative particle number concentration persists in final size interval
C        (a result of precision-related difficulties with aggregation calculations)
         IF (PN(NOPSI,INODE).LT.0.0) THEN
C
C          specify zero particle number concentration in final size interval [m-3]
           PN(NOPSI,INODE) = 0.0
C
C          calculate 3rd moment [micron3.m-3]
           SCAL(INODE,1,IM3) = 0.0
C
           DO 220 I = 1,NOPSI
             SCAL(INODE,1,IM3) = SCAL(INODE,1,IM3)+((PAVE3(I)**3.0)*
     &                           PN(I,INODE))
 220       CONTINUE
C
C          if aggregation-related calculations performed after effects of nucleation
C          established
           IF (GENSET2.EQ.1) THEN
C
C            assign excess solid to appropriate maximum size class
             IF (KSTEP+1.LT.NOPSI) THEN
               PN(KSTEP+1,INODE) = PN(KSTEP+1,INODE)+(((MOM3I(INODE)+(
     &                             (6.0*DPPT(INODE))/(PI*RHO*1.0E-18))
     &                             )-SCAL(INODE,1,IM3))/(
     &                             PAVE3(KSTEP+1)**3.0))  ! [m-3]
             ELSE
               PN(NOPSI,INODE) = PN(NOPSI,INODE)+(((MOM3I(INODE)+((6.0
     &                           *DPPT(INODE))/(PI*RHO*1.0E-18)))-
     &                           SCAL(INODE,1,IM3))/(PAVE3(NOPSI)**3.0
     &                           ))  ! [m-3]
             END IF
C
C          if aggregation-related calculations performed before effects of nucleation
C          established
           ELSE IF (GENSET2.EQ.2) THEN
```

```
C
C             assign excess solid to appropriate maximum size class
              IF (KSTEP.LT.NOPSI) THEN
                PN(KSTEP,INODE) = PN(KSTEP,INODE)+(((MOM3I(INODE)+((6.0
     &                           *DPPT(INODE))/(PI*RHO*1.0E-18)))-
     &                           SCAL(INODE,1,IM3))/(PAVE3(KSTEP)**3.0
     &                           ))  ! [m-3]
              ELSE
                PN(NOPSI,INODE) = PN(NOPSI,INODE)+(((MOM3I(INODE)+((6.0
     &                           *DPPT(INODE))/(PI*RHO*1.0E-18)))-
     &                           SCAL(INODE,1,IM3))/(PAVE3(NOPSI)**3.0
     &                           ))  ! [m-3]
              END IF
C
            END IF
C
          END IF
C
C         calculate 3rd moment [micron3.m-3]
          SCAL(INODE,1,IM3) = 0.0
C
          DO 230 I = 1,NOPSI
            SCAL(INODE,1,IM3) = SCAL(INODE,1,IM3)+((PAVE3(I)**3.0)*
     &                          PN(I,INODE))
 230      CONTINUE
C
C         if particle number concentrations incorrectly reflect absence of precipitate
C         (a result of precision-related difficulties with aggregation calculations)
          IF (SCAL(INODE,1,IM3).EQ.0.0.AND.(MOM3I(INODE)+((6.0*
     &       DPPT(INODE))/(PI*RHO*1.0E-18))).GT.0.0) THEN
C
C           if aggregation-related calculations performed after effects of nucleation
C           established
            IF (GENSET2.EQ.1) THEN
C
C             specify arbitrary positive particle number concentration in appropriate
C             maximum size class
              IF (KSTEP+1.LT.NOPSI) THEN
                PN(KSTEP+1,INODE) = 1.0  ! [m-3]
              ELSE
                PN(NOPSI,INODE) = 1.0  ! [m-3]
              END IF
C
C           if aggregation-related calculations performed before effects of nucleation
C           established
            ELSE IF (GENSET2.EQ.2) THEN
C
C             specify arbitrary positive particle number concentration in appropriate
C             maximum size class
              IF (KSTEP.LT.NOPSI) THEN
                PN(KSTEP,INODE) = 1.0  ! [m-3]
              ELSE
                PN(NOPSI,INODE) = 1.0  ! [m-3]
              END IF
C
            END IF
C
C           calculate 3rd moment [micron3.m-3]
            SCAL(INODE,1,IM3) = 0.0
C
            DO 240 I = 1,NOPSI
              SCAL(INODE,1,IM3) = SCAL(INODE,1,IM3)+((PAVE3(I)**3.0)*
     &                            PN(I,INODE))
 240        CONTINUE
C
          END IF
C
C         ensure calculated particle number concentrations reflect correct change in
C         precipitate yield
          IF (SCAL(INODE,1,IM3).GT.0.0) THEN
C
            DO 250 I = 1,NOPSI
              PN(I,INODE) = PN(I,INODE)*((MOM3I(INODE)+((6.0*
     &                     DPPT(INODE))/(PI*RHO*1.0E-18)))/
     &                     SCAL(INODE,1,IM3))  ! [m-3]
 250        CONTINUE
C
```

```
          END IF
C
C         if dilution effect isolated, then distributed and teamed with time-dependent
C         feed source term
          IF (GENSET3.EQ.3) THEN
C
C           within feed cell(s)
            IF (FEED(INODE).EQ.2) THEN
C
C             calculate redistributed particle number concentrations [m-3]
              DO 260 I = 1,NOPSI
                PN(I,INODE) = PN(I,INODE)*((VOL(INODE)/GEOMVOL)*(
     &                        TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL
     &                        )*(FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*
     &                        TOTALVOL)
 260          CONTINUE
C
C             estimate logarithm of equilibrium species concentrations as logarithm of
C             current redistributed species concentrations (supplied as initial
C             estimate to subroutine NS11AD) [log(kmol.m-3)]
              DO 270 I = 1,11
C
                IF (10.0**X(I).EQ.0.0) THEN
                  X(I) = -1000.0
                ELSE
                  X(I) = LOG10((10.0**X(I))*((VOL(INODE)/GEOMVOL)*(
     &                    TOTALVOL-FEEDVOL*DT)+(VOL(INODE)/FEEDPVOL)*(
     &                    FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*TOTALVOL))
                END IF
C
 270          CONTINUE
C
C           within each remaining cell in flow domain
            ELSE
C
C             calculate redistributed particle number concentrations [m-3]
              DO 280 I = 1,NOPSI
                PN(I,INODE) = PN(I,INODE)*((VOL(INODE)/GEOMVOL)*(
     &                        TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/
     &                        GEOMVOL)*TOTALVOL)
 280          CONTINUE
C
C             estimate logarithm of equilibrium species concentrations as logarithm of
C             current redistributed species concentrations (supplied as initial
C             estimate to subroutine NS11AD) [log(kmol.m-3)]
              DO 290 I = 1,11
C
                IF (10.0**X(I).EQ.0.0) THEN
                  X(I) = -1000.0
                ELSE
                  X(I) = LOG10((10.0**X(I))*((VOL(INODE)/GEOMVOL)*(
     &                    TOTALVOL-FEEDVOL*DT))/((VOL(INODE)/GEOMVOL)*
     &                    TOTALVOL))
                END IF
C
 290          CONTINUE
C
            END IF
C
C           calculate 3rd moment [micron3.m-3]
            SCAL(INODE,1,IM3) = 0.0
C
            DO 300 I = 1,NOPSI
              SCAL(INODE,1,IM3) = SCAL(INODE,1,IM3)+((PAVE3(I)**3.0)*
     &                            PN(I,INODE))
 300        CONTINUE
C
C           calculate precipitate yield [kmol.m-3]
            SCAL(INODE,1,IPPT) = (PI*SCAL(INODE,1,IM3)/6.0)*RHO*1.0E-18
C
C           calculate total component concentrations (excluding solid species; used in
C           subroutine CALFUN to close mass balance) [kmol.m-3]
C           H.+
            CC(1) = (10.0**X(1))-(10.0**X(5))-(2.0*(10.0**X(6)))-(10.0
     &              **X(9))-(10.0**X(10))-(3.0*(10.0**X(11)))
C           Ni.2+
            CC(2) = (10.0**X(2))+(10.0**X(6))+(10.0**X(7))+(10.0**X(8))
```

```
     &              +(10.0**X(9))+(2.0*(10.0**X(10)))+(10.0**X(11))
C           Na.+
            CC(3) = 10.0**X(3)
C           Cl.-
            CC(4) = (10.0**X(4))+(2.0*(10.0**X(7)))+(10.0**X(8))
C
C           calculate logarithm of current Ni(OH)2 solid concentration (supplied to
C           subroutine NS11AD and returned unchanged) [log(kmol.m-3)]
            IF (SCAL(INODE,1,IPPT).GT.0.0) THEN
              X(12) = LOG10(SCAL(INODE,1,IPPT))
            ELSE
              X(12) = -1000.0
            END IF
C
C           specify input parameters required by subroutine NS11AD
C           number of equations; number of unknowns (species concentrations)
            N = 12
C           X-step used to calculate estimates of Jacobian (partial derivatives)
C           matrix numerically
            STEP = 1.0E-12
C           generous estimate of 'distance' between initial approximation and required
C           solution
            STPMAX = 5.0
C           maximum number of calls of subroutine CALFUN
            MAXFUN = 500000000
C           no printing
            IPRINT = 0
C           required accuracy (maximum sum of square errors)
            ACC = 1.0E-20
C
C           exclude solid species from equilibrium calculation
            SPECSET1 = 1
C
C           if nickel present in solution
            IF (CC(2).GT.0.0) THEN
C             include nickel species in equilibrium calculation
              SPECSET2 = 1
C           if no nickel present in solution
            ELSE
C             exclude nickel species from equilibrium calculation
              SPECSET2 = 2
            END IF
C
C           call subroutine NS11AD to resolve equilibrium state of system
            CALL NS11AD(N,X,F,A,STEP,STPMAX,ACC,MAXFUN,IPRINT,WO,IW)
C
C           calculate species activities
            DO 310 I = 1,11
              ACTIV(I,INODE) = 10.0**LAS(I)
 310        CONTINUE
C
C           calculation supersaturation
            SCAL(INODE,1,ISUPER) = ((ACTIV(2,INODE))*((ACTIV(5,INODE))
     &                             **2.0))/KSP
C           calculate pH
            SCAL(INODE,1,IPH) = -LAS(1)
C           calculate ionic strength [kmol.m-3]
            SCAL(INODE,1,ISTION) = STION
C           calculate Ni.2+ activity
            SCAL(INODE,1,IANI) = ACTIV(2,INODE)
C           calculate total dissolved nickel concentration [kmol.m-3]
            SCAL(INODE,1,ITDNC) = (10.0**X(2))+(10.0**X(6))+(10.0**X(7)
     &                           )+(10.0**X(8))+(10.0**X(9))+(2.0*(
     &                           10.0**X(10)))+(10.0**X(11))
C
          END IF
C
        END IF
C
C       calculate 0th [m-3], 1st [micron.m-3], 2nd [micron2.m-3] and 3rd [micron3.m-3]
C       moments
        SCAL(INODE,1,IM0) = 0.0
        SCAL(INODE,1,IM1) = 0.0
        SCAL(INODE,1,IM2) = 0.0
        SCAL(INODE,1,IM3) = 0.0
C
        DO 320 I = 1,NOPSI
```

```
          SCAL(INODE,1,IM0) = SCAL(INODE,1,IM0)+PN(I,INODE)
          SCAL(INODE,1,IM1) = SCAL(INODE,1,IM1)+(PAVE1(I)*PN(I,INODE))
          SCAL(INODE,1,IM2) = SCAL(INODE,1,IM2)+((PAVE2(I)**2.0)*
     &                        PN(I,INODE))
          SCAL(INODE,1,IM3) = SCAL(INODE,1,IM3)+((PAVE3(I)**3.0)*
     &                        PN(I,INODE))
 320    CONTINUE
C
C       calculate solids volume fraction
        SCAL(INODE,1,ISVF) = (PI*SCAL(INODE,1,IM3)/6.0)*1.0E-18
C       calculate precipitate yield [kmol.m-3]
        SCAL(INODE,1,IPPT) = (PI*SCAL(INODE,1,IM3)/6.0)*RHO*1.0E-18
C
        IF (SCAL(INODE,1,IM0).GT.0.0) THEN
C         calculate average particle diameter [micron]
          SCAL(INODE,1,IPA) = (SCAL(INODE,1,IM3)/SCAL(INODE,1,IM0))**
     &                        (1.0/3.0)
C         calculate coefficient of variation
          SCAL(INODE,1,ICV) = ((SCAL(INODE,1,IM0)*SCAL(INODE,1,IM2)/(
     &                        SCAL(INODE,1,IM1)**2.0))-1.0)**(1.0/2.0)
        END IF
C
C       write output from specified cell in 'bulk' zone to file fort.97 at specified
C       time intervals (every 10 s)
        IF (ANINT(100.0*TIME)/1000.0.EQ.ANINT(TIME/10.0).AND.INODE.EQ.
     &     2222) THEN
C
          WRITE(97,*) 'TIME',TIME
          WRITE(97,*) 'SIZE CLASS','PARTICLE NUMBER CONCENTRATION'
C
          DO 330 I = 1,NOPSI
            WRITE(97,*) I,PN(I,INODE)
 330      CONTINUE
C
          WRITE(97,*) '0TH MOMENT',SCAL(INODE,1,IM0)
          WRITE(97,*) '1ST MOMENT',SCAL(INODE,1,IM1)
          WRITE(97,*) '2ND MOMENT',SCAL(INODE,1,IM2)
          WRITE(97,*) '3RD MOMENT',SCAL(INODE,1,IM3)
          WRITE(97,*) 'SOLIDS VOLUME FRACTION',SCAL(INODE,1,ISVF)
          WRITE(97,*) 'PRECIPITATE YIELD',SCAL(INODE,1,IPPT)
C
          IF (SCAL(INODE,1,IM0).GT.0.0) THEN
            WRITE(97,*) 'AVERAGE PARTICLE DIAMETER',SCAL(INODE,1,IPA)
            WRITE(97,*) 'COEFFICIENT OF VARIATION',SCAL(INODE,1,ICV)
          END IF
C
          WRITE(97,*) 'SUPERSATURATION',SCAL(INODE,1,ISUPER)
          WRITE(97,*) 'PH',SCAL(INODE,1,IPH)
          WRITE(97,*) 'IONIC STRENGTH',SCAL(INODE,1,ISTION)
          WRITE(97,*) 'NI.2+ ACTIVITY',SCAL(INODE,1,IANI)
          WRITE(97,*) 'TOTAL DISSOLVED NICKEL CONCENTRATION',
     &                SCAL(INODE,1,ITDNC)
          WRITE(97,*) 'GRID CELL VOLUME',VOL(INODE)
          WRITE(97,*) '"REAL" TOTAL VOLUME',TOTALVOL
C
        END IF
C
C       update user scalars used to store speciation estimate for following time step
C       [log(kmol.m-3)]
C       H.+
        SCAL(INODE,1,IS1) = X(1)
C       Ni.2+
        SCAL(INODE,1,IS2) = X(2)
C       Na.+
        SCAL(INODE,1,IS3) = X(3)
C       Cl.-
        SCAL(INODE,1,IS4) = X(4)
C       OH.-
        SCAL(INODE,1,IS5) = X(5)
C       Ni(OH)2 aqueous
        SCAL(INODE,1,IS6) = X(6)
C       NiCl2 aqueous
        SCAL(INODE,1,IS7) = X(7)
C       NiCl.+
        SCAL(INODE,1,IS8) = X(8)
C       NiOH.+
        SCAL(INODE,1,IS9) = X(9)
```

```
C       Ni2OH.3+
        SCAL(INODE,1,IS10) = X(10)
C       Ni(OH)3.-
        SCAL(INODE,1,IS11) = X(11)
C
C       update user scalars representing dissolved species concentrations [kmol.m-3]
C       H.+
        SCAL(INODE,1,IH) = 10.0**X(1)
C       Ni.2+
        SCAL(INODE,1,INI) = 10.0**X(2)
C       Na.+
        SCAL(INODE,1,INA) = 10.0**X(3)
C       Cl.-
        SCAL(INODE,1,ICL) = 10.0**X(4)
C       OH.-
        SCAL(INODE,1,IOH) = 10.0**X(5)
C       Ni(OH)2 aqueous
        SCAL(INODE,1,INIOH2) = 10.0**X(6)
C       NiCl2 aqueous
        SCAL(INODE,1,INICL2) = 10.0**X(7)
C       NiCl.+
        SCAL(INODE,1,INICL) = 10.0**X(8)
C       NiOH.+
        SCAL(INODE,1,INIOH) = 10.0**X(9)
C       Ni2OH.3+
        SCAL(INODE,1,INI2OH) = 10.0**X(10)
C       Ni(OH)3.-
        SCAL(INODE,1,INIOH3) = 10.0**X(11)
C
C       update user scalars representing particle number concentrations [m-3]
C       1st size interval
        SCAL(INODE,1,I1) = PN(1,INODE)
C       2nd size interval
        SCAL(INODE,1,I2) = PN(2,INODE)
C       3rd size interval
        SCAL(INODE,1,I3) = PN(3,INODE)
C       4th size interval
        SCAL(INODE,1,I4) = PN(4,INODE)
C       5th size interval
        SCAL(INODE,1,I5) = PN(5,INODE)
C       6th size interval
        SCAL(INODE,1,I6) = PN(6,INODE)
C       7th size interval
        SCAL(INODE,1,I7) = PN(7,INODE)
C       8th size interval
        SCAL(INODE,1,I8) = PN(8,INODE)
C       9th size interval
        SCAL(INODE,1,I9) = PN(9,INODE)
C       10th size interval
        SCAL(INODE,1,I10) = PN(10,INODE)
C       11th size interval
        SCAL(INODE,1,I11) = PN(11,INODE)
C       12th size interval
        SCAL(INODE,1,I12) = PN(12,INODE)
C       13th size interval
        SCAL(INODE,1,I13) = PN(13,INODE)
C
C       stop if negative species concentration(s) inputted to CFX-4.3 Solver
        IF (SCAL(INODE,1,IH).LT.0.0.OR.SCAL(INODE,1,INI).LT.0.0.OR.
     &     SCAL(INODE,1,INA).LT.0.0.OR.SCAL(INODE,1,ICL).LT.0.0.OR.
     &     SCAL(INODE,1,IOH).LT.0.0.OR.SCAL(INODE,1,INIOH2).LT.0.0.OR.
     &     SCAL(INODE,1,INICL2).LT.0.0.OR.SCAL(INODE,1,INICL).LT.0.0.OR
     &     .SCAL(INODE,1,INIOH).LT.0.0.OR.SCAL(INODE,1,INI2OH).LT.0.0.
     &     OR.SCAL(INODE,1,INIOH3).LT.0.0) THEN
C         write output to file fort.99
          WRITE(99,*) 'ERROR: NEGATIVE SPECIES CONCENTRATION(S) '//
     &                'INPUTTED TO CFX-4.3 SOLVER'
          WRITE(99,*) 'CELL',INODE
C         stop with message output
          STOP 'NEGATIVE SPECIES CONCENTRATION(S) INPUTTED TO CFX-4.3 SO
     &LVER'
        END IF
C
C       stop if negative particle number concentration(s) inputted to CFX-4.3 Solver
        IF (SCAL(INODE,1,I1).LT.0.0.OR.SCAL(INODE,1,I2).LT.0.0.OR.
     &     SCAL(INODE,1,I3).LT.0.0.OR.SCAL(INODE,1,I4).LT.0.0.OR.
     &     SCAL(INODE,1,I5).LT.0.0.OR.SCAL(INODE,1,I6).LT.0.0.OR.
```

```
     &     SCAL(INODE,1,I7).LT.0.0.OR.SCAL(INODE,1,I8).LT.0.0.OR.
     &     SCAL(INODE,1,I9).LT.0.0.OR.SCAL(INODE,1,I10).LT.0.0.OR.
     &     SCAL(INODE,1,I11).LT.0.0.OR.SCAL(INODE,1,I12).LT.0.0.OR.
     &     SCAL(INODE,1,I13).LT.0.0) THEN
C         write output to file fort.99
          WRITE(99,*) 'ERROR: NEGATIVE PARTICLE NUMBER '//
     &                'CONCENTRATION(S) INPUTTED TO CFX-4.3 SOLVER'
          WRITE(99,*) 'CELL',INODE
C         stop with message output
          STOP 'NEGATIVE PARTICLE NUMBER CONCENTRATION(S) INPUTTED TO CF
     &X-4.3 SOLVER'
        END IF
C
C       calculate maximum pH
        PHMAX = MAX(PHMAX,SCAL(INODE,1,IPH))
C       calculate minimum pH
        PHMIN = MIN(PHMIN,SCAL(INODE,1,IPH))
C
 50   CONTINUE
C
C     initialise overall 0th [m-3], 1st [micron.m-3], 2nd [micron2.m-3] and 3rd
C     [micron3.m-3] moments
      OVERMOM0 = 0.0
      OVERMOM1 = 0.0
      OVERMOM2 = 0.0
      OVERMOM3 = 0.0
C
C     write output to file fort.99
      WRITE(99,*) 'SIZE CLASS','OVERALL PARTICLE NUMBER CONCENTRATION'
C
C     within each size interval
      DO 340 I = 1,NOPSI
C
C       initialise overall particle number concentration [m-3]
        OVERN(I) = 0.0
C
C       call utility routine IPALL to return cell-centre addresses of those cells
C       occupied by tank contents (i.e. all interior cells)
        CALL IPALL('*','*','BLOCK','CENTRES',IPT,NPT,CWORK,IWORK)
C
C       within each cell in flow domain
        DO 350 J = 1,NPT
C
          INODE = IPT(J)
C
C         if dilution effect distributed
          IF (GENSET3.EQ.1.OR.GENSET3.EQ.3) THEN
C
C           calculate overall particle number concentration [m-3]
            OVERN(I) = OVERN(I)+(PN(I,INODE)*VOL(INODE)/GEOMVOL)
C
C         if dilution effect isolated
          ELSE IF (GENSET3.EQ.2) THEN
C
C           within feed cell(s)
            IF (FEED(INODE).EQ.2) THEN
C             calculate overall particle number concentration [m-3]
              OVERN(I) = OVERN(I)+(PN(I,INODE)*((VOL(INODE)*INITVOL/
     &                  GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL-
     &                  INITVOL))/TOTALVOL)
C           within each remaining cell in flow domain
            ELSE
C             calculate overall particle number concentration [m-3]
              OVERN(I) = OVERN(I)+(PN(I,INODE)*(VOL(INODE)*INITVOL/
     &                  GEOMVOL)/TOTALVOL)
            END IF
C
          END IF
C
 350    CONTINUE
C
C       calculate overall 0th [m-3], 1st [micron.m-3], 2nd [micron2.m-3] and 3rd
C       [micron3.m-3] moments
        OVERMOM0 = OVERMOM0+OVERN(I)
        OVERMOM1 = OVERMOM1+(PAVE1(I)*OVERN(I))
        OVERMOM2 = OVERMOM2+((PAVE2(I)**2.0)*OVERN(I))
        OVERMOM3 = OVERMOM3+((PAVE3(I)**3.0)*OVERN(I))
```

```
C
C       write output to file fort.99
        WRITE(99,*) I,OVERN(I)
C
 340  CONTINUE
C
C     calculate overall solids volume fraction
      OVERSVF = (PI*OVERMOM3/6.0)*1.0E-18
C     calculate overall precipitate yield [kmol.m-3]
      OVERPPT = (PI*OVERMOM3/6.0)*RHO*1.0E-18
C
C     write output to file fort.99
      WRITE(99,*) 'OVERALL 0TH MOMENT',OVERMOM0
      WRITE(99,*) 'OVERALL 1ST MOMENT',OVERMOM1
      WRITE(99,*) 'OVERALL 2ND MOMENT',OVERMOM2
      WRITE(99,*) 'OVERALL 3RD MOMENT',OVERMOM3
      WRITE(99,*) 'OVERALL SOLIDS VOLUME FRACTION',OVERSVF
      WRITE(99,*) 'OVERALL PRECIPITATE YIELD',OVERPPT
C
      IF (OVERMOM0.GT.0.0) THEN
C       calculate overall average particle diameter [micron]
        OVERPAVE = (OVERMOM3/OVERMOM0)**(1.0/3.0)
C       write output to file fort.99
        WRITE(99,*) 'OVERALL AVERAGE PARTICLE DIAMETER',OVERPAVE
C       calculate overall coefficient of variation
        OVERCV = ((OVERMOM0*OVERMOM2/(OVERMOM1**2.0))-1.0)**(1.0/2.0)
C       write output to file fort.99
        WRITE(99,*) 'OVERALL COEFFICIENT OF VARIATION',OVERCV
      END IF
C
C     initialise overall total dissolved nickel concentration [kmol.m-3]
      OVERTDNC = 0.0
C
C     call utility routine IPALL to return cell-centre addresses of those cells
C     occupied by tank contents (i.e. all interior cells)
      CALL IPALL('*','*','BLOCK','CENTRES',IPT,NPT,CWORK,IWORK)
C
C     within each cell in flow domain
      DO 360 I = 1,NPT
C
        INODE = IPT(I)
C
C       if dilution effect distributed
        IF (GENSET3.EQ.1.OR.GENSET3.EQ.3) THEN
C
C         calculate overall total dissolved nickel concentration [kmol.m-3]
          OVERTDNC = OVERTDNC+(SCAL(INODE,1,ITDNC)*VOL(INODE)/GEOMVOL)
C
C       if dilution effect isolated
        ELSE IF (GENSET3.EQ.2) THEN
C
C         within feed cell(s)
          IF (FEED(INODE).EQ.2) THEN
C           calculate overall total dissolved nickel concentration [kmol.m-3]
            OVERTDNC = OVERTDNC+(SCAL(INODE,1,ITDNC)*((VOL(INODE)*
     &                 INITVOL/GEOMVOL)+(VOL(INODE)/FEEDPVOL)*(TOTALVOL
     &                 -INITVOL))/TOTALVOL)
C         within each remaining cell in flow domain
          ELSE
C           calculate overall total dissolved nickel concentration [kmol.m-3]
            OVERTDNC = OVERTDNC+(SCAL(INODE,1,ITDNC)*(VOL(INODE)*
     &                 INITVOL/GEOMVOL)/TOTALVOL)
          END IF
C
        END IF
C
 360  CONTINUE
C
C     write output to file fort.99
      WRITE(99,*) 'OVERALL TOTAL DISSOLVED NICKEL CONCENTRATION',
     &            OVERTDNC
      WRITE(99,*) 'TANK GEOMETRY VOLUME',GEOMVOL
      WRITE(99,*) '"REAL" TOTAL VOLUME',TOTALVOL
      WRITE(99,*) 'MAXIMUM PH',PHMAX
      WRITE(99,*) 'MINIMUM PH',PHMIN
C
C     write output to file fort.98 at specified time intervals (every 10 s)
```

```
      IF (ANINT(100.0*TIME)/1000.0.EQ.ANINT(TIME/10.0)) THEN
C
        WRITE(98,*) 'TIME',TIME
        WRITE(98,*) 'SIZE CLASS','OVERALL PARTICLE NUMBER '//
     &              'CONCENTRATION'
C
        DO 370 I = 1,NOPSI
          WRITE(98,*) I,OVERN(I)
 370    CONTINUE
C
        WRITE(98,*) 'OVERALL 0TH MOMENT',OVERMOM0
        WRITE(98,*) 'OVERALL 1ST MOMENT',OVERMOM1
        WRITE(98,*) 'OVERALL 2ND MOMENT',OVERMOM2
        WRITE(98,*) 'OVERALL 3RD MOMENT',OVERMOM3
        WRITE(98,*) 'OVERALL SOLIDS VOLUME FRACTION',OVERSVF
        WRITE(98,*) 'OVERALL PRECIPITATE YIELD',OVERPPT
C
        IF (OVERMOM0.GT.0.0) THEN
          WRITE(98,*) 'OVERALL AVERAGE PARTICLE DIAMETER',OVERPAVE
          WRITE(98,*) 'OVERALL COEFFICIENT OF VARIATION',OVERCV
        END IF
C
        WRITE(98,*) 'OVERALL TOTAL DISSOLVED NICKEL CONCENTRATION',
     &              OVERTDNC
        WRITE(98,*) 'TANK GEOMETRY VOLUME',GEOMVOL
        WRITE(98,*) '"REAL" TOTAL VOLUME',TOTALVOL
        WRITE(98,*) 'MAXIMUM PH',PHMAX
        WRITE(98,*) 'MINIMUM PH',PHMIN
C
      END IF
C
C     if simulation to stop if contents of stirred tank deemed to be uniformly
C     distributed
      IF (GENSET4.EQ.2) THEN
C
C       if criteria for uniformity met
        IF (KSTEP.GT.0.AND.PHMAX-PHMIN.LE.0.05) THEN
C         write output to file fort.99
          WRITE(99,*) 'INFO: SPATIAL UNIFORMITY ACHIEVED'
C         stop with message output
          STOP 'SPATIAL UNIFORMITY ACHIEVED'
        END IF
C
      END IF
C
C     specify zero volumetric feed rate (corrected for use in subroutine USRSRC)
C     [m3.s-1]
      IF (KSTEP.EQ.FEEDSTEP) FEEDVOL = 0.0
C
C+++++++++++++++++ END OF USER AREA 5 +++++++++++++++++++++++++++++++++++
C
      RETURN
C
      END
C
C*********************************************************************
C
C Subroutine CALFUN is called by subroutine NS11AD to compute the values of the
C functions for which zeroes are to be found.  These functions represent the
C equilibrium relationships and mass balances to be satisfied.  The Davies equation,
C deemed suitable for the relatively low ionic strengths encountered, is used to
C calculate the activity coefficients which relate species concentrations and
C activities.  Molarity, as opposed to the technically correct molality, is used as
C the measure of concentration.  The employment of logarithms is essentially a scaling
C exercise, since the performance of subroutine NS11AD is improved if all function
C values are similar in magnitude.  The same applies to the species concentration
C variables.  Necessary parameters are inputted.
C
      SUBROUTINE CALFUN(N,X,F)
C
      DOUBLE PRECISION LKA,Z
      DOUBLE PRECISION F,X
      DOUBLE PRECISION CC,KSP
      DOUBLE PRECISION LACS,LAS,STION
      DOUBLE PRECISION CC0,LKA0
C
      INTEGER I
```

```
      INTEGER N
      INTEGER SPECSET1,SPECSET2
C
      COMMON /UCSPEC1/KSP
      COMMON /UCSPEC2/CC
      COMMON /UCSPEC3/LAS,STION
      COMMON /UCSPEC4/SPECSET1,SPECSET2
C
      DIMENSION LKA(12),Z(12)
      DIMENSION F(12),X(12)
      DIMENSION CC(4)
      DIMENSION LACS(12),LAS(12)
      DIMENSION CC0(4),LKA0(12)
C
C     specify logarithm of equilibrium constant for species formation reactions
C     at temperature of interest (25 deg C); values specified for reactions as
C     presented (i.e. only species designated as components combine to form
C     additional species)
C     H2O = OH.- + H.+
      LKA(5) = -13.998
C     Ni.2+ + 2H2O = Ni(OH)2 aqueous + 2H.+
      LKA(6) = -19.0
C     Ni.2+ + 2Cl.- = NiCl2 aqueous
      LKA(7) = 0.96
C     Ni.2+ + Cl.- = NiCl.+
      LKA(8) = 0.399
C     Ni.2+ + H2O = NiOH.+ + H.+
      LKA(9) = -9.86
C     2Ni.2+ + H2O = Ni2OH.3+ + H.+
      LKA(10) = -10.7
C     Ni.2+ + 3H2O = Ni(OH)3.- + 3H.+
      LKA(11) = -30.0
C
C     calculate logarithm of Ni(OH)2 solubility product at temperature of interest
C     (25 deg C)
C     Ni(OH)2 solid = Ni.2+ + 2OH.-
      LKA(12) = LOG10(KSP)
C
C     specify species charges
C     H.+
      Z(1) = 1.0
C     Ni.2+
      Z(2) = 2.0
C     Na.+
      Z(3) = 1.0
C     Cl.-
      Z(4) = -1.0
C     OH.-
      Z(5) = -1.0
C     NiCl.+
      Z(8) = 1.0
C     NiOH.+
      Z(9) = 1.0
C     Ni2OH.3+
      Z(10) = 3.0
C     Ni(OH)3.-
      Z(11) = -1.0
C
C     calculate ionic strength (charged species only) [kmol.m-3]
      STION = 0.0
C
      DO 10 I = 1,5
        STION = STION+(0.5*(Z(I)**2.0)*(10.0**X(I)))
 10   CONTINUE
C
      DO 20 I = 8,11
        STION = STION+(0.5*(Z(I)**2.0)*(10.0**X(I)))
 20   CONTINUE
C
C     calculate 'charge' of uncharged species (for substitution in the Davies
C     equation) as square root of absolute value of product of positive and negative
C     constituent ion charges
C     Ni(OH)2 aqueous
      Z(6) = 2.0**0.5
C     NiCl2 aqueous
      Z(7) = 2.0**0.5
C
```

```
C     calculate logarithm of activity coefficient of dissolved species using the
C     Davies equation
      DO 30 I = 1,11
        LACS(I) = -0.5092*(Z(I)**2.0)*(((STION**0.5)/(1.0+(STION**0.5))
     &            )-(0.3*STION))
 30   CONTINUE
C
C     calculate logarithm of species activities
      DO 40 I = 1,11
        LAS(I) = LACS(I)+X(I)
 40   CONTINUE
C
C     calculate appropriate relationship between current species activities
C     H2O = OH.- + H.+
      LKA0(5) = LAS(5)+LAS(1)
C
C     if nickel species included in equilibrium calculation
      IF (SPECSET2.EQ.1) THEN
C       Ni.2+ + 2H2O = Ni(OH)2 aqueous + 2H.+
        LKA0(6) = LAS(6)+(2.0*LAS(1))-LAS(2)
C       Ni.2+ + 2Cl.- = NiCl2 aqueous
        LKA0(7) = LAS(7)-LAS(2)-(2.0*LAS(4))
C       Ni.2+ + Cl.- = NiCl.+
        LKA0(8) = LAS(8)-LAS(2)-LAS(4)
C       Ni.2+ + H2O = NiOH.+ + H.+
        LKA0(9) = LAS(9)+LAS(1)-LAS(2)
C       2Ni.2+ + H2O = Ni2OH.3+ + H.+
        LKA0(10) = LAS(10)+LAS(1)-(2.0*LAS(2))
C       Ni.2+ + 3H2O = Ni(OH)3.- + 3H.+
        LKA0(11) = LAS(11)+(3.0*LAS(1))-LAS(2)
C     if nickel species excluded from equilibrium calculation
      ELSE IF (SPECSET2.EQ.2) THEN
        LKA0(6) = LKA(6)
        LKA0(7) = LKA(7)
        LKA0(8) = LKA(8)
        LKA0(9) = LKA(9)
        LKA0(10) = LKA(10)
        LKA0(11) = LKA(11)
      END IF
C
C     if solid species excluded from equilibrium calculation
      IF (SPECSET1.EQ.1) THEN
        LKA0(12) = LKA(12)
C     if solid species included in equilibrium calculation
      ELSE IF (SPECSET1.EQ.2) THEN
C       Ni(OH)2 solid = Ni.2+ + 2OH.-
        LKA0(12) = LAS(2)+(2.0*LAS(5))
      END IF
C
C     calculate total component concentrations based on current species concentrations
C     [kmol.m-3]
C     if solid species excluded from equilibrium calculation
      IF (SPECSET1.EQ.1) THEN
C       H.+
        CC0(1) = (10.0**X(1))-(10.0**X(5))-(2.0*(10.0**X(6)))-(10.0**
     &           X(9))-(10.0**X(10))-(3.0*(10.0**X(11)))
C       Ni.2+
        CC0(2) = (10.0**X(2))+(10.0**X(6))+(10.0**X(7))+(10.0**X(8))+(
     &           10.0**X(9))+(2.0*(10.0**X(10)))+(10.0**X(11))
C     if solid species included in equilibrium calculation
      ELSE IF (SPECSET1.EQ.2) THEN
C       H.+
        CC0(1) = (10.0**X(1))-(10.0**X(5))-(2.0*(10.0**X(6)))-(10.0**
     &           X(9))-(10.0**X(10))-(3.0*(10.0**X(11)))-(2.0*(10.0**
     &           X(12)))
C       Ni.2+
        CC0(2) = (10.0**X(2))+(10.0**X(6))+(10.0**X(7))+(10.0**X(8))+(
     &           10.0**X(9))+(2.0*(10.0**X(10)))+(10.0**X(11))+(10.0**
     &           X(12))
      END IF
C
C     Na.+
      CC0(3) = 10.0**X(3)
C     Cl.-
      CC0(4) = (10.0**X(4))+(2.0*(10.0**X(7)))+(10.0**X(8))
C
C     calculate function for which zero is to be found
```

```
C     satisfy mass balances
      DO 50 I = 1,4
        F(I) = CC0(I)-CC(I)   ! [kmol.m-3]
 50   CONTINUE
C
C     satisfy equilibrium relationships
      DO 60 I = 5,12
        F(I) = LKA0(I)-LKA(I)
 60   CONTINUE
C
      RETURN
C
      END
C
C***********************************************************************
C
C Subroutines NS11AD and MB01CD and functions FD05AD, ID05AD and ZA02AS were acquired
C together, upon application, from HSL Archive.  Any changes to the original routines
C are indicated.
C
C***********************************************************************
C
C Subroutine NS11AD is repeatedly called by subroutine USRTRN to resolve the
C equilibrium state of the system (i.e. to perform the necessary speciation
C calculations).  Computationally, this task amounts to solving a system of a number
C of non-linear algebraic equations or functions (equilibrium and mass balance
C relationships) in an equal number of unknowns (species concentrations).  The
C solution method uses the ideas of Newton-Raphson and Steepest Descent, coupled with
C Broyden's method for improving Jacobian matrices.  An initial estimate of the
C solution (i.e. the set of equilibrium species concentrations) is supplied, as is a
C reasonable step size to be used for approximating derivatives (Jacobian matrix) by
C finite differences.  Subroutine CALFUN is called to calculate each of the function
C values for each updated set of species concentrations.  In addition, subroutine
C MB01CD is invoked to evaluate the inverse of the Jacobian approximation.  A solution
C is accepted when a specified accuracy is achieved.  The simulation is stopped if
C this calculation fails.
C
C COPYRIGHT (c) 1993 AEA Technology and
C Council for the Central Laboratory of the Research Councils
C
C None of the comments in this Copyright notice between the lines
C of asterisks shall be removed or altered in any way.
C
C This Package is intended for compilation without modification,
C so most of the embedded comments have been removed.
C
C ALL USE IS SUBJECT TO LICENCE. For full details of an HSL ARCHIVE
C Licence, see http://hsl.rl.ac.uk/archive/cou.html
C
C Please note that for an HSL ARCHIVE Licence:
C
C 1. The Package must not be copied for use by any other person.
C    Supply of any part of the library by the Licensee to a third party
C    shall be subject to prior written agreement between AEA
C    Technology plc and the Licensee on suitable terms and conditions,
C    which will include financial conditions.
C 2. All information on the Package is provided to the Licensee on the
C    understanding that the details thereof are confidential.
C 3. All publications issued by the Licensee that include results obtained
C    with the help of one or more of the Packages shall acknowledge the
C    use of the Packages. The Licensee will notify the Numerical Analysis
C    Group at Rutherford Appleton Laboratory of any such publication.
C 4. The Packages may be modified by or on behalf of the Licensee
C    for such use in research applications but at no time shall such
C    Packages or modifications thereof become the property of the
C    Licensee. The Licensee shall make available free of charge to the
C    copyright holder for any purpose all information relating to
C    any modification.
C 5. Neither CCLRC nor AEA Technology plc shall be liable for any
C    direct or consequential loss or damage whatsoever arising out of
C    the use of Packages by the Licensee.
C
C***********************************************************************
C
C       Toolpack tool decs employed.
C       NS11AD a modified version of NS01AD with integer workspace
C       argument IW added.
```

```
C       SAVE statements added.
C       NS11CD reference removed.
C       Arg dimensions set to *.
C       DFLOAT -> DBLE.
C       Modified for obsolescent features (Feb 1997)
C
C  EAT 21/6/93 EXTERNAL statement put in for block data so will work on VAXs.
C
      SUBROUTINE NS11AD(N,X,F,AJINV,DSTEP,DMAX,ACC,MAXFUN,IPR,WO,IW)
C
C     modify original routine - include new variable
      DOUBLE PRECISION DENOM
C
      DOUBLE PRECISION ACC,DMAX,DSTEP
      INTEGER IPR,MAXFUN,N
      DOUBLE PRECISION AJINV(N,N),F(*),WO(*),X(*)
      INTEGER IW(N)
      DOUBLE PRECISION ANMULT,DD,DM,DMM,DMULT,DN,DS,DSS,DTEST,DW,FMIN,
     +                 FNP,FSQ,PA,PJ,SP,SPNS,SS,TINC
      INTEGER I,IC,IPRINT,IS,ITER,J,K,KK,KS,MAXC,MW,ND,NDC,NF,NT,NTEST,
     +        NW,NX
      EXTERNAL CALFUN,MB01CD
      EXTERNAL NS11CD
      INTRINSIC DABS,DBLE,DMAX1,DMIN1,DSQRT,MOD
      COMMON /NS11BD/LP,LPD,IERR
      INTEGER IERR,LP,LPD
      SAVE /NS11BD/
      MAXC = 0
      IPRINT = IPR
      IERR = 0
      ITER = 0
      IF (LP.LE.0) IPRINT = 0
C      NT = N + 4
C
C     modify original routine - increase NT to prevent error due to failure to improve
C     residuals
      NT = N + 400000000
C
      NTEST = NT
      DTEST = DBLE(N+N) - 0.5
      NX = N*N
      NF = NX + N
      NW = NF + N
      MW = NW + N
      NDC = MW + N
      ND = NDC + N
      FMIN = 0.0D0
      DD = 0.0D0
      DSS = DSTEP*DSTEP
      DM = DMAX*DMAX
      DMM = 4.0D0*DM
      IS = 5
      TINC = 1.0D0
      IF (IPRINT.NE.0) WRITE (LP,FMT=86)
   86 FORMAT ('1')
    1 MAXC = MAXC + 1
      CALL CALFUN(N,X,F)
      FSQ = 0.0D0
      DO 2 I = 1,N
        FSQ = FSQ + F(I)*F(I)
    2 CONTINUE
      IF(FSQ-ACC.GT.0) GO TO 4
    3 IF (IPRINT.EQ.0) GO TO 5
      WRITE (LP,FMT=7) MAXC
    7 FORMAT (/,/,/,5X,'THE FINAL SOLUTION CALCULATED BY NS11A REQUIRED'
     +       ,I5,' CALLS OF CALFUN, AND IS')
      WRITE (LP,FMT=8) (I,X(I),F(I),I=1,N)
    8 FORMAT (/,/,4X,'I',7X,'X(I)',12X,'F(I)',/,/, (I5,2D24.16))
      WRITE (LP,FMT=9) FSQ
    9 FORMAT (/,5X,'THE SUM OF SQUARES IS',D24.16)
C     5 RETURN
C
C     modify original routine - stop simulation if error in speciation calculation
    5 IF (IERR.EQ.0) THEN
        RETURN
      ELSE
        WRITE(99,*) IERR,'ERROR IN SPECIATION CALCULATION'
```

```
        STOP 'ERROR IN SPECIATION CALCULATION'
      END IF
C
    4 GO TO (10,11,11,10,11) IS
   10 IF(FSQ-FMIN.LT.0) GO TO 15
      IF(DD-DSS.GT.0) GO TO 11
      NTEST = NTEST - 1
      IF(NTEST.LT.0) GO TO 13
      IF(NTEST.GT.0) GO TO 11
      IF (LPD.GT.0) WRITE (LPD,FMT=16) NT
      IERR = 1
   16 FORMAT (/,/,/,5X,'ERROR RETURN FROM NS11A BECAUSE',I5,
     +       ' CALLS OF CALFUN FAILED TO IMPROVE THE RESIDUALS')
   17 DO 18 I = 1,N
        X(I) = WO(NX+I)
        F(I) = WO(NF+I)
   18 CONTINUE
      FSQ = FMIN
      GO TO 3
   13 IF (LPD.GT.0) WRITE (LPD,FMT=19)
      IERR = 2
   19 FORMAT (/,/,/,5X,'ERROR RETURN FROM NS11A BECAUSE F(X) ',
     +       'FAILED TO DECREASE USING A NEW JACOBIAN')
      GO TO 17
   15 NTEST = NT
   11 IF(MAXFUN-MAXC.GT.0) GO TO 22
      IF (LPD.GT.0) WRITE (LPD,FMT=23) MAXC
      IERR = 3
   23 FORMAT (/,/,/,5X,'ERROR RETURN FROM NS11A BECAUSE THERE HAVE BEEN'
     +       ,I5,' CALLS OF CALFUN')
      IF(FSQ-FMIN.LT.0) GO TO 3
      GO TO 17
   22 IF (IPRINT.LE.0) GO TO 24
      IF (MOD(MAXC,IPRINT).NE.0) GO TO 24
      WRITE (LP,FMT=26) MAXC
   26 FORMAT (/,/,/,5X,'AT THE',I5,' TH CALL OF CALFUN WE HAVE')
      WRITE (LP,FMT=8) (I,X(I),F(I),I=1,N)
      WRITE (LP,FMT=9) FSQ
   24 GO TO (27,28,29,87,30) IS
   30 FMIN = FSQ
      DO 31 I = 1,N
        WO(NX+I) = X(I)
        WO(NF+I) = F(I)
   31 CONTINUE
   32 IC = 0
      IS = 3
   33 IC = IC + 1
      X(IC) = X(IC) + DSTEP
      GO TO 1
   29 K = IC
      DO 34 I = 1,N
        WO(K) = (F(I)-WO(NF+I))/DSTEP
        K = K + N
   34 CONTINUE
      X(IC) = WO(NX+IC)
      IF(IC-N.LT.0) GO TO 33
      K = 0
      DO 36 I = 1,N
        DO 37 J = 1,N
          K = K + 1
          AJINV(I,J) = WO(K)
          WO(ND+K) = 0.0D0
   37   CONTINUE
        WO(NDC+K+I) = 1.0D0
        WO(NDC+I) = 1.0D0 + DBLE(N-I)
   36 CONTINUE
      CALL MB01CD(AJINV,N,N,IW,F)
   38 DS = 0.0D0
      DN = 0.0D0
      SP = 0.0D0
      ITER = ITER + 1
      IF (IPRINT.GE.0) GO TO 6
      IF (MOD(ITER,-IPRINT).NE.0) GO TO 6
      WRITE (LP,FMT=25) ITER
   25 FORMAT (/,/,/,5X,'AT THE',I5,
     + ' TH ITERATION THE LEAST CALCULATED SUM OF SQUARES IS AS FOLLOWS'
     +       )
```

```
      WRITE (LP,FMT=8) (I,WO(NX+I),WO(NF+I),I=1,N)
      WRITE (LP,FMT=9) FMIN
    6 DO 39 I = 1,N
        X(I) = 0.0D0
        F(I) = 0.0D0
        K = I
        DO 40 J = 1,N
          X(I) = X(I) - WO(K)*WO(NF+J)
          F(I) = F(I) - AJINV(I,J)*WO(NF+J)
          K = K + N
   40   CONTINUE
        DS = DS + X(I)*X(I)
        DN = DN + F(I)*F(I)
        SP = SP + X(I)*F(I)
   39 CONTINUE
      IF(FMIN*FMIN-DMM*DS.LE.0) GO TO 41
      GO TO (43,43,44) IS
   44 IF (LPD.GT.0) WRITE (LPD,FMT=45)
      IERR = 4
   45 FORMAT (/,/,/,5X,'ERROR RETURN FROM NS11A BECAUSE A NEARBY ',
     +        'STATIONARY POINT OF F(X) IS PREDICTED')
      GO TO 17
   43 NTEST = 0
      DO 46 I = 1,N
        X(I) = WO(NX+I)
   46 CONTINUE
      GO TO 32
   41 IS = 2
      IF(DN-DD.GT.0) GO TO 48
      DD = DMAX1(DN,DSS)
      DS = 2.5D-1*DN
      TINC = 1.0D0
      IF(DN-DSS.GE.0) GO TO 58
      IS = 4
      GO TO 80
   48 K = 0
      DMULT = 0.0D0
      DO 51 I = 1,N
        DW = 0.0D0
        DO 52 J = 1,N
          K = K + 1
          DW = DW + WO(K)*X(J)
   52   CONTINUE
        DMULT = DMULT + DW*DW
   51 CONTINUE
      DMULT = DS/DMULT
      DS = DS*DMULT*DMULT
      IF(DS-DD.LT.0) GO TO 53
      IF(DD.GT.0) GO TO 56
      DD = DMAX1(DSS,DMIN1(DM,DS))
      DS = DS/ (DMULT*DMULT)
      GO TO 41
   56 ANMULT = 0.0D0
      DMULT = DMULT*DSQRT(DD/DS)
      GO TO 98
   53 SP = SP*DMULT
C      ANMULT = (DD-DS)/ ((SP-DS)+DSQRT((SP-DD)**2+ (DN-DD)* (DD-DS)))
C
C     modify original routine - introduce and monitor DENOM to prevent floating error
      DENOM = ((SP-DS)+DSQRT((SP-DD)**2+ (DN-DD)* (DD-DS)))
      IF (DENOM.EQ.0.0) THEN
        DENOM = 1.0E-5
        WRITE(99,*) 'INFO: ZERO DENOMINATOR CORRECTED'
      END IF
      ANMULT = (DD-DS)/DENOM
C
      DMULT = DMULT* (1.0D0-ANMULT)
   98 DN = 0.0D0
      SP = 0.0D0
      DO 57 I = 1,N
        F(I) = DMULT*X(I) + ANMULT*F(I)
        DN = DN + F(I)*F(I)
        SP = SP + F(I)*WO(ND+I)
   57 CONTINUE
      DS = 2.5D-1*DN
      IF(WO(NDC+1)-DTEST.LE.0) GO TO 58
      IF(SP*SP-DS.LT.0) GO TO 60
```

```
      GO TO 58
   50 IS = 2
   60 DO 61 I = 1,N
        X(I) = WO(NX+I) + DSTEP*WO(ND+I)
        WO(NDC+I) = WO(NDC+I+1) + 1.0D0
   61 CONTINUE
      WO(ND) = 1.0D0
      DO 62 I = 1,N
        K = ND + I
        SP = WO(K)
        IF (N.LT.2) GO TO 62
        DO 63 J = 2,N
          WO(K) = WO(K+N)
          K = K + N
   63   CONTINUE
        WO(K) = SP
   62 CONTINUE
      GO TO 1
   58 SP = 0.0D0
      K = ND
      DO 64 I = 1,N
        X(I) = DW
        DW = 0.0D0
        DO 65 J = 1,N
          K = K + 1
          DW = DW + F(J)*WO(K)
   65   CONTINUE
        GO TO (68,66) IS
   66   WO(NDC+I) = WO(NDC+I) + 1.0D0
        SP = SP + DW*DW
        IF(SP-DS.LE.0) GO TO 64
        IS = 1
        KK = I
        X(1) = DW
        GO TO 69
   68   X(I) = DW
   69   WO(NDC+I) = WO(NDC+I+1) + 1.0D0
   64 CONTINUE
      WO(ND) = 1.0D0
      IF(KK-1.LE.0) GO TO 70
      KS = NDC + KK*N
      DO 72 I = 1,N
        K = KS + I
        SP = WO(K)
        DO 73 J = 2,KK
          WO(K) = WO(K-N)
          K = K - N
   73   CONTINUE
        WO(K) = SP
   72 CONTINUE
   70 DO 74 I = 1,N
        WO(NW+I) = 0.0D0
   74 CONTINUE
      SP = X(1)*X(1)
      K = ND
      IF (N.LT.2) GO TO 99
      DO 75 I = 2,N
        DS = DSQRT(SP* (SP+X(I)*X(I)))
        DW = SP/DS
        DS = X(I)/DS
        SP = SP + X(I)*X(I)
        DO 76 J = 1,N
          K = K + 1
          WO(NW+J) = WO(NW+J) + X(I-1)*WO(K)
          WO(K) = DW*WO(K+N) - DS*WO(NW+J)
   76   CONTINUE
   75 CONTINUE
   99 SP = 1.0D0/DSQRT(DN)
      DO 77 I = 1,N
        K = K + 1
        WO(K) = SP*F(I)
   77 CONTINUE
   80 FNP = 0.0D0
      K = 0
      DO 78 I = 1,N
        X(I) = WO(NX+I) + F(I)
        WO(NW+I) = WO(NF+I)
```

```
      DO 79 J = 1,N
        K = K + 1
        WO(NW+I) = WO(NW+I) + WO(K)*F(J)
79    CONTINUE
      FNP = FNP + WO(NW+I)**2
78  CONTINUE
    GO TO 1
27  DMULT = 9.0D-1*FMIN + 1.0D-1*FNP - FSQ
    IF(DMULT.GE.0) GO TO 81
    DD = DMAX1(DSS,2.5D-1*DD)
    TINC = 1.0D0
    IF(FSQ-FMIN.LT.0) GO TO 83
    GO TO 28
81  SP = 0.0D0
    SS = 0.0D0
    DO 84 I = 1,N
      SP = SP + DABS(F(I)* (F(I)-WO(NW+I)))
      SS = SS + (F(I)-WO(NW+I))**2
84  CONTINUE
    PJ = 1. + DMULT/ (SP+DSQRT(SP*SP+DMULT*SS))
    SPNS = 4.0D0
    SP = DMIN1(SPNS,TINC,PJ)
    TINC = PJ/SP
    DD = DMIN1(DM,SP*DD)
    GO TO 83
87  IF(FSQ-FMIN.GE.0) GO TO 50
83  FMIN = FSQ
    DO 88 I = 1,N
      SP = X(I)
      X(I) = WO(NX+I)
      WO(NX+I) = SP
      SP = F(I)
      F(I) = WO(NF+I)
      WO(NF+I) = SP
      WO(NW+I) = -WO(NW+I)
88  CONTINUE
    IF(IS-1.GT.0) GO TO 50
28  DO 89 I = 1,N
      X(I) = X(I) - WO(NX+I)
      F(I) = F(I) - WO(NF+I)
89  CONTINUE
    K = 0
    DO 90 I = 1,N
      WO(MW+I) = X(I)
      WO(NW+I) = F(I)
      DO 91 J = 1,N
        WO(MW+I) = WO(MW+I) - AJINV(I,J)*F(J)
        K = K + 1
        WO(NW+I) = WO(NW+I) - WO(K)*X(J)
91    CONTINUE
90  CONTINUE
    SP = 0.0D0
    SS = 0.0D0
    DO 92 I = 1,N
      DS = 0.0D0
      DO 93 J = 1,N
        DS = DS + AJINV(J,I)*X(J)
93    CONTINUE
      SP = SP + DS*F(I)
      SS = SS + X(I)*X(I)
      F(I) = DS
92  CONTINUE
    DMULT = 1.0D0
    IF(DABS(SP)-0.1*SS.GE.0) GO TO 95
    DMULT = 8.0D-1
95  PJ = DMULT/SS
    PA = DMULT/ (DMULT*SP+ (1.-DMULT)*SS)
    K = 0
    DO 96 I = 1,N
      SP = PJ*WO(NW+I)
      SS = PA*WO(MW+I)
      DO 97 J = 1,N
        K = K + 1
        WO(K) = WO(K) + SP*X(J)
        AJINV(I,J) = AJINV(I,J) + SS*F(J)
97    CONTINUE
96  CONTINUE
```

```
      GO TO 38
      END
      BLOCK DATA NS11CD
      COMMON /NS11BD/LP,LPD,IERR
      INTEGER IERR,LP,LPD
      SAVE /NS11BD/
      DATA LP/6/,LPD/6/
      END
C
C***********************************************************************
C
C Subroutine MB01CD is called by subroutine NS11AD to evaluate the inverse of the real
C matrix of estimated partial derivatives.  The method is one of simple Gaussian
C elimination with row interchanges and double precision accumulation of inner
C products.
C
C COPYRIGHT (c) 1992 AEA Technology
C
C None of the comments in this Copyright notice between the lines
C of asterisks shall be removed or altered in any way.
C
C This Package is intended for compilation without modification,
C so most of the embedded comments have been removed.
C
C ALL USE IS SUBJECT TO LICENCE. For full details of an HSL ARCHIVE
C Licence, see http://hsl.rl.ac.uk/archive/cou.html
C
C Please note that for an HSL ARCHIVE Licence:
C
C 1. The Package must not be copied for use by any other person.
C    Supply of any part of the library by the Licensee to a third party
C    shall be subject to prior written agreement between AEA
C    Technology plc and the Licensee on suitable terms and conditions,
C    which will include financial conditions.
C 2. All information on the Package is provided to the Licensee on the
C    understanding that the details thereof are confidential.
C 3. All publications issued by the Licensee that include results obtained
C    with the help of one or more of the Packages shall acknowledge the
C    use of the Packages. The Licensee will notify the Numerical Analysis
C    Group at Rutherford Appleton Laboratory of any such publication.
C 4. The Packages may be modified by or on behalf of the Licensee
C    for such use in research applications but at no time shall such
C    Packages or modifications thereof become the property of the
C    Licensee. The Licensee shall make available free of charge to the
C    copyright holder for any purpose all information relating to
C    any modification.
C 5. Neither CCLRC nor AEA Technology plc shall be liable for any
C    direct or consequential loss or damage whatsoever arising out of
C    the use of Packages by the Licensee.
C
C***********************************************************************
C
C       DATE 7 Dec 1992
C       Toolpack tool decs employed.
C       MB01ED reference removed.
C       ZERO and ONE made PARAMETER.
C       FM02AD and W1 references removed.
C       SAVE statements added.
C
C  EAT 21/6/93 EXTERNAL statement put in for block data so will work on VAXs.
C
      SUBROUTINE MB01CD(A,M,IA,IND,C)
      DOUBLE PRECISION ZERO,ONE
      PARAMETER (ZERO=0.0D0,ONE=1.0D0)
      INTEGER IA,M
      DOUBLE PRECISION A(IA,M),C(M)
      INTEGER IND(M)
      DOUBLE PRECISION AMAX,DIV,SCPROD,STO,WO
      INTEGER I,I1,IMAX,IPROD,ISTO,IW,J,J1,K,M1
      INTRINSIC DABS
      COMMON /MB01DD/LP,IFLAG
      INTEGER IFLAG,LP
      EXTERNAL MB01ED
      SAVE /MB01DD/
      IFLAG = 0
      IF (M-1) 50,2,3
    2 IF (A(1,1).EQ.ZERO) GO TO 60
```

```
      A(1,1) = ONE/A(1,1)
      GO TO 99
    3 M1 = M - 1
      AMAX = ZERO
      DO 32 I = 1,M
        IND(I) = I
        IF (DABS(A(I,1))-DABS(AMAX)) 32,32,31
   31   AMAX = A(I,1)
        IMAX = I
   32 CONTINUE
      IF (AMAX.EQ.ZERO) GO TO 60
      DO 41 J = 1,M1
        IF (IMAX-J) 35,35,33
   33   IW = IND(IMAX)
        IND(IMAX) = IND(J)
        IND(J) = IW
        DO 34 K = 1,M
          WO = A(IMAX,K)
          A(IMAX,K) = A(J,K)
          A(J,K) = WO
   34   CONTINUE
   35   J1 = J + 1
        IF (J.EQ.1) GO TO 38
C IBMD IGNORE RECRDEPS
        DO 37 I = J1,M
          SCPROD = ZERO
          DO 137 IPROD = 1,J - 1
            SCPROD = SCPROD + A(J,IPROD)*A(IPROD,I)
  137     CONTINUE
          A(J,I) = A(J,I) - SCPROD
   37   CONTINUE
   38   DIV = ONE/AMAX
        DO 440 I = J1,M
          A(I,J) = A(I,J)*DIV
  440   CONTINUE
C IBMD IGNORE RECRDEPS
        DO 40 I = J1,M
          SCPROD = ZERO
          DO 139 IPROD = 1,J
            SCPROD = SCPROD + A(I,IPROD)*A(IPROD,J+1)
  139     CONTINUE
          A(I,J+1) = A(I,J+1) - SCPROD
   40   CONTINUE
        AMAX = ZERO
        DO 240 I = J1,M
          IF (DABS(A(I,J1))-DABS(AMAX)) 240,240,239
  239     AMAX = A(I,J1)
          IMAX = I
  240   CONTINUE
        IF (AMAX.EQ.ZERO) GO TO 60
   41 CONTINUE
      DO 13 I1 = 1,M1
        I = M + 1 - I1
        C(I-1) = -A(I,I-1)
        DO 1011 J = I - 2,1,-1
          SCPROD = ZERO
          DO 1109 IPROD = J + 1,I - 1
            SCPROD = SCPROD + A(IPROD,J)*C(IPROD)
 1109     CONTINUE
          C(J) = -A(I,J) - SCPROD
 1011   CONTINUE
        DO 12 K = 1,I - 1
          A(I,K) = C(K)
   12   CONTINUE
   13 CONTINUE
      WO = ONE/A(M,M)
      DO 320 J = 1,M - 1
        C(J) = A(M,J)
  320 CONTINUE
      C(M) = ONE
      DO 21 J = 1,M
        A(M,J) = C(J)*WO
   21 CONTINUE
      DO 122 I = M - 1,1,-1
        WO = ONE/A(I,I)
        DO 120 J = 1,I - 1
          SCPROD = ZERO
```

```
          DO 118 IPROD = I + 1,M
            SCPROD = SCPROD + A(I,IPROD)*A(IPROD,J)
  118     CONTINUE
          C(J) = A(I,J) - SCPROD
  120   CONTINUE
        SCPROD = ZERO
        DO 1118 IPROD = I + 1,M
          SCPROD = SCPROD + A(I,IPROD)*A(IPROD,I)
 1118   CONTINUE
        C(I) = ONE - SCPROD
        DO 2120 J = I + 1,M
          SCPROD = ZERO
          DO 2118 IPROD = I + 1,M
            SCPROD = SCPROD + A(I,IPROD)*A(IPROD,J)
 2118     CONTINUE
          C(J) = -SCPROD
 2120   CONTINUE
        DO 121 J = 1,M
          A(I,J) = C(J)*WO
  121   CONTINUE
  122 CONTINUE
      DO 26 I = 1,M
   23   IF (IND(I)-I) 24,26,24
   24   J = IND(I)
        DO 25 K = 1,M
          STO = A(K,I)
          A(K,I) = A(K,J)
          A(K,J) = STO
   25   CONTINUE
        ISTO = IND(J)
        IND(J) = J
        IND(I) = ISTO
        GO TO 23
   26 CONTINUE
      GO TO 99
   50 IF (LP.GT.0) WRITE (LP,FMT=55)
   55 FORMAT (51H ERROR RETURN FROM MB01CD BECAUSE M IS NOT POSITIVE)
      IFLAG = 1
      GO TO 99
C    60 IF (LP.GT.0) WRITE (LP,FMT=65)
C    65 FORMAT (52H ERROR RETURN FROM MB01CD BECAUSE MATRIX IS SINGULAR)
C       IFLAG = 2
C
C     modify original routine - suppress repeated message output
   60 IF (LP.GT.0) GO TO 65
   65 IFLAG = 2
C
   99 RETURN
      END
      BLOCK DATA MB01ED
      COMMON /MB01DD/LP,IFLAG
      INTEGER IFLAG,LP
      SAVE /MB01DD/
      DATA LP/6/
      END
C
C***********************************************************************
C
C Function FD05AD may be called to access real-valued machine constants relating to
C floating-point storage and arithmetic for double precision computations.
C
C COPYRIGHT (c) 1993 AEA Technology
C       DATE 21 Jan 1993
C       Toolpack tool decs employed.
C       SAVE statement added.
C 1/10/98 DC(3) not initialized to avoid SUN f90 failure
C
      DOUBLE PRECISION FUNCTION FD05AD(INUM)
C----------------------------------------------------------------
C  Real constants for: IEEE double precision (8-byte arithmetic).
C
C  Obtained from H.S.L. subroutine ZE02AM.
C  Nick Gould and Sid Marlow, Harwell Laboratory, April 1988.
C----------------------------------------------------------------
C     .. Scalar Arguments ..
      INTEGER INUM
C
```

```
C     .. Local Arrays ..
      DOUBLE PRECISION DC(5)
C
C     .. Save statement ..
      SAVE DC
C
C     .. Data statements ..
C
C  DC(1) THE SMALLEST POSITIVE NUMBER: 1.0 + DC(1) > 1.0.
C  DC(2) THE SMALLEST POSITIVE NUMBER: 1.0 - DC(2) < 1.0.
C  DC(3) THE SMALLEST NONZERO +VE REAL NUMBER.
C  DC(4) THE SMALLEST FULL PRECISION +VE REAL NUMBER.
C  DC(5) THE LARGEST FINITE +VE REAL NUMBER.
C
      DATA DC(1)/2.2204460492504D-16/
      DATA DC(2)/1.1102230246253D-16/
C     DATA DC(3)/4.9406564584126D-324/
      DATA DC(4)/2.2250738585073D-308/
      DATA DC(5)/1.7976931348622D+308/
C
C     .. Executable Statements ..
      IF (INUM.LE.0 .OR. INUM.GE.6) THEN
        WRITE(6, 2000)INUM
        STOP
      ELSE IF (INUM.EQ.3) THEN
        FD05AD = DC(4)/2.0D0**52
      ELSE
        FD05AD = DC(INUM)
      END IF
C
      RETURN
C
 2000 FORMAT (' INUM =',I3,' OUT OF RANGE IN FD05AD.',
     +       ' EXECUTION TERMINATED.')
C
      END
C
C*********************************************************************
C
C Function ID05AD may be called to access integer-valued machine constants relating to
C floating-point storage and arithmetic for double precision computations.
C
C COPYRIGHT (c) 1993 AEA Technology
C       DATE 4 Feb 1993
C       Toolpack tool decs employed.
C       SAVE statements added.
C
      INTEGER FUNCTION ID05AD(INUM)
C-----------------------------------------------------------------
C  Integer constants for: IEEE double precision (8-byte arithmetic).
C
C  Obtained from H.S.L. subroutine ZE02AM.
C  Nick Gould and Sid Marlow, Harwell Laboratory, April 1988.
C-----------------------------------------------------------------
C     .. Scalar Arguments ..
      INTEGER INUM
C
C     .. Local Arrays ..
      INTEGER IC(10)
C
C     .. Save statement ..
      SAVE IC
C
C     .. Data statements ..
C
C  IC(1) THE BASE (RADIX) OF THE FLOATING-POINT ARITHMETIC.
C  IC(2) THE NUMBER OF BASE IC(1) DIGITS IN THE SIGNIFICAND.
C  IC(3) THE NUMBER OF BITS USED FOR THE EXPONENT
C  IC(4) = 0 FLOATING-POINT ADDITION CHOPS, = 1 IT ROUNDS.
C  IC(5) = 0 A GUARD DIGIT IS NOT USED FOR *, = 1 IT IS.
C  IC(6) LARGEST -VE INTEGER:1.0 + DBLE(IC(1))**IC(6) > 1.0.
C  IC(7) LARGEST -VE INTEGER:1.0 - DBLE(IC(1))**IC(7) < 1.0.
C  IC(8) LARGEST -VE INTEGER: DBLE(IC(1))**IC(8) > 0.0.
C  IC(9) LARGEST -VE INTEGER: REAL(IC(1))**IC(9) IS NORMAL.
C  IC(10) LARGEST +VE INTEGER: REAL(IC(1))**IC(10) FINITE.
C
      DATA IC(1)/2/
```

```
      DATA IC(2)/53/
      DATA IC(3)/11/
      DATA IC(4)/1/
      DATA IC(5)/1/
      DATA IC(6)/-52/
      DATA IC(7)/-53/
      DATA IC(8)/-1074/
      DATA IC(9)/-1022/
      DATA IC(10)/1023/
C
C     .. Executable Statements ..
      IF (INUM.LE.0 .OR. INUM.GE.11) THEN
        WRITE(6, 2000)INUM
        STOP
C
      ELSE
        ID05AD = IC(INUM)
      END IF
C
      RETURN
C
 2000 FORMAT (' INUM =',I3,' OUT OF RANGE IN ID05AD.',
     +       ' EXECUTION TERMINATED.')
C
      END
C
C***********************************************************************
C
C Function ZA02AS may be called to access the CPU clock reading.
C
C COPYRIGHT (c) 1993 AEA Technology and
C Council for the Central Laboratory of the Research Councils
C       Toolpack tool decs employed.
C
      REAL FUNCTION ZA02AS(X)
C
C  Gives the CPU time in seconds for a Generic Unix machine.
C  This version is appropriate for Compaq/Dec, HP, Silicon Graphics,
C  Sun and Intel(Linux) machines, using default compilers.
C
C     .. Scalar Arguments ..
      REAL X
C
C     .. Local Arrays ..
      REAL RRTIME(2)
C
C     .. External Functions ..
      REAL ETIME
      EXTERNAL ETIME
C
C     .. Executable Statements ..
C
C     DTIME CAN BE USED FOR TIME DIFFERENCE.
      ZA02AS = ETIME(RRTIME)
      RETURN
C
      END
C
C***********************************************************************
C
C Subroutine USRCVG is called by CFX-4.3 Solver at the end of each outer iteration to
C check the value of the logical flag which indicates whether or not the problem is
C satisfactorily converged.  Currently, the number of required iterations is simply
C specified.  As experience demonstrates necessary, this number is increased for a
C selected initial period.  Specification of a particular residual tolerance could
C prove to be a little hit-or-miss if highly variable species and particle number
C concentrations are the only residuals calculated (i.e. the only transport equations
C solved); a further difficulty is that the process modelled is semi-batch.  However,
C the system's chlorine content is constant, and, although the implementation of most
C dilution options makes it appear to CFX-4.3 Solver as if this quantity is slowly
C decreasing, setting a suitable tolerance on the sum of the residuals of the chlorine
C species concentrations (inserting appropriate stoichiometric coefficients) is
C nevertheless a viable alternative.  A more rigorous, structured approach such as
C this potentially yields improvements in accuracy and efficiency.
C
      SUBROUTINE USRCVG(U,V,W,P,VFRAC,DEN,VIS,TE,ED,RS,T,H,RF,SCAL,XP,
     +                  YP,ZP,VOL,AREA,VPOR,ARPOR,WFACT,CONV,IPT,IBLK,
```

```
     +                  IPVERT,IPNODN,IPFACN,IPNODF,IPNODB,IPFACB,CMETH,
     +                  MNSL,MXSL,RDFC,RESOR,URFVAR,LCONVG,WORK,IWORK,
     +                  CWORK)
C
C***********************************************************************
C
C   THIS SUBROUTINE ALLOWS USERS TO MONITOR CONVERGENCE, ALTER
C   UNDER RELAXATION FACTORS, REDUCTION FACTORS ETC
C   AND WRITE SOLUTION DATA AS A FUNCTION OF ITERATION
C
C   >>> IMPORTANT                                                  <<<
C   >>>                                                           <<<
C   >>> USERS MAY ONLY ADD OR ALTER PARTS OF THE SUBROUTINE WITHIN  <<<
C   >>> THE DESIGNATED USER AREAS                                  <<<
C
C***********************************************************************
C
C   THIS SUBROUTINE IS CALLED BY THE FOLLOWING SUBROUTINE
C         CUSR  CVGTST
C
C***********************************************************************
C
C   CREATED
C      09/12/88  ADB
C   MODIFIED
C      08/08/91  IRH  NEW STRUCTURE
C      03/09/91  IRH  ADD CONV TO ARGUMENT LIST
C      23/09/91  IRH  ADD USEFUL COMMON BLOCKS
C      29/11/91  PHA  UPDATE CALLED BY COMMENT, ADD RF ARGUMENT,
C                     CHANGE LAST DIMENSION OF RS TO 6 AND IVERS TO 2
C      03/06/92  PHA  ADD PRECISION FLAG AND CHANGE IVERS TO 3
C      07/07/92  IRH  CORRECT EXAMPLE
C      23/11/93  CSH  EXPLICITLY DIMENSION IPVERT ETC.
C      03/02/94  PHA  CHANGE FLOW3D TO CFDS-FLOW3D
C      03/03/94  FHW  CORRECTION OF SPELLING MISTAKE
C      22/08/94  NSW  MOVE 'IF(IUSED.EQ.0) RETURN' OUT OF USER AREA
C      19/12/94  NSW  CHANGE FOR CFX-F3D
C      25/03/96  NSW  CORRECT MAXIMUM VELOCITY EXAMPLE
C      02/07/97  NSW  UPDATE FOR CFX-4
C
C***********************************************************************
C
C   SUBROUTINE ARGUMENTS
C
C     U       - U COMPONENT OF VELOCITY
C     V       - V COMPONENT OF VELOCITY
C     W       - W COMPONENT OF VELOCITY
C     P       - PRESSURE
C     VFRAC   - VOLUME FRACTION
C     DEN     - DENSITY OF FLUID
C     VIS     - VISCOSITY OF FLUID
C     TE      - TURBULENT KINETIC ENERGY
C     ED      - EPSILON
C     RS      - REYNOLD STRESSES
C     T       - TEMPERATURE
C     H       - ENTHALPY
C     RF      - REYNOLD FLUXES
C     SCAL    - SCALARS (THE FIRST 'NCONC' OF THESE ARE MASS FRACTIONS)
C     XP      - X COORDINATES OF CELL CENTRES
C     YP      - Y COORDINATES OF CELL CENTRES
C     ZP      - Z COORDINATES OF CELL CENTRES
C     VOL     - VOLUME OF CELLS
C     AREA    - AREA OF CELLS
C     VPOR    - POROUS VOLUME
C     ARPOR   - POROUS AREA
C     WFACT   - WEIGHT FACTORS
C
C     IPT     - 1D POINTER ARRAY
C     IBLK    - BLOCK SIZE INFORMATION
C     IPVERT  - POINTER FROM CELL CENTRES TO 8 NEIGHBOURING VERTICES
C     IPNODN  - POINTER FROM CELL CENTRES TO 6 NEIGHBOURING CELLS
C     IPFACN  - POINTER FROM CELL CENTRES TO 6 NEIGHBOURING FACES
C     IPNODF  - POINTER FROM CELL FACES TO 2 NEIGHBOURING CELL CENTRES
C     IPNODB  - POINTER FROM BOUNDARY CENTRES TO CELL CENTRES
C     IPFACB  - POINTER FROM BOUNDARY CENTRES TO BOUNDARY FACES
C
C     CMETH   - SOLUTION METHOD
```

```
C      MNSL   - MINIMUM NUMBER OF SWEEPS
C      MXSL   - MAXIMUM NUMBER OF SWEEPS
C      RDFC   - REDUCTION FACTORS REQUIRED
C      RESOR  - NON LINEAR RESIDUALS
C      URFVAR - UNDER RELAXATION FACTORS
C    * LCONVG - LOGICAL CONVERGENCE FLAG
C
C      WORK   - REAL WORKSPACE ARRAY
C      IWORK  - INTEGER WORKSPACE ARRAY
C      CWORK  - CHARACTER WORKSPACE ARRAY
C
C    SUBROUTINE ARGUMENTS PRECEDED WITH A '*' ARE ARGUMENTS THAT MUST
C    BE SET BY THE USER IN THIS ROUTINE.
C
C    NOTE THAT OTHER DATA MAY BE OBTAINED FROM CFX-4 USING THE
C    ROUTINE GETADD, FOR FURTHER DETAILS SEE THE VERSION 4
C    USER MANUAL.
C
C*********************************************************************
C
      DOUBLE PRECISION U
      DOUBLE PRECISION V
      DOUBLE PRECISION W
      DOUBLE PRECISION P
      DOUBLE PRECISION VFRAC
      DOUBLE PRECISION DEN
      DOUBLE PRECISION VIS
      DOUBLE PRECISION TE
      DOUBLE PRECISION ED
      DOUBLE PRECISION RS
      DOUBLE PRECISION T
      DOUBLE PRECISION H
      DOUBLE PRECISION RF
      DOUBLE PRECISION SCAL
      DOUBLE PRECISION XP
      DOUBLE PRECISION YP
      DOUBLE PRECISION ZP
      DOUBLE PRECISION VOL
      DOUBLE PRECISION AREA
      DOUBLE PRECISION VPOR
      DOUBLE PRECISION ARPOR
      DOUBLE PRECISION WFACT
      DOUBLE PRECISION CONV
      DOUBLE PRECISION RDFC
      DOUBLE PRECISION RESOR
      DOUBLE PRECISION URFVAR
      DOUBLE PRECISION WORK
      DOUBLE PRECISION SMALL
      DOUBLE PRECISION SORMAX
      DOUBLE PRECISION TIME
      DOUBLE PRECISION DT
      DOUBLE PRECISION DTINVF
      DOUBLE PRECISION TPARM
      LOGICAL LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP,LRECT,LCYN,LAXIS,
     +        LPOROS,LTRANS
      LOGICAL LCONVG
C
      CHARACTER*(*) CMETH,CWORK
C
C+++++++++++++++++ USER AREA 1 +++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS EXPLICITLY DECLARED VARIABLES
C
C+++++++++++++++++ END OF USER AREA 1 +++++++++++++++++++++++++++++++++
C
      COMMON /ALL/NBLOCK,NCELL,NBDRY,NNODE,NFACE,NVERT,NDIM,
     +       /ALLWRK/NRWS,NIWS,NCWS,IWRFRE,IWIFRE,IWCFRE,/ADDIMS/NPHASE,
     +       NSCAL,NVAR,NPROP,NDVAR,NDPROP,NDXNN,NDGEOM,NDCOEF,NILIST,
     +       NRLIST,NTOPOL,/CHKUSR/IVERS,IUCALL,IUSED,/DEVICE/NREAD,
     +       NWRITE,NRDISK,NWDISK,/IDUM/ILEN,JLEN,/LOGIC/LDEN,LVIS,
     +       LTURB,LTEMP,LBUOY,LSCAL,LCOMP,LRECT,LCYN,LAXIS,LPOROS,
     +       LTRANS,/MLTGRD/MLEVEL,NLEVEL,ILEVEL,/RESID/IRESID,NRESID,
     +       /SGLDBL/IFLGPR,ICHKPR,/SPARM/SMALL,SORMAX,NITER,INDPRI,
     +       MAXIT,NODREF,NODMON,/TRANSI/NSTEP,KSTEP,MF,INCORE,
     +       /TRANSR/TIME,DT,DTINVF,TPARM
C
C+++++++++++++++++ USER AREA 2 +++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DECLARE THEIR OWN COMMON BLOCKS
```

```
C .    THESE SHOULD START WITH THE CHARACTERS 'UC' TO ENSURE
C      NO CONFLICT WITH NON-USER COMMON BLOCKS
C
C---- COMMON BLOCK FOR EXAMPLE IN USER AREA 6
C      COMMON /UC1/ VELOLD
C
C+++++++++++++++++ END OF USER AREA 2 +++++++++++++++++++++++++++++++++
C
       DIMENSION CMETH(NVAR,NPHASE),MNSL(NVAR,NPHASE),MXSL(NVAR,NPHASE),
      +          RDFC(NVAR,NPHASE),RESOR(NVAR,NPHASE),URFVAR(NVAR,NPHASE)
       DIMENSION U(NNODE,NPHASE),V(NNODE,NPHASE),W(NNODE,NPHASE),
      +          P(NNODE,NPHASE),VFRAC(NNODE,NPHASE),DEN(NNODE,NPHASE),
      +          VIS(NNODE,NPHASE),TE(NNODE,NPHASE),ED(NNODE,NPHASE),
      +          RS(NNODE,NPHASE,6),T(NNODE,NPHASE),H(NNODE,NPHASE),
      +          RF(NNODE,NPHASE,4),SCAL(NNODE,NPHASE,NSCAL)
       DIMENSION XP(NNODE),YP(NNODE),ZP(NNODE),VOL(NCELL),AREA(NFACE,3),
      +          VPOR(NCELL),ARPOR(NFACE,3),WFACT(NFACE),
      +          CONV(NFACE,NPHASE),IPT(*),IBLK(5,NBLOCK),
      +          IPVERT(NCELL,8),IPNODN(NCELL,6),IPFACN(NCELL,6),
      +          IPNODF(NFACE,4),IPNODB(NBDRY,4),IPFACB(NBDRY),IWORK(*),
      +          WORK(*),CWORK(*)
C
C+++++++++++++++++ USER AREA 3 +++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DIMENSION THEIR ARRAYS
C
C---- AREA FOR USERS TO DEFINE DATA STATEMENTS
C
C+++++++++++++++++ END OF USER AREA 3 +++++++++++++++++++++++++++++++++
C
C---- STATEMENT FUNCTION FOR ADDRESSING
      IP(I,J,K) = IPT((K-1)*ILEN*JLEN+ (J-1)*ILEN+I)
C
C---- VERSION NUMBER OF USER ROUTINE AND PRECISION FLAG
C
      IVERS = 3
      ICHKPR = 2
C
C+++++++++++++++++ USER AREA 4 +++++++++++++++++++++++++++++++++++++++++
C---- TO USE THIS USER ROUTINE FIRST SET IUSED=1
C
      IUSED = 1
C
C+++++++++++++++++ END OF USER AREA 4 +++++++++++++++++++++++++++++++++
C
      IF (IUSED.EQ.0) RETURN
C
C---- FRONTEND CHECKING OF USER ROUTINE
      IF (IUCALL.EQ.0) RETURN
C
C+++++++++++++++++ USER AREA 5 +++++++++++++++++++++++++++++++++++++++++
C
C---- EXAMPLE: (TEST ON MAX OF MASS RESIDUAL AND ENTHALPY RESIDUAL)
C
C      CALL GETVAR('USRCVG','P     ',IPRES)
C      CALL GETVAR('USRCVG','H     ',IH)
C
C      URESM=0.0
C      DO 10 IPHASE=1,NPHASE
C      URESM=MAX(URESM,RESOR(IPRES,IPHASE),RESOR(IH,IPHASE))
C 10   CONTINUE
C
C      LCONVG = URESM .LT. 1.0E-5
C
C---- END OF EXAMPLE
C
C     specify desired number of CFX-4.3 Solver iterations (increased for defined
C     initial period)
      IF (KSTEP.LE.50) THEN
        LCONVG = NITER.EQ.50
      ELSE
        LCONVG = NITER.EQ.10
      END IF
C
C+++++++++++++++++ END OF USER AREA 5 +++++++++++++++++++++++++++++++++
C
C
C+++++++++++++++++ USER AREA 6 +++++++++++++++++++++++++++++++++++++++++
```

```
C
C---- EXAMPLE: MONITOR CHANGE IN MAXIMUM VELOCITY
C              ADJUST UNDER RELAXATION ACCORDINGLY
C
C      VELMAX=0.0
C      DO 20 IPHASE=1,NPHASE
C  USE IPALL TO FIND 1D ADDRESSES OF ALL CELL CENTRES
C      CALL IPALL('*','*','BLOCK','CENTRES',IPT,NPT,CWORK,IWORK)
C  LOOP OVER ALL CELL CENTRE LOCATIONS IN FLOW DOMAIN
C        DO 30 I=1,NPT
C  USE ARRAY IPT TO GET ADDRESS
C        INODE=IPT(I)
C        VELMAX=MAX(VELMAX,ABS(U(INODE,IPHASE)),ABS(V(INODE,IPHASE)),
C     +             ABS(W(INODE,IPHASE)))
C 30     CONTINUE
C 20   CONTINUE
C
C      IF (NITER.GT.1) THEN
C        DVEL=(VELMAX-VELOLD)/VELMAX
C        URFMIN=0.01
C        URFMAX=0.8
C        URF=(1.0-DVEL)*URFMAX+DVEL*URFMIN
C        WRITE(NWRITE,100)NITER,DVEL,URF
C        CALL GETVAR('USRCVG','U     ',IU)
C        CALL GETVAR('USRCVG','V     ',IV)
C        CALL GETVAR('USRCVG','W     ',IW)
C        DO 40 IPHASE=1,NPHASE
C        URFVAR(IU,IPHASE)=URF
C        URFVAR(IV,IPHASE)=URF
C        URFVAR(IW,IPHASE)=URF
C 40     CONTINUE
C      ENDIF
C
C      VELOLD=VELMAX
C
C---- END OF EXAMPLE
C
C+++++++++++++++++ END OF USER AREA 6 +++++++++++++++++++++++++++++++++
C
      RETURN
C
      END
C
C*************************************************************************
C
C Subroutine USRSRC is called by CFX-4.3 Solver twice each outer iteration for each
C transport equation to be solved.  For the user scalar equations representing the
C Na.+ and OH.- dissolved species concentrations, appropriate source terms are
C calculated (such addition of terms is executed at the first call).  These sources
C simulate feed addition and are applied to those cells selected and identified as
C feed patch cells.  The associated water (volume) is neglected - an attempt to
C account for its effect is made in the dilution calculations in subroutine USRTRN.
C
      SUBROUTINE USRSRC(IEQN,ICALL,CNAME,CALIAS,AM,SP,SU,CONV,U,V,W,P,
     +                  VFRAC,DEN,VIS,TE,ED,RS,T,H,RF,SCAL,XP,YP,ZP,VOL,
     +                  AREA,VPOR,ARPOR,WFACT,IPT,IBLK,IPVERT,IPNODN,
     +                  IPFACN,IPNODF,IPNODB,IPFACB,WORK,IWORK,CWORK)
C
C*************************************************************************
C
C   UTILITY SUBROUTINE FOR USER-SUPPLIED SOURCES
C
C   >>> IMPORTANT                                                    <<<
C   >>>                                                            <<<
C   >>> USERS MAY ONLY ADD OR ALTER PARTS OF THE SUBROUTINE WITHIN  <<<
C   >>> THE DESIGNATED USER AREAS                                  <<<
C
C*************************************************************************
C
C   THIS SUBROUTINE IS CALLED BY THE FOLLOWING SUBROUTINES
C      CUSR  SCDF  SCDS  SCED  SCENRG  SCHF  SCMOM  SCPCE  SCSCAL
C      SCTE  SCVF
C
C*************************************************************************
C
C   CREATED
C      08/03/90  ADB
```

```
C   MODIFIED
C      04/03/91  ADB  ALTERED ARGUMENT LIST.
C      28/08/91  IRH  NEW STRUCTURE
C      28/09/91  IRH  CHANGE EXAMPLE + ADD COMMON BLOCKS
C      10/02/92  PHA  UPDATE CALLED BY COMMENT, ADD RF ARGUMENT,
C                     CHANGE LAST DIMENSION OF RS TO 6 AND IVERS TO 2
C      03/06/92  PHA  ADD PRECISION FLAG AND CHANGE IVERS TO 3
C      23/11/93  CSH  EXPLICITLY DIMENSION IPVERT ETC.
C      07/12/93  NSW  INCLUDE CONV IN ARGUMENT LIST AND CHANGE IVERS
C                     TO 4
C      03/02/94  PHA  CHANGE FLOW3D TO CFDS-FLOW3D
C      03/03/94  FHW  CORRECTION OF SPELLING MISTAKE
C      08/03/94  NSW  CORRECT SPELLING
C      09/08/94  NSW  CORRECT SPELLING.
C                     MOVE 'IF(IUSED.EQ.0) RETURN' OUT OF USER AREA.
C                     INCLUDE COMMENT ON MASS SOURCES.
C      19/12/94  NSW  CHANGE FOR CFX-F3D
C      02/07/97  NSW  UPDATE FOR CFX-4
C
C*********************************************************************
C
C   SUBROUTINE ARGUMENTS
C
C     IEQN   - EQUATION NUMBER
C     ICALL  - SUBROUTINE CALL
C     CNAME  - EQUATION NAME
C     CALIAS - ALIAS OF EQUATION NAME
C     AM     - OFF DIAGONAL MATRIX COEFFICIENTS
C     SU     - SU IN LINEARISATION OF SOURCE TERM
C     SP     - SP IN LINEARISATION OF SOURCE TERM
C     CONV   - CONVECTION COEFFICIENTS
C     U      - U COMPONENT OF VELOCITY
C     V      - V COMPONENT OF VELOCITY
C     W      - W COMPONENT OF VELOCITY
C     P      - PRESSURE
C     VFRAC  - VOLUME FRACTION
C     DEN    - DENSITY OF FLUID
C     VIS    - VISCOSITY OF FLUID
C     TE     - TURBULENT KINETIC ENERGY
C     ED     - EPSILON
C     RS     - REYNOLD STRESSES
C     T      - TEMPERATURE
C     H      - ENTHALPY
C     RF     - REYNOLD FLUXES
C     SCAL   - SCALARS (THE FIRST 'NCONC' OF THESE ARE MASS FRACTIONS)
C     XP     - X COORDINATES OF CELL CENTRES
C     YP     - Y COORDINATES OF CELL CENTRES
C     ZP     - Z COORDINATES OF CELL CENTRES
C     VOL    - VOLUME OF CELLS
C     AREA   - AREA OF CELLS
C     VPOR   - POROUS VOLUME
C     ARPOR  - POROUS AREA
C     WFACT  - WEIGHT FACTORS
C
C     IPT    - 1D POINTER ARRAY
C     IBLK   - BLOCK SIZE INFORMATION
C     IPVERT - POINTER FROM CELL CENTRES TO 8 NEIGHBOURING VERTICES
C     IPNODN - POINTER FROM CELL CENTRES TO 6 NEIGHBOURING CELLS
C     IPFACN - POINTER FROM CELL CENTRES TO 6 NEIGHBOURING FACES
C     IPNODF - POINTER FROM CELL FACES TO 2 NEIGHBOURING CELL CENTRES
C     IPNODB - POINTER FROM BOUNDARY CENTRES TO CELL CENTRES
C     IPFACB - POINTER FROM BOUNDARY CENTRES TO BOUNDARY FACESS
C
C     WORK   - REAL WORKSPACE ARRAY
C     IWORK  - INTEGER WORKSPACE ARRAY
C     CWORK  - CHARACTER WORKSPACE ARRAY
C
C   SUBROUTINE ARGUMENTS PRECEDED WITH A '*' ARE ARGUMENTS THAT MUST
C   BE SET BY THE USER IN THIS ROUTINE.
C
C   NOTE THAT WHEN USING MASS SOURCES, THE FLOWS THROUGH MASS FLOW
C   BOUNDARIES ARE UNCHANGED. THE USER SHOULD THEREFORE INCLUDE AT
C   LEAST ONE PRESSURE BOUNDARY FOR SUCH A CALCULATION.
C
C   NOTE THAT OTHER DATA MAY BE OBTAINED FROM CFX-4 USING THE
C   ROUTINE GETADD, FOR FURTHER DETAILS SEE THE VERSION 4
C   USER MANUAL.
```

```
C
C*************************************************************************
C
      DOUBLE PRECISION AM
      DOUBLE PRECISION SP
      DOUBLE PRECISION SU
      DOUBLE PRECISION CONV
      DOUBLE PRECISION U
      DOUBLE PRECISION V
      DOUBLE PRECISION W
      DOUBLE PRECISION P
      DOUBLE PRECISION VFRAC
      DOUBLE PRECISION DEN
      DOUBLE PRECISION VIS
      DOUBLE PRECISION TE
      DOUBLE PRECISION ED
      DOUBLE PRECISION RS
      DOUBLE PRECISION T
      DOUBLE PRECISION H
      DOUBLE PRECISION RF
      DOUBLE PRECISION SCAL
      DOUBLE PRECISION XP
      DOUBLE PRECISION YP
      DOUBLE PRECISION ZP
      DOUBLE PRECISION VOL
      DOUBLE PRECISION AREA
      DOUBLE PRECISION VPOR
      DOUBLE PRECISION ARPOR
      DOUBLE PRECISION WFACT
      DOUBLE PRECISION WORK
      DOUBLE PRECISION SMALL
      DOUBLE PRECISION SORMAX
      DOUBLE PRECISION TIME
      DOUBLE PRECISION DT
      DOUBLE PRECISION DTINVF
      DOUBLE PRECISION TPARM
      LOGICAL LDEN,LVIS,LTURB,LTEMP,LBUOY,LSCAL,LCOMP,LRECT,LCYN,LAXIS,
     +        LPOROS,LTRANS
C
      CHARACTER*(*) CWORK
      CHARACTER CNAME*6,CALIAS*24
C
C+++++++++++++++++ USER AREA 1 ++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS EXPLICITLY DECLARED VARIABLES
C
      DOUBLE PRECISION FEEDCONC,FEEDVOL
      DOUBLE PRECISION FEEDPVOL,GEOMVOL,TOTALVOL
C
      INTEGER I,INODE
      INTEGER GENSET3
C
C+++++++++++++++++ END OF USER AREA 1 +++++++++++++++++++++++++++++++++
C
      COMMON /ALL/NBLOCK,NCELL,NBDRY,NNODE,NFACE,NVERT,NDIM,
     +       /ALLWRK/NRWS,NIWS,NCWS,IWRFRE,IWIFRE,IWCFRE,/ADDIMS/NPHASE,
     +       NSCAL,NVAR,NPROP,NDVAR,NDPROP,NDXNN,NDGEOM,NDCOEF,NILIST,
     +       NRLIST,NTOPOL,/CHKUSR/IVERS,IUCALL,IUSED,/DEVICE/NREAD,
     +       NWRITE,NRDISK,NWDISK,/IDUM/ILEN,JLEN,/LOGIC/LDEN,LVIS,
     +       LTURB,LTEMP,LBUOY,LSCAL,LCOMP,LRECT,LCYN,LAXIS,LPOROS,
     +       LTRANS,/MLTGRD/MLEVEL,NLEVEL,ILEVEL,/SGLDBL/IFLGPR,ICHKPR,
     +       /SPARM/SMALL,SORMAX,NITER,INDPRI,MAXIT,NODREF,NODMON,
     +       /TRANSI/NSTEP,KSTEP,MF,INCORE,/TRANSR/TIME,DT,DTINVF,TPARM
C
C+++++++++++++++++ USER AREA 2 ++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DECLARE THEIR OWN COMMON BLOCKS
C     THESE SHOULD START WITH THE CHARACTERS 'UC' TO ENSURE
C     NO CONFLICT WITH NON-USER COMMON BLOCKS
C
      COMMON /UCFEED1/FEEDCONC,FEEDVOL
      COMMON /UCFEED2/FEEDPVOL,GEOMVOL,TOTALVOL
      COMMON /UCFEED3/GENSET3
C
C+++++++++++++++++ END OF USER AREA 2 +++++++++++++++++++++++++++++++++
C
      DIMENSION AM(NCELL,6,NPHASE),SP(NCELL,NPHASE),SU(NCELL,NPHASE),
     +          CONV(NFACE,NPHASE)
C
```

```
      DIMENSION U(NNODE,NPHASE),V(NNODE,NPHASE),W(NNODE,NPHASE),
     +          P(NNODE,NPHASE),VFRAC(NNODE,NPHASE),DEN(NNODE,NPHASE),
     +          VIS(NNODE,NPHASE),TE(NNODE,NPHASE),ED(NNODE,NPHASE),
     +          RS(NNODE,NPHASE,6),T(NNODE,NPHASE),H(NNODE,NPHASE),
     +          RF(NNODE,NPHASE,4),SCAL(NNODE,NPHASE,NSCAL)
C
      DIMENSION XP(NNODE),YP(NNODE),ZP(NNODE),VOL(NCELL),AREA(NFACE,3),
     +          VPOR(NCELL),ARPOR(NFACE,3),WFACT(NFACE),IPT(*),
     +          IBLK(5,NBLOCK),IPVERT(NCELL,8),IPNODN(NCELL,6),
     +          IPFACN(NCELL,6),IPNODF(NFACE,4),IPNODB(NBDRY,4),
     +          IPFACB(NBDRY),IWORK(*),WORK(*),CWORK(*)
C
C+++++++++++++++++ USER AREA 3 ++++++++++++++++++++++++++++++++++++++++
C---- AREA FOR USERS TO DIMENSION THEIR ARRAYS
C
C---- AREA FOR USERS TO DEFINE DATA STATEMENTS
C
C+++++++++++++++++ END OF USER AREA 3 ++++++++++++++++++++++++++++++++++
C
C---- STATEMENT FUNCTION FOR ADDRESSING
      IP(I,J,K) = IPT((K-1)*ILEN*JLEN+ (J-1)*ILEN+I)
C
C---- VERSION NUMBER OF USER ROUTINE AND PRECISION FLAG
C
      IVERS = 4
      ICHKPR = 2
C
C+++++++++++++++++ USER AREA 4 ++++++++++++++++++++++++++++++++++++++++
C---- TO USE THIS USER ROUTINE FIRST SET IUSED=1
C
      IUSED = 1
C
C+++++++++++++++++ END OF USER AREA 4 ++++++++++++++++++++++++++++++++++
C
      IF (IUSED.EQ.0) RETURN
C
C---- FRONTEND CHECKING OF USER ROUTINE
      IF (IUCALL.EQ.0) RETURN
C
C---- ADD TO SOURCE TERMS
      IF (ICALL.EQ.1) THEN
C
C+++++++++++++++++ USER AREA 5 ++++++++++++++++++++++++++++++++++++++++
C
C---- EXAMPLE (HEAT SOURCE) ADD 100W PER UNIT VOLUME IN BLOCK
C     'BLOCK-NUMBER-2'
C
C  USE IPREC TO FIND ADDRESSES
C
C     CALL IPREC('BLOCK-NUMBER-2','BLOCK','CENTRES',IPT,ILEN,JLEN,KLEN,
C    +           CWORK,IWORK)
C
C  FIND VARIABLE NUMBER FOR ENTHALPY
C     CALL GETVAR('USRSRC','H     ',IVAR)
C  IF ENTHALPY EQUATION ADD SOURCE TERMS
C     IF (IVAR.EQ.IEQN) THEN
C  LOOP OVER PATCH
C     DO 103 K = 1, KLEN
C       DO 102 J = 1, JLEN
C         DO 101 I = 1, ILEN
C  USE STATEMENT FUNCTION IP TO GET ADDRESSES
C          INODE = IP(I,J,K)
C  ADD HEAT SOURCE
C          SU(INODE,1)=SU(INODE,1)+100.0*VOL(INODE)
C 101      CONTINUE
C 102    CONTINUE
C 103   CONTINUE
C     ENDIF
C
C---- END OF EXAMPLE
C
C     if Na.+ or OH.- species concentration equation
      IF (CALIAS.EQ.'NAION'.OR.CALIAS.EQ.'OHION') THEN
C
C       call utility routine IPALL to return cell-centre addresses of those cells
C       occupying feed patch
        CALL IPALL('USER3D_FEED','USER3D','PATCH','CENTRES',IPT,NPT,
```

```
     &       CWORK,IWORK)
C
C       within each cv I in feed patch
        DO 10 I = 1,NPT
C
          INODE = IPT(I)
C
C         if dilution effect isolated, then distributed and teamed with time-dependent
C         feed source term
          IF (GENSET2.EQ.3) THEN
C           calculate feed source to be added [kg.kmol.m-3.s-1]
            SU(INODE,1) = SU(INODE,1)+(DEN(INODE,1)*FEEDVOL*FEEDCONC*
     &                    VOL(INODE)/FEEDPVOL)*GEOMVOL/TOTALVOL
C         if feed source term time-independent
          ELSE
C           calculate feed source to be added [kg.kmol.m-3.s-1]
            SU(INODE,1) = SU(INODE,1)+DEN(INODE,1)*FEEDVOL*FEEDCONC*
     &                    VOL(INODE)/FEEDPVOL
          END IF
C
 10     CONTINUE
C
        END IF
C
C+++++++++++++++++ END OF USER AREA 5 ++++++++++++++++++++++++++++++++++++
C
      END IF
C
C---- OVERWRITE SOURCE TERMS
      IF (ICALL.EQ.2) THEN
C
C++++++++++++++++ USER AREA 6 ++++++++++++++++++++++++++++++++++++++++++
C
C---- EXAMPLE (HEAT SOURCE) OVERWRITE WITH 100W PER UNIT VOLUME IN
C                          ALL INTERIOR CELLS
C
C     CALL GETVAR('USRSRC','E     ',IVAR)
C
C     IF (IVAR.EQ.IEQN) THEN
C  USE IPALL TO FIND ID ADDRESSES OF ALL CELL CENTRES
C       CALL IPALL('*','*','BLOCK','CENTRES',IPT,NPT,CWORK,IWORK)
C  LOOP OVER ALL INTERIOR CELLS
C       DO 200 I=1,NPT
C  USE ARRAY IPT TO GET ADDRESS
C        INODE=IPT(I)
C  OVERWRITE SOURCE TERMS
C        SU(INODE,1)=100.0*VOL(INODE)
C 200 CONTINUE
C       ENDIF
C
C---- END OF EXAMPLE
C
C++++++++++++++++ END OF USER AREA 6 ++++++++++++++++++++++++++++++++++++
C
      END IF
C
      RETURN
C
      END
```

Appendix

B

Supplementary simulation results

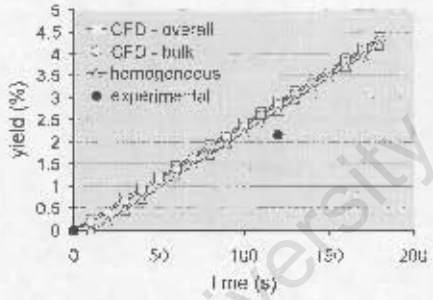

Figure B.1 Comparison of experimentally measured and various simulated precipitate yield profiles for the 'base' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

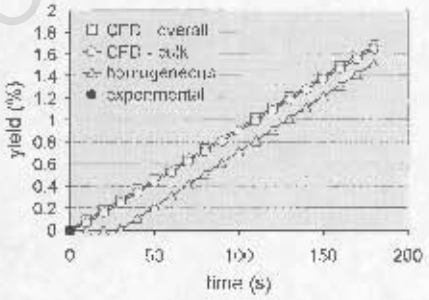

Figure B.2 Comparison of experimentally measured and various simulated precipitate yield profiles for the 'rate L' scenario (simulations employ kinetic 'equilibrium mimic' nucleation; lack of applicable experimental data acknowledged).

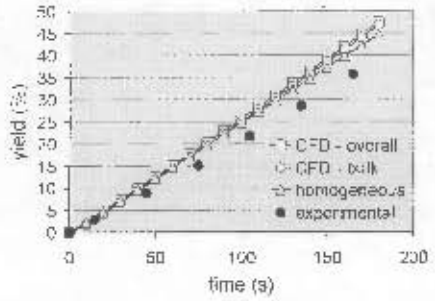

Figure B.3 Comparison of experimentally measured and various simulated precipitate yield profiles for the 'rate H' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

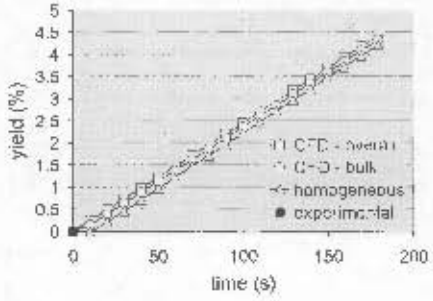

Figure B.4 Comparison of experimentally measured and various simulated precipitate yield profiles for the 'stir L' scenario (simulations employ kinetic 'equilibrium mimic' nucleation; lack of applicable experimental data acknowledged).

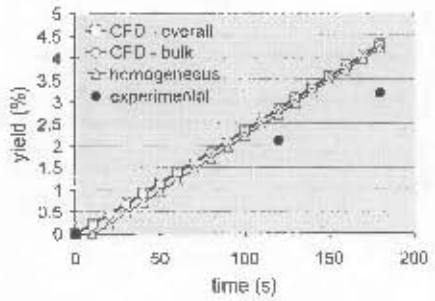

Figure B.5 Comparison of experimentally measured and various simulated precipitate yield profiles for the 'stir H' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

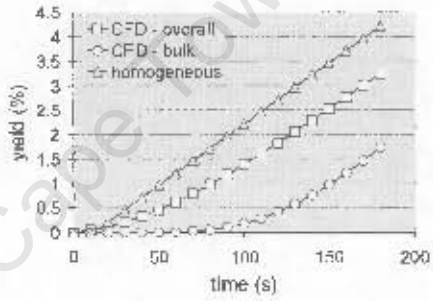

Figure B.6 Comparison of various simulated precipitate yield profiles for the 'stir 0' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

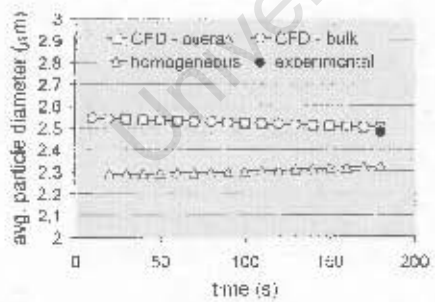

Figure B.7 Comparison of experimentally measured and various simulated average particle diameter profiles for the 'base' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

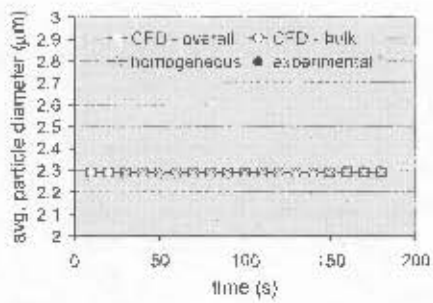

Figure B.8 Comparison of experimentally measured and various simulated average particle diameter profiles for the 'rate L' scenario (simulations employ kinetic 'equilibrium mimic' nucleation; lack of applicable experimental data acknowledged).

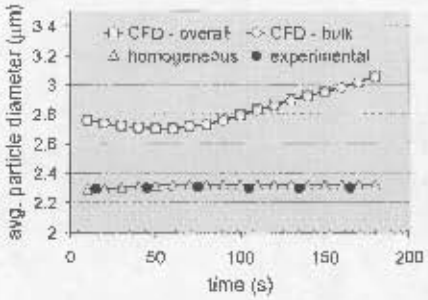

Figure B.9 Comparison of experimentally measured and various simulated average particle diameter profiles for the 'rate H' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

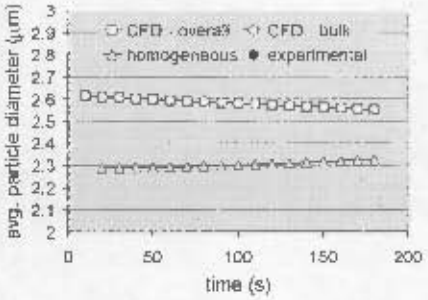

Figure B.10 Comparison of experimentally measured and various simulated average particle diameter profiles for the 'stir L' scenario (simulations employ kinetic 'equilibrium mimic' nucleation; lack of applicable experimental data acknowledged).

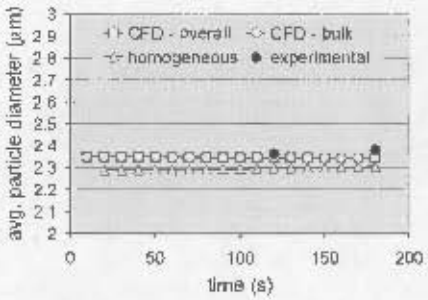

Figure B.11 Comparison of experimentally measured and various simulated average particle diameter profiles for the 'stir H' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

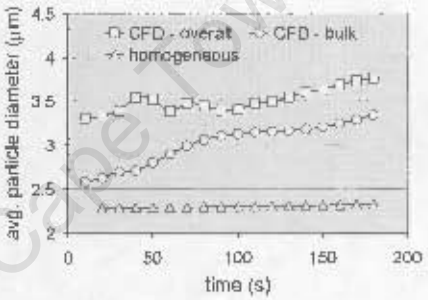

Figure B.12 Comparison of various simulated average particle diameter profiles for the 'stir 0' scenario (simulations employ kinetic 'equilibrium mimic' nucleation).

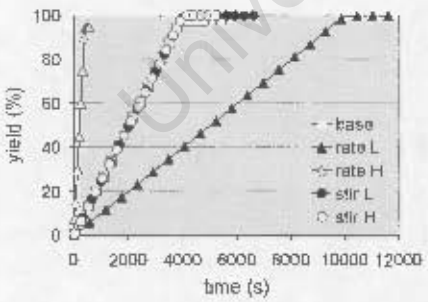

Figure B.13 Profiles of precipitate yield predicted by homogeneous simulations employing a kinetic 'equilibrium mimic' description of nucleation.

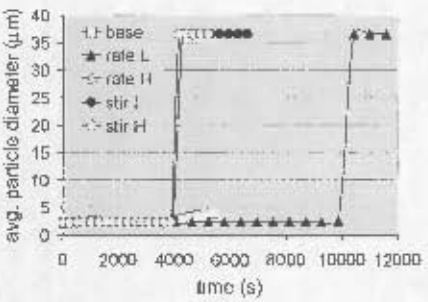

Figure B.14 Profiles of average particle diameter predicted by homogeneous simulations employing a kinetic 'equilibrium mimic' description of nucleation.

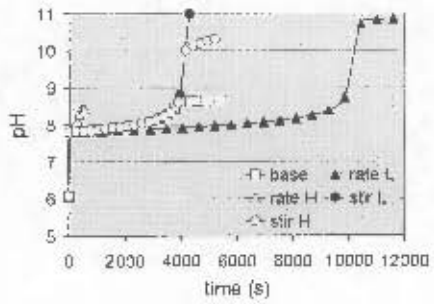

Figure B.15 Profiles of pH predicted by homogeneous simulations employing a kinetic 'equilibrium mimic' description of nucleation.

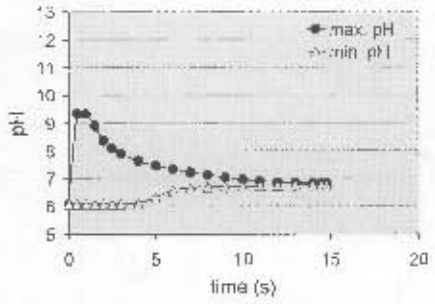

Figure B.16 Profiles of maximum and minimum pH predicted by CFD simulation aimed at determining macromixing time for the 'base' scenario.

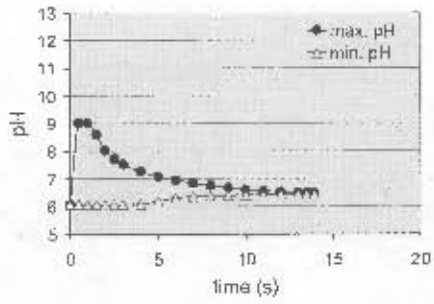

Figure B.17 Profiles of maximum and minimum pH predicted by CFD simulation aimed at determining macromixing time for the 'rate L' scenario.

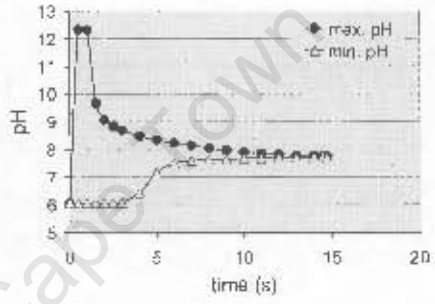

Figure B.18 Profiles of maximum and minimum pH predicted by CFD simulation aimed at determining macromixing time for the 'rate H' scenario.

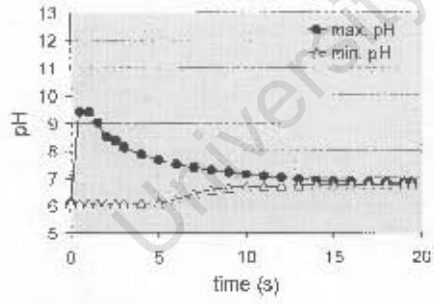

Figure B.19 Profiles of maximum and minimum pH predicted by CFD simulation aimed at determining macromixing time for the 'stir L' scenario.

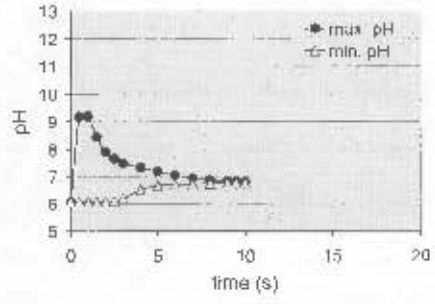

Figure B.20 Profiles of maximum and minimum pH predicted by CFD simulation aimed at determining macromixing time for the 'stir H' scenario.

Notation

a	effective diameter of hydrated ion, Eq. (2.5) (Å)
a	constant, Eq. (2.21)
A	Debye-Hückel constant ($kg^{1/2}.mol^{-1/2}$)
A	constant, Eq. (2.7) ($m^{-3}.s^{-1}$)
a_i	activity of species i
A_T	total particle surface area per unit volume of suspension, Eq. (2.19) ($\mu m^2.m^{-3}$)
B	constant, Eq. (2.5) ($kg^{1/2}.mol^{-1/2}.Å^{-1}$)
B	birth rate ($m^{-3}.\mu m^{-1}.s^{-1}$)
B_0	nucleation rate ($m^{-3}.s^{-1}$)
c_μ	constant, Eq. (2.32)
D	death rate ($m^{-3}.\mu m^{-1}.s^{-1}$)
D_ε	dielectric constant, Eq. (2.4) ($F.m^{-1}$)
d_{imp}	impeller diameter, Eq. (2.21) (m)
D_T	turbulent diffusivity, Eq. (2.22) ($m^2.s^{-1}$)
e	electronic charge (C)
E	engulfment parameter, Eq. (2.24) (s^{-1})
g	kinetic order for growth, Eq. (2.9)
G	linear growth rate ($\mu m.s^{-1}$)
I	ionic strength ($mol.kg^{-1}_{H_2O}$)
J	nucleation rate ($m^{-3}.s^{-1}$)
k	turbulent kinetic energy ($m^2.s^{-2}$)
k_A	area shape factor, Eq. (2.19)
K_a	equilibrium constant
k_B	Boltzmann's constant ($J.K^{-1}$)

k_g growth constant, Eq. (2.9)

k_l length shape factor, Eq. (2.18)

k_n nucleation constant, Eq. (2.8)

K_{sp} solubility product

k_v volume shape factor, Eq. (2.20)

L eddy size, Eq. (2.23) (m)

L particle size (length) (μm)

L_i lower bound of size interval i (μm)

L_i' appropriate mean size in size interval i for calculating j^{th} moment (μm^j)

L_T total particle length per unit volume of suspension, Eq. (2.18) ($\mu m.m^{-3}$)

M_i molality of species i ($mol.kg^{-1}_{H_2O}$)

m_j j^{th} moment of population density function ($\mu m^j.m^{-3}$)

n population density function ($m^{-3}.\mu m^{-1}$)

n kinetic order for nucleation, Eq. (2.8)

N stirrer speed, Eq. (2.21) (s^{-1})

N_A Avogadro's number (mol^{-1})

n_i equilibrium quantity of species i (mol)

N_i particle number concentration in size interval i (m^{-3})

n_i' equilibrium quantity of component i (mol)

$n_{i,0}$ initial quantity of species i (mol)

$n_{i,0}'$ initial quantity of component i (mol)

N_T total particle number per unit volume of suspension, Eq. (2.17) (m^{-3})

$[Ni]_{ppt}$ precipitate molar density ($kmol_{Ni}.m^{-3}$)

Q_{feed} feed addition rate, Eq. (2.22) ($m^3.s^{-1}$)

r ratio of upper and lower bounds of each size interval

S supersaturation

S_ϕ source (or sink) of ϕ ($kg_{[\phi]}.m^{-3}.s^{-1}$)

Sc Schmidt number

t time (s)

T tank diameter, Sec. 3.2 (m)

T temperature (K)

t_C circulation time, Eq. (2.21) (s)

t_D turbulent diffusion time scale, Eq. (2.22) (s)

t_E engulfment time constant, Eq. (2.24) (s)

t_K molecular diffusion time scale, Eq. (2.25) (s)

t_S inertial-convective mixing time scale, Eq. (2.23) (s)

$\overline{u}$ magnitude of velocity close to feed point, Eq. (2.22) ($m.s^{-1}$)

$\mathbf{v}$ vector particle phase-space velocity, Eq. (2.11) ($m.s^{-1}$)

$\mathbf{v}$ velocity vector ($m.s^{-1}$)

v_n 'nucleus' volume (m^3)

V_m molecular volume, Eq. (2.7) (m^3)

V_t total volume of vessel contents, Eq. (2.21) (m^3)

V_T total particle volume per unit volume of suspension, Eq. (2.20) ($\mu m^3.m^{-3}$)

X molar extent of reaction

z_i charge on species i

β aggregation kernel ($m^3.s^{-1}$)

β_0 size-independent portion of aggregation kernel ($m^3.s^{-1}.[f(L,\lambda)]^{-1}$)

δ Dirac delta function

δt averaging time scale, Eq. (2.27) (s)

Δt simulation time-step size (s)

ε turbulent kinetic energy dissipation rate ($m^2.s^{-3}$)

ϕ representative scalar, Eqs. (2.26) – (2.30)

Φ as defined in Eq. (4.35) ($[f(L,\lambda)].m^{-6}$)

Φ_1 as defined in Eq. (4.26) ($[f(L,\lambda)].m^{-3}$)

Φ_2 as defined in Eq. (4.27) ($\mu m^3.m^{-3}$)

Φ_3 as defined in Eq. (4.28) ($[f(L,\lambda)].m^{-6}$)

Φ_A as defined in Eq. (4.29) ($\mu m^{-1}.m^{-3}$)

γ_i activity coefficient of species i

γ' surface energy, Eq. (2.7) ($J.m^{-2}$)

Γ_ϕ molecular diffusivity of ϕ ($kg.m^{-1}.s^{-1}$)

$\Gamma_{\phi,eff}$ effective diffusivity of ϕ ($kg.m^{-1}.s^{-1}$)

$\Gamma_{\phi,T}$ turbulent diffusivity of ϕ ($kg.m^{-1}.s^{-1}$)

λ variable of integration (particle size (length)) (μm)

μ viscosity ($kg.m^{-1}.s^{-1}$)

μ_T turbulent viscosity ($kg.m^{-1}.s^{-1}$)

ν kinematic viscosity ($m^2.s^{-1}$)

v_i stoichiometric coefficient for species i

ρ fluid density ($kg.m^{-3}$)

ρ_0 solvent density, Eq. (2.4) ($kg.m^{-3}$)

τ turbulent mixing time scale (s)

$\bar{}$ Reynolds-averaged quantity, Sec. 2.3.3.1

$'$ fluctuating component quantity, Sec. 2.3.3.1