Multi-level techniques for the solution of the Kinetic Equations in condensing flows

Simon Glazenborg

July 2, 2010

Abstract

The Kinetic Equation can be used to predict the droplet size distribution in condensing flows. Each of the kinetic equations describes the number density of an n-cluster as a function of time. The n-cluster, i.e. a droplet consisting of n monomers, can either grow towards an $(n+1)$-cluster or decay towards an $(n-1)$-cluster, changing the droplet size distribution accordingly.

Solution of the Kinetic Equation requires a considerable computational effort, since typically cluster sizes up to a hundred million monomers have to be included in a numerical simulation. In the past decades in many fields in science it has been shown that multi-level concepts can lead to very efficient algorithms to solve (systems of) partial differential equations numerically.

As a first step a multi-level solver is developed for single-component condensing flows described by the Kinetic Equation. The target is to identify and solve complications before the more complex multi-component condensation problems are considered. Performance, accuracy and stability of the solver is analyzed, and results are compared with results of common numerical solution techniques.

Samenvatting

De Kinetische Vergelijking kan worden toegepast om een druppelgrootteverdeling te bepalen in condenserende stromingen. Elke kinetische vergelijking beschrijft de volumedichtheid van een n-cluster als een functie van de tijd. Een n-cluster, d.w.z. een druppel bestaande uit n monomeren, kan ofwel groeien naar een $(n+1)$-cluster of afnemen naar een $(n-1)$-cluster, zodat de druppelgrootteverdeling verandert.

Het oplossen van de Kinetische Vergelijking vergt een aanzienlijke hoeveelheid rekenkracht, aangezien in het algemeen clustergroottes tot wel een honderd miljoen monomeren moeten worden beschouwd in een numerieke simulatie. In veel takken van de wetenschap is in de afgelopen decennia aangetoond dat multi-level concepten kunnen leiden tot efficiënte algoritmes voor het oplossen van (stelsels van) partiële differentiaal vergelijkingen.

Als eerste stap is een multi-level solver ontwikkeld voor enkel-components condenserende stromingen, beschreven door de Kinetische Vergelijking. Het doel is om moeilijkheden te identificeren en op te lossen voordat complexere multi-components condensatie problemen worden beschouwd. De prestatie, nauwkeurigheid en stabiliteit van de solver wordt geanalyseerd, en resultaten worden vergeleken met de resultaten van gangbare numerieke oplosmethoden.

PREFACE

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Condensing flows

Everywhere in nature we encounter condensation. It is the phenomena we see as the wet dew on grasslands early in the morning, clouds in the sky above us, or the mist of our breath leaves behind on a cold glass. It is the process of a fluid transforming from its gaseous phase to its liquid phase. When we step into our cars on a cold humid morning, the water vapor in the atmosphere comes into contact with the cold windshield, turning into liquid droplets which we see as a fine mist. However, merely describing condensation as the transition of a fluid from its gaseous to its fluid phase does not do justice to the complex physics that take place during this process.

Condensation is often used for the combined processes of nucleation and growth (though this is not an exact definition). For nucleation we can be distinguish two cases: (*a*) homogeneous nucleation, where the supersaturation of a vapor is such that stable clusters of vapor molecules (or droplets) can be formed from the vapor, and (*b*) heterogeneous nucleation, where the nucleation process is set off by small aerosols in the vapor to which vapor molecules attach and grow to a cluster. Although condensation can take place in any vapor, it can only be guaranteed that droplets in the vapor will grow when it is supersaturated. The probability of nucleation in an undersaturated vapor is very small. Characteristic therefore to the condensation process is the supersaturation defined as:

$$S = \frac{\rho_v}{\rho_s(T)} \tag{1.1}$$

where ρ_v is the vapor density, and ρ_s is the saturated vapor density which is a function of the temperature. For $S > 1$ a vapor is said to be supersaturated. The minimum cluster size for clusters to be stable is referred to as the critical cluster size. Smaller and larger clusters are referred to as sub- and supercritical, respectively. A cluster must cross a thermodynamic energy barrier to become supercritical and grow further into a droplet.

The most common form of condensation is heterogenous nucleation where foreign particles act as condensation nuclei. Homogeneous nucleation is mostly encountered in industrial applications with high levels of supersaturation, where foreign particles are absent so that nuclei are formed from the vapor phase itself. Typically low levels of supersaturation are encountered in nature, for which the energy barrier of the former is much lower than that of the latter.

Finally, if multiple substances are present in the vapor, clusters can be formed from molecules from each substance. For a single substance the process is referred to as single-component condensation, whereas for multiple substances it is referred to as multi-component condensation. In nature and in industrial applications both are encountered where the physics of the latter are considerably more complex compared to the former.

1.2 Models

When modeling condensation the liquid phase can be described using a droplet size distribution or DSD. Such a distribution describes the total amount of each separate cluster consisting of n vapor molecules, i.e. an n-cluster, present in the liquid phase. If multiple condensable components exist, the DSD is a multidimensional distribution consisting of all possible molecule combinations. The formulation of the master equation for condensation to obtain the DSD, is based on work of, amongst others, Becker & Döring [1], and the Szilard model of condensation [2]. In there the growth and evaporation of droplets is based on the interaction with single molecules (monomers). As this is a kinetically driven process, the master equation is referred to as the Kinetic Equation (KE).

For single-component condensation, approximations to the KE are the General Dynamic equation (GDE) and the Fokker-Planck equation (FPE), which are first and second order approximations respectively. These have been investigated and compared to the KE by Holten & van Dongen [3] and Sidin [4]. Also a stationary diffusion flux model (SDF) model based on the Fokker-Planck equation was developed by van Putten & Kalikmanov [5]. These models aim to be computationally less expensive than the formally exact KE, however, at the expense of accuracy. Especially for small clusters the approximations lead to inaccurate results, rendering the solution of the KE in certain cases indispensable.

1.3 Motivation and objective

Aside from model assumptions, the solution of the Kinetic Equation is considered to be an accurate description of the condensation process. However, the model requires large computation times for relatively simple condensation processes. The Kinetic Equations for multi-component flows require even larger computational effort, making simulation of practical applications of such flows not feasible at present.

Therefore, in this thesis the numerical solution process is investigated. To improve the numerical solution process, the potential of Multigrid techniques is investigated which in many applications in science have substantially enhanced the performance of numerical solvers. A great deal of literature on the concepts and analysis of Multigrid is available from writers such as, Brandt [6], Venner & Lubrecht [7] and Briggs *et al.* [8]. It will be investigated if Multigrid techniques can be applied to the Kinetic Equation and if their application is promising or not. For this purpose single-component condensation is considered initially, as a fundamental basis for the solution of multi-component condensing flows.

1.4 Outline

The outline of this thesis is as follows:

- In chapter 2, the Kinetic Equation for single-component condensation is introduced. It is presented as a transient form used for the solution of a nucleation pulse experiment, and an advected form used to compute the condensation in a nozzle.
- In chapter 3, the evaluation of the KE is treated, and a discrete set of equations is derived that is to be solved by the Multigrid solver.
- In chapter 4, the concepts of Multigrid are introduced and applied to the equations of the KE.
- In chapter 5, the results of the nucleation pulse experiment are presented.
- In chapter 6, the results of the nozzle flow are presented.
- Finally, in chapter 7, this work is concluded by a discussion of the results followed by recommendations for future research.

CHAPTER 2

THE KINETIC EQUATION

The Kinetic Equation (KE) is based on the description of condensation as a kinetically driven process. It relies on the Szilard approach which assumes that a droplet may grow or decay by gaining or shedding one vapor molecule (monomer) at a time. The solution of the KE is a Droplet Size Distribution (DSD) of the number densities of each separate droplet of a specific composition. In this first investigation of the potential of using multilevel techniques for the solution of the KE, single-component condensation is considered. This is a fundamental first step to identify and solve complications before the more complex multi-component condensation problems are considered.

In this chapter the general KE for single-component condensation will be treated first. Then two specific forms of the KE will be discussed for the application to a nucleation pulse experiment, and a nozzle flow experiment.

2.1 General form

The KE essentially is a mass conservation law for droplets consisting of a integer number of vapor molecules, where the dimer is the smallest possible droplet. With single-component condensation the droplets consist of a single condensable component, characterizing droplets by the amount of monomers n they consist of.

For the flows considered in this work, the concentration of vapor molecules is assumed to be much higher than that of the droplets. This justifies the assumption of the KE that the droplets only interact with monomers. This Szilard approach in condensation is depicted schematically in figure 2.1.

Assuming a no-slip condition, which requires the vapor molecules and droplets to assume the flow velocity $\mathbf{u}$, the KE for an advected vapor is given by:

$$\frac{\partial c_n}{\partial t} + \frac{\partial}{\partial x_j}(u_j c_n) = J_{n-1} - J_n \quad \text{for } n = 2, 3, \dots \tag{2.1}$$

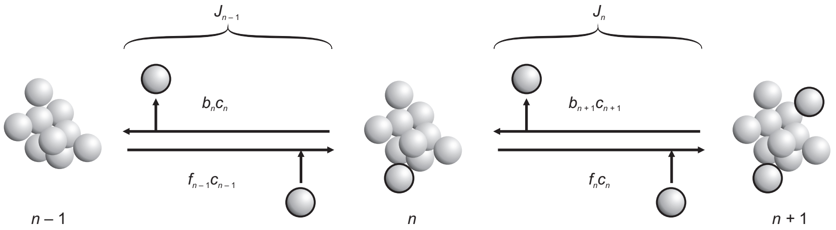


Figure 2.1: Schematic representation of the Szilard model for a single component condensing flow

where c_n is the volumetric number density of droplets consisting of n monomers (n-mers). The condensation flux J_n contains the combined effect of a forward condensation rate f_n and a backward evaporation rate b_{n+1}:

$$J_n = f_n c_n - b_{n+1} c_{n+1}. \tag{2.2}$$

The net flux for cluster n therefore yields:

$$J_{n-1} - J_n = f_{n-1} c_{n-1} - (b_n + f_n) c_n + b_{n+1} c_{n+1}. \tag{2.3}$$

The forward rate is the product of the sticking probability α_n, the collision frequency per unit area, and the cluster surface area:

$$f_n = \alpha_n \cdot c_1 \sqrt{\frac{k_B T}{2\pi m_1}} \cdot a_1 \sqrt{\frac{n+1}{n}} (n^{1/3} + 1)^2 \tag{2.4}$$

where m_1 and a_1 denote the mass and effective surface area of a single molecule. For simplicity the value of α_n will be set to unity, although it has been shown that for small droplets, α_n varies strongly with n (see Sidin [4]). For sufficiently large droplets the forward rate behaves as $f_n \sim n^{2/3}$.

The backward rate b_n is obtained from the detailed balance equation: at equilibrium ($S = 1$, denoted by superscript 'eq') all J_n equal zero, so equation (2.3) becomes:

$$b_{n+1} = f_n \frac{c_n^{eq}}{c_{n+1}^{eq}}. \tag{2.5}$$

The Courtney corrected form of the equilibrium number density is given by:

$$c_n^{eq} = c_1^s \exp\big(- g(n) \big) \tag{2.6}$$

where instead of c_1 the the monomer number density at saturation c_1^s defined as c_1/S is used. The function $g(n)$ is the dimensionless Gibbs free energy of formation for a droplet of size n defined as:

$$-g(n) = \frac{\Delta G_n}{k_B T_n} \tag{2.7}$$

The Courtney model [9] for the Gibbs free energy is applied here:

$$-g(n) = n\ln S - \Theta n^{2/3} \quad \text{for } n > 1, \tag{2.8}$$

where Θ is the dimensionless surface energy defined as:

$$\Theta \equiv \frac{a_1\sigma}{k_B T_n}, \tag{2.9}$$

with σ the surface tension. This results in the following relation for the backward evaporation rate:

$$b_{n+1} = \frac{f_n}{S}\exp\{-\Theta[n^{2/3} - (n+1)^{2/3}]\} \tag{2.10}$$

Furthermore, the flows in this work are considered to be isothermal, i.e. the droplet temperature is assumed to be equal to the mixture temperature.

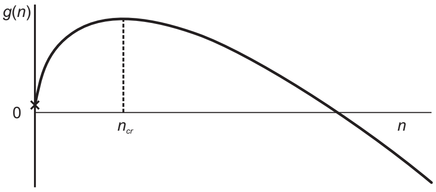


Figure 2.2: Dimensionless Gibbs free energy $g(n)$ as a function of droplet size n. For $n = 0$, $g(n)$ is nonzero.

The function $g(n)$ has a maximum at the critical cluster size $n_{cr} = \left(\frac{2\Theta}{3\ln S}\right)^3$, depicted in figure 2.2. However, this model for ΔG_n was originally obtained for spherical droplets, so that it becomes inaccurate for small clusters. This can be seen from the figure as a discrepancy at $n = 0$, where $g(n)$ is nonzero, so that $c_1^{eq} = c_1\exp(-\Theta)$.

The Classical Nucleation Theory [4] utilizes c_1^{eq} as the value of the boundary condition c_1, where Courtney uses c_1^s. Therefore, when comparing results the boundary value has to be corrected by a factor $\exp(-\Theta)$.

2.2 Experiments

2.2.1 Nucleation pulse

The first model problem that will be treated is the nucleation pulse experiment. The experiment is similar to the one carried out by van Putten & Kalikmanov [5]. The problem consists of a stepwise constant pressure-temperature profile comprising a region with high supersaturation, followed by a region with low supersaturation. In this case there is no advection.

The nucleation pulse will be modeled here with a one-way coupling between the thermodynamic variables and the n-mer concentrations. To close the problem formulation an initial solution and two boundary conditions for the monomer number density and the largest droplet number density are required.

2.2.2 Nozzle flow

As a practical application of the KE, the flow through a Laval nozzle will be considered. In this case advection cannot be neglected, and the general KE has to be considered. However, the flow in the nozzle can be considered to be of a quasi-one dimensional nature, for which the KE can be brought to a Lagrangian form similar to the form obtained for the zero advection case. This will be treated in chapter 3. Numerical treatment is then similar to the nucleation pulse experiment.

Furthermore, this experiment will also be one-way coupled, for which the flow variables u, p and T will be extracted from the results of a flow solver. Latent heat release and the effects on the mixture density by condensation will be neglected.

CHAPTER 3

EVALUATION OF THE KINETIC EQUATION IN CONDENSING FLOW

In this chapter the numerical evaluation procedures for the KE will be derived for the two numerical experiments treated in chapter 2. A closed set of equations is derived for the KE for which a multi-level algoritm will be developed in chapter 4. This set of equations will also be subjected to a thorough analysis to get a full understanding of the system's behavior.

3.1 Nucleation pulse

3.1.1 Kinetic Equation

For the nucleation pulse experiment the transient KE without advection is considered:

$$\frac{\partial c_n}{\partial t} = f_{n-1}c_{n-1} - (b_n + f_n)c_n + b_{n+1}c_{n+1} \quad \text{for } n = 2, 3, \ldots \tag{3.1}$$

With the state vector $\mathbf{c} = (c_2, \ldots, c_{N-1})^T$, (3.1) can be written as:

$$\frac{d\mathbf{c}}{dt} = \mathbf{S} + \mathsf{A}\mathbf{c} \tag{3.2}$$

where the matrix A is defined as:

$$\mathsf{A} = \begin{bmatrix} -(f_2+b_2) & b_3 & 0 & \cdot & 0 & 0 & 0 \\ f_2 & -(f_3+b_3) & b_4 & \cdot & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & f_{N-3} & -(f_{N-2}+b_{N-2}) & b_{N-1} \\ 0 & 0 & 0 & \cdot & 0 & f_{N-2} & -(f_{N-1}+b_{N-1}) \end{bmatrix}. \tag{3.3}$$

The vector **S** contains the contributions from the monomers ($n = 1$) and the largest droplets ($n = N$), and is given by:

$$\mathbf{S} = (f_1 c_1, 0, \ldots, 0, f_N c_N)^T. \tag{3.4}$$

The matrix A is a tridiagonal matrix, for which many efficient solution methods exist. In view of our objective to develop a multi-level method a relaxation based solver is required. Simple and effective relaxation methods are Jacobi and Gauss-Seidel relaxation. Both methods require a diagonally dominant system of equations in order to converge. This property is investigated in appendix A, where it is shown that a diagonally dominant system is obtained in most situations.

3.1.2 Boundary conditions

The number densities at the edges of the domain c_1 and c_N are boundary conditions. Firstly it is assumed that the formation of droplets does not affect the monomer density, i.e. c_1 is constant and the number density of the largest droplet is set to zero, i.e. $c_N = 0$. The monomer number density is obtained from the partial vapor density by $c_1 = \rho_1/m_1$.

Secondly, when the monomer number density is not sufficiently large compared to the droplet number density the assumption of a fixed c_1 no longer holds. It is then more realistic to account for the depletion of monomers when droplets grow. Especially for nucleation pulses of long duration effects of depletion cannot be neglected and have to be accounted for. Therefore the total amount of monomers γ has to be preserved:

$$\frac{d\gamma}{dt} \tag{3.5}$$

with

$$\gamma \equiv \sum_{n=1}^{N} n c_n. \tag{3.6}$$

The saturation S is defined as:

$$S = \frac{p_1}{p_{sat}(T)} \tag{3.7}$$

with p_{sat} is the saturated vapor pressure, which is a function of the temperature. Using $p_1 = c_1 k_b T$ equation (3.6) can be written as:

$$c_{sat} S + \sum_{n=2}^{N} n c_n = \gamma \tag{3.8}$$

with c_{sat} the saturated monomer number density. This relation replaces the Dirichlet boundary condition for c_1.

3.2 Nozzle flow

3.2.1 Conservation of mass

Consider the moving volume $V(t)$ inside the nozzle, depicted in figure 3.1. The volume is advected with the flow, hence the mass contained in that volume does not change:

$$\frac{d}{dt}\iiint_{V(t)} \rho dV = 0. \tag{3.9}$$

In this case ρ is the mixture density. Applying the Leibniz-Reynolds transport theorem yields:

$$\iiint_{V(t)} \frac{\partial \rho}{\partial t} dV + \iint_{\partial V(t)} \rho u_j n_j dS = 0 \tag{3.10}$$

where ∂V is the boundary of V, and $\mathbf{n}$ the outward normal.

In the nozzle, $V(t)$ is the space enclosed by the nozzle wall and the areas $A(x,t)$ and $A(x+\Delta x,t)$, where Δx is a arbitrarily small positive increment along the nozzle axis x.

The first integral in equation (3.10) can be approximated as:

$$\iiint_{V(t)} \frac{\partial \rho}{\partial t} dV \approx \frac{\partial \rho}{\partial t} \iiint_{V(t)} dV = \frac{\partial \rho}{\partial t} V \approx \frac{\partial \rho}{\partial t} A \Delta x \tag{3.11}$$

where higher order terms in Δx have been neglected. The second integral can be written as:

$$\iint_{\partial V(t)} \rho u_j n_j dS = \iint_{A(x+\Delta x)} \rho u dS - \iint_{A(x)} \rho u dS = \Delta(\rho u A) \tag{3.12}$$

with:

$$\Delta(\rho u A) \equiv (\rho u A)_{x+\Delta x} - (\rho u A)_x. \tag{3.13}$$

Substitution into (3.10), division by Δx and taking the limit of $\Delta x \to 0$, yields the quasi-one dimensional mass conservation law applicable to nozzle flow:

$$A\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u A)}{\partial x} = 0. \tag{3.14}$$

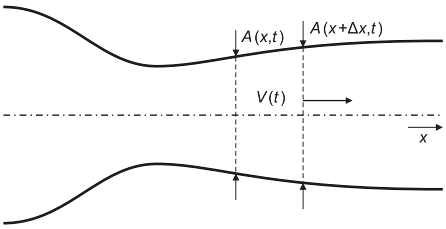


Figure 3.1: Moving volume $V(t)$ inside a Laval nozzle

3.2.2 Kinetic Equation

The advected KE for three dimensional flow is defined as:

$$\frac{\partial c_n}{\partial t} + \frac{\partial (c_n u_j)}{\partial x_j} = J_{n-1} - J_n \quad \text{for } n = 2, 3, \ldots \tag{3.15}$$

where it is assumed that a no-slip condition applies, which requires the vapor molecules and droplets to move with the flow velocity u. Integration over the advected volume $V(t)$ yields:

$$\iiint_{V(t)} \left\{ \frac{\partial c_n}{\partial t} + \frac{\partial (c_n u_j)}{\partial x_j} - (J_{n-1} - J_n) \right\} dV = 0. \tag{3.16}$$

Separation of the advection term and application of the divergence theorem yields:

$$\iiint_{V(t)} \left\{ \frac{\partial c_n}{\partial t} - (J_{n-1} - J_n) \right\} dV + \iint_{\partial V(t)} c_n u_j n_j dS = 0. \tag{3.17}$$

The volume and surface integral can be approximated in the limit of $\Delta x \to 0$, yielding a quasi-one dimensional form of the KE for nozzle flow:

$$A \frac{\partial c_n}{\partial t} + \frac{\partial (c_n u A)}{\partial x} = A(J_{n-1} - J_n). \tag{3.18}$$

At this point the specific number density $\hat{c}_n$ is introduced, i.e. the number of n-mers per unit mass:

$$\boxed{\hat{c}_n = \frac{c_n}{\rho}} \tag{3.19}$$

Substituting this new variable into (3.18), the KE can be rewritten as:

$$A \frac{\partial \rho \hat{c}_n}{\partial t} + \frac{\partial (\rho \hat{c}_n u A)}{\partial x} = A(J_{n-1} - J_n). \tag{3.20}$$

Application of the product rule yields:

$$\hat{c}_n \left[A \frac{\partial \rho}{\partial t} + \frac{\partial (\rho u A)}{\partial x} \right] + \rho A \left[\frac{\partial \hat{c}_n}{\partial t} + u \frac{\partial \hat{c}_n}{\partial x} \right] = A(J_{n-1} - J_n). \tag{3.21}$$

The first bracketed term is recognized as the mass conservation law for the mixture and equals zero. Division by ρA finally yields:

$$\begin{aligned} \frac{D\hat{c}_n}{Dt} &= \frac{J_{n-1}}{\rho} - \frac{J_n}{\rho} \\ &= f_{n-1}\hat{c}_{n-1} - (b_n + f_n)\hat{c}_n + b_{n+1}\hat{c}_{n+1} \end{aligned} \tag{3.22}$$

It was noted that (3.22) is very similar to the transient KE, equation (3.1). It can be brought to a similar matrix form as the transient KE . Inter-droplet fluxes can also be computed in the same fashion. Note that with advection, the forward and backward rates, which are functions of $p(x(t))$ and $T(x(t))$, are influenced by the location in the nozzle, hence the flow velocity $u(x(t))$.

3.2.3 *Mixture thermodynamics*

The pressures in the nozzle are assumed to be moderate, so that the vapor and inert carrier gas can be treated as perfect gases, governed by the ideal gas law:

$$p_i = \rho_i R_i T \tag{3.23}$$

where the subscript i will be replaced by 'c' for the carrier gas and by 'v' for the vapor.

The amount of liquid in the mixture is characterized by the liquid mass fraction g, defined as:

$$g \equiv \frac{m_l}{m_c + m_v + m_l} \qquad 0 \leq g \leq g_{max} \tag{3.24}$$

where g_{max} is the total mass fraction of the condensable component:

$$g_{max} \equiv \frac{m_l + m_v}{m_c + m_v + m_l}. \tag{3.25}$$

The liquid mass fraction and total mass fraction are obtained from the specific number densities by:

$$g = m_1 \sum_{n=2}^{N} n\hat{c}_n \tag{3.26a}$$

$$g_{max} = m_1 \sum_{n=1}^{N} n\hat{c}_n \tag{3.26b}$$

where only droplets of $n \geq 2$ are counted in the liquid phase.

As slip with respect to the flow is neglected for all mixture components, and diffusion of vapor is neglected, the composition is constant along streamlines, and therefore, the material derivative of g_{max} is zero:

$$\frac{D}{Dt}(g_{max}) = 0. \tag{3.27}$$

Therefore, when uniformly specified at the inflow boundaries, g_{max} is a global constant in the flow domain. At each point in the nozzle, when g and ρ are known, the partial densities can be approximated by:

$$\rho_c = (1 - g_{max})\rho \tag{3.28a}$$

$$\rho_v = (g_{max} - g)\rho. \tag{3.28b}$$

The mean density of the liquid then follows from:

$$\rho_l = g\rho. \tag{3.29}$$

Assuming $g << 1$, the liquid density ρ_l becomes negligible so that the mixture density can be approximated by:

$$\rho \approx \rho_c + \rho_v, \tag{3.30}$$

Then, application of the ideal gas law to the remaining gas mixture yields:

$$p = \rho R T \tag{3.31}$$

where R is the specific gas constant for the gas mixture defined as:

$$R = (1 - g_{max})R_c + (g_{max} - g)R_v. \tag{3.32}$$

3.2.4 Depletion

When the monomer specific number density is assumed to be constant, the supersaturation S of the mixture will remain unchanged. In case of nozzle flow effects of depletion cannot be neglected.

As was mentioned above, the total mass fraction of the condensable component g_{max} is constant throughout the flow. From this an extra condition arises for the monomer specific number density $\hat{c}_1$. Equation (3.26b) can be written as:

$$\hat{c}_1 + \sum_{n=2}^{N} n\hat{c}_n = \frac{g_{max}}{m_1} \tag{3.33}$$

where the right hand side is a constant. Using $p_1 = c_1 k_b T$ and (3.19), equation (3.7) can be written as:

$$S = \frac{\rho k_b T}{p_{sat}(T)}\hat{c}_1, \tag{3.34}$$

so with depletion of the vapor, the supersaturation is changed and the forward and backward rates in the KE are affected.

3.3 Time & space marching methods

Both the transient KE without advection and the Lagrangian form of the quasi one dimensional KE, are first order ordinary differential equations (ODE). The time-marching schemes considered in this work are:

- 4th order Runge Kutta (RK4), which is an explicit time integration method with fourth order accuracy. It is very fast since it does not require matrix inversion to solve the discrete set of equations. However, for stiff equations a relatively restrictive strong CFL condition applies, requiring very small time integration steps for stable time integration.

- Backward Euler, which is a fully-implicit time integration method with first order accuracy. It is not bounded by a CFL condition and is therefore stable for arbitrary time integration steps. However, because of the first order accuracy, large time steps will produce inaccurate results.
- Trapezoidal method, which is a semi-implicit time integration method that achieves second order accuracy by averaging forward and backward Euler time integration. It is bounded by a CFL condition, but due to its implicit nature it is much less severe than encountered with RK4. A downside is that it can produce numerical oscillations in the solution.

Multigrid is based on relaxation, and is therefore designed to efficiently solve implicit systems of equations. Furthermore, implicit methods are favored for the solution of the KE because larger time integration steps are allowed compared to explicit methods. The majority of the simulations is carried out using the trapezoidal method, because of its second order accuracy. The other methods are used to verify the results and act as benchmarks.

3.3.1 Discretization

As was mentioned above the trapezoidal method averages the forward and backward Euler methods, achieving second order accuracy. It does however suffer from numerical oscillations which can be very persistent for large time steps. These oscillations are carefully monitored to ensure correct results. Below the discretization is carried out for the transient KE without advection considered for the nucleation pulse experiment. Discretization for the Lagrangian form for nozzle flow is analogous to this form.

With the state vector $\mathbf{c} = (c_2^{m+1}, \ldots, c_{N-1}^{m+1})^T$, the trapezoidal method computes the solution at the next time step as follows:

$$\mathbf{c}^{m+1} = \mathbf{c}^m + \frac{\Delta t}{2}\Big[(\mathsf{A}^m \mathbf{c}^m + \mathbf{S}^m) + (\mathsf{A}^{m+1}\mathbf{c}^{m+1} + \mathbf{S}^{m+1})\Big]. \tag{3.35}$$

The next time step is obtained from *known* present time variables and *unknown* next time variables. The matrix A is known at all times since it only depends on time-dependent quantities as the monomer number density, temperature[1] and saturation[2], which are specified for the experiment sans depletion.

Writing equation (3.35) as

$$\Big(\mathsf{I} - \tfrac{1}{2}\mathsf{A}^{m+1}\Delta t\Big)\mathbf{c}^{m+1} = \mathbf{g}, \tag{3.36}$$

the right hand side $\mathbf{g}$ becomes:

$$\mathbf{g} = \big(g_2 + \tfrac{f_1}{2}\Delta t\, c_1^{m+1}\,,\, g_3\,,\, \ldots\,,\, g_{N-2}\,,\, g_{N-1} + \tfrac{b_N}{2}\Delta t\, c_N^{m+1}\big)^T \tag{3.37}$$

---

[1] For nozzle flow the temperature is extracted from the results of a flow solver.

[2] With depletion the saturation is computed with the monomer number density c_1 which is an additional variable.

with

$$g_n = c_n^m + \frac{\Delta t}{2}\left[f_{n-1}^m c_{n-1}^m - (b_n^m + f_n^m)c_n^m + b_{n+1}^m c_{n+1}^m\right]. \tag{3.38}$$

Here, c_1^{m+1} is either fixed, or a variable governed by the auxiliary depletion equation. In all cases a Dirichlet boundary condition is used for the largest droplet: $c_N^{m+1} = 0$.

With respect to the continuous system of equations (3.2), two changes are brought about by discretization. Firstly, the system matrix A is multiplied by a factor $\Delta t/2$. Secondly, implicit integration adds the identity matrix I. The combination of both changes constitute an improvement of the diagonal dominance of the system. This extends the range of applicability for Gauss Seidel relaxation, which requires this property for stability.

In contrast to fully implicit schemes, the trapezoidal method is not unconditionally stable. In appendix B the CFL condition of this scheme is investigated for the KE. It turns out that, for stable time integration, the following condition must be obeyed:

$$f_n + b_n \geq f_{n-1} + b_{n+1}, \tag{3.39}$$

which is equivalent to the diagonal dominance condition discussed in appendix A.

3.4 Grouping of droplets

In order to reduce the computational effort required for solving the KE, it is common to reduce the number of kinetic equations by grouping ranges of droplets into bins. The way droplets are grouped into bins is presented in figure 3.2.

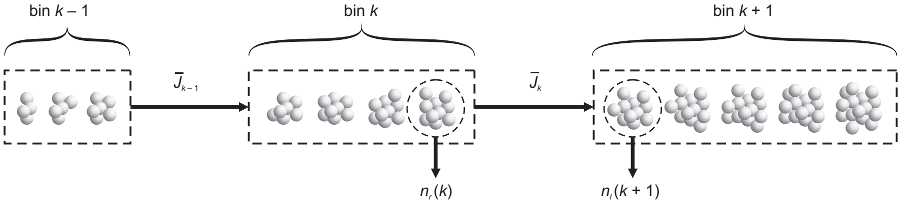


Figure 3.2: Grouping of droplets into bins

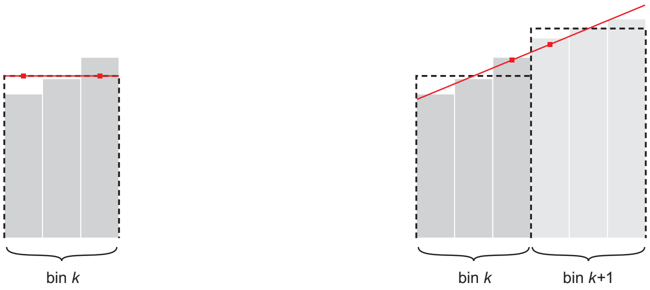
Because of the similarities between the nucleation pulse and nozzle experiment, the KE for bins is only derived for the former. The KE without advection for droplet bins is given by:

$$\frac{d\bar{c}_k}{dt} = \frac{1}{w_k}(\bar{J}_{k-1} - \bar{J}_k) \tag{3.40}$$

where $\bar{c}_k$ is the mean number density in bin k and w_k is number of different droplet sizes in bin k, i.e. the bin size. The fluxes are defined similar to the original KE where:

$$\bar{J}_k = J_{n_r(k)} = f_{n_r(k)}c_{n_r(k)} - b_{n_l(k+1)}c_{n_l(k+1)}. \tag{3.41}$$

This formulation requires the number densities at the neighboring droplet sizes of $n_r(k)$ and $n_l(k+1)$, with the latter being the smallest droplet in bin $k+1$. These can be obtained by interpolation from the mean number densities, which is depicted in figure 3.3.



(a) Zero-th order interpolation (b) First order interpolation

Figure 3.3: Using interpolation to obtain edge number densities. Dotted lines are mean number densities in bin, red line is the interpolation and red dots are the interpolated edge number densities.

The KE for bins will produce interpolation errors compared to the general KE. In appendix C a thorough analysis of this error is performed. This analysis shows that the error of interpolation is 1 order higher than that of the interpolation used. Especially high order error components contribute much to the interpolation error. Therefore a low order interpolation can only be used when the DSD is smooth.

The bin distribution used in this work is an exponentially increasing bin size, for bin totals ranging 2000 to 100000. This distribution yields unit bin widths for small droplets where the DSD is very rough, and very large bins for large droplets where the DSD is relatively smooth.

3.4.1 Zero-th order interpolation

With a zero-th order interpolation, the number densities of the neighboring droplets are directly obtained from their mean bin values, i.e.:

$$c_{n_r(k)} = \bar{c}_k \quad \text{and} \quad c_{n_l(k+1)} = \bar{c}_{k+1}. \tag{3.42}$$

The total flux into bin k then becomes:

$$\bar{J}_{k-1} - \bar{J}_k = f_{n_r(k-1)}\bar{c}_{k-1} - (b_{n_l(k)} + f_{n_r(k)})\bar{c}_k + b_{n_l(k+1)}\bar{c}_{k+1}. \tag{3.43}$$

With the statevector $\bar{\mathbf{c}} = (\bar{c}_2, \dots, \bar{c}_{N-1})^T$ the matrix A of the KE becomes:

$$\mathsf{A} = \begin{bmatrix} -\frac{f_2+b_2}{w_2} & \frac{b_3}{w_2} & 0 & \cdot & 0 & 0 & 0 \\ \frac{f_2}{w_3} & -\frac{f_3+b_3}{w_3} & \frac{b_4}{w_3} & \cdot & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \frac{f_{N-3}}{w_{N-2}} & -\frac{f_{N-2}+b_{N-2}}{w_{N-2}} & \frac{b_{N-1}}{w_{N-2}} \\ 0 & 0 & 0 & \cdot & 0 & \frac{f_{N-2}}{w_{N-1}} & -\frac{f_{N-1}+b_{N-1}}{w_{N-1}} \end{bmatrix} \tag{3.44}$$

where

$$f_k = f_{n_r(k)} \quad \text{and} \quad b_k = b_{n_l(k)}. \tag{3.45}$$

3.4.2 First order interpolation

A 1^{st} order interpolation uses two neighboring bins to produce an estimate for the number densities in the middle. The number densities of the droplets at the inner bin edges are a function of the 2 adjacent bins. The number densities at the inner edges become:

$$c_{n_r(k)} = \frac{\bar{c}_k(w_{k+1}+1) + \bar{c}_{k+1}(w_k - 1)}{w_k + w_{k+1}} \tag{3.46a}$$

$$c_{n_l(k+1)} = \frac{\bar{c}_k(w_{k+1}-1) + \bar{c}_{k+1}(w_k + 1)}{w_k + w_{k+1}}. \tag{3.46b}$$

Then the total flux into bin k becomes:

$$\begin{aligned} \bar{J}_{k-1} - \bar{J}_k = & \frac{(w_k+1)f_{n_r(k-1)} - (w_k-1)b_{n_l(k)}}{w_{k-1}+w_k}\bar{c}_{k-1} \\ & + \frac{(w_{k-1}-1)f_{n_r(k-1)} - (w_{k-1}+1)b_{n_l(k)}}{w_{k-1}+w_k}\bar{c}_k \\ & + \frac{(w_{k+1}-1)b_{n_l(k+1)} - (w_{k+1}+1)f_{n_r(k)}}{w_k+w_{k+1}}\bar{c}_k \\ & + \frac{(w_k+1)b_{n_l(k+1)} - (w_k-1)f_{n_r(k)}}{w_k+w_{k+1}}\bar{c}_{k+1}. \end{aligned} \tag{3.47}$$

However, application of 1^{st} order interpolation causes instability with the KE. For supercritical clusters the KE behaves as an advection equation, whereas for subcritical clusters the KE behaves more as a advection-diffusion equation. The current method lacks ‘upwinding’ and a limiter for interpolation, which in other advection schemes is the case. This causes instabilities for supercritical clusters.

3.4.3 Depletion

As was mentioned above, the assumption that the monomer number density is constant is not valid when very large droplets are formed, which is the case for a nucleation pulse of long enough duration and in nozzle flow.

For grouped droplets, equation (3.6) is only valid for the first bins in the domain which have unit bin size. Larger bins have to account for the total amount of monomers contained within all droplets in those bins.

Therefore the following sum can be used:

$$\sum_{k=1}^{K} w_k \bar{n}_k \bar{c}_k = \gamma \tag{3.48}$$

where $\bar{n}_k$ is the average monomer count per droplet in bin k defined as:

$$\bar{n}_k = \sum_{n=n_l(k)}^{n_r(k)} \frac{n}{w_k} = \frac{n_l(k) + n_r(k)}{2}. \tag{3.49}$$

This basically is a zero-th order interpolation using the average number densities in the bin centers. However, for very large bins the approximation error will be considerable.

Now, consider the k-space continuous. Then a more accurate description of the amount of monomers contained in the droplets can be obtained by numerical integration of the continuous function $\bar{n}_k \bar{c}_k$, with the average number densities $\bar{c}_k$ and the average droplet sizes $\bar{n}_k$ given in each bin k.

Using the trapezoidal rule for the numerical integration, the area under a function $f(k)$ is approximated as:

$$\int_a^b f(k)dk = (b-a)\frac{f(a)+f(b)}{2}, \tag{3.50}$$

with:

$$f(k) = \bar{n}_k \bar{c}_k. \tag{3.51}$$

Then the integration over the entire domain can be written as the sum of all local integrals:

$$\sum_{k=1}^{K-1} \int_{\bar{n}_k}^{\bar{n}_{k+1}} f(k)dk = \sum_{k=1}^{K-1} (\bar{n}_{k+1} - \bar{n}_k) \frac{\bar{n}_k \bar{c}_k + \bar{n}_{k+1} \bar{c}_{k+1}}{2}. \tag{3.52}$$

Then, a correction is made as the function values at the lower and upper boundaries of the domain of integration are only taken into account partially. This requires addition of the missing parts of the lower and upper boundaries, yielding:

$$(\bar{n}_2 - \bar{n}_1)\frac{\bar{n}_1 \bar{c}_1}{2} + \sum_{k=1}^{K-1} (\bar{n}_{k+1} - \bar{n}_k) \frac{\bar{n}_k \bar{c}_k + \bar{n}_{k+1} \bar{c}_{k+1}}{2} + (\bar{n}_K - \bar{n}_{K-1}) \frac{\bar{n}_K \bar{c}_K}{2} = \gamma, \tag{3.53}$$

with γ the total amount of monomers to be preserved. For the limit of unit bin sizes, i.e. $(\bar{n}_{k+1} - \bar{n}_k) \to 1$, equation (3.53) is equivalent to (3.6).

CHAPTER 4

MULTIGRID

The potential of Multigrid methods is best experessed by the following conceptual statement:

> *"The amount of computational work to obtain a solution should be proportional to the amount of real physical information in the computed system."*

Often a lot of computational work is done for marginal improvement or to follow changing effects. In such cases there must be a better way to achieve the same goal. Examples are iterative processes necessary for the solution of equations arising from various physical problems. These have rapid initial convergence, but tend to stall when the error in the solution changes very little from one iteration to the next.

Multigrid can be used to improve the convergence speed of such processes by using multiple grids on which the problem is represented. These different grids serve to represent the different scales of the error. By treating the error components on an appropriate grid the fast initial convergence of the underlying iterative process can be preserved.

In this chapter a Multigrid solver will be constructed for the Kinetic Equation. The KE without advection, discretized with the trapezoidal rule, will be used for its development, but the approach will also be applicable to the other problems due to the similarities. At the basis of the solver lies a relaxation process which will be treated first. Next the different grids necessary to effectively reduce the error will be investigated. It is shown how depletion can be accounted for in a global way by means of a global constraint. Finally, when all components of the Multigrid solver for the KE are ready the solver can be constructed, which is thoroughly analyzed to ensure that the numerical process is optimal.

4.1 Gauss Seidel relaxation

Relaxation is the process of considering each individual equation in a system separately, and solving it for a particular unknown. In the description one need not consider the full system matrix A, only the separate equations. For this purpose the problem is written in the following way:

$$L\mathbf{c} = \mathbf{g} \tag{4.1}$$

where, in case of the KE, L is a linear operator which acts on $\mathbf{c}$, the vector of unknowns.

All equations for unknowns c_n are scanned in a prescribed order. Given an approximation to the solution, the local unknown c_n is changed (relaxed), so that it satisfies the local equation. Relaxation of all unknowns is called a relaxation sweep. With Gauss Seidel relaxation, the new values of local unknowns c_n are directly used when relaxing subsequent equations. It therefore matters in which order the equations are scanned. Often the equations are relaxed in order of increasing index, which is referred to as lexicographic ordering.

In the case of the KE, the treatment of the interior points is different from relaxation of the boundary points. When depletion is neglected there are two Dirichlet boundary conditions. When depletion *is* considered, the auxiliary depletion equation must be solved as well. Both will therefore be discussed separately.

4.1.1 Interior relaxation

After discretization using the trapezoidal rule equation (3.36) was obtained. Then, interior equations of can be written as:

$$L_n\mathbf{c} = g_n \quad \text{for} \quad 2 \leq n \leq N-1 \tag{4.2}$$

where L_n is n^{th} row vector of L.

Gauss Seidel relaxation is performed as follows. Let $\tilde{c}_n$ denote the current approximation to c_n^{m+1}. For each droplet size n a new approximation to c_n^{m+1} is computed according to:

$$\bar{c}_n := \tilde{c}_n + \omega\delta_n, \tag{4.3}$$

where ω is the relaxation factor and δ_n is such that for $\omega = 1$ the local equation is exactly solved:

$$\delta_n = \left(\frac{\partial L_n\mathbf{c}}{\partial c_n}\right)^{-1} r_n, \tag{4.4}$$

and where r_n is the local residual. For the KE, δ_n is:

$$\delta_n = \left[1 + \frac{\Delta t}{2}\left(b_n^{m+1} + f_n^{m+1}\right)\right]^{-1} r_n, \tag{4.5}$$

and r_n, for lexicographic ordering, is defined as:

$$r_n = g_n - \left\{ \tilde{c}_n - \frac{\Delta t}{2} \left[f_{n-1}^{m+1} \bar{c}_{n-1} - \left(b_n^{m+1} + f_n^{m+1} \right) \tilde{c}_n + b_{n+1}^{m+1} \tilde{c}_{n+1} \right] \right\}. \tag{4.6}$$

Note the updated value $\bar{c}_{n-1}$ in the computation of the residual.

Given the neighboring values $\bar{c}_{n-1}$ and $\tilde{c}_{n+1}$, for $\omega = 1$ the n^{th} equation is the discrete equation:

$$-f_{n-1}^{m+1} \bar{c}_{n-1} + \left[1 + \frac{\Delta t}{2} \left(b_n^{m+1} + f_n^{m+1} \right) \right] \bar{c}_n - b_{n+1}^{m+1} \tilde{c}_{n+1} = g_n. \tag{4.7}$$

For $\omega < 1$ the relaxation process is called *damped*, and effectively underrelaxes the local equation.

The solution of the iterative method is considered to be converged when the residual is reduced to machine accuracy.

In appendix D the performance of interior relaxation is analyzed by means of a Local Mode analysis. It is determined how each separate error component is reduced by a single relaxation sweep. From the analysis insight in the effectiveness and stability of the relaxation process is obtained.

4.1.2 Boundary relaxation

When depletion is taken into account, the Dirichlet boundary condition is replaced by equation (3.8). This equation is completely different from the interior equations. It is a large scale equation and a change in S and thereby c_1 will result in a change of all coefficients of the interior equations. An equation which has such a global effect is called a 'global constraint', and must be treated separately from the interior relaxation.

Similar to the interior equations, this equation can be relaxed as follows:

$$\bar{S} = \tilde{S} + \omega \delta \tag{4.8}$$

where δ is given by:

$$\delta = \frac{r}{c_{sat}} \tag{4.9}$$

and the residual is defined as:

$$r = \gamma - \left(c_{sat} \tilde{S} + \sum_{n=2}^{N} n \tilde{c}_n \right). \tag{4.10}$$

However, relaxation of a global equation may frustrate the convergence of the interior relaxation process, and must therefore be applied with caution. Underrelaxation with $\omega < 1$ may be required to successfully account for depletion.

4.2 Multigrid

The Local Mode analysis in appendix D, shows that for Gauss Seidel relaxation the maximum of the graphs is located at $\theta = 0$. This means that the low frequency error components in the residual are reduced the least effective. On the other hand, high frequency components are reduced very effectively, especially for large n. Therefore, after a few relaxation sweeps the error is relatively smooth, and can be represented accurately on a coarser grid. This is the concept of Multigrid methods, which aim to accelerate error reduction on the target grid, by using coarser grids.

4.2.1 Correction Scheme

Starting with the discretized full KE on the target grid h, which is represented in the following manner:

$$L^h \mathbf{c}^h = \mathbf{g}^h \tag{4.11}$$

with L^h the linear operator that acts on the number density vector $\mathbf{c}^h$. For a given approximation $\tilde{\mathbf{c}}^h$, the exact solution is defined as:

$$\mathbf{c}^h \equiv \tilde{\mathbf{c}}^h + \mathbf{v}^h \tag{4.12}$$

where $\mathbf{v}^h$ is the numerical error which is related to the residual in the following way:

$$L^h \mathbf{v}^h = \mathbf{r}^h. \tag{4.13}$$

The residual can be computed for a given $\tilde{\mathbf{c}}^h$, so the right hand side of equation (4.13) is a known quantity. In fact, it is the same equation as for the original problem for a different right hand side.

After the error $\mathbf{v}^h$ is smoothed enough by a few relaxation sweeps, the error can be represented on the coarse grid H, i.e. $\mathbf{v}^H$. To solve this error a representation of equation (4.13) on the coarse grid is defined:

$$L^H \mathbf{v}^H = I_h^H \mathbf{r}^h \tag{4.14}$$

where L^H is the coarse grid operator (discussed in section 4.3.2), I_h^H is a restriction operator from the fine to the coarse grid (discussed in section 4.3.1) and $\mathbf{v}^H$ is a coarse grid approximation to the fine error $\mathbf{v}^h$.

Equation (4.14) can be solved by the same iterative process as used on the target grid. Assuming $\mathbf{v}^H$ is obtained, it can be used to correct the current approximation $\tilde{\mathbf{c}}^h$ by:

$$\bar{\mathbf{c}}^h := \tilde{\mathbf{c}}^h + I_h^H \mathbf{v}^H \tag{4.15}$$

where I_h^H is a interpolation operator from the coarse to the fine grid.

4.2.2 Full Approximation Scheme

The Correction Scheme discussed above assumes a linear operator. When the problem is non-linear, the Correction Scheme must be replaced by the more general Full Approximation Scheme. The non-linear problem can be written as:

$$L^h\langle \mathbf{c}^h \rangle = \mathbf{g}^h \tag{4.16}$$

where the notation $L^h\langle \mathbf{c}^h \rangle$ is used to indicate that the operator L^h depends on the solution $\mathbf{c}^h$ and works on $\mathbf{c}^h$. The smoothing properties are the same.

After a few relaxation sweeps an approximate solution $\tilde{\mathbf{c}}^h$ is obtained, with a residual defined as:

$$\mathbf{r}^h = \mathbf{g}^h - L^h\langle \tilde{\mathbf{c}}^h \rangle. \tag{4.17}$$

Substitution of the error definition (4.12) in (4.17) yields:

$$\mathbf{r}^h \equiv \mathbf{g}^h - L^h\langle \mathbf{c}^h - \mathbf{v}^h \rangle. \tag{4.18}$$

In contrast to the Correction Scheme, now the error cannot be treated separately from the solution. Therefore the full equation must be used to approximate the error on the coarse grid. The problem that is treated on the coarse grid is obtained when substituting (4.12) and (4.17) in (4.16):

$$L^h\langle \tilde{\mathbf{c}}^h + \mathbf{v}^h \rangle = L^h\langle \tilde{\mathbf{c}}^h \rangle + \mathbf{r}^h. \tag{4.19}$$

This equation can be represented on the coarse grid as:

$$L^H\langle \hat{\mathbf{c}}^H \rangle = \hat{\mathbf{g}}^H \tag{4.20}$$

with

$$\hat{\mathbf{c}}^H = I_h^H(\tilde{\mathbf{c}}^h + \mathbf{v}^h) \tag{4.21}$$

and

$$\hat{\mathbf{g}}^H = L^H\langle I_h^H \tilde{\mathbf{c}}^h \rangle + I_h^H \mathbf{r}^h. \tag{4.22}$$

After a good approximation to $\hat{\mathbf{c}}^H$ is found on the coarse grid, the target grid problem is corrected by:

$$\bar{\mathbf{c}}^h := \tilde{\mathbf{c}}^h + I_H^h(\tilde{\mathbf{c}}^H - I_h^H \tilde{\mathbf{c}}^h) \tag{4.23}$$

The difference from CS to FAS is that where for CS the error $\mathbf{v}^h$ appears explicitly in the equations, for FAS it is defined as $(\tilde{\mathbf{c}}^H - I_h^H \tilde{\mathbf{c}}^h)$.

4.2.3 Multi-level cycle

The coarse grid corrections discussed above, can be used recursively so that each subsequent grid is corrected by its next coarser grid. When enough grids are used, the coarsest grid becomes small enough to solve equation (4.14) or (4.20) up to the level of machine accuracy with little effort.

This application leads to a flow diagram of relaxation, coarsening and refining, traversing the grids (levels) referred to as a V-cycle, depicted in figure 4.1. Here ν_0, ν_1 and ν_2 are the number of relaxation sweeps performed at each point in the multilevel cycle:

- In the downward leg of the cycle, ν_1 relaxation sweeps are performed on each level to smooth the error such that it can sufficiently accurately be represented on the next coarser level.
- At the coarsest level, ν_0 relaxations are performed to *exactly* solve the coarse grid problem. Another possibility is to use a direct solver at this point.
- In the upward leg, ν_2 relaxations are performed after correction at each level to remove possible high frequency errors introduced by the interpolation.

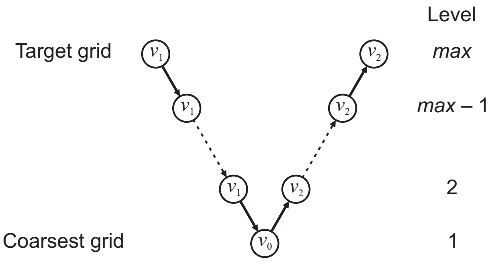


Figure 4.1: Recursive use of coarse grid correction in a V(ν_1, ν_2)-cycle. The coarsest grid is at level 1.

The generalization of the V-cycle is the γ-cycle, where the coarsest level is visited γ times before refining to ensure that the coarsest level is solved accurately enough. $\gamma = 2$ is referred to as a W-cycle. The limit for large γ is the two level cycle.

4.3 Coarsening

Consider the variable $\mathbf{r}^h$, on grid h. This vector contains n residual values r_n^h. A coarse grid vector can be defined $\mathbf{r}^H$, containing only half the number of variables, say only the fine grid variables r_n^h with odd indices n. It then has to be determined how the fine grid is *restricted* to the coarse grid, i.e. how the coarse grid values are related to the fine grid values. This is done in a restriction operator I_h^H.

When the coarse grid problem is solved the solution must be used to correct the fine grid solution. However, the coarse grid solution is only defined in coarse grid points. So, to obtain values in each of the fine grid points an interpolation must be defined, with an interpolation operator I_H^h, which is related to the restriction operator.

In general there are 2 types of coarsening:

– Vertex centered coarsening; coarse grid points are chosen to coincide with fine grid points.

– Cell centered coarsening; coarse grid cells are defined as a combination of fine grid cells.

The choice of the coarsening must represent the nature of the problem at hand in order to obtain coarse grid equations similar to the target grid equations. When badly chosen, the coarse grid equations will not be similar and possibly more complex, preventing the use of the same iterative process for the coarse grid as for the target grid. The subsequent subsections will discuss the coarsening of the KE.

4.3.1 Inter-grid operators

Vertex centered coarsening

With vertex coarsening, fine grid vertices and coarse grid vertices coincide. The value of a coarse grid variable is a combination of the values of fine grid variables, depending on the order of interpolation. An often used restriction method is *full weighting*, displayed in figure 4.2 for 1D. Each interior coarse grid point is assigned half the value of its coinciding fine grid point, and one quarter of the neighbors of that fine grid point. The corresponding linear interpolation, gives each coinciding fine grid point the full value of the coarse grid, and gives the intermediate fine grid points half the value of the adjacent coarse grid points.

– The restriction operator I_h^H can be expressed in a stencil:

$$I_h^H = \frac{1}{4}\left[1\ 2\ 1\right] \tag{4.24}$$

and results in the following relation for the interior coarse grid points $\mathbf{r}^H = I_h^H \mathbf{r}^h$:

$$r_N^H = \left(1 \cdot r_{2N-1}^h + 2 \cdot r_{2N}^h + 1 \cdot r_{2N+1}^h\right)/4. \tag{4.25}$$

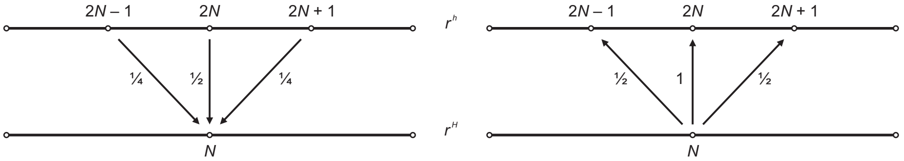


Figure 4.2: Vertex centered coarsening for the interior grid points. Restriction (left) and interpolation (right).

- The interpolation operator I_H^h is expressed in a similar manner:

$$I_H^h = \frac{1}{2}\left[1\,2\,1\right] \tag{4.26}$$

  which gives the following relations for the fine grid points $\mathbf{r}^h = I_H^h \mathbf{r}^H$:

$$r_{2N}^h = r_N^H \qquad 0 \le N \le n/2 \tag{4.27a}$$

$$r_{2N+1}^h = \left(r_N^H + r_{N+1}^H\right)/2 \qquad 0 \le N < n/2. \tag{4.27b}$$

Equation (4.25) cannot be used at the boundaries, since for those points r_{2N-1}^h and r_{2N+1}^h are not defined. When boundary values are really needed, injection could be used or the stencil from (4.25) without the contributions of the points that lie outside the domain.

Cell centered coarsening

With cell centered coarsening the target grid is coarsened in 2 ways:

- For the interior grid cells each pair of fine grid cells is combined into a single coarse grid cell.
- For the boundary cells no coarsening is applied, so that the coarse grid variables are equal to the target grid variables.

The coarse grid variable is situated in the center of a coarse grid cell consisting of a combination of fine grid cells. The case of a coarse grid cell consisting of 2 fine grid cells is depicted in figure 4.3. Note how the cell boundaries coincide on fine and coarse grid.

In the restriction of the interior cells, each coarse grid cell variable is the average of its coinciding fine grid cell variables. In the interpolation the coarse cell variables are *injected* to coinciding fine grid cell variables.

- The restriction operator I_h^H in this case becomes:

$$I_h^H = \frac{1}{2}\left[1\,1\right]. \tag{4.28}$$

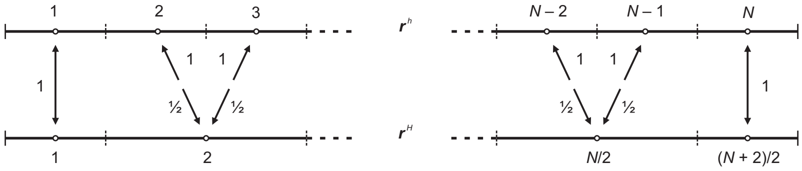


Figure 4.3: Restriction and interpolation using cell centered coarsening.

This results in the following equation relating the coarse grid cell values to the fine grid values:

$$r_N^H = \left(1 \cdot r_{2(N-1)}^h + 1 \cdot r_{2(N-1)+1}^h\right)/2. \tag{4.29}$$

– The interpolation operator I_H^h is:

$$I_H^h = \left[1\ 1\right], \tag{4.30}$$

which gives the following relations for the fine grid cell values obtained from coarse grid values:

$$r_{2(N-1)}^h = r_N^H \qquad 2 \le N \le n/2 \tag{4.31a}$$

$$r_{2(N-1)+1}^h = r_N^H \qquad 2 \le N \le n/2. \tag{4.31b}$$

The boundary cells maintain a fixed unit size. Injection is used in both cases, so that:

$$r_1^h = r_1^H \tag{4.32a}$$

$$r_N^h = r_{(N+2)/2}^H. \tag{4.32b}$$

4.3.2 Coarse grid operator

The KE on the target grid was defined in the following way:

$$L^h \mathbf{c}^h = \mathbf{g}^h. \tag{4.33}$$

The coarse grid problem can be derived by means of the restriction operator I_h^H:

$$I_h^H \left(L^h \mathbf{c}^h\right) = I_h^H \mathbf{g}^h \tag{4.34}$$

Substitution of $\mathbf{c}^h = I_H^h \mathbf{c}^H$ yields:

$$I_h^H \left(L^h I_H^h \mathbf{c}^H\right) = I_h^H \mathbf{g}^h \Rightarrow \left(I_h^H L^h I_H^h\right) \mathbf{c}^H = \mathbf{g}^H \tag{4.35}$$

Therefore the coarse grid operator L^H can be defined as:

$$\boxed{L^H = I_h^H L^h I_H^h} \tag{4.36}$$

This kind of coarsening is called Galerkin coarsening. In practice the coarse grid operator can be obtained very easily from what looks like a complicated matrix multiplication. Like the target grid operator, the coarse grid operator essentially gives weights to each coarse grid variable to define the coarse grid problem. These weights can be obtained by setting a single coarse grid variable to 1, and setting the others to zero. Then, using the restriction of the fine grid operators working on the interpolation of the coarse grid variables, the weight for the 1 on the coarse grid is readily obtained.

The coarse grid operators for each coarsening type are derived below.

Vertex centered coarsening

Consider figure 4.4, which depicts the contributions to the coarse grid operator in the coarse grid point N. The fine grid operators at fine grid points $2N-1$, $2N$ and $2N+1$ are defined as:

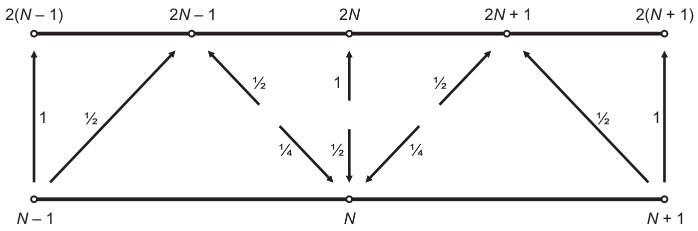


Figure 4.4: Contributions to the coarse grid operator through restriction and interpolation for vertex centered coarsening.

$$L_{2N-1}^h = L^h\left(c_{2(N-1)}^h, c_{2N-1}^h, c_{2N}^h\right) \tag{4.37a}$$

$$L_{2N}^h = L^h\left(c_{2N-1}^h, c_{2N}^h, c_{2N+1}^h\right) \tag{4.37b}$$

$$L_{2N+1}^h = L^h\left(c_{2N}^h, c_{2N+1}^h, c_{2(N+1)}^h\right) \tag{4.37c}$$

where:

$$L_n^h = c_n^h - \frac{\Delta t}{2}\left[f_{n-1}c_{n-1}^h - \left(b_n + f_n\right)c_n^h + b_{n+1}c_{n+1}^h\right] \tag{4.38}$$

Then, applying the interpolation operator I_H^h yields:

$$L_{2N-1}^h = L^h\left(c_{N-1}^H, \tfrac{c_{N-1}^H + c_N^H}{2}, c_N^H\right) \tag{4.39a}$$

$$L_{2N}^h = L^h\left(\tfrac{c_{N-1}^H + c_N^H}{2}, c_N^H, \tfrac{c_N^H + c_{N+1}^H}{2}\right) \tag{4.39b}$$

$$L_{2N+1}^h = L^h\left(c_N^H, \tfrac{c_N^H + c_{N+1}^H}{2}, c_{N+1}^H\right) \tag{4.39c}$$

Finally, restricting these 3 fine grid operators to the coarse grid, the coarse grid operator becomes:

$$\boxed{L_N^H = \frac{1}{4}L_{2N-1}^h + \frac{1}{2}L_{2N}^h + \frac{1}{4}L_{2N+1}^h} \tag{4.40}$$

This formulation can easily be implemented in a Multigrid solver by means of a recurrence relation, constructing the operator on each grid. However, the forward and backward rates in the Kinetic Equation, which are defined in between grid points on the target grid, now get a different physical meaning on the coarse grid. Hence the coarse grid operator is in fact not similar to the target grid operator, and it may behave differently. Beneficial properties of the fine grid operator for relaxation, such as diagonal dominance, may not be inherited by the coarse grid equations. It may therefore be complicated to find a suitable and stable relaxation process on the coarse grid.

In appendix D coarse grid relaxation is analyzed through Local Mode analysis. Analysis points out that Gauss Seidel fails at the coarser grids, due to a changed coarse grid operator. Therefore to solve the coarse grid problem obtained via vertex centered coarsening, a different solution procedure should be used. A possibility is to use Kaczmarz relaxation which alters to system so that it becomes diagonally dominant, and subsequently applies Gauss Seidel relaxation. Convergence of Kaczmarz however, is a lot slower than Gauss Seidel and is therefore not preferred.

Cell centered coarsening

For vertex centered coarsening, the coarse grid operator can also be constructed using Galerkin coarsening. Figure 4.5 displays the locations that are involved in the definition of the coarse grid operator in point N.

The fine grid operators in points $2(N-1)$ and $2(N-1)+1$ are:

$$L_{2(N-1)}^h = L^h\left(c_{2(N-2)+1}^h, c_{2(N-1)}^h, c_{2(N-1)+1}^h\right) \tag{4.41a}$$

$$L_{2(N-1)+1}^h = L^h\left(c_{2(N-1)}^h, c_{2(N-1)+1}^h, c_{2N}^h\right) \tag{4.41b}$$

where as before:

$$L_n^h = c_n^h - \frac{\Delta t}{2}\left[f_{n-1}c_{n-1}^h - (b_n + f_n)c_n^h + b_{n+1}c_{n+1}^h\right] \tag{4.42}$$

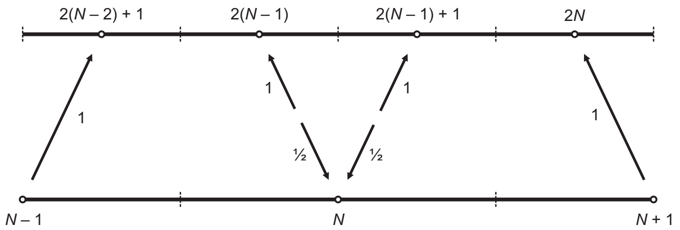


Figure 4.5: Contributions to the coarse grid operator through restriction and interpolation for binning.

Applying the interpolation operator I_H^h yields:

$$L_{2(N-1)}^h = L^h\left(c_{N-1}^H, c_N^H, c_N^H\right) \tag{4.43a}$$

$$L_{2(N-1)+1}^h = L^h\left(c_N^H, c_N^H, c_{N+1}^H\right) \tag{4.43b}$$

Restriction to the coarse grid gives:

$$\boxed{L_N^H = \left(L_{2(N-1)}^h + L_{2(N-1)+1}^h\right)/2} \tag{4.44}$$

As with vertex centered coarsening, the operator can be easily implemented numerically through recurrence.

Also, upon substitution of (4.42) in (4.44) one obtains:

$$L_N^H = c_N^H - \frac{1}{2}\frac{\Delta t}{2}\left[f_{2(N-2)+1}c_{N-1}^H - \left(b_{2(N-1)} + f_{2(N-1)+1}\right)c_N^H + b_{2N}c_{N+1}^H\right] \tag{4.45}$$

Recognizing the prefactor $\frac{1}{2}$ as reflecting inter-grid ratio of cell sizes $\frac{1}{w}$, this operator looks very similar to the KE for bins:

$$L_N^{bin} = \bar{c}_N - \frac{1}{w_N}\frac{\Delta t}{2}\left[f_{n_r(N-1)}\bar{c}_{N-1} - \left(b_{n_l(N)} + f_{n_r(N)}\right)\bar{c}_N + b_{n_l(N+1)}\bar{c}_{N+1}\right] \tag{4.46}$$

where $\bar{c}_N$ is the average bin density, and $n_l(N)$ and $n_r(N)$ are the indices of the left and right droplets in the N^{th} bin.

In essence, Galerkin coarsening applied to the KE with the restriction and interpolation operators considered here, results in the KE for bins on the coarse grid. Further coarsening will result in bin sizes equal to a power of the base bin size, and a similar coarse grid operator. This is interesting since the coarse grid operator now is a representation of the formulation of the original KE, thereby preserving beneficial properties for numerical solving.

The Local Mode analysis performed in appendix D confirms that the coarse grid equations are similar to the target grid equations. Gauss Seidel relaxation performance is similar on the coarse grid compared to the target grid. This enables the use of Gauss Seidel relaxation on all grids in the multi-level cycle.

4.4 Implementation verification

The Local Mode analysis used to analyze the relaxation process, is not only useful to determine convergence behavior. The error amplification factors obtained, actually are a good measure to which the numerical performance can be verified. In fact, the results should not only be a guideline, but rather a goal for the numerical performance. Often when numerical performance is worse than the results predicted by the Local Mode analysis, there is an error in the implementation.

In this section the performance of the Multigrid program is verified against predictions of the Local Mode analysis. It is that Local Mode analysis only considers error reduction on each independent grid. Interpolation effects are not considered, and can be taken into account by performing a two-level analysis. However, for low v_1, nu_2 Multigrid cycles, effects of interpolation are minimal and the convergence rate predicted by Local Mode analysis is a good estimate for the cycle performance.

4.4.1 Relaxation

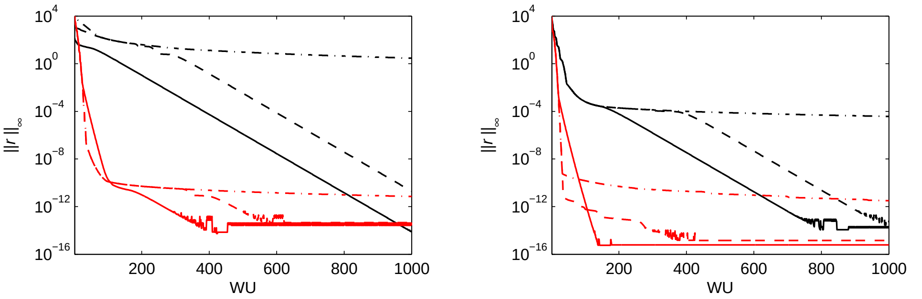
When Local Mode analysis is not performed the performance can be analyzed using the results from the solver. To analyze multi-level performance, first some insight in relaxation performance is required. For this purpose a number of iterations is performed on the system, with a Fourier mode serving as an initial guess:

$$v_n = \sin\left(\frac{k\pi(n-1)}{N}\right), \quad 1 \le n \le N+1, \quad 1 \le k \le \ N-1 \tag{4.47}$$

Here k is the wave number, which specifies the number of half wavelengths on the grid.

As each grid point represents a droplet size n, relaxing the Kinetic Equation is not possible in the sense of different droplet size differences (different Δn). This is because inter-droplet fluxes are defined for $\Delta n = 1$. Comparing the full KE to a coarser bin form of the KE is not an option, as a uniform transition from the single droplets to droplets bins does not scale the inter-droplet fluxes accordingly (especially for small n) . Therefore, the only option of exploring relaxation on different grids is to extend the domain. This is done on 3 grids with 10^2, 10^3 and 10^4 points (maximum droplet size). The initial guesses on the different domains should be comparable. Therefore wave numbers used for the initial guesses are fractions of the maximum droplet size N, i.e. $k = \frac{1}{100}N, \frac{1}{4}N, \frac{1}{2}N, \frac{99}{100}N$.

The logarithm of the maximum norm of the residual is plotted as a function of the number of iterations for different wave numbers in figure 4.6. Clearly, high frequency errors are reduced more efficiently than low frequency errors. This conclusion was already obtained from the Local Mode analysis.



(a) $k_1 = \frac{1}{100}N$ (black), $k_3 = \frac{1}{2}N$ (red)

(b) $k_2 = \frac{1}{4}N$ (black), $k_4 = \frac{99}{100}N$ (red)

Figure 4.6: Maximum norm of residual ($||r||_\infty$) vs. number of iterations for different domain sizes: $N = 10^2$ (solid), $N = 10^3$ (dashed) and $N = 10^4$ (dash-dot). High frequency errors are reduced more efficiently.

Also, on the smaller domains, after some iterations the residual steadily decreases to machine accuracy. This is due to the asymmetry of the operator, discussed in appendix E. It transfers residuals downstream for $n > n_{cr}$, i.e. supercritical clusters. What essentially happens is that, when the residuals reach the end of the domain, they disappear. In this investigation only 1000 iterations have been performed, which was enough for the information to travel form the left to the right boundary for the smaller domains, but not for $N = 10^4$. If relaxation where continued, this would also have been the case for the largest domain.

An arbitrary initial guess consists of many Fourier modes. Relaxation in that case will have a high initial convergence rate, because of the efficient reduction of the high frequent modes. However, the low frequency components will not be reduced as effectively and will dominate convergence after a few iterations.

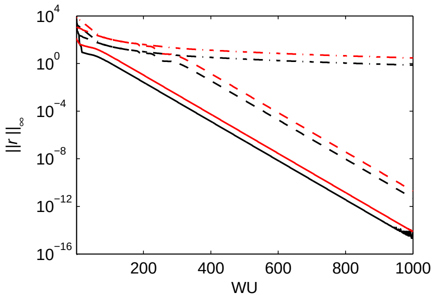


Figure 4.7: Relaxation all 4 modes (black) and the single lowest mode (red). Initial convergence is rapid because of efficient reduction of high frequency modes, but eventually slows down to the speed of the remaining low frequency mode.

In figure 4.7 residual reduction for the low wave number $k_1 = \frac{1}{100}N$ is compared to residual reduction for an initial guess that combines all 4 wave numbers, i.e.:

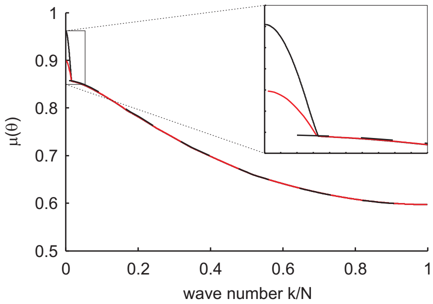
$$v_n^0 = \frac{1}{4}\left[\sin\left(\frac{k_1\pi(n-1)}{N}\right) + \sin\left(\frac{k_2\pi(n-1)}{N}\right) + \sin\left(\frac{k_3\pi(n-1)}{N}\right) + \sin\left(\frac{k_4\pi(n-1)}{N}\right)\right] \tag{4.48}$$

From the figure it is clear that initial convergence is faster for the combined modes, but after a few iterations it slows down to the same speed as for the lowest frequency mode. In other words, after the a few relaxations the error is smooth as only the low frequent component remains.

The residual reduction for each mode can be described in the following manner:

$$||r^{(i)}(\theta)||_\infty = \mu^i(\theta)||r^{(0)}(\theta)||_\infty \tag{4.49}$$

where $\mu(\theta)$ is a mode dependent convergence factor. This is the numerical equivalent of the error amplification factor obtained by the Local Mode analysis. Figure 4.8 shows the values for different wave numbers k.



k	$N = 10^2$	$N = 10^3$	$N = 10^4$
$\frac{1}{100}N$	0.8572	0.8825	0.9167
$\frac{1}{4}N$	0.7594	0.7583	0.7582
$\frac{1}{2}N$	0.6653	0.6653	0.6653
$\frac{99}{100}N$	0.5972	0.5972	0.5972

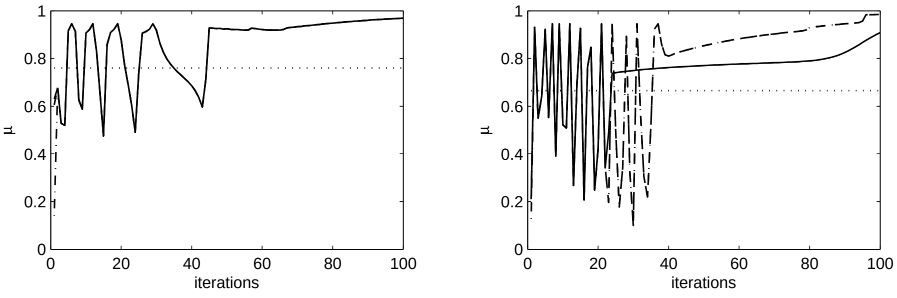
Figure 4.8: Error amplification factors $\mu(\theta)$ obtained from Local Mode analysis for different domain sizes $N = 10^2$ (dashed black), $N = 10^3$ (solid red) and $N = 10^4$ (solid black). Error reduction is similar for different domain sizes except for low frequency modes, which converge slower on large domains.

The size of the domain has virtually no effect on the error reduction. Most error components have their maximum error amplification factor situated at the beginning of the domain. However, for low frequency components this is not the case. These components have 2 maxima in the n direction, 1 at the beginning of the domain and one at $n = N$. For a small domain, the first maximum is the largest. For a large N the second maximum is larger, appearing as a "bump" in the graph. Increasing the domain size will cause this maximum to approach 1, worsening error reduction for low frequency components.

The error amplification factor $\mu(\theta)$ can be compared to the measured residual reduction, defined as:

$$\tilde{\mu} = \frac{||r^{(i+1)}||_\infty}{||r^{(i)}||_\infty} \tag{4.50}$$

Both are displayed in figure 4.9 for the modes $k_2 = \frac{1}{4}N$ and $k_3 = \frac{1}{2}N$. For the first iterations the measured reduction factor is very oscillatory. However, its mean shows good correlation with the predicted error amplification factor from the Local Mode analysis.



(a) $k_2 = \frac{1}{4}N$ (b) $k_3 = \frac{1}{2}N$

Figure 4.9: Measured residual reduction $\tilde{\mu}$ vs. predicted error amplification factor μ (dotted).

Summarizing, as for any relaxation process applied to more or less elliptic problems, the convergence of the solution is impeded by an ineffective reduction of low frequency modes present in the error. This is when multilevel algorithms are effective.

4.4.2 Two level cycle

To test the effectiveness of the coarse grid correction a two level cycle is is considered. The two level cycle consists of:

- ν_1 relaxations on the target grid h.
- Restriction of residuals to coarse grid H, and construction of coarse grid operator L^H.
- Solve coarse grid problem *exactly*, using a direct solver.
- Correction of the target grid solution with the coarse grid result by interpolation.
- ν_2 relaxations on the target grid.

The coarse grid problem needs to be solved exactly to ensure that the coarse grid correction will be optimal. To measure the computational work, a Work Unit is defined as the equivalent 1 relaxation sweep on the target grid. Solving the coarse grid problem approximately costs 3 WUs.

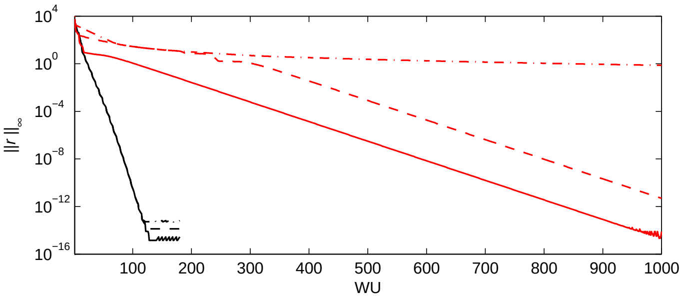


Figure 4.10: Performance of multilevel cycling (black) vs. relaxation (red). Performance is measured in WUs (1 WU = 1 relaxation sweep on the finest grid). Results for domain sizes $N = 10^2$ (solid), $N = 10^3$ (dashed) and $N = 10^4$ (dash-dot).

The two level cycle is executed using $\nu_1 = 2$ and $\nu_2 = 1$, so a total of 6 WUs is required for a single correction cycle, where solving the coarse grid exactly is assumed to require 3WUs. The initial guess consists of Fourier modes with wave numbers $k = \frac{1}{100}N, \frac{1}{4}N, \frac{1}{2}N$ and $\frac{99}{100}N$.

Figure 4.10 shows the performance of two level cycles vs. single grid relaxations. With relaxation the residual reduction is slowed down by the ineffective reduction of low frequency modes. With the two level cycle, these error components are transferred to the coarse grid where they can be reduced more efficiently. The only components that are to be resolved on the target grid are the high frequent components with $\pi/2 \le |\theta| \le \pi$.

The residual reduction of 1 cycle can be estimated using the results from the Local Mode analysis. Similar to single grid relaxation, the maximum error amplification factor dictates the residual reduction rate. With only the high frequent components to be resolved on the target grid, the asymptotic smoothing rate is defined as:

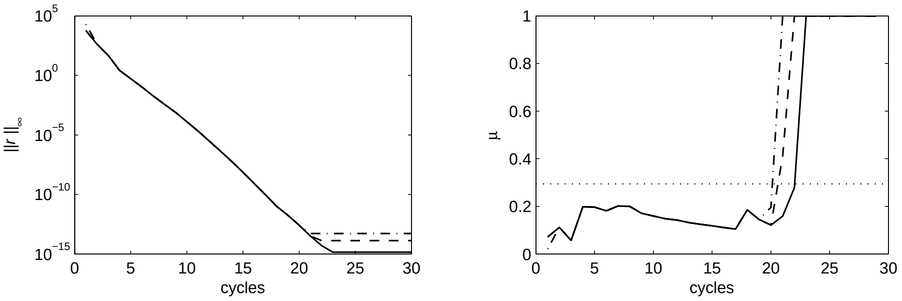
$$\bar{\mu} = \max_{\pi/2 \le |\theta| \le \pi} \mu(\theta) \tag{4.51}$$

In the coarse grid correction cycle a total of $\nu_1 + \nu_2$ relaxations are preformed on the fine grid, therefore the total residual reduction of a coarse grid correction is the amplification factor μ_c, defined as:

$$\mu_c = \bar{\mu}^{\nu_1 + \nu_2} \tag{4.52}$$

Figure 4.11 displays the reduction of the residual per cycle. The measured amplification factor $\tilde{\mu}$ is even lower than the predicted amplification factor μ_c.

This can be explained by the asymmetry in the operator that enhances error reduction by moving residuals downstream , off the grid. This effect is stronger on coarser grids and for smaller domains.



(a) Residual max norm per cycle (b) Measured $\tilde{\mu}$ vs. predicted μ_c

Figure 4.11: Residual reduction using a two level cycle. The measured amplification factors are even better than the predicted value.

4.5 Global constraint

The depletion equation is a global conservation law, which affects all interior equations equally. Because of its nature as a global constraint, it is relaxed separately.

Global conditions need not be treated at all on the target grid, see [6]. In the multiscale approach it is important to realize that there can be no error-smoothing related to such single conditions. All that has to be done is to transfer the residual of the condition to the next coarser grid. For a nonlinear condition FAS should be used, whereas linear conditions can be treated with CS if the constraint is defined in terms of the error. As the coarsest grid is much smaller than the target grid, information of its application is propagated through the grid much faster.

4.5.1 Coarsening

With cell centered coarsening only the interior grid is coarsened, whereas the boundary is not coarsened at all. The restriction and interpolation in between grids is carried out through injection, i.e.:

$$I_h^H = I_H^h = 1 \quad \Rightarrow \quad r_1^H = r_1^h \tag{4.53}$$

Using Galerkin coarsening, the coarse grid operator is defined as:

$$L_1^H = I_H^h L_1^h I_h^H = L_1^h \tag{4.54}$$

with:

$$L_1^h = c_{sat}S^h + \sum_{n=2}^{N} nc_n^h \tag{4.55}$$

The saturation is injected so that $S^H = S^h$. The droplet number densities are averaged however so that $c_n^H = (c_{2(n-1)}^h + c_{2(n-1)+1}^h)/2$. The coarse grid operator then yields:

$$\begin{aligned} L_1^{2h} &= c_{sat}S^{2h} + \sum_{n=2}^{N} [2(n-1) + 2(n-1) + 1] c_n^{2h} \\ &= c_{sat}S^{2h} + \sum_{n=2}^{N} [4(n-1) + 1] c_n^{2h} \end{aligned} \tag{4.56}$$

The prefactor n in the target grid operator sum represents the amount of monomers in a droplet c_n^h. Coarsening automatically yields a prefactor which corresponds to the total amount of monomers in droplet bin c_n^H.

Consider a droplet on level $k - l$, with target level k. The total amount of droplets in a bin is:

$$n_{bin(n)} = (n_{l(n)} + n_{r(n)})\frac{w}{2} \tag{4.57}$$

where $n_{l(n)}$ and $n_{r(n)}$ are the amounts of monomers in the leftmost and rightmost droplet in bin n with size $w = 2^{k-l}$. These amounts correspond to the location of these droplets on the target grids, which in turn can be specified in terms of the bin location n:

$$n_{l(n)} = w(n-1) - (w-2) \tag{4.58}$$

$$n_{r(n)} = w(n-1) + 1 \tag{4.59}$$

The total amount of monomers in each bin then becomes:

$$n_{bin(n)} = (n - \tfrac{3}{2})w^2 + \tfrac{3}{2}w \tag{4.60}$$

yielding the coarse grid operator:

$$L_1^{k-l} = c_{sat}S^{k-l} + \sum_{n=2}^{N} \left[(n - \tfrac{3}{2})w^2 + \tfrac{3}{2}w\right] c_n^{k-l} \tag{4.61}$$

As explained above the global constraint is only applied on the coarse grid. Implementation can be done conveniently using FAS. The coarse grid problem with FAS is:

$$L_1^{k-1}(\mathbf{c}^{k-1}) = L_1^{k-1}(I_k^{k-1}\mathbf{c}^k) + r^k \tag{4.62}$$

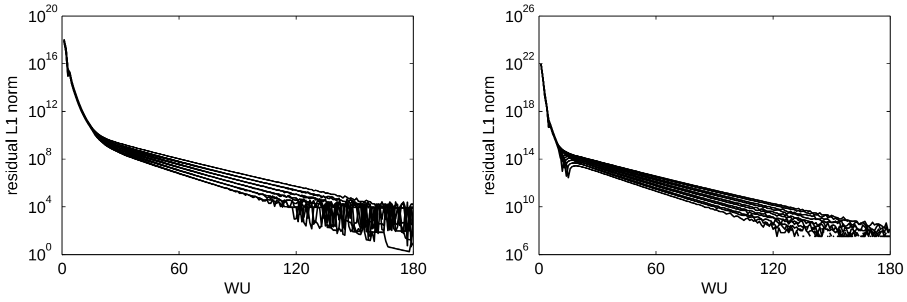
with $\mathbf{c} = (S, c_2, \ldots, c_N)^T$.

4.5.2 Application

With the global constraint is defined for each grid level, its application can be investigated. In contrast to interior relaxation, which smoothes the error in the interior of the domain, relaxation of the global constraint is treated separately.

A global change to the error, e.g. a change in magnitude, will not affect the error smoothing process and thus is allowed. In case relaxation of the global constraint introduces a local change, the interior relaxation process could be frustrated. This can however be counterbalanced by applying a certain amount of underrelaxation to the application of the global constraint, so that interior relaxation remains convergent.

In figure 4.12 residual reduction for the interior of the domain and for the global constraint are displayed. With the KE the global constraint can be applied without underrelaxation after each relaxation sweep. This does not disrupt the error smoothing process, implying that local changes are small enough not to upset interior relaxation.



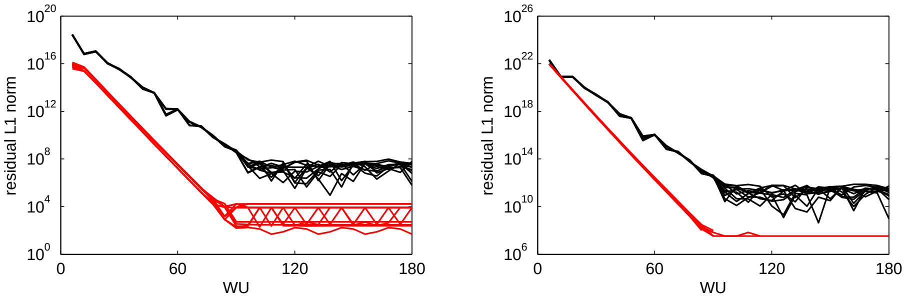
(a) Interior residual

(b) Global residual

Figure 4.12: Residual reduction during computation of 15^{th} to 25^{th} time step ($\Delta t = 10^{-7}$ s, $N = 8194$). 180 sweeps per time step, $\omega = 1$. Global constraint is applied after each interior relaxation sweep.

In a multi-level cycle, the global constraint should only be relaxed on the target grid or on the coarsest grid. Applying the global constraint on an intermediate level will interfere with the correction tot the finer level. Figure 4.13 shows the residual reduction with a global constraint either applied at the target level, or at the coarsest level.

Apparently, application of the global constraint at the target grid yields the best performance. As with relaxation the global constraint can be applied after each cycle without the need for underrelaxation. The reason that its application is more effective on the target level than on the coarsest level is because the constraint requires an integral quantity to be preserved. This integral quantity is best represented on the target grid. Uniform coarsening of the KE results in a poor approximation on the coarsest grid, especially for small n where the



(a) Interior residual

(b) Global residual

Figure 4.13: Residual reduction during computation of 15th to 25th time step ($\Delta t = 10^{-7}$ s, $N = 8194$). 30 cycles per time step, $\omega = 1$. The global constraint either applied at the coarsest level (black), or at the target level (red).

DSD typically has very steep gradients.

The theoretical benefit of fast information propagation on coarser, smaller grids, doe not play a role with the KE. The changes in the supersaturation S and therefore c_1 are relatively small, so that the solution in the rest of the domain is unaffected. Larger time-steps would perhaps induce larger changes, for which the speed of information propagation becomes more significant.

4.6 Multi-level cycle extensions

A single multi-level cycle requires $O(N)$ operations. In his work Venner [7] describes that the amount of cycles required to converge to the level of the discretization error is $O(\ln N)$, so the total amount of work equals $O(N \ln N)$ operations. However, this estimate is based on the assumption that the initial error is $O(1)$. If this initial error is reduced, so would the amount of required operations. Two extensions to a coarse grid correction cycle program, which aim to produce this improved initial guess, are Full Multigrid (FMG) and the F-cycle. Both concepts and the application to the KE are treated below.

4.6.1 *Full Multigrid*

With Full Multigrid, or FMG, the solution on coarse grid H is used as a starting solution on the fine grid with $h = H/2$. Then, the initial error equals the discretization error on grid H instead of $O(1)$. For a discretization error of order p, the initial error is $O(H^p)$, so that the number of cycles M required for convergence is:

$$\mu^M H^p \le h^p \tag{4.63}$$

and so

$$M \geq \frac{p \ln(2)}{\ln(1/\mu)} \tag{4.64}$$

where μ is the error reduction per cycle. Since μ is independent of the number of grid points N the required amount of cycles is also independent of N, instead of $O(\ln N)$ as before. Therefore the total amount of work becomes $O(N)$ operations.

The process of using a coarse grid solution as a starting solution for the fine grid is applied recursively with FMG, which is displayed in figure 4.14. Since the coarse grids already are initialized for the multi-level cycle, FMG can be implemented very easily. Starting from the coarsest grid, a starting solution is supplied to the successive finer grid. Then a multi-level cycle is performed to obtain a solution converged to the level of the discretization error, which is represented by the double circles.

Attention has to be paid to the interpolation of a coarse grid solution to a fine grid starting solution, since in most cases a higher order of interpolation is required than used in the multi-level cycle.

4.6.2 F-cycle

For time dependent problems, instead of computing each time step separately, the solution of the previous time step can act as a basis to which only the changes in time are considered. Consider the following transient problem, where $L^{h,m}$ is a differential operator defined on grid h at time m, with $\mathbf{c}^{h,m}$ the solution to be computed and $\mathbf{g}^{h,m}$ a right-hand side function:

$$L^{h,m}\mathbf{c}^{h,m} = \mathbf{g}^{h,m}. \tag{4.65}$$

Using the solution from the previous time $m-1$ as an approximation for the solution at the current time, the residual can be defined as:

$$\mathbf{r}^{h,m} = \mathbf{g}^{h,m} - L^{h,m}\mathbf{c}^{h,m-1} \tag{4.66}$$

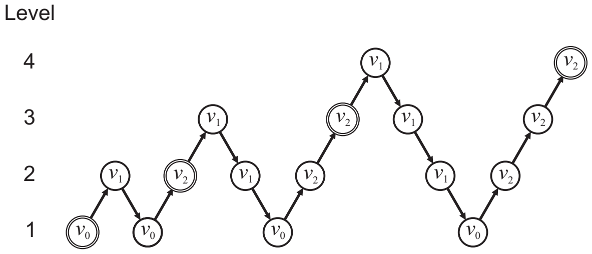


Figure 4.14: FMG algorithm, 1 V(v_1, v_2)-cycle per refinement

Defining:

$$\delta^{h,m} = \mathbf{c}^{h,m} - \mathbf{c}^{h,m-1} \tag{4.67}$$

equation (4.65) can be written as:

$$L^{h,m}(\mathbf{c}^{h,m-1} + \delta^{h,m}) = L^{h,m}\mathbf{c}^{h,m-1} + \mathbf{r}^{h,m}. \tag{4.68}$$

In this way the information of the previous time step is used for the calculation for the current time step. Solution of the problem can be done using standard CS or FAS cycles. However as with FMG, the F-cycle aims to produce an initial error on the target grid equal to the discretization error on the coarse grid.

Therefore equation (4.68) is solved on the coarsest grid first. Hence solving $\hat{\mathbf{c}}^{H,m}$ from:

$$L^{H,m}(\hat{\mathbf{c}}^{H,m}) = L^{H,m}(I_h^H\mathbf{c}^{h,m-1}) + I_h^H\mathbf{r}^{h,m} \tag{4.69}$$

an approximation $\hat{\mathbf{c}}^{h,m}$ is calculated according to:

$$\hat{\mathbf{c}}^{h,m} = \mathbf{c}^{h,m-1} + I_H^h(\hat{\mathbf{c}}^{H,m} - \mathbf{c}^{h,m-1}). \tag{4.70}$$

Then a normal CS or FAS cycle can be used to reduce the error up to the level of the discretization error. As with FMG, the process described above can be applied recursively as depicted in figure 4.15. Note the first downward leg where zero relaxations are performed, as the target grid solution on the previous time step is purely restricted to the coarsest grid.

4.6.3 Application to the KE

With an initial guess having an error equal to the truncation error on the coarse grid, $O(1)$ cycles should be required to converge the solution on the target grid. However, it is sometimes required to perform additional correction cycles after the FMG or F-cycle algorithm are performed. Therefore, performance is only improved if the work of FMG or an F-cycle plus the additional work of using coarse grid correction cycles to converge the solution on the target grid, is less

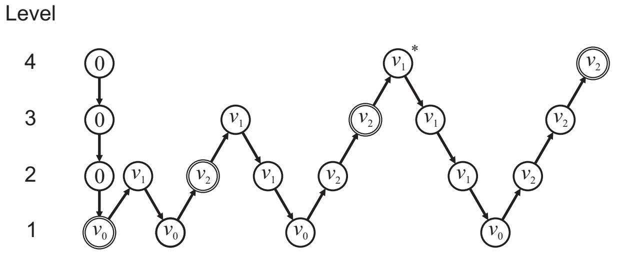
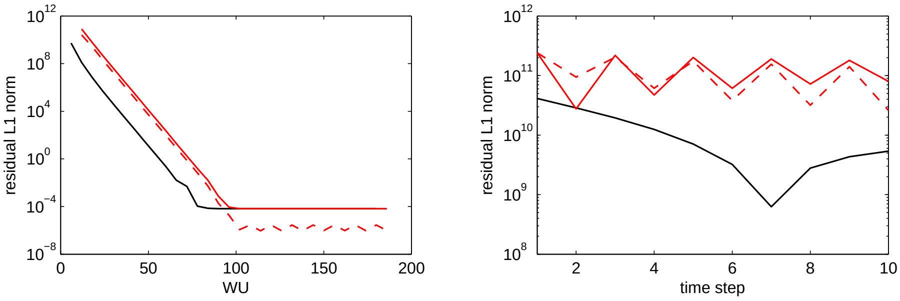


Figure 4.15: $F - V(v_1, v_2)$ cycle algorithm

than the work of $O(\ln N)$ coarse grid correction cycles ordinarily required to obtain a converged solution.

However, for the current multilevel implementation of the KE performance does not seem to improve. Figure 4.16 shows the performance of the solver for starting solutions obtained from (a) the previous time step, (b) FMG and (c) an F-cycle.



(a) Residual reduction of the 10th time step

(b) Residual of starting solution

Figure 4.16: Residual as a function of work for different starting solutions: previous time step (black), FMG (solid red), F-cycle (dashed red). FMG and the F-cycle require approximately 6 WU, and provide a worse starting solution than the previous time step.

Looking at the initial residual norms, it is clear that the previous time step produces a more accurate starting solution than obtained from FMG or the F-cycle. Also FMG and the F-cycle require extra work to obtain the starting solution, causing the residual reduction to be delayed. To explain this behavior the starting solution and its initial error are investigated.

As is explained above, a performance gain from FMG and the F-cycle relies on the assumption that the initial error on the fine grid is equal to the discretization error on the coarse grid. First, as the KE is not a continuous problem, there is no discretization error on the target grid. Its solution is the exact solution. Therefore the error on the coarse grids should not be defined as a discretization error, but rather an approximation error with respect to the target grid.

The current multi-level implementation coarsens the grid in a uniform fashion, for all n. The coarsening corresponds to a zero-th order interpolation used for bins. In appendix C it was already pointed out that the approximation error is minimal when the DSD is smooth. However, especially for small n, the DSD has very steep gradients, resulting in large approximation errors when coarsening is applied in that region. Therefore, both FMG and F-cycles will always produce a significant error in the starting solution on the target grid when using uniform coarsening.

This approximation error could perhaps be reduced to some extend by using a higher order of interpolation between different grid levels. However, even when the coarse grid solution can yield an accurate approximation to the fine grid solution, the grid cells at the beginning of the domain should not be coarsened at all. Grid coarsening is only allowed for large n where the DSD becomes relatively smooth.

The conclusion is that with the current coarsening, the most efficient approach is to use the solution from the previous time step as a starting solution for the current time step and use simple cycling to improve it to the required accuracy level. Since for nucleation experiments the solution only seems to change at the nucleation front, the previous time step already is accurate for a large part of the domain.

CHAPTER 5

RESULTS: NUCLEATION PULSE EXPERIMENT

In this chapter the results for the nucleation pulse experiment are discussed, both for a fixed monomer number and depletion. The case consists of a stepwise constant pressure-temperature profile comprising a region with high supersaturation (*region I*), followed by a region with low supersaturation (*region II*). The experiment settings are equal to the experiment carried out by van Putten *et al.* [5], for which the test case conditions are given in table 5.1. Properties of water are as obtained by Wölk & Strey [10]. The time step sizes for accurate time integration are $1 \cdot 10^{-8}$ s, which is determined from a convergence analysis treated in appendix F.

Table 5.1: Nucleation pulse test case conditions for water with nonequilibrium vapor molar fraction $y_w = 7 \cdot 10^{-3}$.

Parameter	*Region I* ($t \leq 50\mu$s)	*Region II* (50μs$< t \leq 300\mu$s)
p(kPa)	77.0	115.5
T(K)	240.0	270.0
S	14.34	1.67
n_{cr}	23	1885

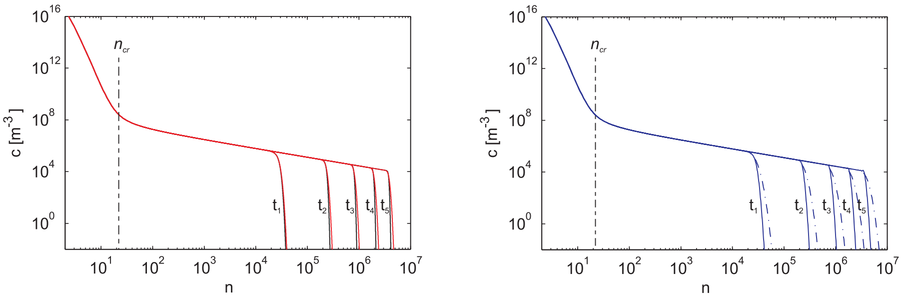
To compare his results to the Classical Nucleation Theory, van Putten used the equilibrium number density $c_1^{eq} = c_1^S \exp(-\Theta)$ as a boundary condition. However, for the computation of the forward rates he did not apply this correction so there is an inconsistency between the two. Furthermore, van Putten uses a slightly different definition for the forward rate in the limit of $n \to 1$. The inconsistency and forward rates as applied in his experiment were also applied here to validate the results.

5.1 Fixed monomer density

5.1.1 Results

The experiment is carried out using the KE for grouped droplets, as described in section 3.4. The largest droplet size is fixed at $N = 5 \cdot 10^8$, for which a total of 2050, 16386 or 131074 bins are used. This corresponds to 12, 15 or 18 levels in the multi-level cycle respectively, with in all cases a minimum of 3 bins on the coarsest grid in.

In section 3.4.2 it was noted that the 1st order interpolation can be unstable due to the lack of upwinding and a limiter. However, a stable simulation was achieved for 2050 bins. The results of this simulation is presented here next to the results obtained by zero-th order interpolation.



(a) 131074 (black) and 16386 bins (red)

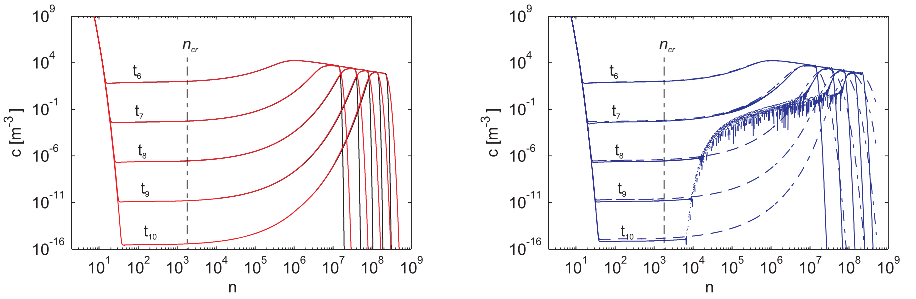
(b) 2050 bins using first (solid) and zero-th order interpolation (dash-dot)

Figure 5.1: Multi-level results for nucleation pulse in region I at $t_1 = 10\ \mu$s, $t_2 = 20\ \mu$s, $t_3 = 30\ \mu$s, $t_4 = 40\ \mu$s and $t_5 = 50\ \mu$s. CS V(2,1)-cycles.

Figure 5.1 shows the results in region I of the nucleation pulse, computed with 5 CS V(2,1)-cycles per time step. All simulations produce similar droplet size distributions, except at the nucleation front. There it seems that the approximation of the full KE with the grouping of droplets results in an overestimation of the nucleation front. When more bins are used, the overestimation is reduced.

Also, from the simulations with 2050 bins, the higher 1st order interpolation shows to be much more accurate than the zero-th order interpolation. 2050 bins with 1st order interpolation is as accurate as 16386 bins with zero-th order interpolation, a significant increase in accuracy. However, small oscillations around the front indicate that instabilities are present.

The analysis of the approximation error in appendix C showed that a zero-th order interpolation can only be used when the gradients in the DSD are close to zero. At the nucleation front this clearly is not the case, resulting in a large approximation error in this area.



(a) 131074 (black) and 16386 bins (red)

(b) 2050 bins using 1^{st} (solid) and zero-th order interpolation (dash-dot)

Figure 5.2: Multi-level results for nucleation pulse in region II at $t_6 = 100\ \mu$s, $t_7 = 150\ \mu$s, $t_8 = 200\ \mu$s, $t_9 = 250\ \mu$s and $t_{10} = 300\ \mu$s. FAS V(2,1)-cycles.

In figure 5.2 the results are plotted for region II of the nucleation pulse. Again, the approximation by bins tends to overestimate the nucleation front. Also, the 1^{st} order interpolation results in more significant errors for t_8 to t_{10}. The instabilities only occur in the supercritical region of the DSD. However, the prediction of the growth appears not to suffer from these instabilities.

5.1.2 Performance

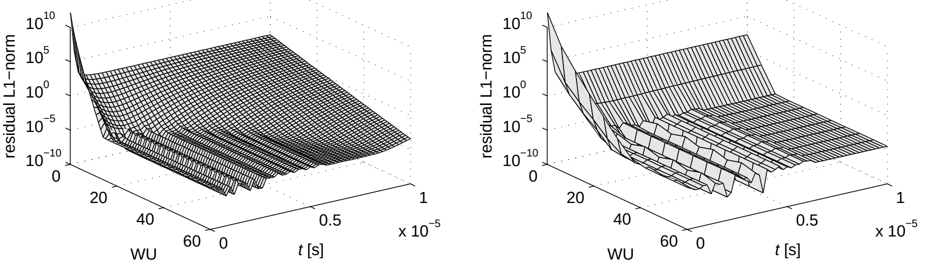
In this section the performance of the multilevel solver is analyzed. For this the nucleation pulse experiment is considered for the first 10 μs. The performance of the multi-level solver is compared to the performance of relaxation and a direct solver using RK4.

The multi-level and relaxation based solvers require $\Delta t = 1 \cdot 10^{-8}$ s for accurate computation, as derived in appendix F. The direct solver is bounded by a CFL condition requiring $\Delta t = 5 \cdot 10^{-11}$ s. This CFL condition becomes stronger when the nucleation front advances and the droplet growth speed increases.

For an implicit solver to have an advantage over a direct solver, the amount of work per time step should not exceed the amount of work of an equivalent amount of direct time steps. In this case, each implicit time step is equal to 200 direct time steps. Considering a single relaxation sweep on the target grid and computation of a single direct time step as $O(N)$ operations, or 1 Work Unit (WU), then computation of an implicit time step should not exceed 200 WUs.

Full KE

Figure 5.3 displays the residual reduction of the implicit solver using relaxation and V-cycles for the full KE. It is clear that both methods are well within the limit of 200 WUs, resulting in a performance increase compared to the direct solver.



(a) Relaxation (b) V-cycles

Figure 5.3: Interior residual reduction by implicit trapezoidal solver ($\Delta t = 10^{-8}$) for the full KE.

Relaxation is surprisingly effective. Instead of the characteristic initially large but eventually stagnating convergence speeds, residuals are reduced in $O(N\ln(N))$ operations, especially for initial droplet growth. In fact, at the initial stages of condensation, relaxation outperforms the multi-level solver. This steady convergence behavior with relaxation indicates that the error consists of a few error components which are of high frequency, so that relaxation is effective.

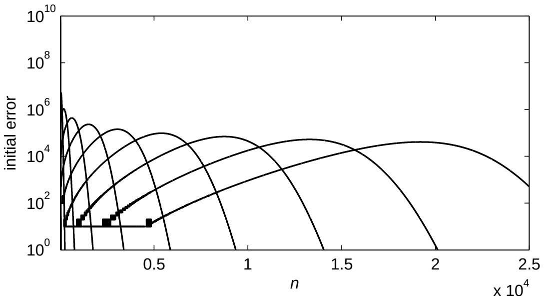


Figure 5.4: Initial errors at $t = 1, 2, \ldots, 10$ μs for full KE. Initial errors exist at the front of the DSD and become of lower frequency as the front advances.

To confirm this suspicion, the initial errors at times $t = 1, 2, \ldots, 10$ μs are plotted in figure 5.4. This reveals that the error only exists at the front of the DSD, which for small Δt is a small portion of the domain, i.e. a high frequent error component. When the DSD advances, droplet growth speed increases and therefore the error is present on a larger portion of the domain. This reduces the error component frequency, resulting in a slower convergence speed observed with relaxation for large t.

Only in the case that the DSD is advanced far enough, the multi-level solver gives a performance improvement. The low frequent errors are reduced as effective as the high frequent errors. For a larger time step size, the performance improvement of the multi-level solver will become better due to the lower error component frequency, and increased error complexity.

KE for grouped droplets

In case bins are use to group the droplets, the above argument that the frequency of the initial error components decreases for large t does not hold. The bin size is increased exponentially with n which is suited to the exponential advancement of the nucleation front. Therefore, the front advancement per time step is more or less constant in the bin-space.

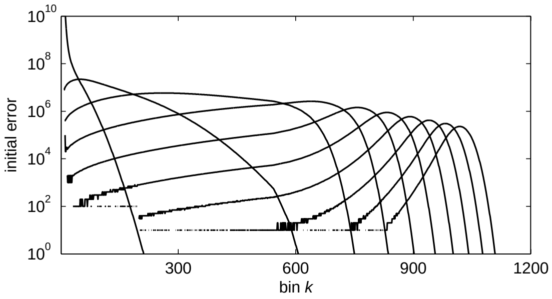


Figure 5.5: Initial errors at $t = 1, 2, \ldots, 10$ μs for binned KE. Initial errors exist at the front of the DSD but remain of high frequency as the increasing bin size matches the exponential growth rate.

This is confirmed by figure 5.5, for which the same simulation is repeated for 2050 bins. In this case the initial error at the front of the DSD remains of high frequency as the increasing bin size matches the increasing growth rate. In fact, frequency of the initial error component seems to increase due to the very large bins for larger droplets.

Because of the high frequency nature of the initial error component, relaxation remains effective when the simulation progresses. This is confirmed by figure 5.6. Also the observation that the frequency of the error component increases over time, is confirmed by the increased effectiveness of relaxation.

The multi-level solver also benefits from the high frequency as relaxation lies at its basis. The advantage of the correction cycle however is lost, as most of the residual is reduced in the first few relaxation sweeps.

5.2 Depletion

5.2.1 Results

The assumption that the monomer number density is unaffected, is only valid for a very short nucleation pulse. When the duration of the pulse is increased more droplets are able to cross the critical droplet size, allowing more large droplets to be formed. To analyze the impact with depletion the nucleation pulse experiment is repeated. In this case however, only region I of the pulse is considered for a duration of 100 μs.

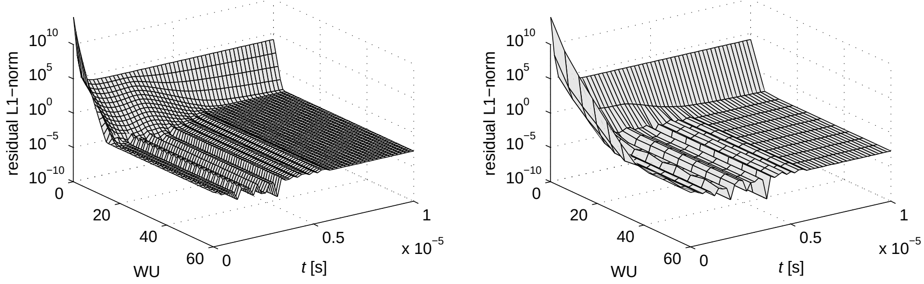
Furthermore, depletion must be accounted for by the additional depletion equation (3.8). This equation is considered as a global constraint and is treated separately from interior relaxation, as explained in section 4.5. In the multilevel program the global constraint is only treated on the target grid.

Figure 5.7 displays the results for the 100 μs nucleation pulse. This experiment is performed using 15 levels with 3 bins on the coarsest level, corresponding to 16386 bins.

Clearly depletion cannot be neglected for a pulse of long duration. Immediately the supersaturation S drops from 14.34 to 13.83. This however is caused by the DSD changing from the starting solution containing only monomers, to a solution containing monomers and small droplets. Should a saturated equilibrium distribution have been used this drop would not have been as significant.

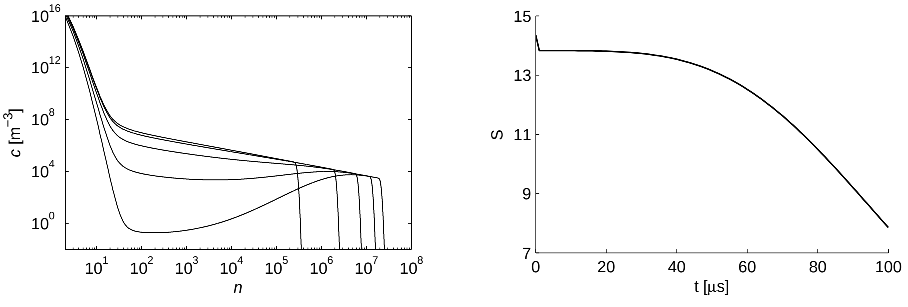
After the initial drop the supersaturation is constant up to about 20 μs. From that point on, the effect of formation of large droplets on the vapor density can be noticed. Corresponding to the drop in the supersaturation the critical cluster size n_{cr} increases, reducing the probability of a droplet crossing this thermodynamic barrier. At 100 μs the supersaturation has dropped to 7.86 so that very few supercritical droplets are formed.

Next the case of total depletion is considered. For this purpose a nucleation



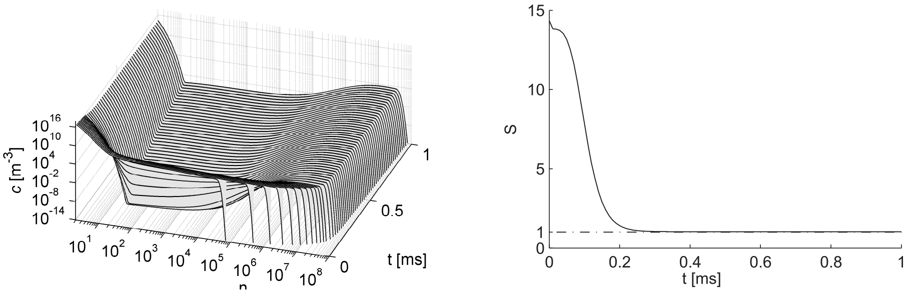
(a) Relaxation (b) V-cycles

Figure 5.6: Interior residual reduction by implicit trapezoidal solver ($\Delta t = 10^{-8}$) for the binned KE.



(a) Droplet size distribution (b) Supersaturation S

Figure 5.7: Multi-level results for depletion in region I at $t_1 = 20$ μs, $t_2 = 40$ μs, $t_3 = 60$ μs, $t_4 = 80$ μs and $t_5 = 100$ μs. CS V(2,1)-cycles.



(a) Droplet size distribution (b) Supersaturation S

Figure 5.8: Nucleation pulse of 1ms. The vapor is depleted until an equilibrium at $S = 1$ is reached.

pulse in region I is considered for a duration of 1 ms. The largest droplet considered is $N = 10^9$, for which 16386 bins are used on the target level. Results are displayed in figure 5.8.

The results show that after 0.2 ms an equilibrium situation $S = 1$ is reached. At this time the critical cluster size n_{cr} is large enough to prevent the formation of new supercritical clusters. The growth of existing supercritical droplets is stopped because at equilibrium $J_n = J_{n+1}$.

5.2.2 Performance

For the nucleation pulse experiment with depletion, only the performance of relaxation and CS V(2,1)-cycles are compared. Since the depletion equation is an implicit equation, a direct solver is not considered. However, it was shown above that the trapezoidal rule solver using relaxation or V-cycles already outperforms a direct RK4 solver. The binned KE using 16386 bins on the target grid was used, with a time step size of $\Delta t = 10^{-8}$ s. The nucleation pulse sans depletion showed that relaxation is quite effective in this case. This could be contributed to the high frequent initial errors on the grid.

With depletion however, the initial errors do not merely exist at the front of the distribution. Because the formation of droplets reduces the monomer density, the lower boundary value and the supersaturation S are affected. The latter also affects the critical droplet size n_{cr}, so that the amount of droplets that become supercritical is altered. All these effects contribute to a more complex difference between subsequent solutions of the KE, creating more complex initial errors.

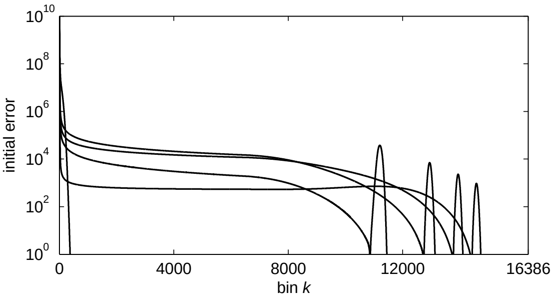


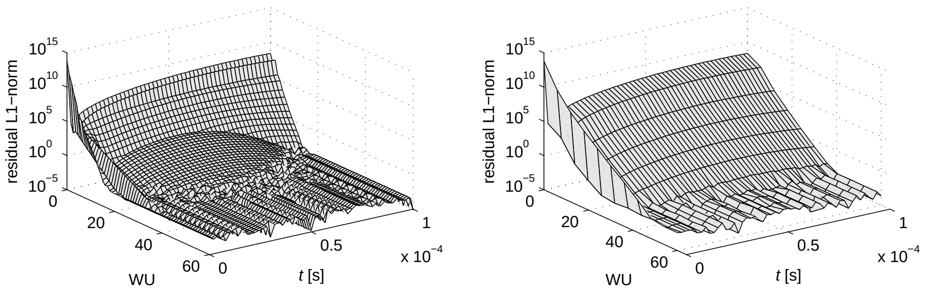
Figure 5.9: Initial errors at $t = 20, 40, \ldots, 100$ μs for binned KE with depletion. Initial errors are more complex compared to fixed monomer density experiment.

In figure 5.9 the initial errors are plotted at t = 20, 40, 60, 80 and 100 μs. The high frequent errors that are present at the front of the DSD are still visible. However, there also is a more complex error visible, spanning a large portion of the domain. It is to be expected that V-cycles are better suited than relaxation to reduce these types of errors due to the low frequent components the errors contain.

Figure 5.10 displays the interior residual reduction of relaxation and V-cycles, where relaxation seems to be most effective. However, the more complex initial error does impact the relaxation speed when compared to figure 5.10. The fast initial reduction is still apparent, but the slower reduction of low frequencies is seen as a ‘belly’ in the surface plot. V-cycles converge to machine accuracy with a steady convergence rate $\bar{\mu}_c$.

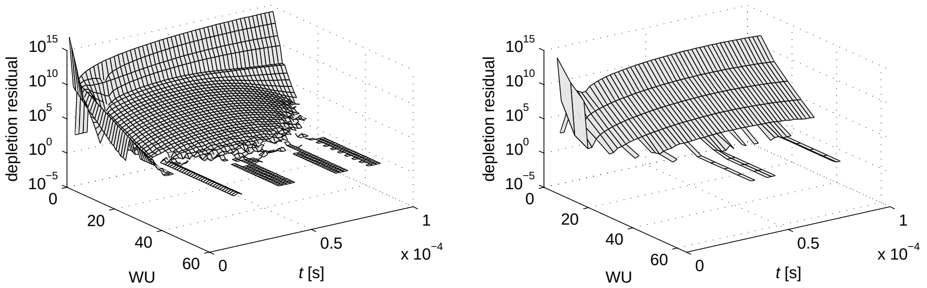
Furthermore, the interior relaxation is influenced by the application of the global constraint. This becomes clear when looking at figure 5.11, which depicts the residual reduction of the depletion equation. There is a clear resemblance between interior residual reduction and depletion residual reduction because both processes are intertwined.

The global constraint was applied before each relaxation sweep at the target grid. For the V-cycle this constitutes 3 applications of the global constraint



(a) Relaxation (b) V(2,1)-cycles

Figure 5.10: Interior residual reduction by implicit trapezoidal solver ($\Delta t = 10^{-8}$) for the binned KE with depletion.



(a) Relaxation (b) V(2,1)-cycles

Figure 5.11: Depletion residual reduction by implicit trapezoidal solver ($\Delta t = 10^{-8}$) for the binned KE with depletion.

per cycle. Each relaxation sweep or V-cycle is designed to reduce the overall residual in the interior of the domain. When the solution c_1 is altered after this, the interior residual changes accordingly. This behavior is even more severe for the KE. Here the residuals for the first equations contribute more to the total residual than the other equations, since the solutions there are several orders higher than in the rest of the domain.

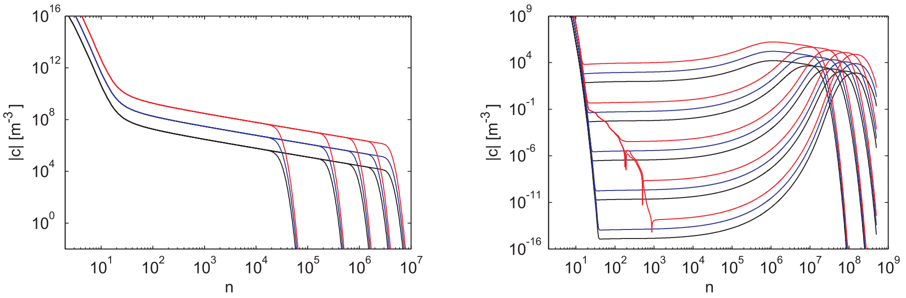
Vice versa, interior relaxation influences the residual at the boundary. Relaxation of the interior equations alters the total amount of monomers in the domain, affecting the residual of the depletion equation. However, it is most affected by the first few equations where droplet number densities are high enough, and in occasions where a lot of large droplets are formed.

As long as the depletion equation can be solved more accurately the interior residual reduction cannot complete. From 5.11 it can be seen that it takes 5 cycles to reduce the depletion equation, so the depletion equation is solved 15 times. This is equal to the amount relaxation sweeps needed to resolve the global constraint, which apply the global constraint with each sweep. with relaxation. Therefore the global constraint is applied much more often per WU with relaxation compared to V-cycles, indicating that the multi-level process is stalled by the global constraint.

The combination of small time steps and a limited number of applications of the global constraint per cycle, results in an inferior performance compared to relaxation. The latter is difficult to change, but a more sophisticated time integration scheme would allow larger time steps for which initial errors presumably would contain more low frequent components for which relaxation is ineffective.

5.3 FAS and the KE

The discrete KE is has a linear operator L. Therefore, the CS and FAS correction to an approximation $\tilde{\mathbf{c}}$ should be equal. However, with the KE some subtleties appear in certain scenarios, so that the solutions obtained by FAS differ from the solution obtained by CS and relaxation. This is investigated here for the nucleation pulse experiment sans depletion.



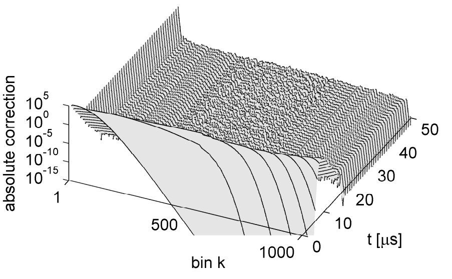
(a) Region I

(b) Region II

Figure 5.12: Comparison of DSDs: relaxation (black), CS (blue, $\mathbf{c} \cdot 10$) and FAS (red, $\mathbf{c} \cdot 100$). Absolute value of $\mathbf{c}$ is taken. Simulations of 2050 bins with zero-th order interpolation.

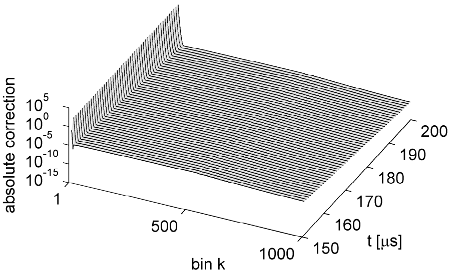
In figure 5.12 the results are displayed for (*a*) relaxation, (*b*) CS and (*c*) FAS. Here the CS and FAS results are multiplied with a factor 10 and 100 respectively for clarity. CS produces the same result as relaxation, but FAS differs in region II of the pulse. Apparently the correction given by the FAS cycle differs from the correction by the CS cycle, which formally should be equal. In figures 5.13 and 5.14 the corrections to the target grid from its next coarser grid are displayed for region I and region II of the experiment, respectively.

In region I the solutions of CS and FAS are similar, as are the corrections.



(a) CS (b) FAS

Figure 5.13: Region I: Absolute corrections of first cycle per time step. CS and FAS corrections are different, resulting in different solutions for small n.



(a) CS (b) FAS

Figure 5.14: Region II: Absolute corrections of first cycle per time step. CS and FAS corrections are different, resulting in different solutions for small n.

With FAS only at the lower boundary it can be seen that some of the boundary correction seems to be transferred into the grid, whereas this is not the case with CS. In region II the differences in the solutions appear for $t > 150\mu s$. From this point on the boundary correction seems to be transferred further into the grid over time. This causes the differences between CS and FAS.

As was explained, for a linear operator, CS and FAS should produce the same correction. The difference however lies the variable that is solved on the coarse grid. CS directly solves the coarse grid error $\mathbf{v}^H$, whereas FAS solves the course grid unknown $I_h^H \tilde{\mathbf{c}}^h + \mathbf{v}^H$. For a known boundary condition the error on the boundary is zero, hence the coarse grid boundary value is zero. In numerics this is much more accurate compared to the boundary value with FAS, which is the boundary condition itself. Therefore digits can be lost with FAS, which for the KE apparently happens when the solution near the boundary is much smaller than the boundary value itself.

CHAPTER 6

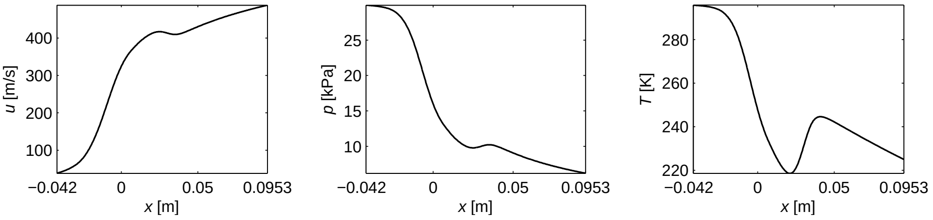
RESULTS: NOZZLE FLOW

In this chapter the condensation in a Laval nozzle is investigated. The flow data from the two-way coupled nozzle flow experiment for single-component condensation performed by Sidin [4] is used for the present one-way coupled experiment. Since the flow in the nozzle can be considered steady-state, the Langrangian form of the advected KE, equation (3.22), is simplified for the steady-state case:

$$\frac{d\hat{c}_n}{dx} = \frac{1}{u}\left[f_{n-1}\hat{c}_{n-1} - (b_n + f_n)\hat{c}_n + b_{n+1}\hat{c}_{n+1}\right] \tag{6.1}$$

This equation is marched in space along the nozzle axis using Euler backward integration. For the trapezoidal method stable space integration was not obtained.

For his experiment Sidin considers a mixture of nitrogen (N_2) and methane (CH_4) as the carrier gas, and heavy water (D_2O) as the condensing component. The mole fraction of methane in the mixture is approximately 4%, and that of D_2O approximately 2.5%, which corresponds with a total D_2O mass fraction of $g_{max} = 0.018$. The material properties of N_2, CH_4 and D_2O are taken from [11]. The velocity, pressure and temperature profiles are displayed in figure 6.1.



(a) Velocity (b) Pressure (c) Temperature

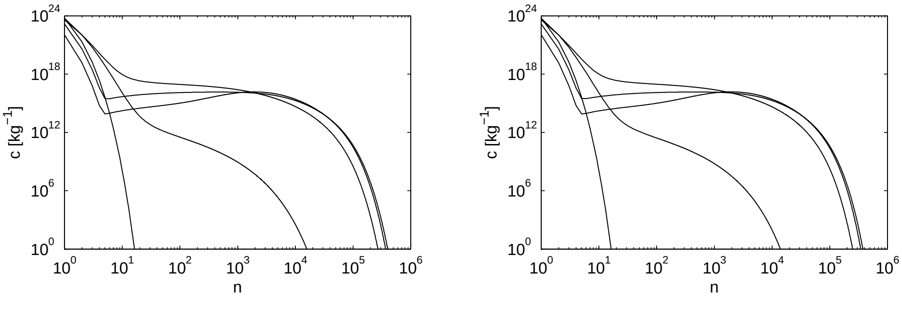
Figure 6.1: Flow variables from condensation experiment in Laval nozzle

Sidin's experiment utilizes the same inconsistency in the monomer density for the computation of the forward rates and the boundary condition as was applied by van Putten (see chapter 5). It was not possible to reproduce this inconsistency for the one-way coupled case. Therefore, the results that are obtained here are different from the results obtained by Sidin. However, the analysis of the multi-level performance is not affected by this.

6.1 Results

For the simulations the largest droplet is taken as $N = 10^6$. The droplet space is approximated using 2050 bins, using zero-th order interpolation. In appendix F a space increment of $\Delta x = 2.8 \cdot 10^{-6}$ m was deemed sufficient for accurate computation for the Euler backward integration. Results are obtained using $\Delta x = 1.0 \cdot 10^{-6}$ m and $\Delta x = 2.7 \cdot 10^{-6}$ m. An equilibrium distribution is used at the nozzle inlet.

The condensation in the nozzle is computed using CS V(2,1)-cycles. The global constraint accounting for depletion (3.33) was applied before each target grid relaxation sweep, which constitutes three applications per V-cycle. With relaxation the supersaturation S, the specific gas constant R and the mixture density ρ are updated after each application of the global constraint. For the V-cycles this is done only once after each cycle.



(a) $\Delta x = 1 \cdot 10^{-5}$ m

(b) $\Delta x = 2.7 \cdot 10^{-6}$ m

Figure 6.2: Droplet size distributions at $x_1 = -0.014$ m, $x_2 = 0.003$ m, $x_3 = 0.011$ m, $x_4 = 0.021$ m and $x_5 = 0.041$ m. CS V(2,1)-cycles.

Figure 6.2 displays the DSD at different stages in the nozzle obtained using different space increments. The solution using the larger space step slightly overestimates the location of the nucleation front, however the difference is negligible. More bins would probably produce more accurate results as was concluded in chapter 5.

The DSD obtained using $\Delta x = 2.7 \cdot 10^{-6}$ m is displayed as a function of x and n in figure 6.3. The contours of the DSD at nozzle locations x_1 to x_5 are

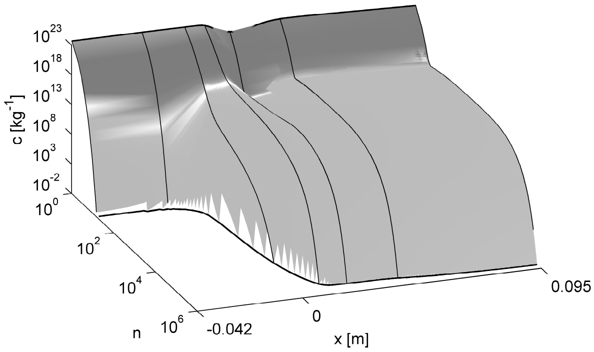
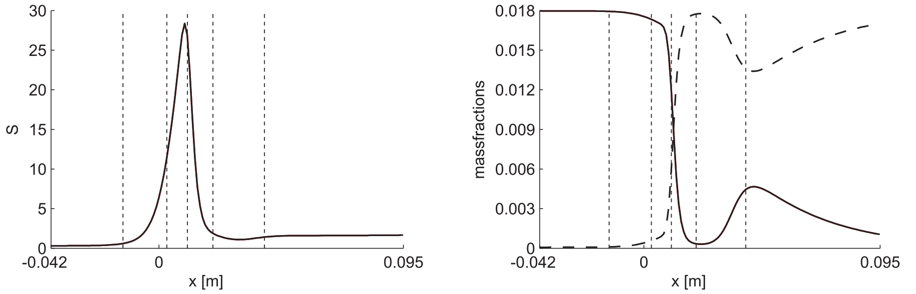


Figure 6.3: Droplet size distribution for $\Delta x = 2.7 \cdot 10^{-6}$ m using CS V(2,1)-cycles. Contours are plotted for $x_1, \ldots, x_5$.



(a) Supersaturation S

(b) Massfractions g_1 (solid) and g (dashed)

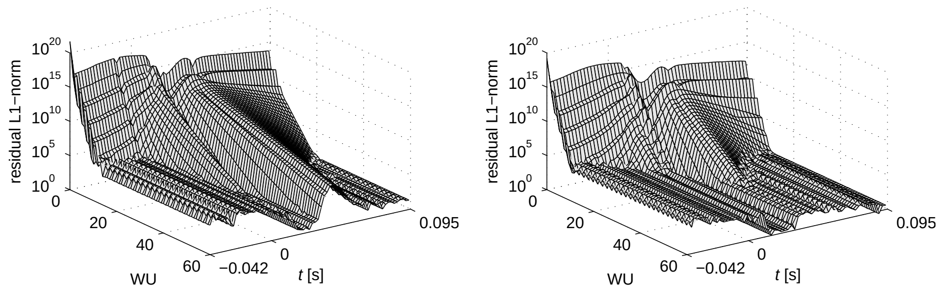
Figure 6.4: Supersaturation and mass fractions of condensable component D_2O in nozzle. Results for $\Delta x = 1 \cdot 10^{-5}$ m and $\Delta x = 2.7 \cdot 10^{-6}$ m are equal.

marked for clarity. Figure 6.4 displays the supersaturation S, and the monomer and liquid massfractions g_1 and g along the nozzle axis.

From the figures it can be seen that condensation starts just before the nozzle throat, around $x_1 = -0.014$ m. The supersaturation achieves its maximum value of 28.6 at $x = 0.01$ m. The vapor is almost entirely depleted at the end of the nozzle. The results for $\Delta x = 1.0 \cdot 10^{-6}$ m and $\Delta x = 2.7 \cdot 10^{-6}$ m are the same since they solely depend on the monomer density $\hat{c}_1$.

6.2 Performance

The performance of the multi-level algorithm is compared to the performance of relaxation. It was already concluded in section 5.1.2 that reduction of the integration step yields initial error components with a higher frequency, making solution by relaxation more effective.



(a) $\Delta x = 1 \cdot 10^{-5}$ m (b) $\Delta x = 2.7 \cdot 10^{-6}$ m

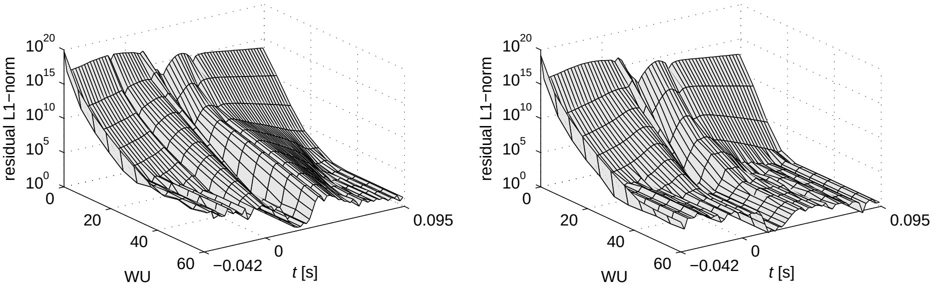
Figure 6.5: Interior residual reduction using relaxation. Initial errors components are of higher frequency for smaller Δx, resulting in a faster residual reduction.

For the nozzle flow experiment this is confirmed by figure 6.5. The observed convergence rates are higher for $\Delta x = 2.7 \cdot 10^{-6}$ m compared to $\Delta x = 1 \cdot 10^{-5}$ m. Furthermore, relaxation becomes less effective when the DSD changes significantly. This is caused by the more complex initial errors which contain more low frequency components.

In figure 6.6 the residual reduction using CS V(2,1)-cycles is displayed. There is no significant performance improvement of the V-cycles compared to relaxations.

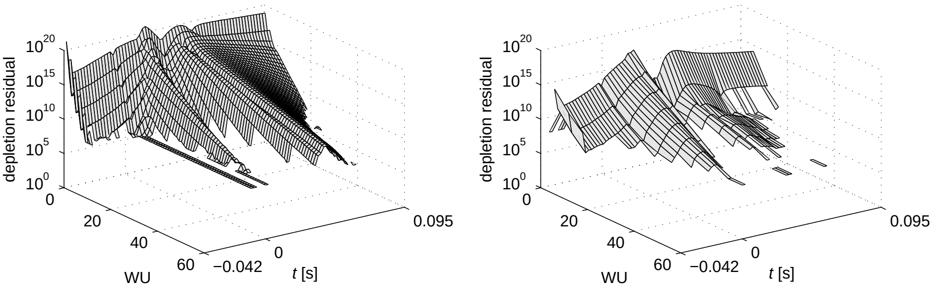
Looking at figure 6.7 it can be seen that the interior residual reduction is stalled by the reduction of the depletion residual. Especially where the DSD changes very little, relaxation outperforms V-cycles because it is able to apply the global constraint more often per WU. Combined with a very small space increment required for accuracy, relaxation is very effective in those parts of the nozzle.

When the DSD changes drastically in the nozzle, where the supersaturation reaches its peak, the performance of V-cycles becomes comparable to relaxation. The initial errors contain significantly more low frequent components, significantly reducing the performance of relaxation but hardly affecting V-cycle performance. If a more accurate space integration scheme is used, larger steps can be made, reducing the number of steps, and increasing the effectiveness of Multigrid.



(a) $\Delta x = 1 \cdot 10^{-5}$ m (b) $\Delta x = 2.7 \cdot 10^{-6}$ m

Figure 6.6: Interior residual reduction using CS V(2,1)-cycles. Initial errors components are of higher frequency for smaller Δx, resulting in a faster residual reduction.



(a) Relaxation (b) V-cycles

Figure 6.7: Depletion residual reduction for $\Delta x = 1 \cdot 10^{-5}$ m.

CHAPTER 7

CONCLUSIONS & RECOMMENDATIONS

The solution of the Kinetic Equation is a computationally expensive task. Especially multi-component condensation simulations based on the KE are not feasible when computing condensing flows. Therefore multi-level techniques, which are proven to enhance the numerical solution process in many physical simulations, are investigated for the application to the KE. As a first step, a working multi-level solver has been developed for one-way coupled single-component condensation experiments. The results and performance were thoroughly analyzed.

Due to their iterative nature, multi-level techniques aim to efficiently solve implicit problems. The KE can much better be solved by implicit methods compared to explicit methods due to very fast propagation of information, i.e. high growth rates, in the droplet space. This puts a relatively restrictive CFL condition on explicit methods requiring very small integration steps. Implicit methods are stable and accurate for much larger steps, compensating the extra work needed to advance from one integration step to the next. Furthermore, monomer depletion introduces an extra implicit condition which can only be resolved by an implicit solver.

However, at first glance the multi-level solver does not improve upon the performance of the Gauss Seidel relaxation that lies at its base. When considering a nucleation pulse without depletion it can be seen that the initial errors at each integration step are only present at the nucleation front. The use of a second order integration scheme limits the allowable integration step for accuracy, which in this case was so small that the advancement of the front is very small. The initial error therefore was only present at a small part of the grid, which is a high frequency error. Multigrid becomes effective when the initial error contains low frequencies, whereas high frequencies can be resolved very effectively by relaxation.

When depletion is included in the simulation, the solution does not merely change at the nucleation front but everywhere on the grid. However, despite of a more complex initial error, containing more low frequency components, relaxation remains the most effective. This is caused by the application of a global constraint which accounts for the depletion of monomers when droplets grow. Interior residual reduction is stalled when the global constraint remains active. Only when the depletion condition is completely satisfied the interior solution can fully converge. The global constraint is applied on the target grid before each relaxation sweep. Compared to relaxation the multi-level solver triggers the global constraint less often per unit of computational work. Therefore the depletion condition is resolved earlier with relaxation compared to the multi-level solver, and the interior residual reduction.

These combined effects resulted in that the nozzle experiment, where environment parameters change at each integration step, did not benefit from the multi-level approach. The first order implicit scheme applied there required too small integration steps to render relaxation ineffective in large parts of the nozzle. In parts of the nozzle with high supersaturation and large droplet growth the multi-level solver was stalled by the depletion condition which required several cycles to be resolved.

Furthermore, a brief look has been given to Full Multigrid (FMG) and F-cycles which aim to improve the performance of a multi-level cycle by providing improved initial guesses. However, these methods succumb to the fact that the solution of the KE itself is badly represented on coarse grids due to the uniform coarsening used in this work. Especially for smaller clusters the KE is badly represented, so a non-uniform coarsening would better suit the solution of the KE.

Finally, it was noticed that numerical treatment of the KE is prone to truncation errors. This was observed with the computation of residual norms and the application of FAS. In the droplet size distribution (DSD), number densities can differ several orders of magnitude. So much even that using 32-bit doubles it was found that the number densities in the DSD cannot be compared to the monomer number density. With FAS this resulted in a loss of digits within the correction, yielding inaccurate results. A brief look has been given to a change of coordinate which scales the solution, but to no avail. If such a transformation can be made it would greatly benefit the numerical process. Another option is use long doubles, preferably on a 64-bit compiler, for which the accuracy is a lot better.

Numerical solution of the KE with Multigrid has potential. However, when the KE is discretized using the integration schemes discussed in this work, relaxation is too effective. The multi-level solver is able to show its potential in some cases, but it mostly loses from relaxation in terms of effectiveness.

It is recommended to investigate discretization using higher order implicit integration schemes which would allow for larger integration steps. Resulting in more low frequent error components, this would seriously impact the performance of relaxation, so that Multigrid may show its real potential. The depletion condition may then also be resolved much quicker relative to the interior error reduction. Also, there are multi-level techniques available which are specifically designed for time-dependent problems. These could be successful in simulations where changes in the DSD are very small over time.

Furthermore, the uniform predefined coarsening used here may not be optimally suited to the KE. Especially when the underlying target grid is non-uniform, as is the case when grouping droplets, a more suitable coarsening could be devised. This could be provided by Algebraic Multigrid (AMG) which constructs the coarse grid by analyzing the coupling between clusters. Stronger couplings are preserved on the coarse grid, whereas weaker couplings can be approximated by coarsening. It recommended to investigate this option.

Finally it is noted that Multigrid is better suited to multi-dimensional problems, where the amount of computational effort is significantly increased with each dimension. The tools developed in this work can serve as a basis for the application of Multigrid to multi-component condensation simulations, for which it is expected that a significant gain in the performance can be achieved.

BIBLIOGRAPHY

[1] Becker, R. and Döring, W. *Ann. Phys.* **416**, 719–752 (1935).

[2] Kashchiev, D. *Nucleation: basic theory with applications*. Butterworth-Heineman, Oxford, (2000).

[3] Holten, V. and van Dongen, M. E. H. *J. Chem. Phys.* **130**(014102) (2009).

[4] Sidin, R. S. R. *Droplet Size Distribution in Condensing Flow*. PhD thesis, University of Twente, (2009).

[5] van Putten, D. S. and Kalikmanov, V. I. *J. Chem. Phys.* **130**(164508) (2009).

[6] Brandt, A. *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*. Number 85. GMD-Studien, (1984).

[7] Venner, C. H. and Lubrecht, A. A. *Multilevel Methods in Lubrication*. Elsevier Tribology Series, (2000).

[8] Briggs, W. L., Henson, V. E., and McCormick, S. F. *A Multigrid Tutorial*. SIAM, (2000).

[9] Courtney, W. G. *J. Chem. Phys.* **35**(2249) (1961).

[10] Wölk, J. and Strey, R. *J. Chem. Phys. B* **105**(47), 11683–11701 (2001).

[11] Heath, C. H., Streletzky, K. A., Wyslouzil, B. E., Wölk, J., and Strey, R. *J. Chem. Phys.* **118**(12), 5465–5473 (2003).

[12] Roy, C. J. *J. Comput. Phys.* **205**(2005), 131–156 (2004).

NOMENCLATURE

Roman

Symbol	**Units**	**Description**
A	[-]	System matrix of the KE
A	[m^2]	Nozzle cross-sectional area
a_1	[m^3]	Effective surface area of monomer
b_n	[s^{-1}]	Backward evaporation rate of n-mer
$\mathbf{c}$	[# m^{-3}]	Volumetric number density vector
$\hat{\mathbf{c}}$	[# kg^{-1}]	Specific number density vector
c_1^S	[# m^{-3}]	Volumetric monomer number density at saturation
c_n^{eq}	[# m^{-3}]	Volumetric equilibrium number density
c_{sat}	[# m^{-3}]	Volumetric monomer number density at saturation
$\mathbf{c}^h$	[-]	Fine grid solution
$\tilde{\mathbf{c}}h$	[-]	Given approximation to $\mathbf{c}^h$
$\bar{\mathbf{c}}^h$	[-]	Corrected approximation to $\mathbf{c}^h$
$\mathbf{c}^H$	[-]	Coarse grid solution
f_n	[s^{-1}]	Forward condensation rate n-mer
$\mathbf{g}^h$	[-]	Fine grid right hand side of discrete KE
g	[-]	Liquid mass fraction
g_1	[-]	Condensable vapor mass fraction
g_{max}	[-]	Total mass fraction of condensable component
ΔG_n	[m^2 kg s^{-2}]	Gibbs free energy of formation
$-g(n)$	[-]	Dimensionless Gibbs free energy of formation
I_h^H	[-]	Restriction operator from fine grid h to coarse grid H
I_H^h	[-]	Interpolation operator from coarse grid H to fine grid h
J_n	[# m^{-3} s^{-1}]	Volumetric condensation flux from cluster n to $n+1$
k_b	[m^2 kg s^{-2} K^{-1}]	Boltzmann constant
L^h	[-]	Discretized operator
L^H	[-]	Coarse grid operator
M	[-]	Number of multi-level cycles required for convergence
m_1	[kg]	Mass of monomer
m_c	[kg]	Mass of carrier gas
m_l	[kg]	Mass of liquid

m_v	[kg]	Mass of condensable vapor
n_{cr}	[-]	Critical cluster size
$\bar{n}_k$	[#}	Average monomer count in bin k for the binned KE
$n_l(k)$	Leftmost droplet in bin k	
$n_r(k)$	Rightmost droplet in bin k	
p	[Pa]	Mixture pressure
p_1	[Pa]	Partial vapor pressure
p_{sat}	[Pa]	Saturated vapor pressure
R	[J kg^{-1} K^{-1}]	Specific gas constant of mixture
R_c	[J kg^{-1} K^{-1}]	Specific gas constant of carries gas
R_v	[J kg^{-1} K^{-1}]	Specific gas constant of condensable vapor
$\mathbf{r}$	[-]	Residual vector
S	[-]	Supersaturation, or saturation ratio
T	[K]	Mixture temperature
t	[s]	Time
Δt	[s]	Discrete time step size
$\mathbf{u}$	[m s^{-1}]	Flow velocity vector
$\mathbf{v}^h$	[-]	Fine grid numerical error
w_k	[-]	Size of bin k for the binned KE
x	[m]	Position along nozzle axis
Δx	[m]	Discrete space step size
y_w	[-]	Vapor molar fraction
u	[m s^{-1}]	Flow velocity in nozzle

Greek

Symbol	**Units**	**Description**
α_n	[-]	Sticking probability of n-mer
γ	[#]	Total monomer count
δ_n	[-]	Correction to a given value of $\tilde{c}_n$
Θ	[-]	Dimensionless surface energy
$\mu(\theta)$	[-]	Error amplification factor of angular component θ for relaxations
$\bar{\mu}$	[-]	Asymptotic smoothing rate
$\tilde{\mu}$	[-]	Measured residual reduction
μ_c	[-]	Total error amplification factor for multi-level cycle
ρ	[kg m^{-3}]	Mixture density
ρ_1	[kg m^{-3}]	Partial density of condensable vapor
ρ_c	[kg m^{-3}]	Partial density of carrier gas
ρ_l	[kg m^{-3}]	Liquid density
ρ_v	[kg m^{-3}]	Partial density of condensable vapor
σ	[N m^{-1}]	Surface tension

ω	[-]	Relaxation factor

APPENDIX A

DIAGONAL DOMINANCE

A very convenient property of the system matrix of an equation set to be solved by numerical methods, is diagonal dominance. A matrix A is defined diagonally dominant if it satisfies the following inequality:

$$|A_{ii}| \geq \sum_{j \neq i} \left|A_{ij}\right| \quad i \in \mathbb{N}, i \geq 1. \tag{A.1}$$

In fact many iterative procedures, such as Gauss Seidel relaxation, rely on this property for the solution to converge. Here it will be investigated if the system matrix of the KE also has this property.

Applying equation (A.1) to matrix A of the KE (chapter 3, equation (3.2.2)) results in the following inequality:

$$|f_n + b_n| \geq |-f_{n-1}| + |-b_{n+1}| \quad \text{for } n = 2, 3, \ldots \tag{A.2}$$

These forward and backward rates are defined as:

$$f_n = C(p,T)\sqrt{\frac{n+1}{n}}(n^{1/3}+1)^2 \quad \text{for } n = 1, 2, \ldots \tag{A.3a}$$

$$b_n = \frac{f_{n-1}}{S}\exp\{\Theta[n^{2/3} - (n-1)^{2/3}]\} \quad \text{for } n = 2, 3, \ldots \tag{A.3b}$$

where C is a function of the pressure and temperature. Since both f_n and b_n are positive functions, (A.2) can be written as:

$$f_n + b_n \geq f_{n-1} + b_{n+1} \quad \text{for } n = 2, 3, \ldots \tag{A.4}$$

The forward rate function f_n is a monotonic increasing function in n which helps to support (A.4). However, it is not clear if this is also true for b_n.

A.1 Supercritical *n*

To get a better insight in inequality (A.4), it is rewritten to the following form:

$$\left[1-\frac{1}{S}\exp\{\Theta[(n+1)^{2/3}-n^{2/3}]\}\right]f_n \geq \left[1-\frac{1}{S}\exp\{\Theta[n^{2/3}-(n-1)^{2/3}]\}\right]f_{n-1}, \qquad (A.5)$$

which can written as:

$$a_n f_n \geq a_{n-1} f_{n-1}, \qquad (A.6)$$

with:

$$a_n = 1-\frac{1}{S}\exp\{\Theta[(n+1)^{2/3}-n^{2/3}]\}. \qquad (A.7)$$

The argument of the exponent in (A.7) is a monotonic decreasing function for $n \geq 1$, which goes to 0 for $n \to \infty$. Ergo, a_n is a monotonic increasing function in n.

However, unlike f_n, a_n is not a strictly positive function. Depending on the values of S and Θ, a_n is smaller than zero for small n. Therefore, (A.6) is only satisfied if:

$$a_n \geq 0. \qquad (A.8)$$

This is a sufficient but not a necessary condition. Still, it is interesting to investigate the scenarios for which this is true.

For $n \gtrsim 5$ equation (A.7) can be approximated as:

$$a_n = 1-\frac{1}{S}\exp\frac{2\Theta}{3(n+1)^{1/3}}. \qquad (A.9)$$

Then, inequality (A.8) always holds for

$$n+1 \geq n_{cr} = \left(\frac{2\Theta}{3\ln S}\right)^3. \qquad (A.10)$$

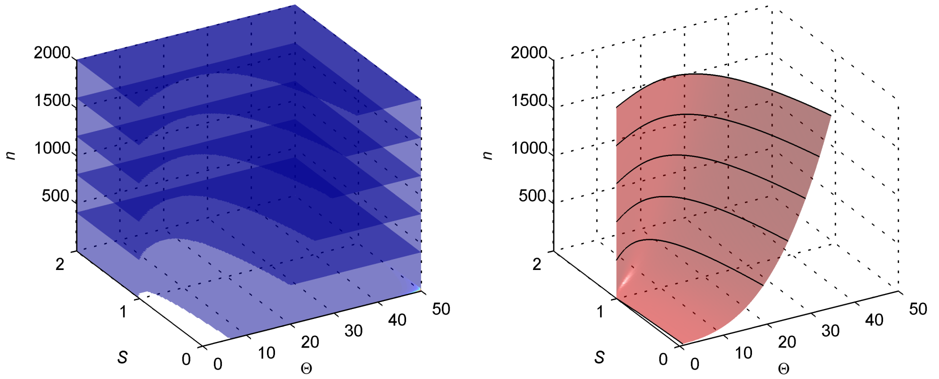
This is a very interesting find, because now it is evident that the system matrix is always diagonally dominant for $n \gtrsim n_{cr}$. Therefore, only for subcritical clusters there is a risk of losing diagonal dominance.

A.2 Subcritical *n*

For subcritical cluster sizes inequality (A.8) does not hold. However, the system can still be diagonally dominant for subcritical clusters if the full inequality is obeyed:

$$U = a_n f_n - a_{n-1} f_{n-1} \geq 0. \qquad (A.11)$$

This is the absolute condition for diagonal dominance.



(a) Volume $U(\Theta, S, n)$ (b) Limiting surface U_{min}

Figure A.1: Left: Blank portions represent combinations of the 3 parameters where the system is not diagonally dominant ($U < 0$). Right: Lower bounding surface for diagonal dominance.

Ideally, this condition should be met for all $n \geq 2$. This would require the product $a_n f_n$ to be a positive, monotonic increasing function, which fully written is:

$$a_n f_n = \sqrt{\frac{n+1}{n}} (n^{1/3} + 1)^2 \left[1 - \frac{1}{S} \exp\{\Theta[(n+1)^{2/3} - n^{2/3}]\} \right]. \tag{A.12}$$

This is a highly nonlinear function for which it is complicated to verify if it is monotonically increasing.

Therefore (A.11) is investigated by evaluation. Figure A.1(a) shows the volume $U = U(\Theta, S, n)$ sliced at different values of n. Portions of the volume where $U < 0$ are blank. Figure A.1(b) shows the limiting surface U_{min}, which is the lower bound for diagonal dominance. In this region, combinations of Θ, S and n result in a non diagonally dominant system.

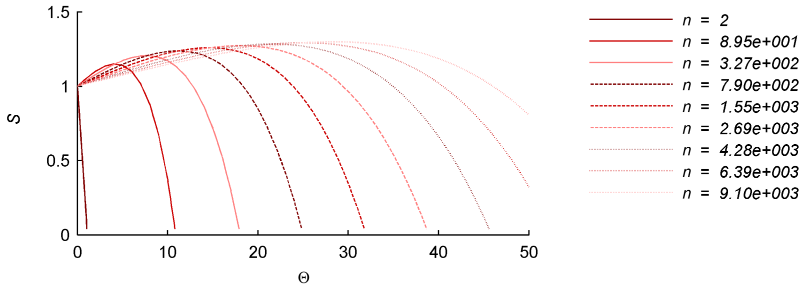
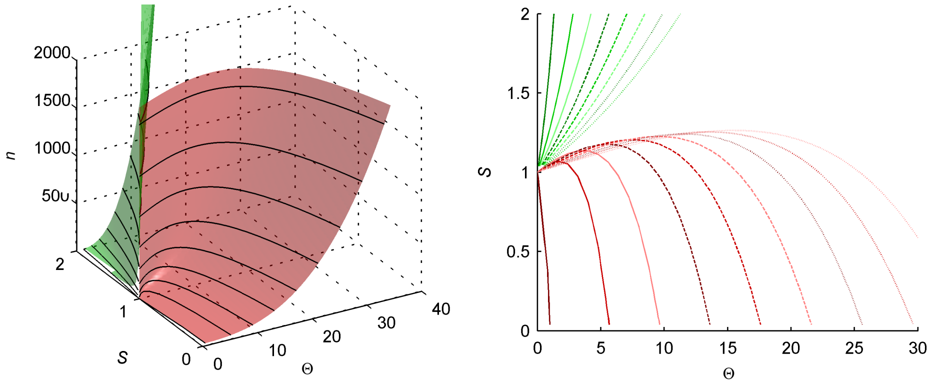


Figure A.2: Curves of U_{min} for different n. Combinations of Θ and S below these curves result in a non diagonally dominant system for specific n

For a constant n the limiting surface U_{min} becomes a curve in the Θ-S plane. This is depicted in figure A.2. From the figure it can be seen that for a certain Θ and S combination, the system may be diagonally dominant for n. For higher values of n diagonal dominance is lost. The system is diagonally dominant for the lower boundary $n = 2$ under almost all conditions, but never completely.



(a) Limiting surfaces $a_n \geq 0$ (green) and $U \geq 0$ (red)

(b) Limiting curves in Θ-S plane

Figure A.3: Space of diagonal dominance. Green is the conservative condition $a_n \geq 0$. Red is the absolute condition $U \geq 0$. Diagonal dominance is not achieved on this side of the red surface.

Next the absolute condition for diagonal dominance (A.11), is compared to the sufficient condition $a_n \geq 0$. In figure A.3 this is plotted for all 3 parameters, similar to surface U_{min}. Indeed, $a_n \geq 0$ is a sufficient condition, as it leaves room for other parameter combinations compared to the necessary condition $U \geq 0$. This is the space between the green and red surface.

Both surfaces intersect at $(\Theta, S, n) = (0, 1, n)$, which indicates that this is a key situation. In reality typical values of Θ are $O(10)$, which gives a diagonally dominant system for all n for $S \gtrsim 1.3$.

APPENDIX B

STABILITY ANALYSIS

The trapezoidal method, being an implicit method, will always have favorable stability to explicit methods. However, where fully implicit methods are unconditionally stable, this is not the case for this semi-implicit method. To determine the conditions for stable time integration, a von Neumann analysis is performed.

For this purpose the spatial part of the solution is decomposed in its Fourier components:

$$c(n,t) = \hat{c}(t)e^{\iota\theta n} \tag{B.1}$$

Substituting this in equation (**??**) of chapter 3 gives:

$$\begin{aligned}\hat{c}^{m+1}e^{\iota\theta n}\Big\{1 - \frac{\Delta t}{2}\Big[f_{n-1}^{m+1}e^{-\iota\theta} - \big(f_n^{m+1} + b_n^{m+1}\big) + b_{n+1}^{m+1}e^{\iota\theta}\Big]\Big\} = \\ \hat{c}^{m}e^{\iota\theta n}\Big\{1 + \frac{\Delta t}{2}\Big[f_{n-1}^{m}e^{-\iota\theta} - \big(f_n^{m} + b_n^{m}\big) + b_{n+1}^{m}e^{\iota\theta}\Big]\Big\}\end{aligned} \tag{B.2}$$

The amplification factor is defined as $A = \frac{\hat{c}^{m+1}}{\hat{c}^m}$. For stability the absolute amplification factor should be smaller than 1:

$$\begin{aligned}|A| &= \left|\frac{1 + \frac{\Delta t}{2}\Big[f_{n-1}^{m}e^{-\iota\theta} - \big(f_n^{m} + b_n^{m}\big) + b_{n+1}^{m}e^{\iota\theta}\Big]}{1 - \frac{\Delta t}{2}\Big[f_{n-1}^{m+1}e^{-\iota\theta} - \big(f_n^{m+1} + b_n^{m+1}\big) + b_{n+1}^{m+1}e^{\iota\theta}\Big]}\right| \\ &= \left|\frac{1 + \frac{\Delta t}{2}z^m}{1 - \frac{\Delta t}{2}z^{m+1}}\right| \leq 1\end{aligned} \tag{B.3}$$

So the trapezoidal method is only stable if $\mathfrak{R}(z^m)$ and $\mathfrak{R}(z^{m+1})$ are smaller than 0. Both cases are analogous so they will be considered simultaneously with their superscripts dropped:

$$\begin{aligned}\mathfrak{R}(z) &= \mathfrak{R}\Big[f_{n-1}e^{-\iota\theta} - \big(f_n + b_n\big) + b_{n+1}e^{\iota\theta}\Big] \\ &= \big(f_{n-1} + b_{n+1}\big)\cos\theta - \big(f_n + b_n\big) \leq 0\end{aligned} \tag{B.4}$$

Since $|\cos\theta| \leq 1$ the inequality becomes:

$$f_n + b_n \geq f_{n-1} + b_{n+1} \tag{B.5}$$

If the above requirement is met for the current and the future time step, time integration is stable. It was already shown in appendix A that this is the case for the majority of the situations.

APPENDIX C

INTERPOLATION ERROR

When bins are used, the calculated state variable is the mean number density of all the droplets inside the bin. A prerequisite for accuracy is an accurate description of the inter-bin fluxes. This requires the number densities of the edge droplets to be accurate. Since these are unknown, they must be obtained from the mean number densities. Figure C.1 illustrates how this can be done by interpolation from the mean bin values.

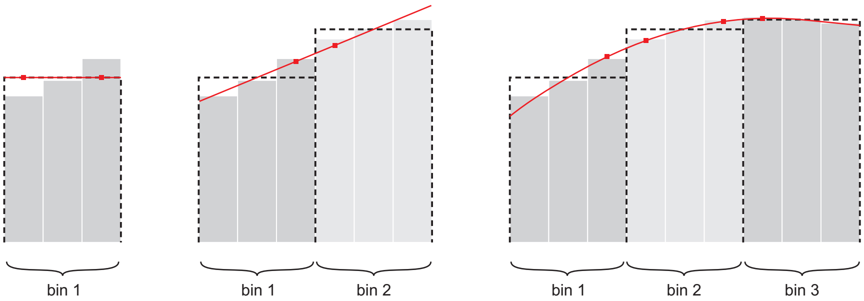


Figure C.1: Derivation of the bin-edge values from the mean bin values using zero-th, 1st and 2nd order interpolation.

The simplest is the zero-th order interpolation. In this case the edge values are determined from the mean bin value of the same bin. With a 1st order interpolation, a 1st order polynomial is fitted through the mean number densities of 2 neighboring bins. The number densities of the neighboring droplets in between the bins are then calculated. The 2nd order interpolation requires 3 bins to fit a 2nd order polynomial. From this polynomial the number densities of the droplets between the bins can then be determined.

The accuracy of the calculated number densities at the edges of bins can be increased when smaller bins and higher orders of interpolation are used. The first is obvious since in the limit of a bin size of 1, the mean number density in the bin *is* the number density of the single droplet it contains. The latter is true since higher order interpolation polynomials are able to represent a more complex DSD. In the subsequent sections the error made by interpolation is estimated for a zero-th and 1^{st} order interpolation.

C.1 Zero-th order interpolation

Assuming all number densities in bin k of size w_k are known. Then a polynomial of order $w_k - 1$ can be fitted through these data points, as depicted in figure C.2. Let Δn be a continuous variable along the bin, with the origin at the first droplet.

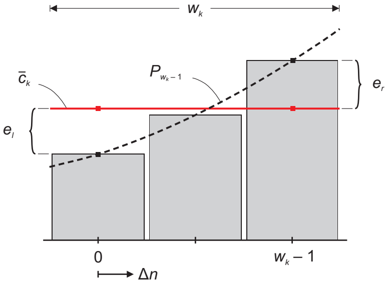


Figure C.2: Error estimation through comparison with a fitted polynomial of order $w_k - 1$.

The droplet number density as a function of Δn can be obtained from the polynomial P_{w_k-1} according to:

$$\tilde{c}(\Delta n) = \sum_{j=0}^{w_k-1} a_j (\Delta n)^j \tag{C.1}$$

The mean number density in the bin is then calculated by:

$$\bar{c}_k = \frac{1}{w_k} \sum_{l=0}^{w_k-1} \tilde{c}(l) \tag{C.2}$$

From these relations the error can be defined as:

$$e(\Delta n) = |\bar{c}_k - \tilde{c}(\Delta n)| \tag{C.3}$$

Substituting (C.1) and (C.2) in (C.3), and evaluating at 0 and $w_k - 1$, gives the error on the left and on the right:

$$e_l = \left| \sum_{j=0}^{w_k-1} \frac{a_j}{w_k} \left[(1-w_k)0^j + \sum_{l=1}^{w_k-1} l^j \right] \right| \tag{C.4a}$$

$$e_r = \left| \sum_{j=0}^{w_k-1} \frac{a_j}{w_k} \left[\sum_{l=0}^{w_k-2} l^j + (1-w_k)(w_k-1)^j \right] \right| \tag{C.4b}$$

For a bin size of $w_k = 1$ the errors are zero, which is as expected since for a unit bin size the mean number density is equal to the droplet number density. For larger bins the error takes on the following form:

$$e = 0 \cdot a_0 + c_1 a_1 + \ldots + c_{w_k-1} a_{w_k-1} \tag{C.5}$$

In essence, the error contains the polynomial coefficients that are of higher order than the zero-th interpolation. Furthermore, the higher order polynomial coefficients contribute more to the error than the lower order coefficients. For instance, e_r for a 3 and 4 droplet bin is:

$$w_k = 3: \quad e_r = |-a_1 - 2\tfrac{1}{3}a_2| \tag{C.6a}$$

$$w_k = 4: \quad e_r = |-1\tfrac{1}{2}a_1 - 5\tfrac{1}{2}a_2 - 18a_3| \tag{C.6b}$$

In order to keep the error small, the higher order polynomial coefficients should be small, as they contribute the most. In other words, larger bins will only have small interpolation errors if the DSD in the bin is smooth. Smooth distributions can be represented by polynomials of low order, resulting in small high order coefficients.

C.2 First order interpolation

With 1^{st} order interpolation 2 bins are used to estimate the number densities of the neighboring droplets in between the bins, as depicted in figure C.3. The bins k and $k+1$ contain w_k and w_{k+1} droplets respectively.

Analogous to the zero-th order interpolation, a polynomial is fitted through the individual droplet number densities, which are assumed to be known. This is a P_{W-1} polynomial:

$$\tilde{c}(\Delta n) = \sum_{j=0}^{W-1} a_j (\Delta n)^j \tag{C.7}$$

where $W = w_k + w_{k+1}$. Also the mean number densities in the bins are:

$$\bar{c}_k = \frac{1}{w_k} \sum_{l=0}^{w_k-1} \tilde{c}(l) \tag{C.8a}$$

$$\bar{c}_{k+1} = \frac{1}{w_{k+1}} \sum_{l=w_k}^{W-1} \tilde{c}(l) \tag{C.8b}$$

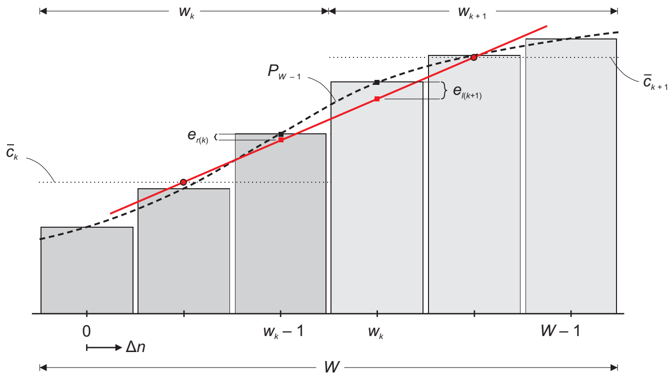


Figure C.3: Error estimation through comparison with a fitted polynomial of order $W-1$.

Now a 1st order polynomial is fitted through the mean number densities, from which the number densities of the inner edge droplets are determined:

$$c_{r(k)} = \frac{\bar{c}_k(w_{k+1}+1)+\bar{c}_{k+1}(w_k-1)}{w_k+w_{k+1}} \tag{C.9a}$$

$$c_{l(k+1)} = \frac{\bar{c}_k(w_{k+1}-1)+\bar{c}_{k+1}(w_k+1)}{w_k+w_{k+1}} \tag{C.9b}$$

Substitution of equations (C.7), (C.8) and (C.9) in the error definition (C.3) gives the errors at the internal edges:

$$e_{r(k)} = \sum_{j=0}^{W-1} a_j \left[\frac{w_{k+1}+1}{w_k W}\sum_{l=0}^{w_k-2} l^j + \left(\frac{w_{k+1}+1}{w_k W}-1\right)(w_k-1)^j + \frac{w_k-1}{w_{k+1}W}\sum_{l=w_k}^{W-1} l^j\right] \tag{C.10a}$$

$$e_{l(k+1)} = \sum_{j=0}^{W-1} a_j \left[\frac{w_{k+1}-1}{w_k W}\sum_{l=0}^{w_k-1} l^j + \left(\frac{w_k+1}{w_{k+1}W}-1\right)w_k^j + \frac{w_k+1}{w_{k+1}W}\sum_{l=w_k+1}^{W-1} l^j\right] \tag{C.10b}$$

Again, when the bin sizes are 1, the errors are zero. Larger bins produce the following error:

$$e = 0\cdot a_0 + 0\cdot a_1 + c_2 a_2 + \ldots + c_{w_k-1}a_{w_k-1} \tag{C.11}$$

Similar to the zero-th order interpolation, the error consists only of polynomial coefficients that are of higher order than the interpolation. Higher orders of interpolation will most likely produce the same results, eliminating even more terms from the error. Also, the higher order polynomial coefficients contribute more to the error than the lower order coefficients, as with the zero-th order case. So it is again concluded that for the error to remain small, the DSD in the bin should be smooth.

APPENDIX D

LOCAL MODE ANALYSIS

D.1 Target grid

To predict the error reduction by relaxation, one can use a Local Mode analysis. First Gauss Seidel relaxation applied to the KE on the target grid, i.e. the original grid is considered. For this purpose the experimental settings for the nucleation pulse as used by van Putten [5] were chosen.

For the analysis the errors before and after relaxation are defined as:

$$\tilde{e}_n = c_n - \tilde{c}_n \tag{D.1a}$$

$$\bar{e}_n = c_n - \bar{c}_n \tag{D.1b}$$

where c_n is the discrete solution of the problem. Relaxation was defined as:

$$\bar{c}_n = \tilde{c}_n + \omega\delta_n \tag{D.2}$$

and substitution of the error definition eventually yields:

$$\bar{e}_n = \tilde{e}_n(1-\omega) + \frac{\omega\Delta t}{2+(b_n+f_n)\Delta t}\left(f_{n-1}\bar{e}_{n-1} + b_{n+1}\tilde{e}_{n+1}\right) \tag{D.3}$$

The relaxation is a local process, i.e. points several meshsizes away affect each other exponentially little. Therefore, in order to analyze relaxation behavior, the error is decomposed into a Fourier series:

$$\tilde{e}_n = \sum_{0<|\theta|<\pi} \tilde{A}(\theta)e^{\iota\theta n} \tag{D.4a}$$

$$\bar{e}_n = \sum_{0<|\theta|<\pi} \bar{A}(\theta)e^{\iota\theta n} \tag{D.4b}$$

where θ is the angular frequency, and A the amplitude of the component with frequency θ. Only components of $|\theta| \leq \pi$ are considered, which are the components that the grid can represent uniquely. Due to the local nature of the error the effect of the boundaries may be disregarded, however, their effect is often felt only a few meshsizes into the domain. If this is not the case a separate analysis can be done for relaxation near the boundary, which will not be treated here.

Assuming that each error component maps onto itself, one can suffice with the analysis of a single component. Substitution of (D.4) in (D.3) then yields:

$$\bar{A}(\theta) = \tilde{A}(\theta)(1-\omega) + \frac{\omega}{1+(b_n+f_n)\frac{\Delta t}{2}}\left[f_{n-1}\frac{\Delta t}{2}\bar{A}(\theta)e^{-\iota\theta} + b_{n+1}\frac{\Delta t}{2}\tilde{A}(\theta)e^{\iota\theta}\right] \tag{D.5}$$

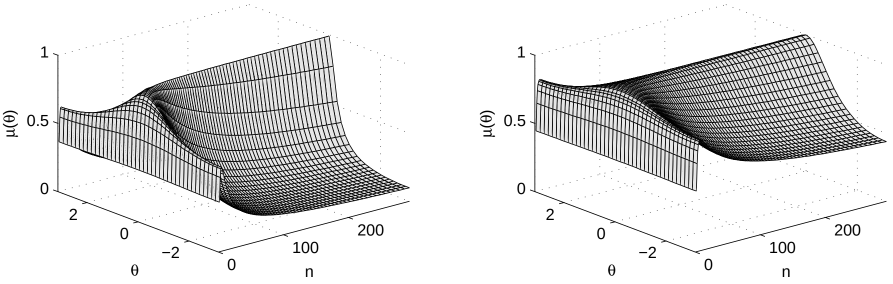
which for $\omega = 1$ gives:

$$\frac{\bar{A}(\theta)}{\tilde{A}(\theta)} = \frac{b_{n+1}\frac{\Delta t}{2}e^{\iota\theta}}{1+(b_n+f_n)\frac{\Delta t}{2} - f_{n-1}\frac{\Delta t}{2}e^{-\iota\theta}} \tag{D.6}$$

The error amplification factor defined as $\mu(\theta) = |\bar{A}(\theta)/\tilde{A}(\theta)|$ then becomes:

$$\mu(\theta) = \frac{b_{n+1}\frac{\Delta t}{2}}{\sqrt{\left[1+(b_n+f_n)\frac{\Delta t}{2}\right]^2 + \left[f_{n-1}\frac{\Delta t}{2}\right]^2 - \left[1+(b_n+f_n)\frac{\Delta t}{2}\right]f_{n-1}\Delta t\cos\theta}} \tag{D.7}$$

Since the forward and backward rates depend on n, the amplification factor is different for each equation. Also, the backward rates b_n depend on the environment variables S and Θ. Thus for each environment and each equation the error reduction will be different.



(a) $S = 14.34 \quad \Theta = 11.40$ (b) $S = 1.67 \quad \Theta = 9.51$

Figure D.1: Local mode analysis for expansion stage (left) and compression stage (right). Error amplification factors are a function of n. Amplification factors stabilize for increasing n.

Figure D.1 displays the error amplification factor as a function of the equation number n for $-\pi \le \theta \le \pi$ during a nucleation pulse. Left the expansion stage, and right the compression stage. The figure shows that high frequency components are reduced more efficiently than low frequency components.

Low frequency components, and component $\theta = 0$ in particular, are reduced the least. For $\theta = 0$ and $n \to \infty$ the amplification factor approaches unity in both cases. However, since the problem has Dirichlet boundary conditions, this component cannot be represented on the grid.

High frequency components are reduced the least around $n \approx 5$ for the expansion stage and $n \approx 8$ for the compression stage. Around these n amplification factors show minima of 0.61 and 0.81 respectively. Certainly in the latter case this is poor. For $n \to \infty$ the minimal amplification factors stabilize to 0.03 and 0.23 for the expansion and compression stage respectively.

D.2 Coarse grids

Similar to the relaxation on the target grid, a Local Mode analysis can be performed for the coarse grid operator L^H. This operator operates on three points in the number density vector $\mathbf{c}^H$, and can therefore be separated into three parts:

$$L_N^H\left(c_{N-1}^H, c_N^H, c_{N+1}^H\right) = L_N^H(1,0,0)c_{N-1}^H + L_N^H(0,1,0)c_N^H + L_N^H(0,0,1)c_{N+1}^H \tag{D.8}$$

Gauss Seidel relaxation can then be written as:

$$\bar{c}_N^H = \tilde{c}_N^H + \frac{\omega}{L_N^H(0,1,0)}\left\{g_N^H - \left[L_N^H(1,0,0)\bar{c}_{N-1}^H + L_N^H(0,1,0)\tilde{c}_N^H + L_N^H(0,0,1)\tilde{c}_{N+1}^H\right]\right\} \tag{D.9}$$

Defining the errors before and after relaxation as:

$$\tilde{e}_N^H = c_N - \tilde{c}_N^H \tag{D.10a}$$

$$\bar{e}_N^H = c_N - \bar{c}_N^H \tag{D.10b}$$

where c_N is the exact solution, and decomposing them in their Fourier components:

$$\tilde{e}_N^H = \sum_{0<|\theta|<\pi} \tilde{A}^H(\theta)e^{\iota\theta N} \tag{D.11a}$$

$$\bar{e}_N^H = \sum_{0<|\theta|<\pi} \bar{A}^H(\theta)e^{\iota\theta N} \tag{D.11b}$$

eventually yields:

$$\bar{A}^H = \tilde{A}^H(1-\omega) - \frac{\omega}{L_N^H(0,1,0)}\left[L_N^H(1,0,0)\bar{A}^H e^{-\iota\theta} + L_N^H(0,0,1)\tilde{A}^H e^{\iota\theta}\right] \tag{D.12}$$

For $\omega = 1$:

$$\frac{\bar{A}^H}{\tilde{A}^H} = -\frac{L_N^H(0,0,1)e^{\iota\theta}}{L_N^H(0,1,0) + L_N^H(1,0,0)e^{-\iota\theta}} \tag{D.13}$$

and:

$$\mu(\theta) = \sqrt{\frac{\left[L_N^H(0,0,1)\right]^2}{\left[L_N^H(0,1,0)\right]^2 + \left[L_N^H(1,0,0)\right]^2 + 2L_N^H(0,1,0)L_N^H(1,0,0)\cos\theta}} \tag{D.14}$$

D.2.1 Vertex centered coarsening

The results for vertex centered coarsening are presented in figure D.2 for the expansion stage of the nucleation pulse experiment. The Local Mode analysis for the target grid h is the same as obtained earlier and is stable. The error amplification factors for low frequency components and $n \to \infty$ approach unity, since equation (D.14) for $\theta = 0$ yields:

$$\mu(\theta) = \left|\frac{L_n^h(0,0,1)}{L_n^h(1,0,0) + L_n^h(0,1,0)}\right| = \left|\frac{b_{n+1}}{f_{n-1} - (b_n + f_n)}\right| \tag{D.15}$$

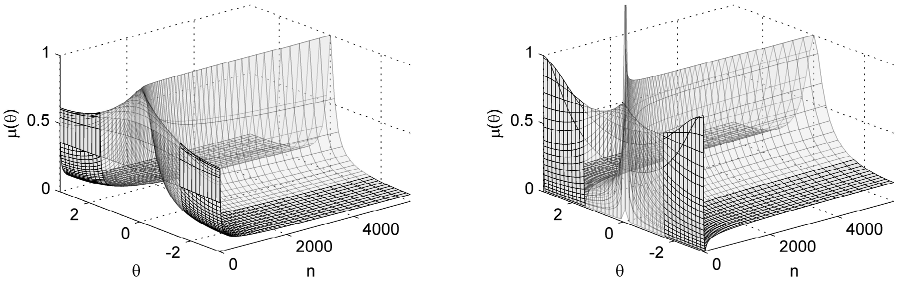
Gauss Seidel relaxations proves to be unstable at coarser grids. At grid level $2h$ high frequent components are amplified rather than reduced, for small n. Also, there is an amplification zone around $\theta = 0$ for small n on all coarse grids.

It should be noted that decreasing the time step Δt has a positive impact on the performance of Gauss Seidel relaxation. Overall error amplification factors are lower for smaller time steps. This supports the earlier finding that a smaller Δt enhances diagonal dominance of the operator, due to the use of an implicit time integration method. This improves the performance of Gauss Seidel relaxation.

D.2.2 Cell centered coarsening

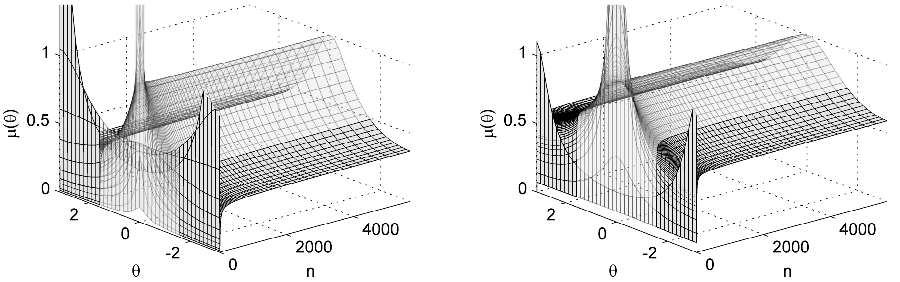
The results for cell centered coarsening are presented in figure D.3. In contrast to the coarse grid operator obtained by vertex centered coarsening, the coarse grid operators obtained in this way are similar to the target grid operator.

As expected, because the coarse and fine grid operators are similar, the error amplification factors are also similar on different grids. This is a desirable property of the coarse grid operator, since Gauss Seidel relaxation remains stable for coarse grids. Cell centered coarsening is therefore used as the basis for the Multigrid algorithm.



(a) Target grid h

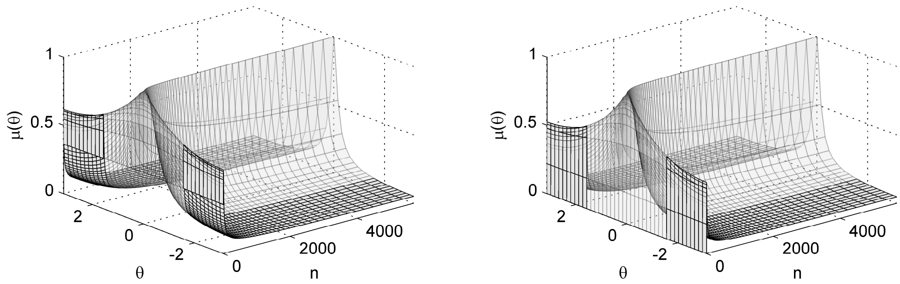
(b) Coarse grid $2h$



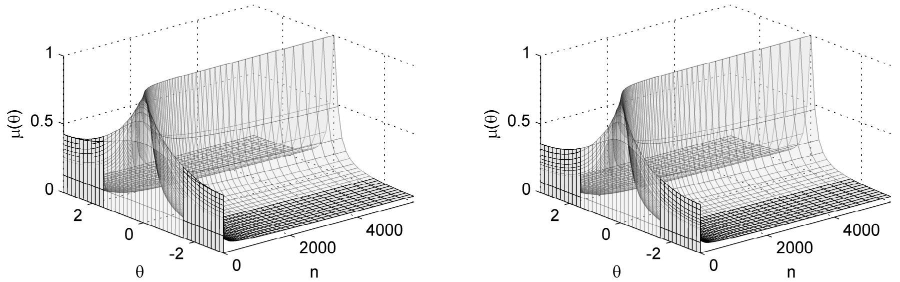
(c) Coarse grid $4h$

(d) Coarse grid $8h$

Figure D.2: Error amplification factors $\mu(\theta, n)$ for different grid levels using vertex centered coarsening. Low frequency components are transparent. Expansion stage of nucleation pulse: $\Delta t = 10^{-7}$ s, $S = 14.34$, $\Theta = 11.40$.



(a) Target grid h (b) Coarse grid $2h$



(c) Coarse grid $4h$ (d) Coarse grid $8h$

Figure D.3: Error amplification factors $\mu(\theta, n)$ for different grid levels using cell centered coarsening. Low frequency components are transparent. Expansion stage of nucleation pulse: $\Delta t = 10^{-7}$ s, $S = 14.34$, $\Theta = 11.40$.

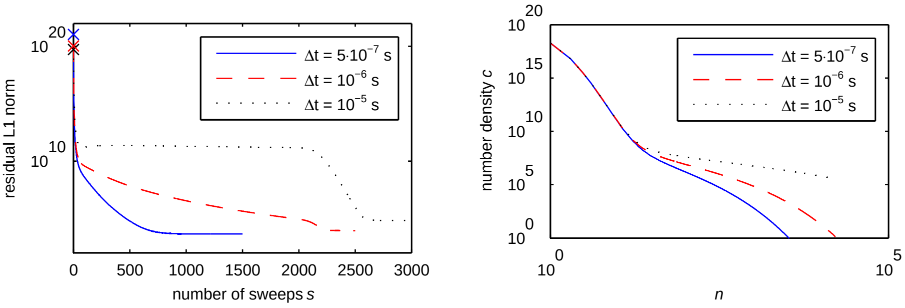
APPENDIX E

OPERATOR ASYMMETRY

In chapter 3 it was established that the operator is diagonally dominant, which is essential for Gauss Seidel relaxation to converge. The order of relaxation was assumed to be lexicographic from the first to the last equation. The effect of this choice is analyzed below for the full KE.

E.1 Lexicographic ordering

First, relaxation performance is tested with the downstream relaxation order. On a grid with $n = 16386$ relaxation is continued up to machine accuracy for different sizes of time steps. The results for the first time step are displayed in figure E.1



(a) residuals for computation 1st time step (b) solution after 1 time step

Figure E.1: Residuals and solutions after 1 time step. The solution converges rapidly initially, but then slows down to a steady decay for large time steps.

Looking at the residuals, a rapid initial decay is observed, which then slowly stalls. After a while the residual suddenly further diminishes to machine accuracy. This is clearly visible for the large Δt, but not so clear for the smaller time step.

To find the cause of this behavior, the local residuals are examined. Figure E.2 shows the local residuals after 1, 750, 1500, 2250 and 3000 sweeps. The rapid initial decay can be contributed to the rapid reduction of the residuals of the first equation, which has the largest residual. However, with the relaxation of a single equation, part of the residual seems to be transferred downstream to the next equation. This transferred residual than travels across the entire grid until it eventually vanishes at the end. This explains the more or less steady decay after the rapid initial convergence, and the sudden drop for the large time steps. For large time steps the maximum of the residual 'lump' is larger than the remaining residual of the first equation, and contributes more to the residual L1 norm.

E.2 Downstream residual transfer

Consider sweep s at some point in the relaxation process. The residual in the equation currently being relaxed is:

$$r_n^s = g_n - \left(L_n^l \bar{c}_{n-1}^s + L_n^c \tilde{c}_n^s + L_{n+1}^r \tilde{c}_{n+1}^s\right), \tag{E.1}$$

where the left, center and right operators in point n are given by:

$$L_n^l = -f_{n-1}\frac{\Delta t}{2} \tag{E.2a}$$

$$L_n^c = 1 + (b_n + f_n)\frac{\Delta t}{2} \tag{E.2b}$$

$$L_n^r = -b_{n+1}\frac{\Delta t}{2}. \tag{E.2c}$$

The previously relaxed variable $\bar{c}_{n-1}^s$ can be expressed as:

$$\bar{c}_{n-1}^s = \tilde{c}_{n-1}^s + \frac{r_{n-1}^s}{L_{n-1}^c} = \tilde{c}_{n-1}^{s-1} + \frac{r_{n-1}^s}{L_{n-1}^c}. \tag{E.3}$$

Also, the variables to be relaxed can be written as:

$$\tilde{c}_n^s = \bar{c}_n^{s-1} = \tilde{c}_n^{s-1} + \frac{r_n^{s-1}}{L_n^c} \tag{E.4a}$$

$$\tilde{c}_{n+1}^s = \bar{c}_{n+1}^{s-1} = \tilde{c}_{n+1}^{s-1} + \frac{r_{n+1}^{s-1}}{L_{n+1}^c}. \tag{E.4b}$$

Substituting (E.3) and (E.4) in (E.1) yields:

$$\begin{aligned} r_n^s =& g_n - \left(L_n^l \bar{c}_{n-1}^{s-1} + L_n^c \tilde{c}_n^{s-1} + L_{n+1}^r \tilde{c}_{n+1}^{s-1}\right) \\ & - \Big(\frac{L_n^l}{L_{n-1}^c} r_{n-1}^s + \frac{L_n^c}{L_n^c} r_n^{s-1} + \frac{L_{n+1}^r}{L_{n+1}^c} r_{n+1}^{s-1}\Big). \end{aligned} \tag{E.5}$$

Recognizing the first part as the residual in equation n from the previous sweep r_n^{s-1}, equation (E.5) reduces to:

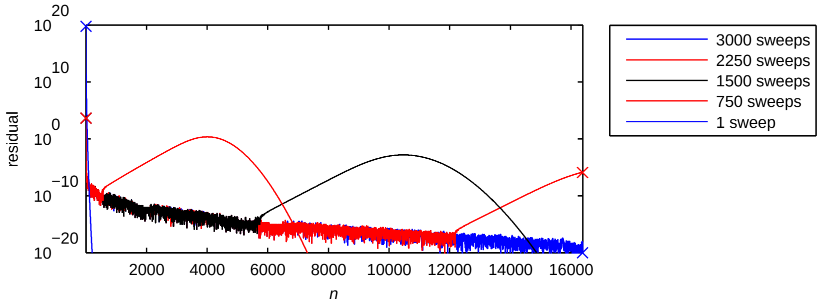
$$r_n^s = -\frac{L_n^l}{L_{n-1}^c} r_{n-1}^s - \frac{L_{n+1}^r}{L_{n+1}^c} r_{n+1}^{s-1}. \tag{E.6}$$

Fully expressing the left, center and right operators then finally yields:

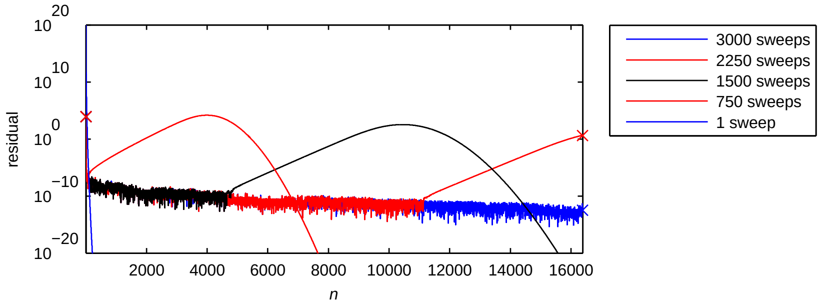
$$r_n^s = \frac{f_{n-1}\frac{\Delta t}{2}}{1+(b_{n-1}+f_{n-1})\frac{\Delta t}{2}} r_{n-1}^s + \frac{b_{n+1}\frac{\Delta t}{2}}{1+(b_{n+1}+f_{n+1})\frac{\Delta t}{2}} r_{n+1}^{s-1}. \tag{E.7}$$

This is an important relation, expressing the residual in the current equation as a function of the residuals in the neighboring points. In case of a symmetric operator, the residual in the current equation would receive equal contributions from the neighboring equations. For an asymmetric operator this is not the case. Then the residual is transferred in a specific direction.

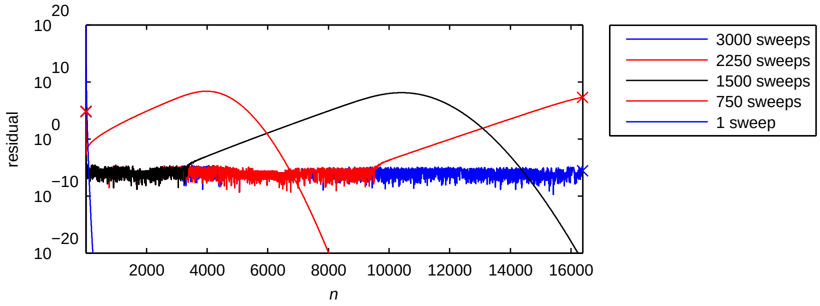
For large n, with the KE the contribution of the left neighbor is larger than that of the right neighbor. However, there is a point at which this behavior is reversed. This happens at the critical cluster size n_{cr} for which there is a minimum in the Gibbs potential. This means that residuals tend move upstream below n_{cr}, and downstream above n_{cr}.



(a) local residuals, $\Delta t = 5 \cdot 10^{-7}$ s



(b) local residuals, $\Delta t = 10^{-6}$ s



(c) local residuals, $\Delta t = 10^{-5}$ s

Figure E.2: Local residuals during computation of first time step. Relaxation of an equation transfers part of the residual to the next equation, creating a residual 'lump' which travels across the grid from left to right.

APPENDIX F

CONVERGENCE ANALYSIS

In this appendix the time-accurate time step size is determined for the nucleation pulse experiment and for the nozzle flow experiment. For discretization methods, a sequence f_n is said to converge to ξ with order p if there exists a constant C such that:

$$|f_n - \xi| < Cn^{-p} \quad \text{for all} \quad n \tag{F.1}$$

where n is the amount of gridpoints. The order p, the exact solution ξ and the constant C can be determined from three numerical solutions on different mesh sizes, as is explained by Roy [12].

However, this is only true is all three solutions fall within the asymptotic range of convergence. Therefore four simulations are performed for both experiments, so that the convergence results can be verified.

To measure time-accuracy, the nucleation front where new droplets are formed is examined. The location of the front is derived from a first order interpolation of the data, by determining the largest formed droplet n_f from $c_{n_f} = 1$.

F.1 Nucleation pulse

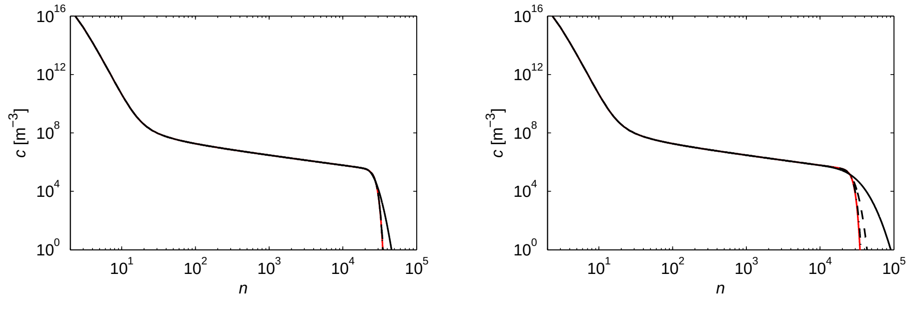
The nucleation pulse experiment is as investigated by van Putten [5]. The Multigrid solver for the full KE is considered, with the largest droplet in the domain $N = 131074$, which equates to 18 levels with 3 bins on the coarsest level in the solver.

Accuracy tests are performed for implicit time integration using (a) 2^{nd} order the trapezoidal method and (b) the 1^{st} order backward Euler method. A pulse of 10 μs duration is considered, using time steps of $\Delta t = 10^{-7}$, 10^{-8}, 10^{-9} and 10^{-10} s.

For this experiment also an explicit RK4 solver was considered. A time step size of $5 \cdot 10^{-11}$ s was required to obtain a stable solution. This is a severe CFL condition for the RK4 solver and it should be noted that it becomes even more restraining for larger droplets were the droplet growth rate increases.

F.1.1 Convergence tests

In figure F.1 the droplet size distribution after 10 μs is displayed for different time step sizes. The solution from $\Delta t = 10^{-10}$ is not shown but here, but it almost perfectly coincides with the solution from $\Delta t = 10^{-9}$. The solution from the RK4 solver is plotted in red for comparison.



(a) Trapezoidal method (b) Euler backward

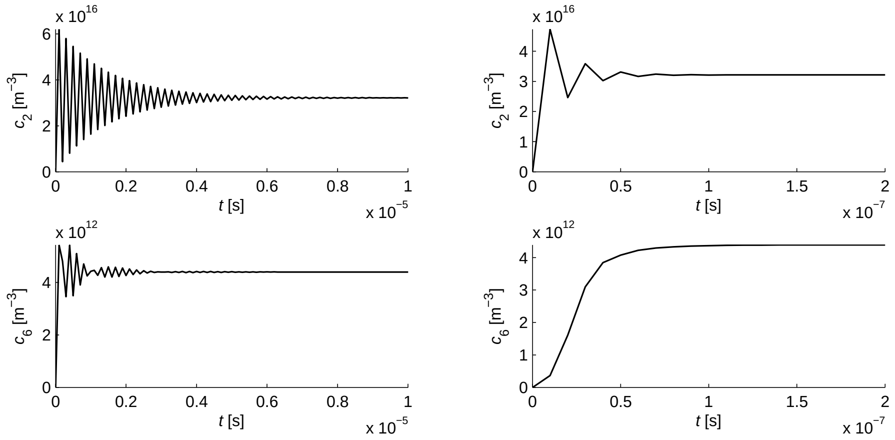
Figure F.1: DSD after 10 μs computed with implicit methods for $\Delta t = 10^{-7}$ (solid), $\Delta t = 10^{-8}$ (dashed) and $\Delta t = 10^{-9}$ (dash-dot). A solution from a direct RK4 solver with $\Delta t = 5 \cdot 10^{-11}$ (red) is plotted for comparison. Large time steps overestimate droplet growth.

Clearly, the risk of an inaccurate solution exists. Large Δt result in an overestimation of droplet growth, causing the front of the distribution to be advanced to far. The solution seems to be accurate for $\Delta t = 10^{-8}$with the trapezoidal method, and for $\Delta t = 10^{-9}$ for Euler backward. This confirms the higher order of the trapezoidal method.

Furthermore for small droplets, computation with the trapezoidal method introduces spurious oscillations inherent to the numerical algorithm. This is displayed in figure F.2, which shows the time histories for droplets $n = 2$ and 6. The oscillations are most significant for the smallest droplets for which, with large enough Δt, the transition from the initial solution $\mathbf{c}_n = \mathbf{0}$ to the next time solutions is very drastic. For values of $\Delta t = 10^{-8}$ the spurious oscillations are small enough so that they are dampened in a few time steps.

F.1.2 Analysis

In this case the sequence will be constructed from the advancement of the front of the DSD, measured as n_f, since this is the area that is most prone to



(a) $\Delta t = 10^{-7}$ s (b) $\Delta t = 10^{-8}$ s

Figure F.2: Numerical oscillations occurring for small droplets with the trapezoidal method. Oscillations are larger and more persistent for large Δt.

inaccuracy. Also, the number of gridpoints in this case is the inverse of the time steps so:

$$|n_f(\Delta t) - n_f^*| < C(\Delta t)^p \quad \text{for all} \quad \Delta t \tag{F.2}$$

where n_f^* is the exact solution of the problem for $\Delta t \rightarrow 0$. The convergence analysis requires three simulations with different amounts of time steps to determine all three parameters. When the parameters are determined, a required Δt_α can be determined for which $n_f = \alpha n_f^*$ and α is the required accuracy.

The convergence analysis is performed with two data sets of 3 simulations: $\Delta t = 10^{-7}, 10^{-8}, 10^{-9}$ and $\Delta t = 10^{-8}, 10^{-9}, 10^{-10}$. This is required to ensure that the parameters are obtained from simulation that are within the range of convergence of the discretization method. If this is not the case, solutions will not behave according the laws above and will give different results.

Table F.1 shows the results from the simulations and the convergence analysis. It seems that the simulation with $\Delta t = 10^{-7}$ is not quite within the range of convergence, yielding different results between data sets. However, the orders predicted with the second data set approach the theoretical orders, indicating results obtained from the second analysis are accurate. The results show that to obtain a 95% accurate solution, the trapezoidal method requires $\Delta t = 2.0 \cdot 10^{-8}$ and Euler backward requires $\Delta t = 1.5 \cdot 10^{-9}$. Indeed, an accurate solution is achieved with $O(10^{-8})$ and $O(10^{-9})$ size time steps respectively, which are considerably larger that the stability limit of RK4.

It is noted that the monitored front n_f, is a value obtained from a 1st order interpolation of the data. This influences the convergence analysis, leading to some inaccuracy in the determined order of the method. However, as the

Table F.1: Convergence analysis for the implicit methods

(a) Largest formed droplet n_f

$\Delta t\,[\mathrm{s}]$	Trapezoidal	Euler back
10^{-7}	45653	90899
10^{-8}	35163	43303
10^{-9}	34746	35882
10^{-8}	35163	43303
10^{-9}	34746	35882
10^{-10}	34741	34863

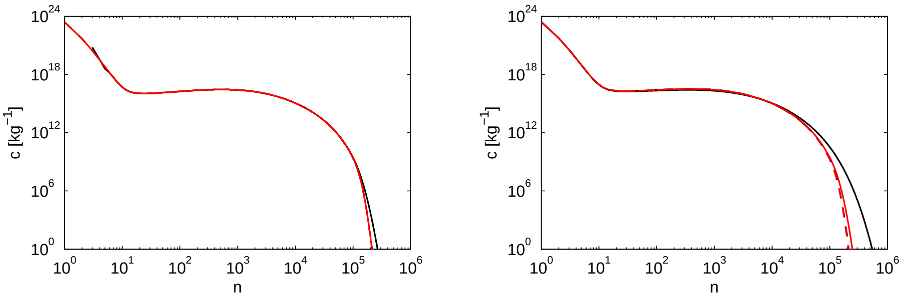
(b) Convergence analysis

	Trapezoidal	Euler back
p	1.4008	0.8072
n_f^*	34729	34512
$\Delta t_{0.95}$	$2.7 \cdot 10^{-8}$	$1.3 \cdot 10^{-9}$
p	1.9461	0.8623
n_f^*	34741	34701
$\Delta t_{0.95}$	$2.0 \cdot 10^{-8}$	$1.5 \cdot 10^{-9}$

orders are quite close to their predicted theoretical values, the impact of interpolation is considered minimal.

F.2 Nozzle flow

The nozzle flow experiment is carried out using the Multgrid solver for grouped droplets. The part of the nozzle considered here is a high saturation region from $x = 0.002$ m to $x = 0.012$ m. The solutions are obtained on grids with $\Delta x = 10^{-4}$, 10^{-5}, 10^{-6} and 10^{-7} m. The largest droplet in the simulation is set to $N = 1 \cdot 10^8$, which is collected in 2050 bins using 0°rder interpolation. Only the trapezoidal and Euler backward methods are considered for time integration.



(a) Trapezoidal method)

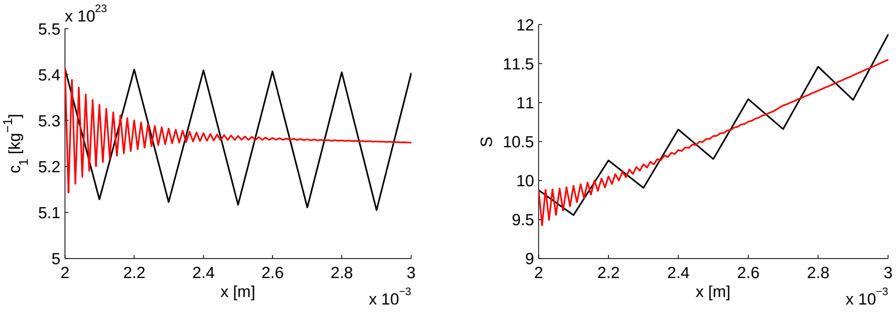
(b) Euler backward

Figure F.3: DSD at $x = 0.012$m - Solutions for $\Delta x = 10^{-4}$ m (solid black), 10^{-5} m (solid red), 10^{-6} m (dashed black) and 10^{-7} m (dashed red).

F.2.1 Convergence tests

Figure F.3 displays the DSDs at $x = 0.012$ m for the different simulations. For the trapezoidal method only the simulation with $\Delta x = 10^{-4}$ m can be clearly distinguished, as the other simulations produce almost the same solution. Backward Euler shows more deviation in the solution for different space step sizes.

With the trapezoidal method, the solutions for $\Delta t = 10^{-4}$ and 10^{-5} m seem to suffer from the earlier encountered numerical oscillations. This can be seen when looking more closely to the specific monomer density $\hat{c}_1$ and the supersaturation S. For $\Delta x = 10^{-4}$ m the oscillations are very persistent, whereas for $\Delta x = 10^{-5}$ m the oscillations are damped after a few steps.



(a) Specific monomer density $\hat{c}_1$ (b) Supersaturation S

Figure F.4: Numerical oscillations occurring in the solutions for larger space steps: $\Delta x = 10^{-4}$ m (black) and 10^{-5} m (red).

F.2.2 Analysis

Table F.2 shows the results from the simulations and the convergence analysis. As with the nucleation pulse experiment the order of the method p, the exact location of the front n_f^* and the required space integration step for 95% accuracy $\Delta x_{0.95}$ are obtained from two data sets.

For both time integration methods there is a good agreement between both data sets, which indicates that all solutions are within the asymptotic range of convergence. The predicted orders of convergence approach the theoretical values of two and one for the trapezoidal and the Euler backward method respectively. For a 95% accurate solution a space step size of $\Delta x = 3.8 \cdot 10^{-5}$ m is required for the former, and $\Delta x = 2.8 \cdot 10^{-6}$ m for the latter.

The numerical oscillations encountered for the trapezoidal method should be small enough to be damped within a few space steps, and should therfore not affect the accuracy of the predicted DSD.

Table F.2: Convergence analysis for the implicit methods

(a) Largest formed droplet n_f

$\Delta t\,[s]$	Trapezoidal	Euler back
10^{-4}	263870	542128
10^{-5}	211103	244008
10^{-6}	210347	211346
10^{-5}	211103	244008
10^{-6}	210347	211346
10^{-7}	210340	207940

(b) Convergence analysis

	Trapezoidal	Euler back
p	1.8442	0.9603
n_f^*	210337	207327
$\Delta x_{0.95}$	$4.1 \cdot 10^{-5}$	$2.7 \cdot 10^{-6}$
p	1.9710	0.9819
n_f^*	210340	207543
$\Delta x_{0.95}$	$3.8 \cdot 10^{-5}$	$2.8 \cdot 10^{-6}$