UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

MATHEMATICAL MODELING OF THE DESTABILIZATION OF ACID IN CRUDE OIL EMULSIONS USING POPULATION BALANCE EQUATION

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

SHAMS EL-DIN EL-FOULY
Norman, Oklahoma
2019

MATHEMATICAL MODELING OF THE DESTABILIZATION OF ACID IN CRUDE OIL EMULSIONS USING POPULATION BALANCE EQUATION

A THESIS APPROVED FOR THE
MEWBOURNE SCHOOL OF PETROLEUM AND GEOLOGICAL ENGINEERING

BY

Dr. Mashhad Fahes, Chair

Dr. Xingru Wu

Dr. Catalin Teodoriu

Acknowledgements

I would like to express my gratitude for my committee chair, Dr. Fahes, for her support and encouragement throughout the course of this research. I would also like to express my gratitude to Dr. Wu and Dr. Teodoriu for serving on my thesis committee. I would also like to thank Dr. Misra for awarding me the summer assistantship in which I have learned valuable information that assisted me in this current work.

And finally thanks to my family and to my friends at University of Oklahoma who have Supported me along the way.

Table of Contents

List of Tables

List of Figures

Abstract

In this current work, a phenomenological mathematical model that utilizes the population balance equation was developed to understand the destabilization process of acid in crude oil emulsions in laboratory conditions. The model considers changes due to advection, diffusion and binary coalescence of the dispersed acid phase. The model studies specifically the evolution of the droplet size distribution as a function of time for the top, next to bottom and bottom layers of a vertical cylinder. The resulting model was a nonlinear hyperbolic intergo-partial differential equation.

As with many complex mathematical models, this model required a numerical solution. The source term modeling birth and death of droplets was numerically solved using Kumar's fixed-pivot technique and converted to a system of partial differential equations. As for the advection-diffusion transport properties, five numerical solutions were examined each suitable under certain model properties. Four of the techniques were standard upwind schemes that are second, third and fourth order accurate, and the fifth was a non-standard method. All these schemes were also optimized to reduce the numerical errors to almost negligible. The non-standard method proved to be superior to the traditionally used upwind schemes for fine and course grids and low and high Reynolds numbers.

Based on this algorithm, the numerical model was solved using initial literature experimental data for acid/oil emulsions and was able to provide a suitable prediction of droplet size distribution profiles for each of the three layers of interest in addition to both dispersed and continuous phase volumes. The model was also found to be a good predictive tool for the initial mean droplet size for early acid/oil emulsion days using an exponential regression model for each time step. The volume fraction of acid was found to be highly sensitive to changes in mean

droplet size but was inconclusive in accessing the effect of standard deviation on the model evolution. The proposed algorithm has also presented evidence of the presence of a cohesive process dominating the system alongside the destabilization process of advection, diffusion and binary coalescence.

CHAPTER 1: INTRODUCTION

1.1 Emulsions

The geochemical interactions between crude oil and injected fluids cause a large number of consequences. In some cases, these interactions could enhance the hydrocarbon recovery (Mehana and El-Monier, 2015; Mehana and Fahes, 2016), but in others, they can sabotage a hydrocarbon production well (James et al. 2018). But one of the most fascinating by product of these interactions and the focus of this work are emulsions. An emulsion is a dispersion of a liquid within another immiscible liquid. In the context of oil and gas, emulsions exist in many forms. The two common types are water-in-oil emulsions and crude oil in water emulsions. **Figure 1** shows the two types of emulsions showing also surfactant molecules.

Emulsions are formed when three criteria are met. The first criterion involves the contact of two immiscible liquids, such as oil and water. The presence of water in a reservoir can be either due to existing water, or due to EOR efforts to pressurize the reservoir (Mehana et al. 2018). The second criterion involves the presence of a surface-active agent such as Asphaltene and Resins that provide a stabilizing effect on emulsions (Michell and Speight 1973). They form films that act as barriers to coalescence, flocculation and settling of the droplets. These films also lower the interfacial tension and are therefore able to minimize the energy required to create emulsions. In addition, they allow the creation of smaller droplets. The third and final criterion is the availability of sufficient turbulence or mixing energy to disperse one liquid into another. Such turbulence is mostly due to shear forces from motion through porous media, valves, pumps, and other production equipment (Wong 2015).

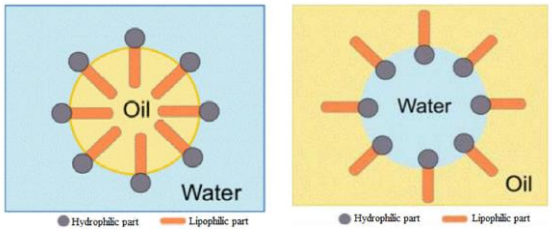


Figure 1 – Water in oil Emulsion and oil in water emulsion (Khan et al. 2011)

The produced emulsion is often separated using gravity settlers. Which are among the least expensive and simplest methods of separation (Barnea 1975). This type of separation is generally known as Liquid-Liquid separation.

Whether emulsions are good or bad depends on the context. Emulsions formed at the wellhead are undesired. But emulsions such as drilling mud are intentionally formed to aid in vital drilling activities such as providing hydrostatic support, cleaning cuttings and cooling drill bits. In addition, heavy oils are sometimes turned into emulsions to reduce its viscosity for economic transportation (Kilpatrick 2012).

1.2 liquid-liquid separation modeling

1.2.1 Early experimental modeling

Barnea and Mizrahi (1975 a, b, c, d) laid the ground work for liquid-liquid separation in four seminal papers. They identified experimentally three layers, the top, next to bottom, and bottom layers of an emulsion is a separation apparatus. The top layer comprised of the continuous phase and bottom layer of the dispersed phase in its homo-phase form, whereas the next to bottom represented the region of dispersion.

The bottom layer is formed when enough droplets undergo binary coalescence forming a separate phase – Binary coalescence is the process in which two droplets merge to form a larger

droplet. After this layer is formed, coalescence with this layer is then referred to as interfacial coalescence.

The region of dispersion is often made up of two layers: a dense packed layer and a sedimentation layer. The sedimentation layer is characterized with high mobility and a probability of binary coalescence. This process proceeds until the coalescence front (i.e. the interface between the bottom layer & the dense packed zone) merges with the sedimentation front (i.e. the interface between the top layer & the sedimentation zone). The resulting state is referred to as the settled state where the two phases separate completely. **Figure 2** shows the three layers observed in an emulsion in its transient state and its settled state after both immiscible liquids have separated completely. Despite the dispersed layer is subdivided into two type of dispersions, it is regarded as a single layer representing the emulsified region.

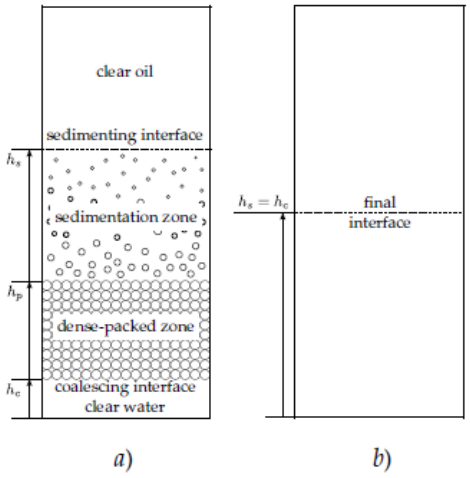


Figure 2 – Liquid-liquid separation in a batch settler and the evolution of front within it. a) transient. b) Final settled state (Barnea and Mizrahi 1975)

1.2.2 Theoretical modeling of emulsions

Barnea and Mizrahi did ground breaking work to understand experimentally the layers that make up an emulsion. On the other hand, Jeelani and Hartland (1986 and 1998) presented

theoretical models to study the kinematics of these fonts in addition to the evolution of the height of the dense-packed layer. Their models depended on various experimental parameters such as settling time, initial droplet size distribution, and single drop coalescing time. Other authors such as Nadiv and Semiat (1995) presented similar models using different parameters such as the coalescence and sedimentation velocity.

The models up to this point study the evolution of the fronts without considering the droplet changes in the sedimentation zone or the dense-packed zone. Henschke et al. (2002) established models that predict the evolution of these fronts by taking into account the droplet changes and deformation in both the sedimentation and dense-packed zones.

All the previously mentioned models are phenomenological models since they depend on either experimental and/or theoretical estimates to model the sedimentation and coalescence. They are also front kinematic models. They study the evolution of all the fonts present in the liquid-liquid separation process.

1.2.3 Conservation laws and liquid-liquid separation modeling

There is another type of phenomenological models that are based instead on the conversation of mass of the dispersed phase, where the dispersed phase is polydisperse (i.e. of diverse droplet sizes). The model considers all possible changes that the dispersed population can undergo. Changes due to settling, diffusion, coalescence, breakage and nucleation. All this while conserving the mass of the dispersed phase. Each of these phenomena are defined in accordance to the properties of the dispersed and continuous phases. This model is commonly referred to as the population balance equation.

A shortcoming of this type of models is that they don't consider the flow of the continuous phase. The continuous phase is considered static. Drumm (2010) establish a model to

couple computational Fluid Dynamics with the population balance equation. In other words, in addition to the PBE solving the conservation of mass of the dispersed phase, this model also attempts to solve the conservation of momentum between the dispersed phase and continuous phase. The output of both of these models is the droplet size distribution of the dispersed phase at each material point.

1.3 The importance of studying emulsion evolution

The emulsion's Droplet Size Distribution (DSD) has a significant impact on both the emulsion stability and viscosity. Emulsions with small droplet sizes for instance promote a more stable emulsion and reduce viscosity (Raikar. et al 2009). Therefore, information in regard to the evolution of the droplet size distribution gives us insight into the evolution of the stability and viscosity of the emulsion.

There are two contexts in which the change is droplet size distribution can be studied. The change can either be studied during the mixing process (i.e. in turbulent conditions) or after the mixing has stopped (i.e. Static or settling conditions). Studying the change during the mixing process is critical to reduce time and resources invested in finding the right emulsion procedure to produce the desired emulsion otherwise done through trial and error (Raikar. Et al 2009). As for studying the change after mixing, as mentioned before, this is important to understand the evolution of the emulsion's stability and viscosity.

1.4 Emulsion destabilization mechanisms

Emulsions are mostly known to be kinetically stable and thermodynamically unstable (Solsvik Et al 2015). Due to this fact, emulsions destabilize through six different mechanisms as seen in **Figure 3**. Sedimentation and creaming are types of gravity segregation mechanisms

driven mostly by gravity forces and differences in density. If the dispersed phase has a density greater than the surrounding fluid, the process of sedimentation sets in. As for when the dispersed phase is lighter, creaming dominates this kind of system (Pena, 2004).

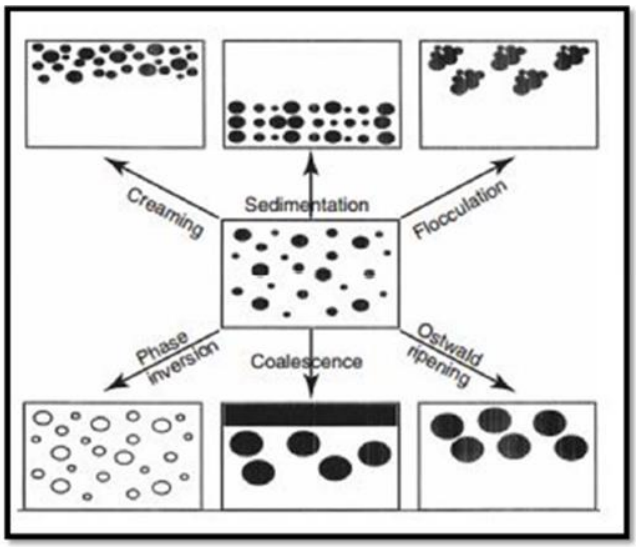


Figure 3 - Emulsion Destabilization Mechanisms (Tadros. 2013)

Other types of destabilization are flocculation and coalescence. Coalescence can be either binary or interfacial. Binary coalescence is when two droplets merge to form a new droplet, and interfacial coalescence is when a droplet(s) merge with an interface. The phenomenon of coalescence and breakage is driven by a balance between the elastic and interfacial energies of droplets as seen in **Figure 4** (Dahiya, 2016).

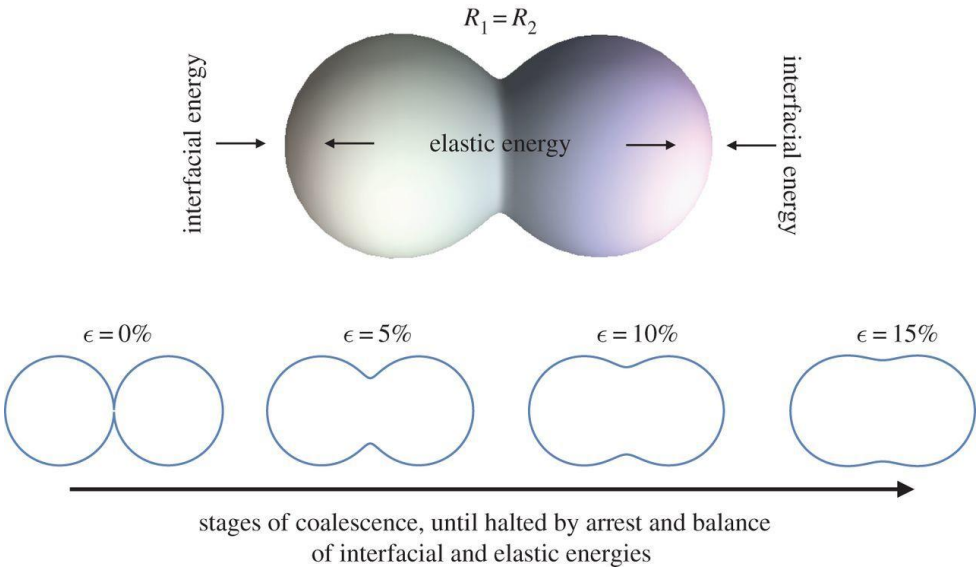


Figure 4 – Interfacial Energy and elastic energy in the binary coalescence phenomena (Dahiya, 2016)

The two remaining mechanisms are phase inversion and Ostwald ripening. In the case of phase inversion, the continuous phase becomes the dispersed phase and dispersed phase becomes the continuous phase. As for Oswald ripening, this phenomenon occurs when droplets grow due to the presence of a non-particulate matter in solution (Tadros. 2013).

1.4.1 Emulsion destabilization mechanisms in laboratory conditions

Despite the heterogeneity of the oil composition (Mehana et al. 2019), there are three primary mechanisms that occur in crude oil emulsions. Since water and acid commonly have densities larger than oil, sedimentation is usually present in unstable emulsions. And since droplets are always in constant motion and colliding, either with themselves or an interface, binary coalescence and interfacial coalescence are usually present as well. However, it has been observed in laboratory experiments on acid in crude oil emulsions that interfacial coalescence does not occur (Scarborough, 2016). Instead, a very viscous and almost solid like layer occurs at

the bottom instead of a layer of acid. It is worth noting that despite the dependence of asphaltene stability on the crude oil composition (Mehana et al. 2019), the asphaltene/acid interactions were neglected. **Figure 5** shows the evolution of such layer as it starts as a fluid layer and evolves into a solid highly viscous layer with no presence of a pure acid layer.

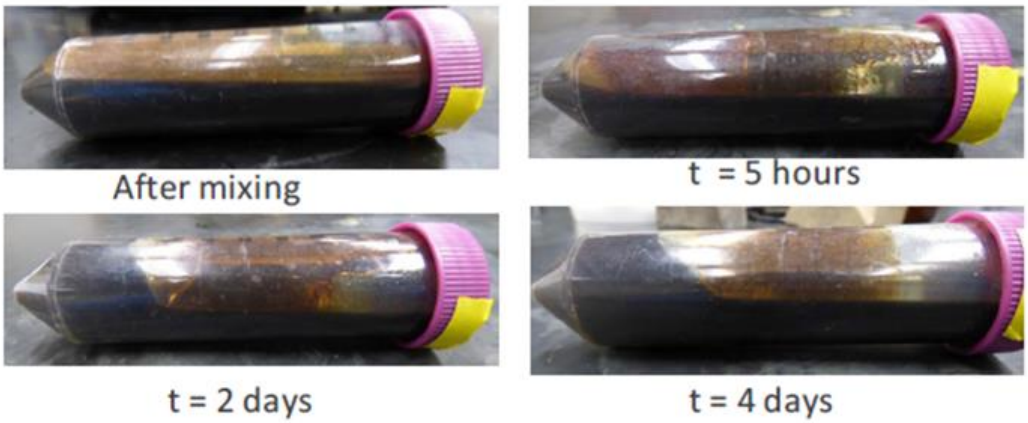


Figure 5– Separation vs. Time of a 40% acid in crude oil emulsion (Scarborough, 2016)

1.5 Purpose of this work

The purpose of this work was to identify the main processes that govern the destabilization mechanisms in acid in crude oil emulsions. The best way to accomplish this task was to track directly the changes in the acid fraction as a function of time. Models devoleped by Jeelani and Hartland (1986 and 1998) and Henschke et al. (2002) are excellent at tracking the front kinematics properties of an emulsion (i.e. the displacement and velocity of the fronts) but do not track the acid fraction changes directly therefore do not identify specifically what destabilization mechanisms are involved in the emulsion. And secondly, acid/oil emulsions in their early days do not have a visible separated layer of oil or a visible separated layer of acid therefore two of the fronts studied by these models do not exist. A model that would best fit the purpose of this work would be the population balance equation. This model not only studies the

dispersed phase directly, but also gives the flexibility in studying any number of possible changes that a dispersed population can undergo, in addition to understanding the impact of each on the destabilization process. The population balance equation was used successfully to study the processes that govern water in crude oil emulsions (Cunha 2008). In this paper, similar methodology was used to understand the effect of mixing acid with crude oil instead of water. The choice to independently code this model gave the flexibility to apply a large range of numerical solutions on the PBE equation.

To accomplish the task of identifying the processes that control acid in oil emulsion evolution, Scarborough's (2016) density measurements were used as a guide since density can be easily converted to acid volume fraction through **equation 1**.

$$\Phi_{Acid} = \frac{\rho - \rho_{Oil}}{\rho_{Acid} - \rho_{Oil}} \quad (1)$$

Scarborough (2016) conducted a series of experiments on five three-inch-long acid in oil emulsion columns examining both the change in viscosity & density in the bottom and next to bottom layers for concentrations of 20%, 30% and 40% acid fractions. **Figure 6** shows a schematic of the experimental methodology of the sampling procedure.

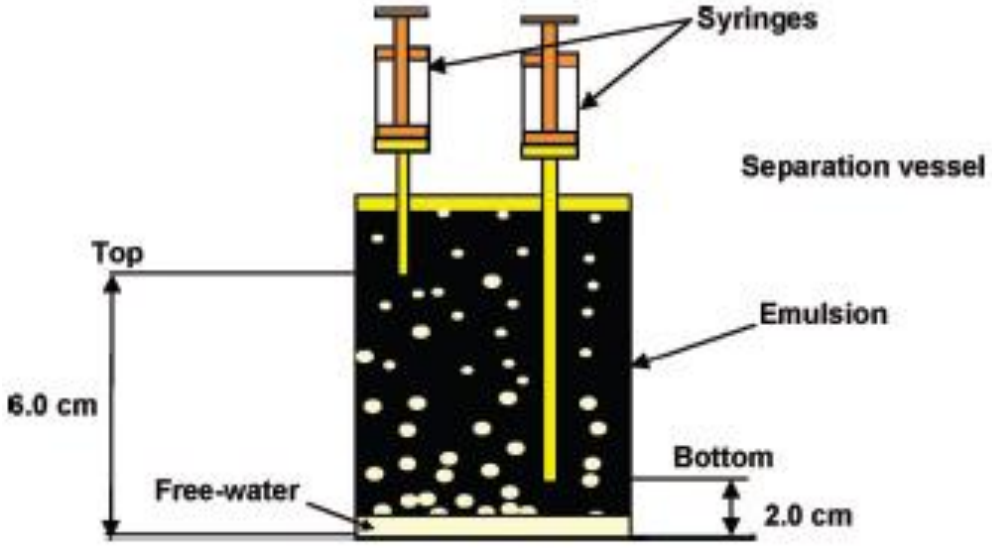


Figure 6 – Emulsion destabilization and sampling in a test tube (Cunha, 2008)

His main objective was to relate the obtained viscosity and density data for the three concentration experiments for the bottom and next to bottom layers to create a cross plot of viscosity and density as shown in **Figure 7.** Viscosities ranged from 29.77 cp to 3187 cp. Since density was measured with time in these experiments, for our purposes, it was used as a proxy of the change in acid fraction as a function of time.

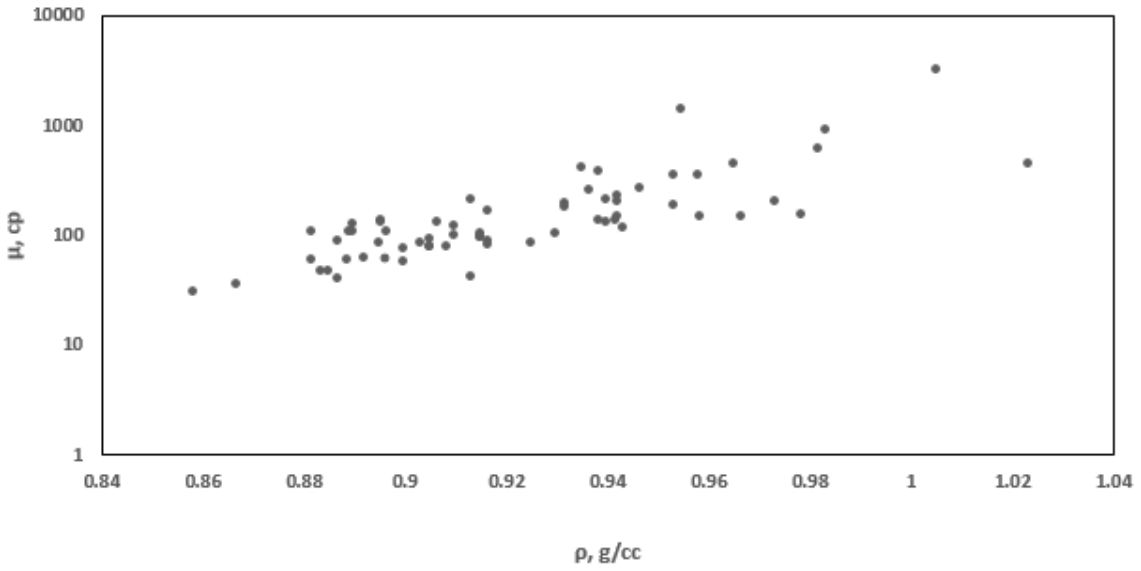


Figure 7 – Cross Plot between viscosity and density for 20%, 30% and 40% acid/oil Emulsions. Scarborough (2016)

1.6 Destabilization modeling

The main output of the PBE is the droplet size distribution (DSD). **Figure 8a** shows an example of a density distribution function. This function gives the fraction that each droplet size makes up from the present population. As a result, the area under the curve of such graph is equal to 1. As for **Figure 8b**, the function provides the exact number of droplets that each diameter contributes. This distribution is called a number distribution. This type of distribution is generated with knowledge of each droplet's volume with the assumption that the droplets are spherical. Therefore, the area under the curve of such graph reflects the volume of the dispersed fluid droplets. Calculating the area under the curve of the droplet size distribution at each time step for a given space would therefore provide the volume fraction at that time and space. With this idea, the volume fraction can be tracked through time.

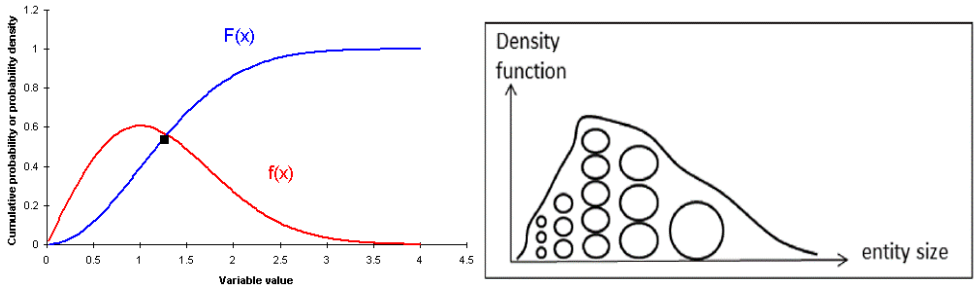


Figure 8 – a) Example of density distribution function and cumulative curve. b) Number distribution of a dispersed droplet size Population (Solsvik and Jakobsen 2015)

Figure 9 provides a bird's eye view of the model process. The model aims at tracking the droplet size distribution changes within 3 layers of a 3in test tube. The test tube will be discretized evenly into three 1in sections, top, next to bottom and bottom layers. Within each of the layers the change in DSD will be tracked using the population balance equation. The model tracks the change in concentration of each droplet as a function of changes due to advection, diffusion and binary coalescence of the dispersed acid phase. The choice of the number of droplets to track will depend on the number of droplets present in the emulsion in addition to their concentrations. This information is usually provided experimentally. Since an experimental DSD was not available at the time of this work, only the change in concentration of 20 common size droplets was considered that fall along a log-normal distribution. Acid in oil emulsions follow commonly a lognormal type distribution with a very small mean droplet size (Opedal 2009). **Figure 10** below shows an example of literature results of DSD of a crude oil emulsion using NMR and microscope. It is important to note that the model can track any number of droplets with any number of concentrations.

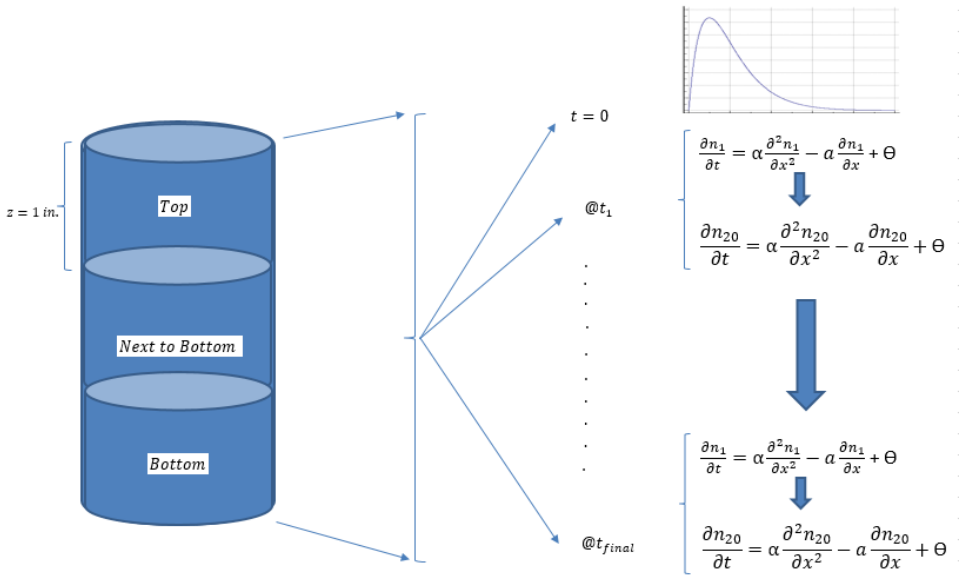


Figure 9 – Bird's Eye View of the Destabilization Modeling

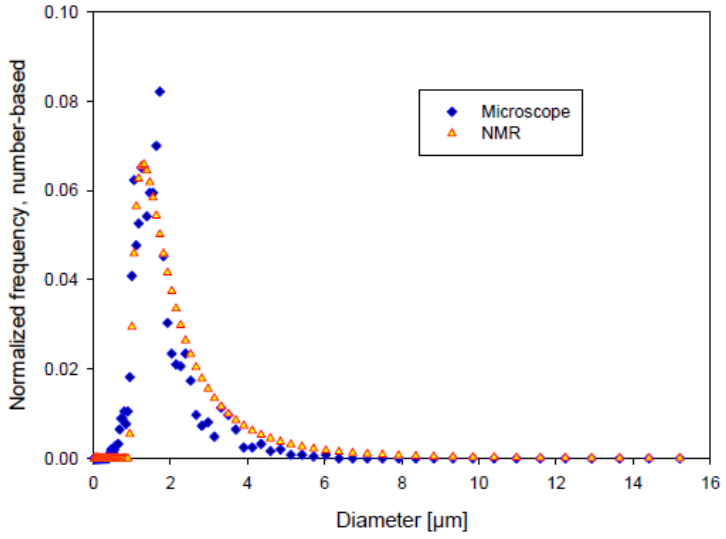


Figure 10 – DSD for a crude oil emulsion from NMR and Microscope (Opedal 2009)

CHAPTER 2: POPULATION BALANCE EQUATION

2.1 Overview

This chapter provides the mathematical model that will be utilized to track the changes in the droplet size distribution for each of the three layers in the test tube. In addition, it will go through a detailed discussion of the numerical techniques used to solve this type of model.

The model used in this work is referred to as the population balance equation. This model is a nonlinear hyperbolic intergo-partial differential equation and is defined in **equation 2**.

$$\frac{\partial n_k}{\partial t} = \frac{\partial}{\partial z}\left(D_z \frac{\partial n_k}{\partial z}\right) - \frac{\partial (w_k n_k)}{\partial z} + \theta_k \qquad (2)$$

Where n_k is the concentration of droplet k at a given time and space volume, D_z is the diffusivity factor, w_k is the settling velocity of a given droplet k, and finally θ_k represents the binary coalescence and breakage of droplet k.

This model is made up of two primary parts, the Advection-Diffusion equation (ADE) and the source term. The Advection-diffusion equation is one of the most challenging and important equations in engineering since it comprises of a superposition of two transport processes. This equation is used in a wide variety of engineering disciplines from computational fluid dynamic to acoustics to heat & mass transfer (Cunha, 2008).

The ADE equation is a second order parabolic partial differential equation made up of a time derivative and of first and second order spatial derivatives. The first order spatial derivative models the change is concentration due to droplet settling mainly driven by the force of gravity. In other words, a change in concentration of any droplet in a layer can be either due to a droplet settling into or out of a layer. As for the second order spatial derivative, it tracks the change in concentration due to droplet diffusion. This is a phenomenon where droplets move from a higher

energy state to a lower energy state. As with settling, a change in concentration in this case can be either due to droplets diffusing into or out of a layer.

In addition to droplets settling and diffusing, the distribution of the population of the dispersed phase can change due to interactions between droplets. There are four different mechanisms the droplets can interact. Droplets can either aggregate, break apart, grow or nucleate. The process of aggregation is when two or more droplets combine to form a larger droplet. As for breakage, a droplet can fragment into two or more smaller droplets.

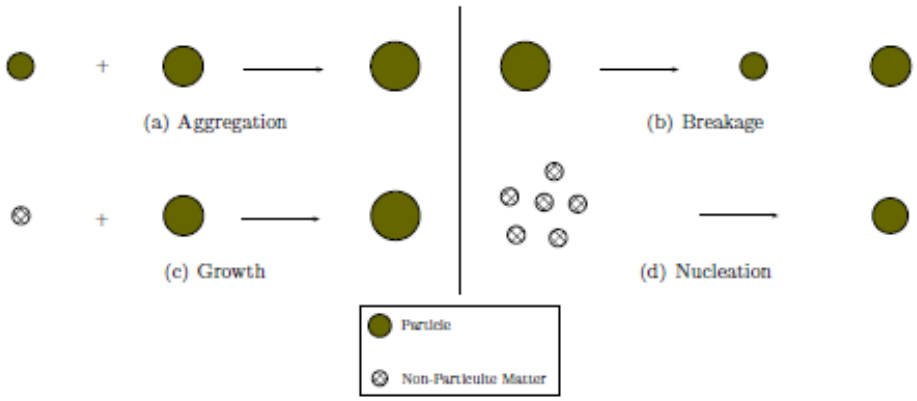


Figure 11 – Different particle formation mechanisms (Kumar 2006)

In some emulsion systems there exists a non-particulate matter when deposited on a droplet will cause it to grow. If this happens, the droplets are said to have undergone the process of growth. Furthermore, non-particulate matter has the ability to combine with itself to form new droplets. This process is referred to as nucleation. A detailed desciption of the exact process of birth and death of pariticles will be discussed later in this work. **Figure 11** summarizes the four mechanisms.

In this work, only the process of aggregation will be studied. Aggregation can result in both the death and birth of droplets. For example, if a droplet k aggregates with another droplet,

droplet k as a result dies to form a new larger droplet. However, two droplets with a combined volume equal to k can merge to form a new k droplet. These two phenomena are defined by theta as seen in **equation 3**.

$$\theta = \frac{1}{2}\int_0^v \beta(v-v',v')n(v-v',z,t)n(v',z,t)dv' - \int_0^\infty \beta(v,v')n(v',z,t)n(v',z,t)dv' \tag{3}$$

Where $v\ and\ v'$ are droplet volumes and β is the collision frequency.

The general population balance equation therefore takes the expanded form in **equation 4** descrbing the change in droplet size distribuation as a function of diffusion, advenction and binary coalescence

$$\frac{\partial n_k}{\partial t} = \frac{\partial}{\partial z}\left(D_z \frac{\partial n_k}{\partial z}\right) - \frac{\partial (w_k n_k)}{\partial z} + \underbrace{\frac{1}{2}\int_0^v \beta(v-v',v')n(v-v',z,t)n(v',z,t)dv'}_{Birth\ Term} - \underbrace{\int_0^\infty \beta(v,v')n(v',z,t)n(v',z,t)dv'}_{Death\ Term} \tag{4}$$

2.2 Advection diffusion equation

The first part of the population balance equation that will require numerical modeling is the advection-diffusion equation. The one-dimensional PDE is given by **equation 5**.

$$\frac{\partial n}{\partial t} = \frac{\partial}{\partial z}\left(D_z \frac{\partial n}{\partial z}\right) - \frac{\partial (wn)}{\partial z} \tag{5}$$

Many techniques have been developed to solve this PDE equation. These techniques are generally categorized as either standard or non-standard finite difference schemes. The most

common standard finite difference schemes include the lax-wendroff, Crank-Nicolson and third and fourth order upwind schemes. As for the non-standard finite difference scheme, this method was developed with the objective to remove the numerical instability commonly present in standard schemes (wand and Roeger 2015). In return, NSFD scheme has a very wide stability range and can handle a wide range of applications.

In this section the stability of five numerical solutions are presented to choose the best model to fit our problem. Four of the techniques will be standard methods that are second, third and fourth order accurate, and the fifth will be a non-standard method. This will be followed by a numerical optimization of the parameters of the chosen method.

The main reason for presenting such a large number of solutions for the ADE equation is because of the large number of properties that can be studied. For some cases under specific properties, one solution might be unstable, but another might produce a perfectly accurate result. This work attempts to provide the best solutions available for the ADE equation and then selects the one best for the set of properties of acid, crude oil, and apparatus used.

2.2.1 General explicit and implicit finite difference methods

The general standard finite difference schemes are derived using Taylor series approximations of the first and second derivatives. The first derivative of advection is approximated using a weighted average between the backward finite difference (BFD) and forward finite difference (FDD) schemes and is given by **equation 6**.

$$\frac{\partial n}{\partial z} = \frac{(1-\gamma)(n_k - n_{k-1}) + \gamma(n_{k+1} - n_k)}{h} \qquad (6)$$

The spatial weight factor is defined as γ. When γ is equal to 1, the first derivative is approximated fully using the FFD and when γ is equal to 0, the first derivative is approximated fully using the BFD.

Furthermore, the temporal dimension in incorporated in this problem using a weighted average between the current time step and future time step. Factoring in the time dimension, the advection term now takes the following form as seen in **equation 7**.

$$\frac{\partial n}{\partial z} = (1-\Phi)\left[\frac{(1-\gamma)(n_k^n - n_{k-1}^n)+\gamma(n_{k+1}^n - n_k^n)}{h}\right] + \Phi\left[\frac{(1-\gamma)(n_k^{n+1} - n_{k-1}^{n+1})+\gamma(n_{k+1}^{n+1} - n_k^{n+1})}{h}\right] \quad (7)$$

Φ is defined as the temporal weight factor. When Φ is equal to 0, the first derivative is defined fully in terms of the current time step. Such approach is defined as explicit. As for the remaining values of Φ, the first derivative in defined implicitly.

Unlike the first derivative, the second derivative of diffusion is approximated fully using the centered finite difference as seen in **equation 8**.

$$\frac{\partial^2 n}{\partial z^2} = \frac{n_{k+1} - 2n_k + n_{k-1}}{h^2} \quad (8)$$

Factoring in the time dimension, the final form of the diffusion term is given by **equation 9.**

$$\frac{\partial^2 n}{\partial z^2} = (1-\Phi)\frac{n_{k+1}^n - 2n_k^n + n_{k-1}^n}{h^2} + \Phi\frac{n_{k+1}^{n+1} - 2n_k^{n+1} + n_{k-1}^{n+1}}{h^2} \quad (9)$$

By substituting the final approximation of the first and second derivative in the original ADE equation and approximating the time derivative using the forward divided finite difference we obtain a general explicit and implicit finite difference method where k is the time step and h is the spatial step size as seen in **equation 10**.

$$\frac{n_k^{n+1} - n_k^n}{k} = Dz\left[(1-\Phi)\frac{n_{k+1}^n - 2n_k^n + n_{k-1}^n}{h^2} + \Phi\frac{n_{k+1}^{n+1} - 2n_k^{n+1} + n_{k-1}^{n+1}}{h^2}\right] - wz\left[(1-\right.$$

$$\left.\Phi)\left(\frac{(1-\gamma)(n_k^n - n_{k-1}^n)+\gamma(n_{k+1}^n - n_k^n)}{h}\right) + \Phi\left(\frac{(1-\gamma)(n_k^{n+1} - n_{k-1}^{n+1})+\gamma(n_{k+1}^{n+1} - n_k^{n+1})}{h}\right)\right] \quad (10)$$

By taking $h^2$and h as a common factor we obtain **equation 11**.

$$n_k^{n+1} - n_k^n = \frac{Dz*k}{h^2}\,[(1-\Phi)n_{k+1}^n - 2n_k^n + n_{k-1}^n + \Phi n_{k+1}^{n+1} - 2n_k^{n+1} + n_{k-1}^{n+1}] -$$

$$\frac{wz*k}{h}[(1-\Phi)(1-\gamma)(n_k^n - n_{k-1}^n) + \gamma(n_{k+1}^n - n_k^n) + \Phi(1-\gamma)(n_k^{n+1} - n_{k-1}^{n+1}) + \gamma(n_{k+1}^{n+1} -$$

$$n_k^{n+1})] \qquad (11)$$

For the purpose of simplification $\frac{Dz*k}{h^2}$ and $\frac{wz*k}{h}$ and defined as s and c, respectively as seen in

equation 12

$$n_k^{n+1} - n_k^n = s\,[(1-\Phi)n_{k+1}^n - 2n_k^n + n_{k-1}^n + \Phi n_{k+1}^{n+1} - 2n_k^{n+1} + n_{k-1}^{n+1}] - c\,[(1 -$$

$$\Phi)(1-\gamma)(n_k^n - n_{k-1}^n) + \gamma(n_{k+1}^n - n_k^n) + \Phi(1-\gamma)(n_k^{n+1} - n_{k-1}^{n+1}) + \gamma(n_{k+1}^{n+1} - n_k^{n+1})]$$

(12)

Through simple algebraic manipulations, the concentration *n* of a droplet k at a future time step *n+1* is given by **equation 13**.

$$n_k^{n+1} = \frac{1}{1-\Phi[c(2\gamma-1)-2s]}((\Phi-1)[c(\gamma-1)-s]n_{k-1}^n + 1 + (\Phi-1)[c(1-2\gamma)+2s]n_k^n + (1 -$$

$$\Phi)[s-c\gamma]n_{k+1}^n + \Phi[s+c(1-\gamma)]n_{k-1}^{n+1} + \Phi[s-\gamma c]n_{k+1}^{n+1}) \qquad (13)$$

The approximated ADE equation can be further simplified as seen in **equation 14**.

$$n_k^{n+1} = \frac{1}{A_o}(A_1 n_{k-1}^n + A_2 n_k^n + A_3 n_{k+1}^n + A_4 n_{k-1}^{n+1} + A_5 n_{k+1}^{n+1}) \qquad (14)$$

$$A_o = 1 - \Phi[c(2\gamma-1)-2s] \qquad A_1 = (\Phi-1)[c(\gamma-1)-s]$$

$$A_2 = 1 + (\Phi-1)[c(1-2\gamma)+2s] \qquad A_3 = (1-\Phi)[s-c\gamma]$$

$$A_4 = \Phi[s+c(1-\gamma)] \qquad A_5 = \Phi[s-\gamma c]$$

Where, c = $w\ k/h$ and s = $D_z\ k/h^2$

2.2.2 Lax-wenhoff and crank-nelson

Manipulations of the spatial and temporal weight factors is what gives rise to the large number of standard explicit and implicit finite difference schemes available.

When$\Phi = 0\ And\ \gamma = \frac{1-c}{2}$, the resulting formula is referred to as the lax-wenhoff scheme and it is given by **equation 15**.

$$n_k^{n+1} = \frac{1}{2}(2s + c + c^2)\, n_{k-1}^n + (1 - 2s - c^2)\, n_k^n + \frac{1}{2}(2s - c + c^2)\ n_{k+1}^n \qquad (15)$$

This is an explicit scheme where the concentration of a droplet k at a future time step is defined in terms of the concentration of k-1, k and k+1 droplet at the current time step. This makes this technique second order accurate. **Figure 12** a visual computation of this method.

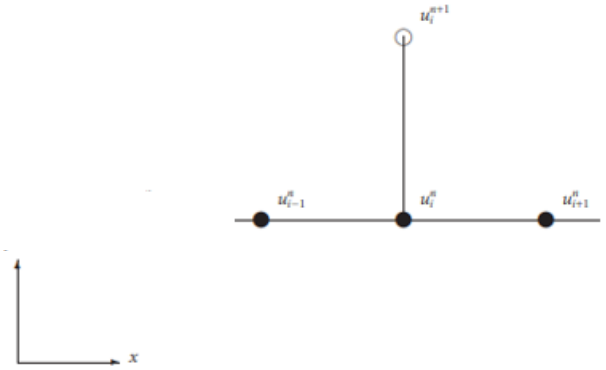


Figure 12 – Computation schematic for explicit scheme

When$\Phi = \frac{1}{2}\ And\ \gamma = \frac{1}{2}$, the resulting formula is referred to as the crank-Nelson and it is given by **equation 16**.

$$n_k^{n+1} = \frac{1}{4(1+s)}[(c + 2s)n_{k-1}^{n+1} - (c - 2s)n_{k+1}^{n+1} + (c + 2s)n_{k-1}^n - (c - 2s)n_{k+1}^n + (4 - 4s)n_k^n]$$

(16)

This is an implicit scheme where the concentration of a droplet k at a future time step is defined in terms of the concentrations at *both* the current and future time steps. If this formula is

rearranged so that the future terms are on the left side and current time step terms are on the right, we obtain **equation 17**.

$$A_4 n_{k-1}^{n+1} + A_o n_k^{n+1} - A_5 n_{k+1}^{n+1} = (A_1 n_{k-1}^n + A_2 n_k^n + A_3 n_{k+1}^n) \quad (17)$$

If this formula is expanded for a k number of droplets, this generates a special type of matrix, a tridiagonal matrix which takes this general form

$$\begin{bmatrix} A_o & -A_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_4 & A_o & -A_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_4 & A_o & -A_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_4 & A_o & -A_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_4 & A_o & -A_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_4 & A_o & -A_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_4 & A_o & -A_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_4 & A_o \end{bmatrix} \begin{Bmatrix} n_1^{n+1} \\ n_2^{n+1} \\ n_3^{n+1} \\ n_4^{n+1} \\ \cdot \\ \vdots \\ n_k^{n+1} \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ \cdot \\ \vdots \\ r_k \end{Bmatrix}$$

$Where\ r_k = (A_1 n_{k-1}^n + A_2 n_k^n + A_3 n_{k+1}^n)$

The coefficients $A_o\ to\ A_5$ and r_k are all known. Therefore, solving this matrix will result in the concentration of droplets at next time step.

It is important to note that since $A_o\ to\ A_5$ are dependent on c, and c is dependent on the droplet velocity, and the droplet velocity is dependent on diameter, $A_o\ to\ A_5$ are different for each droplet size. Therefore, each row in the tridiagonal matrix is generated using a different set of coefficients.

Due to the large number of zeros in this matrix, a very efficient technique that is used to solve this type of matrix is the Thomas algorithm.

2.2.3 Error of the general finite difference scheme

When using numerical models, the concept of stability always needs to be addressed. Every numerical model has certain parameters which make it stable and others that generate

numbers that do not make reasonable sense within the context of the problem being solved. Therefore, a stability criterion needs to be defined and for this general finite difference formula it's given as **equation 18**.

$$0 < s \leq \frac{1-c^2}{2} \qquad (18)$$

Furthermore, within the stability region there exists optimum parameters that reduce the error to almost zero. Finding the optimum parameter within the stability region will be discuss later in this section.

2.2.4 High accuracy formulas

In the previous section the first and second derivative of the advection-diffusion equation were approximated using what are called the conversional finite difference approximations. Where the first derivative is approximated using a first order Taylor series and the second derivative is approximated using a second order Taylor series.

In this section I will introduce a class of derivative approximations called the high accuracy finite difference formulas that would create a more accurate approximation for the ADE equations. These formulas unlike the conventional use a higher order Taylor series to approximate the derivatives. For example, the first derivative instead of being approximated using a first order Taylor series, it is approximated using a higher order Taylor series and the same applies for the second derivative of diffusion.

Third order upwind scheme

The first technique that is developed as a result of the application of the high accuracy formulas is the third order upwind explicit technique. The first and second spatial derivative for this method are approximated as **equation 19** and **20,** respectively.

$$\frac{\partial n}{\partial z} = \left(\frac{2c^2+3c+12s-2}{12}\right)\left(\frac{n_k^n - n_{k-2}^n}{2h}\right) + \left(\frac{2c^2-3c+12s-2}{12}\right)\left(\frac{n_{k+2}^n - n_k^n}{2h}\right)$$

$$+ \left(\frac{4-c^2-6s}{3}\right)\left(\frac{n_{k+1}^n - n_{k-1}^n}{2h}\right) \qquad (19)$$

$$\frac{\partial^2 n}{\partial z^2} = \left(\frac{6s-12sc+2c-2c^3+3c^2}{6s}\right)\left(\frac{n_{k+1}-2n_k+n_{k-1}}{h^2}\right)$$

$$+\left(\frac{12sc-2c+2c^3-3c^2}{12}\right)\left(\frac{n_{k+2}-2n_k+n_{k-2}}{4h^2}\right) \qquad (20)$$

The time derivative is approximated using a forward divided finite difference given by **equation 21**.

$$\frac{\partial n}{\partial t} = \frac{n_k^{n+1} - n_k^n}{k} \qquad (21)$$

Putting the above approximation into the ADE equation with further simplifications, we obtain the **equation 22**:

$$n_k^{n+1} = A_1 n_{k-2}^n + A_2 n_{k-1}^n + A_3 n_k^n + A_4 n_{k+1}^n \qquad (22)$$

$$A_1 = \frac{1}{6}c(c^2 + 6sc - 1) \qquad A_2 = \frac{1}{2}(2c - c^3 - 6sc + 2c + c^2)$$

$$A_3 = \frac{1}{2}(2 - 2c^2 - 4s + 6sc - c + c^3) \quad A_4 = \frac{1}{6}(1 - c)(6s + c^2 - 2c)$$

Where, c = $w\ k/h$ and s = $D_z\ k/h^2$

Compared to the lax-wenhoff, this technique approximates the concentration of a droplet k in terms of four points, k-2, k-1, k and k+1 instead of only three points k-1, k and k+1. This in consequence makes this technique much more accurate. This method is third order accurate. **Figure 13** is visual representation of the computation of this technique.

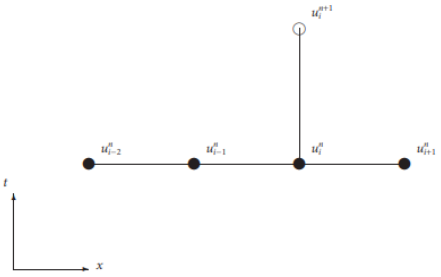


Figure 13 – Computation schematic for third order upwind technique

As with the previous techniques, the stability criteria for this method is given **equation 23.**

$$0 < s \leq \frac{1}{6}\frac{c(2+3c-2c^2)}{(2c-1)} \qquad (23)$$

Fourth order upwind scheme

The high accuracy formulas can be further used to generate a more accurate approximation of the ADE equation by generating a fourth order accurate finite difference scheme that is given by **equation 24**.

$$\frac{\partial n}{\partial z} = \left(\frac{12s + 2c^2 - 3c - 2}{12}\right)\left(\frac{n_{k+2}^n - n_k^n}{2h}\right) + \left(\frac{12s - 2c^2 + 3c - 2}{12}\right)\left(\frac{n_k^n - n_{k-2}^n}{2h}\right)$$

$$- \left(\frac{c^2 + 6s - 4}{3}\right)\left(\frac{n_{k+1}^n - n_{k-1}^n}{2h}\right)$$

$$\frac{\partial^2 n}{\partial z^2} = \left(\frac{-c^4 + 4c^2 - 12s^2 - 12sc^2 + 8s}{6s}\right)\left(\frac{n_{k+1}^n - 2n_k^n + n_{k-1}^n}{h^2}\right)$$

$$+ \left(\frac{c^4 - 4c^2 + 12s^2 + 12sc^2 - 2s}{6s}\right)\left(\frac{n_{k+2}^n - 2n_k^n + n_{k-2}^n}{4h^2}\right) \qquad (24)$$

The time derivative is approximated using a forward divided finite difference given by **equation 25**.

$$\frac{\partial n}{\partial t} = \frac{n_k^{n+1} - n_k^n}{k} \qquad (25)$$

Putting the above approximation into the ADE equation with further simplifications, we obtain the **equation 26**.

$$n_k^{n+1} = A\, n_{k-2}^n + B\, n_{k-1}^n + C\, n_k^n + D\, n_{k+1}^n + E\, n_{k+2}^n \qquad (26)$$

$$A = \frac{1}{24}\begin{pmatrix}12s(s+c^2) + 2s(6c-1)\\ +c(c-1)(c+1)(c+2)\end{pmatrix} \quad B = -\frac{1}{6}\begin{pmatrix}12s(s+c^2) + 2s(3c-4)\\ +c(c-2)(c+1)(c+2)\end{pmatrix}$$

$$C = \frac{1}{4}\begin{pmatrix}12s(s+c^2) - 10s\\ +(c-1)(c-2)(c+1)(c+2)\end{pmatrix} D = -\frac{1}{6}\begin{pmatrix}12s(s+c^2) - 2s(3c+4)\\ +c(c-2)(c-1)(c+2)\end{pmatrix}$$

$$E = \frac{1}{24}\begin{pmatrix}12s(s+c^2) - 2s(6c+1)\\ +c(c-1)(c+1)(c-2)\end{pmatrix}$$

Where, c = $w\ k/h$ and s = $D_z\ k/h^2$

As seen in the above equation, the fourth order Upwind scheme is approximated using 5 grid points k-2, k-1, k, k+1 and k+2 as seen in **Figure 14**.

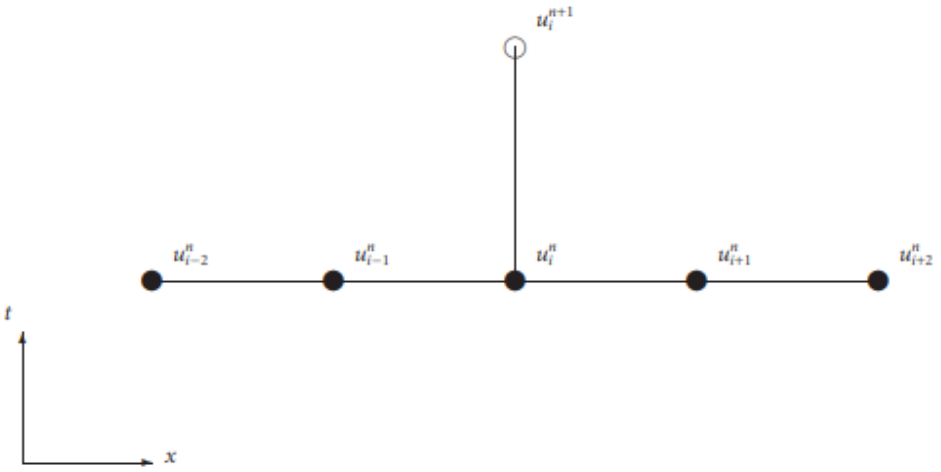


Figure 14 – Computation schematic for fourth order upwind scheme

The stability criteria for the following approximation is given by **equation 27**.

$$0 \le s \le \frac{1}{3} - \frac{1}{6}c^2 + \frac{1}{6}\sqrt{4 + 6c^4} \qquad (27)$$

Mehdi Dehghan (2004) had undergone a study comparing the effectiveness of upwind schemes in approximating the convection-diffusion transport problems. The study included the

second, third, and fourth order upwind schemes. The study concluded that the fourth upwind scheme is by far the most accurate. This should not come as a surprise since not only is the fourth upwind scheme approximated using high accuracy finite difference formulas, but it uses a higher order high accuracy formula than the third order. The second order upwind scheme used in the study is similar to the explicit lax-wenhoff technique.

Another study was made by Appadu (2015) in which the third and fourth order upwind schemes were compared against the non-standard finite difference scheme (NSFD). The study observed that the NSFD was superior to both the third and fourth upwind schemes. This was for both course and fine grids and also low and high Reynolds numbers.

The ability for the NSFD formula to handle high Reynolds numbers is very important to this work. Diffusion in liquids in general are at 10^-6 order of magnitude (i.e. Re = 1,000,000) due to the small velocities of the dispersed droplets. Diffusion can even be lower if the dispersed droplets are in a highly viscous continuous phase similar to crude oil.

High Reynolds numbers can cause difficulties in explicit standard schemes. Furthermore, settling velocities in liquids are at a 10^-4 order of magnitude which can cause additional problems when using explicit standard formulas. Therefore, NSFD and the implicit crank-nelson present two of the most effective methods to be applied to emulsion droplet size distribution analysis.

It is important to note that using a higher order approximation is not always preferable since techniques that are sometimes more accurate also require more calculations and as a result more CPU time. And if a model will track millions of droplets, a few additional calculations per step can compound to very large additional calculations for the entire model. Due to this limitation it is important to define the error tolerance and accepted CPU time for a given model.

2.2.5 Nonstandard finite difference scheme

As mentioned before, the one-dimensional Advection-Diffusion equation is generally given by **equation 28**.

$$\frac{\partial n_k}{\partial t} = \frac{\partial}{\partial z}\left(D_z \frac{\partial n_k}{\partial z}\right) - \frac{\partial (w_k n_k)}{\partial z} \qquad (28)$$

This equation has three possible outcomes or sub-equations depending upon the presence or absence of the transport phenomena being studied. The first outcome is achieved when there is negligible diffusion in a fluid. In this case the ADE equation reduces to **equation 29**.

$$\frac{\partial n_k}{\partial t} + w_k \frac{\partial n_k}{\partial z} = 0 \qquad (29)$$

The second outcome occurs when viscous forces in a fluid cancel out gravitational forces resulting in negligible settling of the dispersed population. The ADE equation is this case reduces to **equation 30**.

$$\frac{\partial n_k}{\partial t} = D_z \frac{\partial^2 n_k}{\partial z^2} \qquad (30)$$

The third and final outcome is when there is no change in the concentration of a population as a function of time is a given volume of space. However, within that volume of space, the dispersed population is free to move – free to settle and free to diffuse. In this case the ADE equation reduces to **equation 31**.

$$w_k \frac{\partial n_k}{\partial z} = D_z \frac{\partial^2 n_k}{\partial z^2} \qquad (31)$$

Equations 31 and 32 have exact known finite difference formulas as seen in **equations 32** and **33**, respectively.

$$\frac{n_k^{n+1} - n_k^n}{k} + w_k \frac{n_k^n - n_{k-1}^n}{h} = 0 \qquad (32)$$

And

$$\frac{n_k^n - n_{k-1}^n}{h} = D_z \left(\frac{n_{k+1}^n - n_k^n + n_{k-1}^n}{D_z h (e^{(\frac{w_k h}{D_z})} - 1)} \right) \quad (33)$$

By combining the above two equations and preforming minor simplification to the resulting equation, the non-standard finite difference formula is defined and is generally given as **equation 34.**

$$n_k^{n+1} = w_k n_{k+1}^n + (1 - \alpha_1 - 2\beta_1) n_k^n + (\alpha_1 + \beta_1) n_{k-1}^n \quad (34)$$

Where, $\alpha_1 = w_k \, k/h$ and $\beta_1 = \alpha_1 / e^{\left(\frac{w_k h}{D_z}\right)} - 1$

The stability criteria for the following approximation is given by **equation 35**.

$$\alpha_1{}^2 \leq \alpha_1 + 2\beta_1 \leq 1 \quad (35)$$

2.2.6 Optimization of numerical parameters of k and h

Now that the best numerical models have been chosen and developed for the advection diffusion equation, the next step is to find the most optimum numerical parameters k and h that would minimize the error. The aim of this section is to find the optimum k value for a given spatial discretization h. h will be taken as 0.00254cm for this problem. This in turn will lead to a 3000-layer discretized test tube. This gives the ability to sample the smallest of changes within a cylindrical test tube both in space and as we will see later we are able to sample the volume fraction in small time increments as well. But lower discretization are possible if such a high resolution is not required.

Non-standard finite difference formula

The numerical error for the NSFD technique is quantified using two approaches. The purpose of the two approaches is to minimize the dissipation and dispersion errors normally encountered in numerical modeling. Each technique approaches the problem differently but both methods yield the same exact graphical result as we'll see later in this chapter. The first approach

is by Tam and shen [22] where they go about optimizing the coefficients α_1 & β_1 of the NSFD formula. As for the second approach, Bogey and Bailly [20] minimize the relative difference between the exact wavenumber and the numerical wavenumber. These approaches are defined as **equations 36** and **37**, respectively.

$$\text{IETAM} = \int_0^{1.1} (RPE_{NSFD} - 1)^2 d\omega \quad (36)$$

$$\text{IEBOGEY} = \int_0^{1.1} |RPE_{NSFD} - 1| \, d\omega \quad (37)$$

RPE stands for the relative phase error which is defined in terms ofα_1, the phase angle, ω and the coefficients of the amplification factor, $\Re(\xi_{NSFD})$ and $\Im(\xi_{NSFD})$ that will be defined later in this section. RPE is defined generally as **equation 38**.

$$RPE_{NSFD} = \frac{1}{\alpha_1 \omega} tan^{-1}\left(\frac{\Im(\xi_{NSFD})}{\Re(\xi_{NSFD})}\right) \quad (38)$$

Since the RPE is defined in terms of the coeffiecnets of the amplification factor, it's important to furthur explain what the amplification factor signifies. The amplification factor is a quanity that reflects the stability of a numerical scheme. It's a measure of the whether the numerical disturbance grows or dampends. If the amplification factor is equal to 1, no change in disturbance is to be expected and the scheme is said to be stable. However, if the amplification factor is greater than or less than 1, this indicates an unstable state. The amplification factor is defined as **equation 39**.

$$\xi_{NSFD} = \Re(\xi_{NSFD}) - \Im(\xi_{NSFD}) * I \quad (39)$$

Where $\Re(\xi_{NSFD})$ and $\Im(\xi_{NSFD})$ are the coefficients and $I = \sqrt{-1}$. The coefficients are defined as **equations 40** and **41**, respectively.

$$\Re(\xi_{NSFD}) = 1 + (\alpha_1 + 2\beta_1) * (cos(\omega) - 1) \quad (40)$$

$$\Im(\xi_{NSFD}) = \alpha_1 * sin(\omega) \quad (41)$$

By substituting back $\Re(\xi_{NSFD})$ & $\Im(\xi_{NSFD})$into the RPE equation, we obtain a more expanded form of the RPE equation as seen in **equation 42**.

$$RPE_{NSFD} = \frac{1}{\alpha_1 * \omega} tan^{-1}\left[\frac{\alpha_1 * sin(\omega)}{1+(\alpha_1+2\beta_1)*(cos(\omega)-1)}\right] \quad (42)$$

Furthermore, by substituting α_1 & $\beta_1$previosuly defined in the NSFD formulation, we obtain the final fully expanded form of the RPE defined in terms of known parameters k, h, D_z and w as seen in **equation 43**.

$$RPE_{NSFD} = \frac{h}{w_k k * \omega} tan^{-1}\left[\frac{\left(\frac{\overline{w}}{h} k\, sin(\omega)\right)}{1+\left[\frac{w}{h}+2\frac{w}{h\, e^{\left(\frac{w_k h}{D_z}\right)}-1}\right]k(cos(\omega)-1)}\right] \quad (43)$$

Given that RPE is now fully developed, it can now be substituted back into the two Error integration formulas defined earlier in this section. The modified equation is given by **equations 44** and **45**, respectively.

$$\text{IETAM} = \int_0^{1.1}\left[\frac{h}{w_k k * \omega} tan^{-1}\left(\frac{\left(\frac{\overline{w}}{h} k\, sin(\omega)\right)}{1+\left[\frac{w}{h}+2\frac{w}{h\, e^{\left(\frac{w_k h}{D_z}\right)}-1}\right]k(cos(\omega)-1)}\right)-1\right]^2 d\omega \quad (44)$$

$$\text{IEBOGEY} = \int_0^{1.1}\left|\frac{h}{w_k k * \omega} tan^{-1}\left(\frac{\left(\frac{\overline{w}}{h} k\, sin(\omega)\right)}{1+\left[\frac{w}{h}+2\frac{w}{h\, e^{\left(\frac{w_k h}{D_z}\right)}-1}\right]k(cos(\omega)-1)}\right)-1\right| d\omega \quad (45)$$

From the following equations it can be clearly shown that for a given combination of numerical parameters k and h, a corresponding error can be quantified. It is important to note that the spatial step size is kept constant and only the time step with be changed. Therefore, the above

integrals will be evaluated at a range of k values for a constant spatial discretization. The integrals are calculated for k values ranging from 0 to 250 seconds.

The IETAM and IEBOGEY are complex integration formula without an analytical solution. To evaluate these integrals a numerical integration technique was used. The numerical technique chosen was Simpson's $\frac{1}{3}$ rule due to its effectiveness to find a very accurate integral in a very efficient manner. **Figures 15** and **16** show the IETAM and IEBOGEY error equations for k values ranging from 0 to 250 seconds

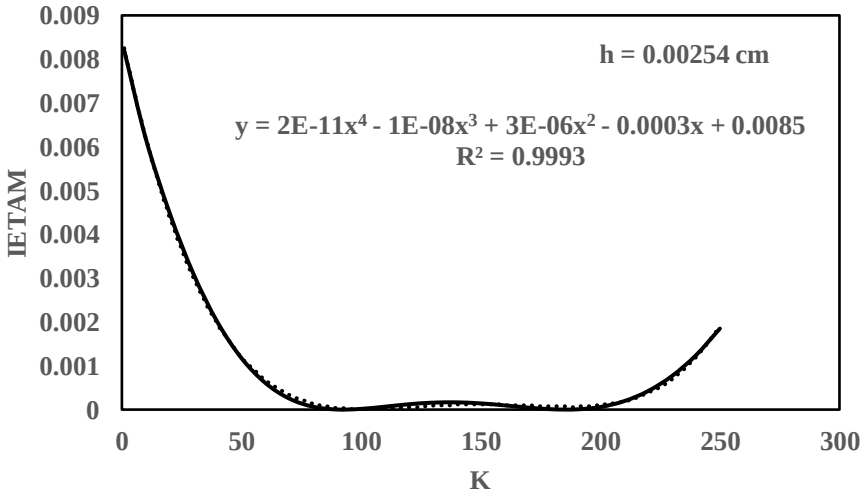


Figure 15 – Plot for IETAM versus k for NSFD when h = 0.00254 cm

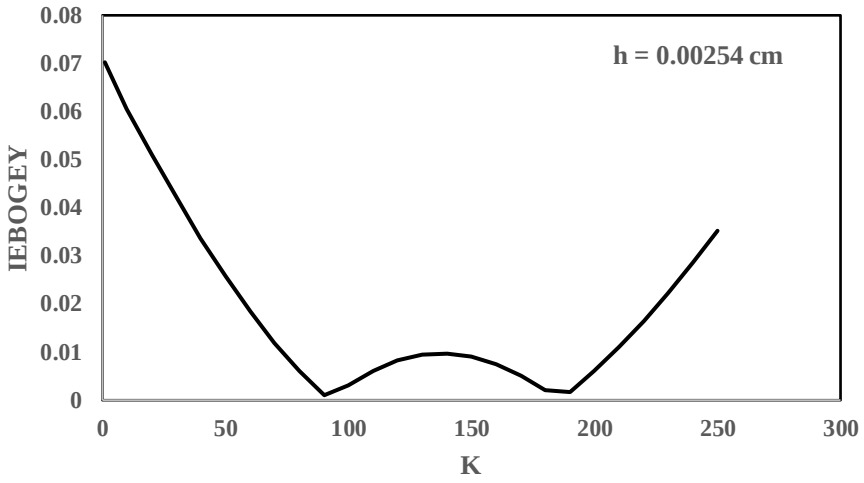


Figure 16 – Plot for IEBODEY versus k for NSFD when h = 0.00254 cm

By performing regression analysis for both the IETAM and IEBOGEY models followed by the use of a root finding technique capable of handling multiple roots, the k values that would minimize the error to close to zero for both IETAM and IEBOGEY were found to be

$$k = 90$$

$$k = 190$$

Since both values of 90 and 190s are capable of minimizing the error to the same degree, the lager time discretization (i.e. k = 190) was chosen for the purpose of minimizing the computational time.

Crank Nelson

Finding the optimum numerical parameters of the crank-Nelson is almost identical to that of the NSFD scheme. The difference is in the development of the RPE as seen in **equations 46** and **47**. The RPE equation can be generated using the same steps outlined above.

$$\text{IETAM} = \int_0^{1.1} (RPE_{CN} - 1)^2 d\omega \qquad (46)$$

$$\text{IEBOGEY} = \int_0^{1.1} |RPE_{CN} - 1| \, d\omega \qquad (47)$$

After the development of the RPE term specific to the crank-nelson in addition to the application of numerical integration techniques, the error vs. time step curves can be generated for both the IETAM and IEBOGEY error analysis. **Figure 17** and **18** show the results of the IETAM and IEBOGEY integrals evaluated for a spatial step size of 0.254 cm and a k range of 0 to 1800. As mentioned before, the NSFD has a large stability criterion which allowed the ability to discretize to up to 3000 layers. But with the crank- nelson, to maintain numerical stability, discretization of the space was limited to 300 layers of the 3 in test tube.

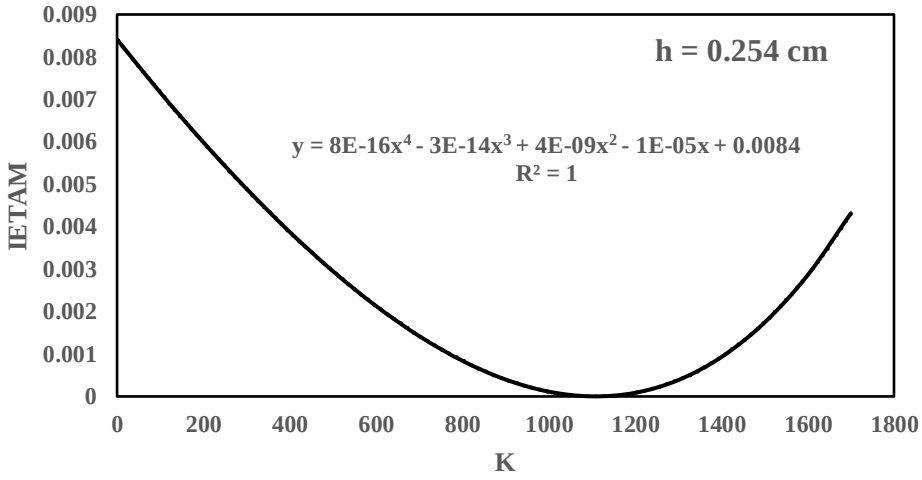


Figure 17 – Plot for IETAM versus k for Crank-Nelson when h = 0.254 cm

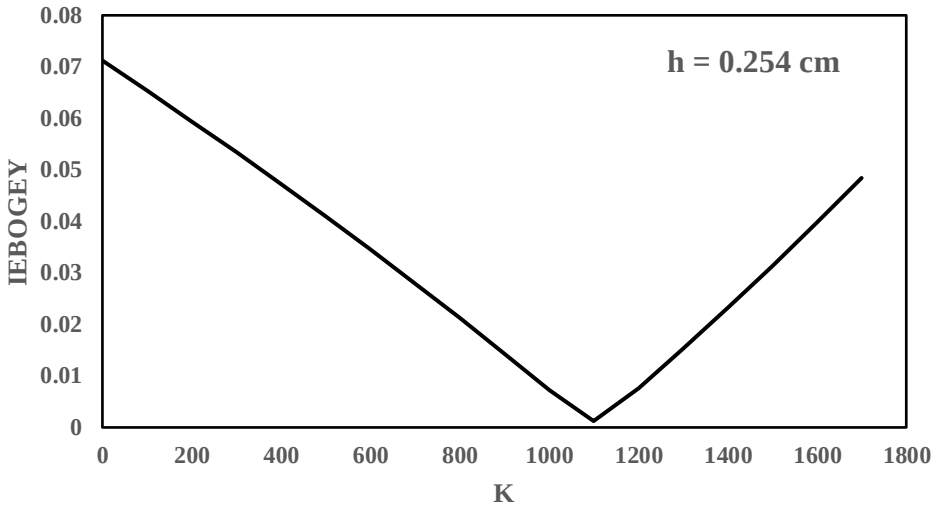


Figure 18 – Plot for IEBODEY versus k for Crank-Nelson when h = 0.254 cm

By performing regression analysis for both the IETAM and IEBOGEY models followed by the use of a root finding, the k value that would minimize the error to close to zero for both IETAM and IEBOGEY were found to be

$$k = 1100s$$

It can be observed that the volume fraction can be sampled accurately every 1100 seconds as oppose to every 90 or 190 seconds as with the NSFW. This is conformation of the differences in stability criterion between the techniques used to solve the ADE equation. If high resolution is highly desired in both space and time then the perfect choice would be the NSFW but if moderate resolution is desired then the crank-nelson would be a good choice.

2.3 Source Term

Droplets within a fluid do not only settle and diffuse but are always in constant motion and as a result are in constant collisions with other droplets. Furthermore, some droplets do not only collide but rather merge and coalesce to form larger droplets.

If we track a droplet k within a fluid where coalescence takes place, two phenomena can be observed. The first phenomenon is the death of droplet k. This occurs when droplet k collides and merges with another droplet. The second phenomenon is the birth of a droplet k. This occurs when two droplets that have a combined volume equal to k collide and merge together.

The net increase or decrease of droplet k is represented by theta which is given as **equation 48**.

$$\Theta_k = \frac{1}{2}\int_0^v \beta(v-v',v')n(v-v',z,t)n(v',z,t)dv' - \int_0^\infty \beta(v,v')n(v',z,t)n(v',z,t)dv' \quad (48)$$

Where $v\ and\ v'$ are droplet volumes of droplets colliding and β is the collision frequency.

The first term represents the number of droplet k that were created and the second term represents the number that died.

2.3.1 Fixed Pivot Technique

As with many differentiation and integration formulas, an exact solution is not always readily available. An exact solution can be either difficult or impossible to obtain. Kumar and Ramkrishana (2008) developed a technique called the fixed Pivot Technique to solve for coalescence, breakage, growth and nucleation numerically. However, in this work only the process of coalescence will be studied. The change in concentration of a droplet k due to coalescence using the fixed pivot technique is defined as seen in **equation 49**.

$$\theta_k = \underbrace{\frac{1}{2}\sum_{i,j}^{i\geq j} \eta_k \beta_{i,j} n_i n_j}_{Birth\ Term} - \underbrace{\sum_{i=1}^{N_p} \beta_{i,k} n_k n_i}_{Death\ Term} \quad (49)$$

Where $\beta_{i,j} and\ \beta_{i,k}$ are collision frequencies, $n_i n_j and\ n_k$ are the concentraton of droplets i, j and k respectivily. And η_k is the contribution factor.

The birth term models the collisons between all the droplets excluding droplet k and how many of these collisons will results in the birth of a k droplet. As for the death term, this term models the collsions of k droplets with all other droplets and how many of these collsions will merge and result in the death of droplet k.

There are two types of collisions that can occur within a fluid, collisions due to Brownian motion and collisions due to differential sedimentation. The collision frequency term incorporates both phenomena for collisions between i and k, and i and j and are both defined as **equations 50** and **51**.

$$\beta_{i,k} = K_2(\beta_{Br(i,k)} + \beta_{DS(i,k)}) \quad (50)$$

$$\beta_{i,j} = K_2(\beta_{Br(i,j)} + \beta_{DS(i,j)}) \quad (51)$$

K_2 is the coalescence coefficient. It is a fraction that quantifies how many of the collisions undergo coalescence. This term is commonly obtained experimentally.

Brownian type collisions occur simply due to the random motion of droplets within a fluid. These collisions are a function of Boltzmann constant, temperature, viscosity of continuous phase and diameter of colliding droplets. Collison frequency due to Brownian motion for collisions between i and k, and i and j are defined as **equations 52** and **53**.

$$\beta_{Br(i,k)} = \frac{2KT}{\mu_c}\left(\frac{1}{d_i} + \frac{1}{d_k}\right)(d_i + d_k) \quad (52)$$

$$\beta_{Br(i,j)} = \frac{2KT}{\mu_c}\left(\frac{1}{d_i} + \frac{1}{d_j}\right)(d_i + d_j) \quad (53)$$

As for differential sedimentation, collisions of this type are a result of differences in settling velocities and are given by **equations 54** and **55**.

$$\beta_{DS(i,k)} = \frac{\pi}{4}(d_i + d_k)^2|w_i - w_k| \quad (54)$$

$$\beta_{DS(i,j)} = \frac{\pi}{4}\left(d_i + d_j\right)^2\left|w_i - w_j\right| \quad (55)$$

Where

$$w_k = \frac{K_1(\rho_c - \rho_d) * d_k^2 * g}{18\mu_c} \quad \text{(Same for i \& j)}$$

In the birth term, after the number of collisions have been quantified and the fraction of these collisions that will coalesce, the fraction that contributes to the k droplet needs to be quantified. This is done through the contribution term η_k which is defined as **equation 56**.

$$\eta_k = \begin{cases} \frac{v_{k+1}+v}{v_{k+1}-v_k} & if: v_k \leq v \leq v_{k+1} \\ \frac{v-v_{k-1}}{v_k-v_{k-1}}, & if: v_{k-1} \leq v \leq v_k \end{cases} \quad (56)$$

Where v is the combined voulme of the droplets colliding. v_k is the volume of droplet k and v_{k+1} is a volume a step size away from v_k defined as $v_{k+1} = qv_k$ where q is geometric descritization parameter. The descritiztion factor contols the courseness and fineness of the grid being used in the numerical solution. The geometric facotor that best optimizes this technique was taken to be q = 1.5.

CHAPTER 3: CODE DEVELOPMENT AND MODEL SIMULATION

3.1 Programming & Simulation of the PBE Equation

Numerical modeling is a very powerful branch of mathematics. It is capable to solve very complex problems using very simple techniques. It is a branch that can differentiate and integrate without the need to use any of the sometimes-complex differentiation and integration rules. It can create best fit models with just knowing a data set. It can create functions where the only thing we know is its derivative and a condition. And best of all, it can generate approximations that have such a low percentage error that they are as good as the true value.

But, there is a limitation to numerical methods. They depend on iterative calculations which can sometimes be impossible to do by hand not to mention tedious. This is where programming comes in and will be the focus of this section. In this section, a step by step outline will be made for creating the population balance equation simulator using the numerical solutions outlined above.

The series of programs described in this section were developed using VBA programming language and are available in the appendix.

3.1.1 Declaration and dimensionalization of vectors and arrays

The first step of the code involves the definition of all the vectors and matrices used. The vectors in this code were defined for diameter, sum and volume of droplets, and NSFD coefficients.

As for the matrices, fourteen matrices were defined and they include and not limited to, droplet concentrations in each of the discretized thickness, droplet concentration is the final 3

layers, collision frequency for ij and ik and source term calculation for each layer. A complete list can be found with the accompanied code.

3.1.2 Data importation, definition of coefficients and boundary conditions

The second step involves importing the diameter data and fluid properties. **Table 1** provides a summary of the fluid properties used. These properties involve the settling constant, fluid viscosities and diffusivity, in addition to the numerical and collision parameters such as h and k. These properties can be changed and as a result a variety of emulsion scenarios can be modeled. The imported diameter data in this model is synthesized. They should ideally be provided from DSD experimental measurement with their accompanying concentrations.

Inputs	
K1	0.081
Rhoc (g/cc)	0.88
Rhod (g/cc)	1.072
Muoc (g/cm*s)	0.25
g (cm/s2)	980.667
Dz (cm2/s)	1E-05
k (sec)	200
h (cm)	2.54
Days Simulated	0.01875
K Boltzman Constant	1.381E-16
T (K)	293
K2 (Binary	0.025
q	1.5

Table 1 – Fluid Properties

After all properties have been imported, NSFD and source term coefficients can be calculated. Coefficients α_1 and $\beta_1$depend on w_k droplet velocity which means they need to be calculating for each droplet size. Therefore, the coefficients are not constant for both the NSFD and source term.

Since the distribution used starts at the minimum possible droplet diameter and ends at the maximum possible droplet diameter, the boundary condition for this model are set to zero.

3.1.3 Generate lognormal DSD and calculate initial distribution

The third step involved generating the lognormal distribution of the initial concentration of the dispersed droplets. Ideally the concentration should be provided experimentally but in this model they will be calculated using a log normal distribution. Depending on how emulsions were prepared and mixed, acid in oil emulsions seem to follow a lognormal droplet size distribution which would simply mean that the average droplet size is very small. This observation might explain the high stability of acid in crude oil emulsions and the sold like layer formed in laboratory conditions.

Given that a density distribution function is calculated, the next step is to convert it into a number distribution function using knowledge of the test tube height and diameter, acid concentration and number and diameter of droplets. The height and diameter of the tube were taken at 3 inch and 1 inch respectively and acid concentration at 30%. **Table 2** shows an example list of droplet sizes that will be tracked within an emulsion sample.

Inputs	
d(mm)	d(cm)
0	0
0.3	0.00003
0.4	0.00004
0.5	0.00005
0.6	0.00006
0.7	0.00007
1	0.0001
2	0.0002
3	0.0003
4	0.0004
5	0.0005
6	0.0006
7	0.0007
8	0.0008
9	0.0009
10	0.001
11	0.0011
12	0.0012
15	0.0015
17	0.0017
20	0.002

Table 2 – Example list of droplet sizes tracked within an emulsion sample

Since the test tube was assumed to be equally mixed, only the number distribution of the initial layer needs to be calculated and the remaining layers were initialized at the same concentration. It is important to note that the initial discretization was 3000 layers following the optimization results but this model is still a 3 layer model to replicate Scarborough's work. **Figure 19** shows the input distribution function at a mean of 1.5 microns and how it compares to literature distributions.

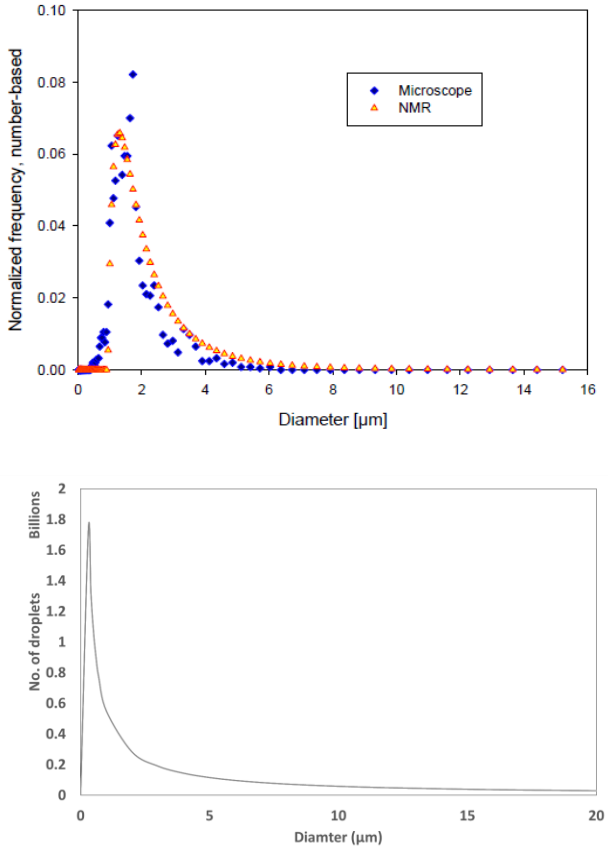


Figure 19 – Input distribution function and how it compares to literature distributions. (Opedal 2009)

3.1.4 Material balance and error calculation

The next step is to calculate the total volume in each layer and entire test tube given the input distribution. This step gives the opportunity to track not only the change in distribution per

layer but also to ensure that total volume of acid before simulation is the same as total volume after simulation and to maintain volume balance throughout the simulation process.

3.1.5 Source term coefficient calculation

The final step before the application of the population balance equation is the calculation of the source term coefficients $\beta_{i,j}$, $\beta_{i,k}$ and η_k. The contribuation term η_k was calculted using Kumar & Rankrishna's fixed piviot technique discribed earlier.

3.1.6 Application of the population balance equation

Given that the initial droplet size distribution and coefficients of both the NSFD technique (or any chosen technique that best fits the problem at hand) and source term are now calculated, the population balance equation can now be applied to track the changes due to advection, diffusion and aggregation.

The advection- diffusion equation models mainly the loss of droplets since the driving transport phenomena is advection. However, this loss is another layer's gain. From layer 2 to the second to last layer, these layers both lose and gain. For this reason, an adjustment needs to be made to each layer to account for the amount of gain the layer experienced from the layer above. **Figure 20** shows a visual representation of this phenomena. Therefore, no application of the PBE will be made to the top and bottom layers.

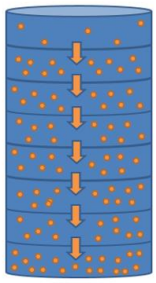


Figure 20 – An illustration showing that the loss of one layer is the gain of another

In addition to the adjustment due to advection, an adjustment due to aggregation has to be made for each time step. The source term will be either a negative number or a positive number. Positive indicates birth of droplet k and negative will indicate the number of droplet k that have died.

The application of the PBE equation will be applied for the 20 droplets in each of the 3000 layers for each of the 190 sec time steps. For experimental comparisons, the final time of the simulation was taken to be 5 days. This therefore will lead to approximately 136.5 Million calculations to generate the final droplet size distribution for each of the 3000 layers.

The 3000 layers are then combined to only three layers by combining the DSD of the top 1000 layers, middle 1000 layers and bottom 1000 Layers. Furthermore, calculating the area under the curve of each of the three DSD gives the volume fraction in the top, next to bottom and bottom layers. The code has been setup in such a way that the volume fraction is outputted for each of the 190 second intervals so that the change of volume fraction for the top, next to bottom and bottom layer can be tracked with time.

CHAPTER 4: RESULTS & DISCUSSIONS

4.1 Effect of Initial Statistical Parameters on Model Evolution

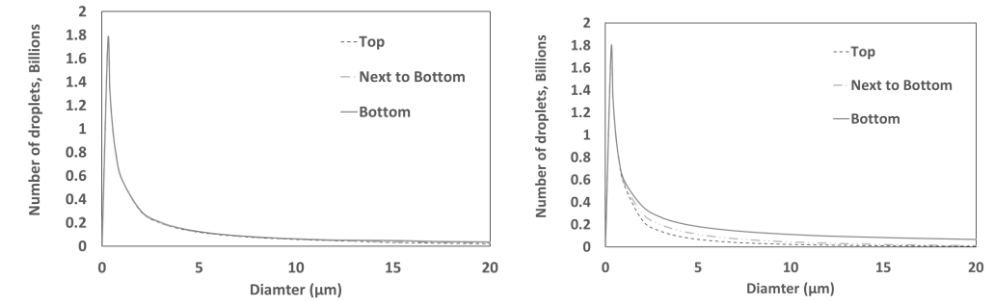
The first objective of the simulated model was to test the sensitivity to changes in mean and standard deviation. The sensitivity to the mean was studied using 1.5, 5, 10 and 15 microns at a constant standard deviation of 5 microns. The sensitivity to standard deviation was tested using 5, 10 and 15 microns at each mean.

4.1.1 Effect of Distribution Mean

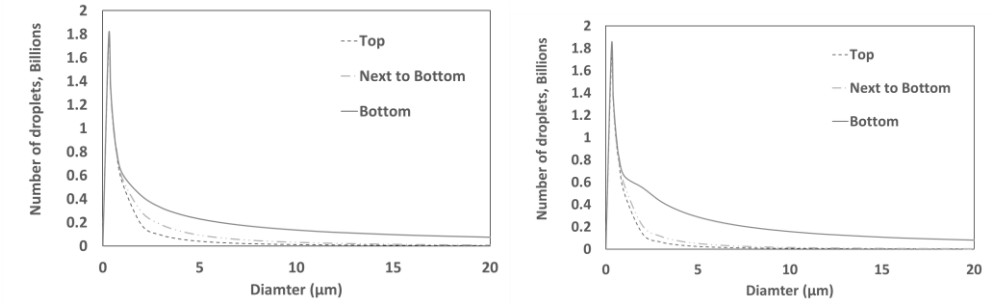
Figure 21 shows DSD evolution for mean of 1.5 microns after 5 days, 3 months, 6 months and 1 year. In the initial 5 days, slight change occurred. As seen in **Figure 22**, the top layer's acid fraction reduced to 24.2% from 30%, next to bottom layer increased slightly at 30.4% and the bottom layer increase to 35.4%.

After 3 months, the top, next to bottom and bottom layers had acid fractions of 8.2%, 17.5% and 64.3% respectively. After 6 months, the top, next to bottom and bottom layers had acid fractions of 4.3%, 11.7% and 73.98% respectively.

The change in the first 3 months is considerable compared to the second 3 months since in the initial months the larger droplets are settling and diffusing at a much faster rate. As from month 3 to month 6, the top layer and next to bottom layer had majority smaller droplets and thus the settling and diffusing speed is much slower. This effect can also be seen between the acid fractions of month 6 and 1 year.

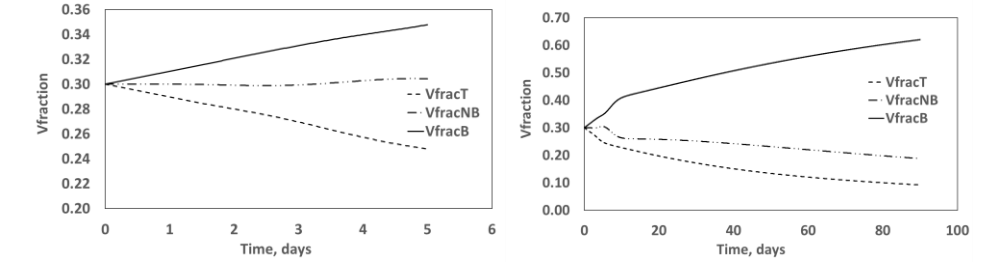


a)5 Days **b)3 month**



c)6 month **d)1 Year**

Figure 21 – Change is DSD with mean of 1.5 and standard deviation of 5 for 5 days, 3 months, 6 months and 1 year.



a)5 Days **b)3 month**

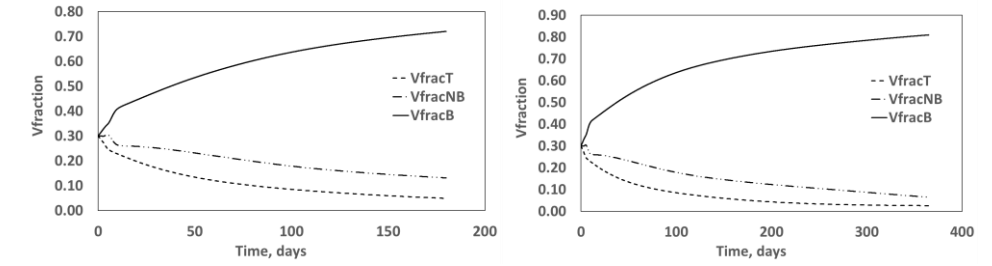


Figure 22 – Change is Acid Fraction for DSD with mean of 1.5 and standard deviation of 5 for 5 days, 3 months, 6 months and 1 year.

By increasing the mean from 1.5 to 5 µm a much different result is obtained. The top, next to bottom and bottom layers after only 5 days had acid fractions of 2.2%, 4.26% and 83.5% respectively as oppose to 24.2%, 30.4% and 35.4% for a mean of 1.5 µm. This is a significant change and implies the high sensitivity of the acid fraction to the mean of the initial droplet size distribution. **Table 3** and **Figure 23** show a summary of acid fractions for mean of 5, 10 and 15 µm for top, next to bottom and bottom layers.

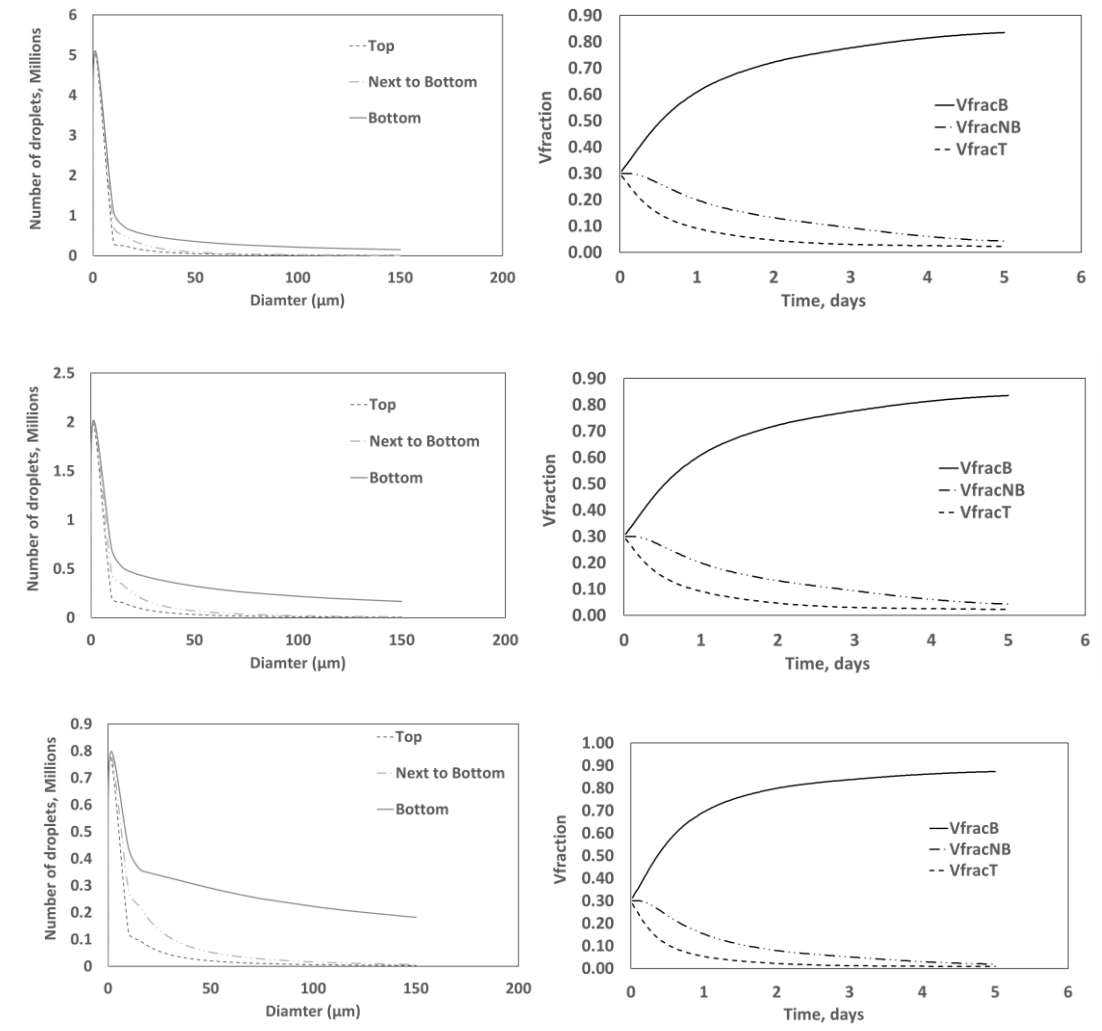


Figure 23 – Change is Acid Fraction for DSD with means of 5, 10 and 15 and standard deviation of 5 at 5 days

Mean, µm	1.5	5	10	15
	Acid Fraction, %			
Top	24.2	2.2	1.36	0.82
Next to Bottom	30.4	4.26	2.83	1.87
Bottom	35.4	83.5	85.82	87.31

Table 3 – Acid fraction as a function of initial DSD mean

4.1.2 Effect of Standard deviation on model evolution

By keeping the mean constant at 5 µm and changing the standard deviation, an interesting observation is made. The acid fraction of the top, next to bottom and bottom layers do not vary very much. The change is not statistically significant. It seems the acid fraction is not sensitive to changes to standard deviation as with droplet mean size. Or perhaps, changes in standard deviation cannot be detected through this type of model. The top layer for example only changes by 0.07% from an increase in standard deviation from 10 to 15 µm as seen in **Table 4**. This insensitivity might also be attributed to the input distribution being a logarithmic distribution. The standard deviation might start to have an effect with other distribution types or with experimental input.

Std, µm	5	10	15
	Acid Fraction, %		
Top	2.24	2.5	2.57
Next to Bottom	4.26	4.6	4.67
Bottom	83.5	82.89	82.77

Table 4 – Acid fraction as a function of initial DSD standard deviation

4.2 DSD Statistical Prediction compared to Experimental Measurements

The main purpose of this work was to identify the processes that govern acid in crude oil emulsions. **Figure 24** shows the evolution of acid fraction in the bottom layer as a function of time for a 30% acid/oil sample. The first 24 hours yielded an average experimental measurement of 29%. On day 2 the acid fraction in the bottom layer increase to 54% and ended up to a final concentration of 75% on day 3.

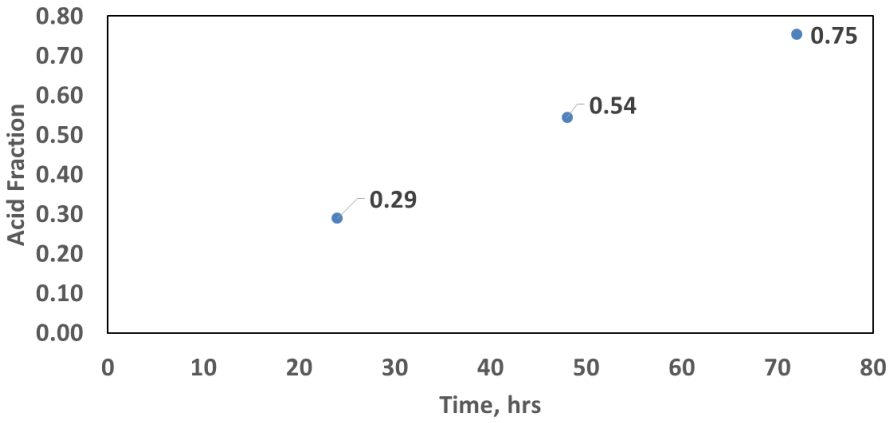


Figure 24 – Change is Acid Fraction for bottom layer for a 30% acid/crude oil emulsion

If the model truly captures the processes in the acid in crude oil emulsion through modeling advection, diffusion and coalsacance, we would be able to predict the initial mean droplet distribution of the emulsion at any given time. And it should be constant for all times. This can be investigated from the previously simulated data for means to generate a mean droplet size vs. acid fraction. And since standard deviation yielded no effect on the volume fraction, this analysis was performed using a constant standard deviation of 5 µm.

Figure 25 shows a regression model of mean versus acid fraction at day 1 for means of 1.5, 5, 10 and 15 µm. The regression model that yeilded the best fit was an expenential model with a constant of 0.2545 and a population growth rate of 5.2418. According to the regression model,

to achieve an acid fraction of 29% at day 1 as in the experimental data, it would require a intial mean droplet size of 1.24 microns. In other words, to achieve no significant change in acid fraction the mean droplet size is required to be very small that neglectable settling and diffusing are occuring. Furthmore, to achieve an acid fraction of 54% at day 2, this requires an intial mean droplet size of 4.86 microns as seen in **Figure 26** using a exponential regression model with a constant of 0.2303 and a population growth rate of 5.6071. And to Achieve a 75% acid fraction at day 3, this would require an initial mean droplet size close to 10 micros. It is important to note that predictions all assume the predomainate processes in the emulsion are settling, diffusion and coalsacance.

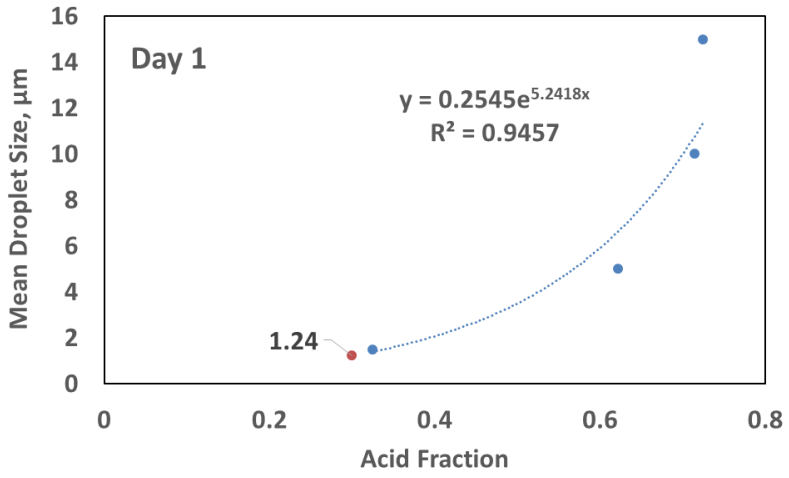


Figure 25 – Change in mean as a function of acid fraction for a 30% acid in crude oil emulsion

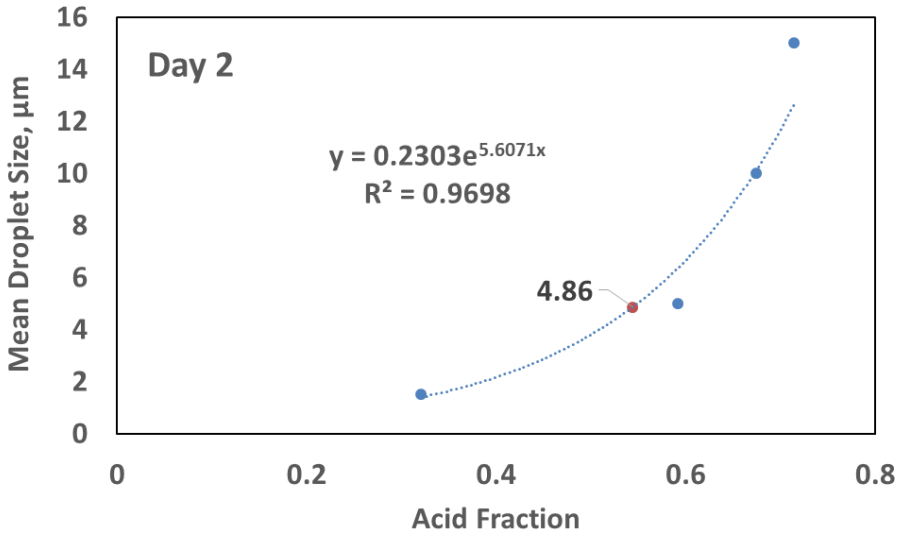


Figure 26 – Change in mean as a function of acid fraction for a 30% acid in crude oil emulsion

Figure 27 shows clearly the comparison between the model data versus experimental data. It can be observed that at day 1 the acid/oil emulsion behaves as an emulsion with an initial mean droplet size distribution of 1.24 microns. But at day 2, the emulsion starts behaving as an emulsion with an initial mean droplet size distribution close to 5 microns. And at day 3, its behavior shifts to a distribution with an initial mean of close to 10 microns.

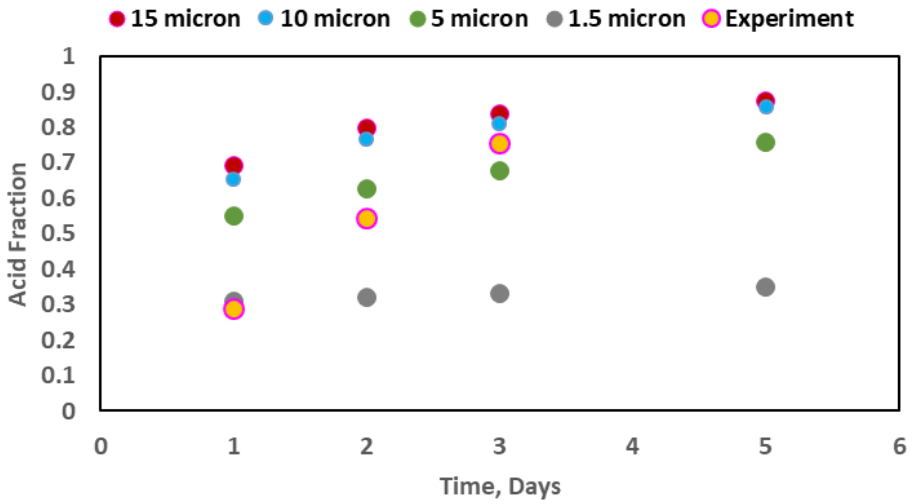


Figure 27 – Acid Fraction as of function of time for a 30% Acid oil emulsion for means of 1.5, 5, 10 and 15 microns generated by PBE model compared with experimental results

A hypothesis that might explain this behavior is that acid droplets might be sticking along the interface and not merging and as a result end up behaving as a larger distribution without the initial distribution changing at all. From the observed experimental data, this process of fusing might be continuing to later days in the emulsion life. This is evidence that there exists strong inter-droplet interactions and furthermore, this might also be evidence of strong interfacial forces. This model started by assuming that acid in crude oil emulsions can be described by modeling the processes of advection, diffusion and aggregation. But when comparing model data to experimental data it is quickly observed that there are additional processes that are occurring in acid/oil emulsions and we hypothesis that predominate mechanisms that are controlling this system are mainly settling and the fusing along the interface.

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The following conclusion are drawn from the population balance equation model development, performance and analysis:

- A phenomenological Mathematical model that utilizes the population balance equation was developed in this paper to identify the processes in acid in crude oil emulsions through modeling advection, diffusion and coalsacance.
- The mean droplet size was found to have an exponential relationship with acid fraction in the early days of an emulsion. The relationship is expected to level off and plateau as the system reaches equilibrium.
- Traditionally, upwind schemes are used to solve the advection-diffusion equation in the PBE models. In the proposed model, the Non Standard Finite difference approximate is used instead. It was superior for both course and fine grids and also low and high Reynolds numbers which makes it unconditionally stable. The ability for the NSFD formula to handle high Reynolds numbers was vital to this work since liquid diffusivity are at 10^-6 order of magnitude (i.e. Re = 1,000,000) due to the small velocities of the dispersed droplets.
- The acid fraction within each of the three layer was found to be highly sensitive to the mean droplet size distribution. For mean of 1.5 microns the acid fraction of the bottom layer increased from 30% to 74% in 1 year time. But with a mean of 5 microns, the acid fraction increased from 30% to 84% in just 5 days.

- The acid fraction was also found to not be effected significantly by changes in the standard deviation. This might be attributed to the input distribution being a logarithmic distribution.
- There is more to the acid oil emulsion evolution than just settling, diffusing and coalsacance. We hypothesis that there are strong inter-droplet interaction within the emulsion.

5.2 Recommendations

Future areas of investigation and study on the topic are listed below:

- Perform measurements on the properties of the acid-oil interface that can inform the droplet-droplet interactions in the model.
- Investigate the effect of coupling momentum conservation with the population balance equation on the droplet size distribution of the dispersed phase.
- Investigate the effect of factoring in interfacial coalescence as part of the crude oil destabilization process.
- Explore population balance equation output accuracy using input experimental DSD data coupled with experimentally derived DSD at five chosen future times.
- Explore the application of the PBE on large systems where momentum and binary coalescence have a much larger effect of the evolution of the droplet size distribution.
- Explore the expansion of the model to account for the remaining destabilization types such as flocculation, Phase inversion, Ostwald ripening and creaming.
- Explore converting the VBA code into a Fortran 90 code. This can be beneficial for much larger systems with a large simulation duration in terms of computation time.

- Study the effect of using different types of experimental input data such as data derived from NMR versus data derived from a 3D optical laser scanning microscope.

References

Almeida, T. C., Larentis, A. L., & Ferraz, H. C. (2015). Evaluation of the Stability of Concentrated Emulsions for Lemon Beverages Using Sequential Experimental Designs. *PLoS One*, 10.

Al-Yaari, M., Hussein, I. A., Al-Sarkhi, A., Abbad, M., & Chang, F. (2013). Pressure Drop Reduction of Stable Emulsions: Role of Aqueous Phase Salinity. *Saudi Arabia Section Technical Symposium and Exhibition.* Al-Khobar, Saudi Arabia: SPE.

Arirachakaran, S., Oglesby, D. K., Malinovsky, M. S., Shoharn, O., & Brill, J. P. (1989). An Analysis of Oil/Water Flow Phenomena in Horizontal Pipes. *SPE Production Operations Symposium* (pp. 155-170). Oklahoma City: SPE.

Bellary, S. A., Siddique, M. H., Samad, A., & Sangwai, J. S. (2017). Effects of crude oil-water emulsions at various water-cut on the performance of the centrifugal pump. *Int. J. Oil, Gas and Coal Technology*, 71-88.

Borja, H., & Castano, R. (1999). Production Optimization by Combined Artifical Lift Systems and Its Application in Two Colombian Fields. *Latin American and Carribbean Petroleum Engineering Conference*. Caracas: Society of Petroleum Engineers .

Brandal, O., Hanneseth, A. D., Hemmingsen, P. V., Sjoblom, J., Kim, S., Rodgers, R., & Marshall, A. (2006). Isolation and Characterization of Naphthenic Acids from a Metal Naphthenate Deposit: Molecular Properties at Oil-Water and Air-Water Interfaces. *Journal of Dispersion Science and Technology* , 295-305.

Brooks, B., & Richmond, H. (1991). Dynamic of Liquid-Liquid Phase Inversion Using Non-ionic Surfactants. *Colloid and Surfaces*, 131-148.

Cao, Y., Zhao, R., Zhang, L., Xu, Z., Jin, Z., Luo, L., & Zhao, S. (2012). Effect of electrolyte and temperature on interfacial tensions of alkylbenzene sulfonate solutions. *Energy Fuels*, 2175– 2181.

Chattopadhyay, A., Ghaicha, L., Shah, D., & Oh, S. (1992). Salt Effects on Monolayers and Their Contribution to Surface Viscosity . *The Journal of Physical Chemistry*, 6509-6513.

Daaou, M., & Bendedouch, D. (2012). Water pH and surfactant addtion effects on the stability of an Algerian crude oil emulsion. *Journal of Saudi Chemical Society*, 333-337.

Dalmazzone, C., Noik, C., & Komunjer, L. (2005). Mechanism of Crude-Oil/Water Interface Destabilization by Silicone Demulsifiers. *Society of Petroleum Engineers*.

Donato, M. (2016). *Experimental Study of Water-Oil Two-Phase Flow in an 8-Stage Electrical Submersible Pump*. Campinas: Universidade Estadual de Campinas.

Dresel, E., & Rose, A. (2010). Chemistry and Origin of Oil and Gas Well Brines in Western Pennsylvania. *Pennsylvania Geologoical Survey*.

Economides, M. J., Hill, D. A., Economides, E. C., & Zhu, D. (2013). *Petroleum Prodction Systems - Second Edition* . Westford: Prentice Hall.

Gafonova, O. V., & Yarranton, H. W. (2001). The Stabilization of Water-in-Hydrocarbon Emulsions by Asphaltenes and Resins. *Journal of Colloid and Interface Science*, 469-478.

Gomez, S., Mansi , M., & Fahes, M. (2018). Quantifying the Non-Monotonic Effect of Salinity on Water-in-Oil Emulsions towards a Better Understanding of Low-Salinity-Water/Oil/Rock Interactions. *Abu Dhabi International Petroleum Exhibition & Conference, 12-15 November, Abu Dhabi, UAE.* Abu Dhabi: Society of Petroleum Engineers.

Griffin, W. (1949). Classification of surface-active agents by HLB. *J. Soc. Cosmet, Chem*, 311-326.

Grondfos. (2018, October 22). *The Centrifugal Pump.* Retrieved from Grundfos: http://machining.grundfos.com

Henríquez, C. J. (2009). *W/O Emulsions: Formulation, Characterization and Destabilization.* Technischen Universität Cottbus zur Erlangung, Caracas.

ICI Americas Inc. (1984). *The HLB System: A Time-saving Guide to Emulsifier Selection.* Wilmington, Delaware: CHEMMUNIQUE.

Johnsen, E. E., & Ronningsen, H. P. (2003). Viscosity of 'live' water-in-crude-oil emulsions: experimental work and validation of correlations. *Journal of Petroleum Science and Engineering* , 23-36.

Jones, T. J., Neustadter, E. L., & Whittingham, K. P. (1978). Water-In-Crude Oil Emulsion Stability And Emulsion Destabilization By Chemical Demulsifiers. *Journal of Canadian Petroleum Technology*, 100-108.

Jones, T. J., Neustadter, E. L., & Whittingham, K. P. (1978). Water-in-crude oil emulsion stability and emulsion destabilization by chemical demulsifiers . *Jcpt*.

Karthikeyan, S., & Ranjith, P. (2007). Degradation Studies on Anionic and Non-Ionic Surfactants by Ozonation . *Journal of Industrial Pollution Control*, 37-42.

Kilpatrick, P. K. (2012, May 30). Water-in-Crude Oil Emulsion Stabilization: Review and Unanswered. *Energy & Feuls, 26*(7), pp 4017–4026. doi:10.1021/ef3003262

Kokal, S. L. (2005). Crude Oil Emulsions: A State-Of-The-Art Review. *Society of Petroleum Engineers*.

Kumari, A. (2016). *Brief Information About Non-Newtonian Fluids and Their Properties*. Retrieved from Buzzle: http://www.buzzle.com/articles/brief-information-about-non-newtonian-fluids.html

Mahmoodian, H., Fasih, M., Moosavi, M., Nalchi, M., Ghalyagizadeh, R., Farkhani, D., . . . Arzpayma, A. (2007). *Study of Emulsion Problem Due to Gaslift and Choose Optimum Gas Rate and Suitable Type and Rate of Demulsifier.* R. I. P. I. - E&P - P. V. T. Departament .

Malkin, A. Y., Rodionava, G., Simon, S., Iiyin, S. O., Arinina, M. P., & Kulichikhin, J. (2016). *Some Compositional Viscosity Correlations for Crude Oils from Russia and Norway* . Moscow: Institute of Petrochemical Synthesis, Russian Academy of Sciences.

Maodong, X., Jianzhong , J., & Xiaomei , P. (2018). Novel Oil-in-Water Emulsions Stabilised by Ionic Surfactant and Similarly Charged Nanoparticles at Very Low Concentrations. *Angewandte Chemie*, 7738-7742.

McLean, J., & Kilpatrick, P. (1997). Effects of asphlatene aggregation in model heptane-toluene mixtures on stability of water-in-oil emulsions. *Colloid Interface*, 23-34.

Mehana, M., & El-Monier, I. (2015). Numerical Investigation of the Osmatic Flow Impact on the Load Recovery and Early Well Performance. *Journal of Petroleum Engineering and Technology*, *5*(03).

Mehana, M., & Fahes, M. (2016, April). The Impact of the Geochemical Coupling on the Fate of Fracturing Fluid, Reservoir Characteristics and Early Well Performance in Shale Reservoirs. In *SPE Kingdom of Saudi Arabia Annual Technical Symposium and Exhibition*. Society of Petroleum Engineers.

Mehana, M., Al Salman, M., & Fahes, M. (2018). Impact of Salinity and Mineralogy on Slick Water Spontaneous Imbibition and Formation Strength in Shale. *Energy & fuels*, *32*(5), 5725-5735.

James Abraham, J., Mehana, M., & Fahes, M. (2018, June). Investigating the Uniformity of In-Situ Asphaltene Deposits in Sandstone Rocks. In *SPE Europec featured at 80th EAGE Conference and Exhibition*. Society of Petroleum Engineers.

Mehana, M., Fahes, M., & Huang, L. (2019). Asphaltene Aggregation in Oil and Gas Mixtures: Insights from Molecular Simulation. *Energy & Fuels*.

Mehana, M., Abraham, J., & Fahes, M. (2019). The impact of asphaltene deposition on fluid flow in sandstone. *Journal of Petroleum Science and Engineering*, *174*, 676-681.

Mohammed, S. (2009). *Characterization and Rheology of Water-in-Oil Emulsion from Deepwater Fields.* MSc. Thesis, Rice University, Department of Chemical Engineering, Houston.

Mooney, M. (1951). The Viscosity of a Concentrated Suspension of Spherical Particle. *Journal of Colloid Science*, *6*, pp 162-170.

Moradi, M., Alvarado, V., & Huzurbazar, S. (2011). Effect of Salinity on Water-in-Crude Oil Emulsion: Evaluation through Drop-Size Distribution Proxy. *Energy & Fuels*, 260-268.

Ngai, T., & Bon, S. A. (2014). *Particle-Stabilized Emulsions and Colloids: Formation and Applications.* Cambridge, UK: The Royal Society of Chemistry.

Pal, R. (1996). Effect of Droplet Size on the Rheology of Emulsions . *AIChE Journal*, 3181-3190.

Pal, R., Yan, Y., & Masliyah, J. (1992). *Emulsions Fundamentals and Application in Petroleum Industry.* Washington: Schramm L.L.

Pena, A. (2004). *Dynamic aspect of emulsion stability,PhD thesis.* Rice University Houston.

Perler, C., Onofrio, P., & Bombard, J. (2012). Study of the Cation and Salinity Effect on Electrocoalescence of Water/Crude Oil Emulsions. *Energy & Fuels*, 6914-6924.

Petroleum Engineering Handbook. (n.d.). In J. R. Fanchi, & L. Lake (Eds.). Society of Petroleum Engineers.

Pickering, S. U. (1907). CXCVI.—Emulsions. *Chemical Society, Transactions*, 2001-2021.

Richardson, E. (1933, April 28). Über die Viskosität von Emulsionen. *Kolloid-Zeitschrift, 65*(1), pp 32-37. doi:10.1007/BF01428855

Rocha, J., Baydak, E., Yarranton, H., Sztukowski, D., Ali-Marcano, V., Gong, L., . . . Zeng , H. (2016). Role of Aqueous Phase Chemistry, Interfacial Film Properties, and Surface Coverage in Stabilizing Water-in-Bitumen Emulsions. *Energy & Fuels*, 5240-5252.

Rønningsen, H. (1995). *Correlations for predicting viscosity.* SPE International Symposium on Oilfield Chemistry. Texas: Society of Petroleum Engineers.

Salager, J.-L. (2002). *Surfactant: Types and Uses.* Merida VENEZUELA: Laboratorio FIRP Escuela de INGENIERIA QUIMICA UNIVERSIDAD de Los ANDES.

Salahshoor, S., Fahes, M., & Teodoriu, C. (2017). A review on the effect of confinement on phase behavior in tight formations. *Journal of Natural Gas Science and Engineering*.

Saputelli, L. (1997). Combined Artifical Lift System - An innovative Apprach . *Latin American and Caribbean Petroleum Engineering* . Rio de Janeiro: Socierty of Petroleum Engineers .

Silset, A. (2008). *Emulsions (w/o and o/w) of Heavy Crude Oils Characterization, Stabilization,Destabilization and Produced Water Quality.* PhD Thesis, Norwegian University of Science and Technology, Trondheim,.

Strassner, J. E. (1968). Effect of pH on interfacial films and stability of crude oil–water emulsions. *Journal of Petroleum Technology*, 303-312.

Subramanian, D., May, N., & Firoozabadi, A. (2017). Functional Molecules and the Stability of Water-in-Crude Oil Emulsions. *Energy & Fuels*, 8967-8977.

Tambe , D., & Sharma, M. (1993). Factors Controlling the stability of Colloid-Stabilized Emulsions. *Colloids and Interface Science*, 244-253.

Umar, A. A., Saaid , B. M., Sulaimon , A. A., & Pilus, R. B. (2018). A review of petroleum emulsions and recent progress on water-in-crude oil emulsions stabilized by natural surfactants and solids. *Journal of Petroleum Science and Engineering*, 673-690.

Umebayashi, Y., Shin, M., Kanzaki, R., & Ishiguro, S. (2006). Metal Ion Complexation in Sufactant Solutions. *Encyclopedia of Surface and Colloid Science*, 3632-3642.

Yan, N., Gray, M. R., & Masliyah, J. H. (2001). On water-in-oil emulsions stabilized by fine solids. *Colloids and Surfaces*, 97-107.

Yang, F., & Tchoukov, P. (2014). Asphaltenes subfractions responsable for stabilizing water-in-crude oil emulsions. *Energy Fuels*, 6897-6904.

Zahid, A., Sandersen, S. B., Stenby, E. H., Solms, V., & Shapiro, A. (2011). Advanced Waterflooding in Chalk Reservoirs: Understanding of Underlying mechanisms. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 281-290.

Zahra, U. (2012). *Modelling Emulsions and Correlations.* TRC-Q PROJECT STATUS REPORT.

Zaki, N. (1997). Surfactant stabilized crude oil-in-water emulsions for pipeline transportation of viscous crude oils. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 19-25.

Zolfaghari, R., Fakhru-Razi, A., Abdullah, L., Elnashaie, S., & Pendashteh, A. (2016). Demulsification techniques of water-in-oil and oil-in-water emulsions in petroleum industry. *Separation and Purification Technology*, 377-407.

Appendix A: Mathematical Models

NSFD_SourceTerm Code

```
Sub NSFD_SOURCETERM()

n = InputBox("What are the number of diameters in the distribution?")
m = InputBox("How many hours would you like to run the simulation?") * 18
u = (7.62 / Cells(10, 6))
Cells(11, 6) = m / 960

'Declaration
Dim D() As Double: Dim Dm() As Double: Dim vd() As Double: Dim nold() As Double
Dim ni() As Double: Dim nf() As Double: Dim nmax() As Double: Dim nnew() As Double
Dim beta() As Double: Dim alphal() As Double: Dim betal() As Double: Dim theta() As Double
Dim mum() As Double

Dim xx() As Double: Dim mu() As Double: Dim bij() As Double: Dim bbrij() As Double
Dim bdsij() As Double: Dim bik() As Double: Dim bbrik() As Double
Dim bdsik() As Double: Dim colAll() As Double: Dim colKonly() As Double

'redimension the arrays while preserving the contents.
ReDim Preserve xx(n + 1, 1) As Double: ReDim Preserve mu(n, n) As Double
ReDim Preserve D(n, 1) As Double: ReDim Preserve Dm(n, 1) As Double: ReDim Preserve vd(n, 1) As Double
ReDim Preserve nold(n + 1, u) As Double: ReDim Preserve nnew(n + 1, u) As Double: ReDim Preserve ni(n + 1, 3) As Double
ReDim Preserve nf(n + 1, 3) As Double: ReDim Preserve nmax(n, 1) As Double: ReDim Preserve beta(n + 1, 1) As Double
ReDim Preserve alphal(n + 1, 1) As Double: ReDim Preserve betal(n + 1, 1) As Double: ReDim Preserve theta(n, u) As Double
ReDim Preserve bij(n, n) As Double: ReDim Preserve bbrij(n, n) As Double: ReDim Preserve bdsij(n, n) As Double
ReDim Preserve bik(n, n) As Double: ReDim Preserve bbrik(n, n) As Double: ReDim Preserve bdsik(n, n) As Double
ReDim Preserve mum(n, n) As Double: ReDim Preserve colAll(n, u) As Double: ReDim Preserve colKonly(n, u) As Double


'Calculate Coefficeints Alphal, beta and betal
    'Coefficeints Boundary Conditions
    alphal(0, 1) = 0
    alphal(21, 1) = 0
    betal(0, 1) = 0
    betal(21, 1) = 0
    beta(0, 1) = 0
    beta(21, 1) = 0
    For i = 1 To n
        D(i, 1) = Cells(4 + i, 3)
        beta(i, 1) = (Cells(3, 6) * (Cells(5, 6) - Cells(4, 6)) * D(i, 1) ^ 2 * Cells(7, 6)) / (18 * Cells(6, 6))
        alphal(i, 1) = beta(i, 1) * (Cells(9, 6) / Cells(10, 6))
        betal(i, 1) = beta(i, 1) * alphal(i, 1) / (Exp((beta(i, 1) * Cells(10, 6)) / Cells(8, 6)) - 1)
    Next i
'Generate Lognorm DSD * Calculate [nold] for any given h in the first layer and intialze all layers
    For i = 1 To n
        Dm(i, 1) = D(i, 1) * 10000
        vd(i, 1) = 4 / 3 * 3.14 * (D(i, 1)) ^ 3
        nold(i, 1) = WorksheetFunction.LogNorm_Dist(Dm(i, 1), (1.5), (5), False)
    Next i

    Vacid = (3.14 / 4) * (2.54 ^ 2) * Cells(10, 6) * 0.3
    vsection = (3.14 / 4) * (2.54 ^ 2) * Cells(10, 6)
    summ = 0
    For i = 1 To n
       x = nold(i, 1) * vd(i, 1)
       summ = summ + x
    Next i

    Multiplier = Vacid / summ

    For j = 1 To u
        For i = 1 To n
            If j = 1 Then
               nold(i, j) = nold(i, j) * Multiplier
            Else
                nold(i, j) = nold(i, 1)
            End If
        Next i
   Next j

    'DSD Boundary Conditions
    For j = 1 To u
        nold(0, j) = 0
        nnew(0, j) = 0
        nold(21, j) = 0
        nnew(21, j) = 0
    Next j
```

```
'Calcualte intial DSD and volume of acid in the tube for error analysis
    summm = 0
    summm1 = 0
    summm2 = 0
    For i = 1 To n
        For j = 1 To u / 3
            ni(i, 1) = ni(i, 1) + nold(i, j)
            summm = summm + nold(i, j) * vd(i, 1)
        Next j
        For j = (u / 3) + 1 To 2 * u / 3
            ni(i, 2) = ni(i, 2) + nold(i, j)
            summm1 = summm1 + nold(i, j) * vd(i, 1)
        Next j
        For j = (2 * u / 3) + 1 To u
            ni(i, 3) = ni(i, 3) + nold(i, j)
            summm2 = summm2 + nold(i, j) * vd(i, 1)
        Next j
        nmax(i, 1) = ni(i, 1) + ni(i, 2) + ni(i, 3)
    Next i

    Cells(26, 13) = summm + summm1 + summm2


'Calculate SOURCE TERM constants for Binary Coalasance Calculations
   'Calculate the collision frequency function for i,j
    For i = 1 To n
       For j = 1 To n
          If j >= i Then
            bbrij(i, j) = (2 * Cells(13, 6) * Cells(12, 6) / Cells(6, 6)) * ((1 / D(i, 1)) + (1 / D(j, 1))) * (D(i, 1) + D(j, 1))
            bdsij(i, j) = 3.14 / 4 * (D(i, 1) + D(j, 1)) ^ 2 * Abs(beta(i, 1) - beta(j, 1))
            bij(i, j) = Cells(14, 6) * (bbrij(i, j) + bdsij(i, j))
          Else
          End If
       Next j
    Next i
    'calculate the collisoin frequency function for i,k
     For k = 1 To n
        For i = 1 To n
          bbrik(i, k) = (2 * Cells(13, 6) * Cells(12, 6) / Cells(6, 6)) * ((1 / D(i, 1)) + (1 / D(k, 1))) * (D(i, 1) + D(k, 1))
          bdsik(i, k) = 3.14 / 4 * (D(i, 1) + D(k, 1)) ^ 2 * Abs(beta(i, 1) - beta(k, 1))
          bik(i, k) = Cells(14, 6) * (bbrik(i, k) + bdsik(i, k))
        Next i
     Next k

     'Calculate Contribuation Term (Based on the Fixed Pivit technique by Kumar & Ramkrishna)

      For k = 1 To n - 1
         For i = 1 To n
            For j = 1 To n
                'Not equal statments are to exclude k from the birth term
                If j <= i And j <> k And i <> k Then
                    Size = vd(i, 1) + vd(j, 1)
                    If vd(k, 1) <= Size And Size < vd(k + 1, 1) Then
                      mu(i, j) = (vd(k + 1, 1) - (vd(i, 1) + vd(j, 1))) / (vd(k + 1, 1) - vd(k, 1)) * Cells(15, 6)
                    End If
                    If j = i Then 'To avoid repitation
                        mu(i, j) = mu(i, j) * 0.5
                    End If
                End If
            Next j
         Next i
      Next k


      For k = 2 To n
         For i = 1 To n
            For j = 1 To n
                If j <= i Then
                    Size = vd(i, 1) + vd(j, 1)
                    If vd(k - 1, 1) <= Size And Size < vd(k, 1) Then
                      mu(i, j) = ((vd(i, 1) + vd(j, 1)) - vd(k - 1, 1)) / (vd(k, 1) - vd(k - 1, 1)) * Cells(15, 6)
                    End If
                    If j = i Then 'To avoid repitation
                        mu(i, j) = mu(i, j) * 0.5
                    End If
                End If
            Next j
         Next i
      Next k

      For i = 1 To n
        For j = 1 To n
            Cells(3 + i, 16 + j) = mu(i, j)
        Next j
      Next i
```

```
'''''''''''''''''''''''''''''''''''''
'Calculate DSD for a given time t
'''''''''''''''''''''''''''''''''''''
    For T = 1 To m
        'Calculate [nnew] using NSFD (Non-Standard Finite Differance)
            For j = 1 To u - 1 'Does not include Last layer
            For i = 1 To n
                nnew(i, j) = (alphal(i - 1, 1) + betal(i - 1, 1)) * nold(i - 1, j)
                + (1 - alphal(i, 1) - 2 * betal(i, 1)) * nold(i, j) + beta(i + 1, 1) * nold(i + 1, 1)
            Next i
            Next j
        'Adjust DSD for each layer to account for loss & gain due to ADVECTION & DIFFUSION
            For j = u To 2 Step -1 'Does not include layer 1
                If j = u Then
                    For i = 1 To n
                        nnew(i, j) = nold(i, j) + (nold(i, j - 1) - nnew(i, j - 1))
                    Next i
                Else
                    For i = 1 To n
                        nnew(i, j) = nnew(i, j) + (nold(i, j - 1) - nnew(i, j - 1))
                    Next i
                End If
            Next j
        'Adjust DSD for each layer to account for DSD change due to BINARY COALASANCE
            'Calculation of source term
            For l = 1 To u
                All = 0
                For i = 1 To n
                    For j = 1 To n
                        If j <= i Then
                            All = All + (bij(i, j) * nnew(i, l) * nnew(j, l))
                        End If
                    Next j
                Next i

                colAll(l, l) = All

                For k = 1 To n
                konly = 0
                    For i = 1 To n
                            konly = konly + bik(i, k) * nnew(i, l) * nnew(k, l)
                    Next i
                    colKonly(k, l) = konly
                Next k


                For k = 1 To n
                    theta(k, l) = colAll(l, l) - colKonly(k, l)
                Next k
            Next l
            'Adjustment of DSD due to Binary coalasance
             For j = 1 To u
                For i = 1 To n
                    nnew(i, j) = nnew(i, j) + theta(i, j)
                Next i
             Next j
       'Place [nold] = [nnew] for next timestep
            For j = 1 To u
                For i = 1 To n
                    nold(i, j) = nnew(i, j)
                Next i
            Next j
    Next T
```

```
'Divide no. of layers into three groups (Top + Next to Bottom + Bottom) & add DSD + Calculate Vacid in each layer
    summ = 0
    summ1 = 0
    summ2 = 0
    For i = 1 To n
        For j = 1 To u / 3
            nf(i, 1) = nf(i, 1) + nnew(i, j)
            summ = summ + nnew(i, j) * vd(i, 1)
        Next j
        For j = (u / 3) + 1 To 2 * u / 3
            nf(i, 2) = nf(i, 2) + nnew(i, j)
            summ1 = summ1 + nnew(i, j) * vd(i, 1)
        Next j
        For j = (2 * u / 3) + 1 To u
            nf(i, 3) = nf(i, 3) + nnew(i, j)
            summ2 = summ2 + nnew(i, j) * vd(i, 1)
        Next j
    Next i

'Display Results
    'Vacid in each layer
    Cells(26, 8) = summ
    Cells(26, 9) = summ1
    Cells(26, 10) = summ2
    'Vfraction of acid in each layer
    Cells(28, 8) = summ / vsection
    Cells(28, 9) = summ1 / vsection
    Cells(28, 10) = summ2 / vsection
    'DSD for Top, Next to Bottom & Bottom
    For i = 1 To n
        Cells(3 + i, 8) = nf(i, 1)
        Cells(3 + i, 9) = nf(i, 2)
        Cells(3 + i, 10) = nf(i, 3)
    Next i
   For j = 1 To u
        Cells(4, 16 + j) = colAll(j, j)
        For i = 1 To n
            Cells(27 + i, 16 + j) = colKonly(i, j)
            Cells(49 + i, 16 + j) = theta(i, j)
        Next i
   Next j
   For i = 1 To 20
        For j = 1 To 20
            Cells(3 + i, 16 + j) = bij(i, j)
            Cells(25 + i, 16 + j) = bik(i, j)
        Next j
   Next i

End Sub
```

Crank Nelson Code

```
Sub Crank_Nelson()

n = InputBox("What are the number of diameters in the distribution?")
m = InputBox("How many days would you like to run the simulation?") * 79
u = (7.62 / Cells(12, 6))

'Declaration
Dim Ao() As Double: Dim A1() As Double: Dim A2() As Double: Dim A3() As Double
Dim A4() As Double: Dim A5() As Double: Dim D() As Double: Dim Dm() As Double
Dim c() As Double: Dim w() As Double: Dim ni() As Double: Dim A() As Double
Dim IM() As Double: Dim r() As Double: Dim nold() As Double: Dim nnew() As Double
Dim nmax() As Double: Dim nf() As Double: Dim vd() As Double

'redimension the arrays while preserving the contents.
ReDim Preserve Ao(n, 1) As Double: ReDim Preserve A1(n, 1) As Double
ReDim Preserve A2(n, 1) As Double: ReDim Preserve A3(n, 1) As Double
ReDim Preserve A4(n, 1) As Double: ReDim Preserve A5(n, 1) As Double
ReDim Preserve D(n, 1) As Double: ReDim Preserve Dm(n, 1) As Double
ReDim Preserve c(n + 1, 1) As Double: ReDim Preserve w(n, 1) As Double
ReDim Preserve ni(n, 3) As Double: ReDim Preserve A(n, n) As Double
ReDim Preserve IM(n, n) As Double: ReDim Preserve r(n, u) As Double
ReDim Preserve nold(n + 1, u) As Double: ReDim Preserve nnew(n + 1, u) As Double
ReDim Preserve nmax(n, 1) As Double: ReDim Preserve nf(n, 3) As Double
ReDim Preserve vd(n + 1, 1) As Double

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Calculate Coefficeints A1, A2, A3, A4 & A5
    S = Cells(8, 6) * (Cells(11, 6) / (Cells(12, 6) ^ 2))
    c(21, 1) = 0

    For i = 1 To n
        D(i, 1) = Cells(3 + i, 3)
        Dm(i, 1) = D(i, 1) * 10000
        vd(i, 1) = 4 / 3 * 3.14 * (D(i, 1)) ^ 3
        nold(i, 1) = WorksheetFunction.LogNorm_Dist(Dm(i, 1), (1.5), (5), False)
        w(i, 1) = (Cells(3, 6) * (Cells(5, 6) - Cells(4, 6)) * D(i, 1) ^ 2 * Cells(7, 6)) / (18 * Cells(6, 6))
        c(i, 1) = w(i, 1) * (Cells(11, 6) / (Cells(12, 6)))
        gamma = (1 - c(i, 1)) / 2
        Ao(i, 1) = 1 - Cells(9, 6) * (c(i, 1) * (2 * gamma - 1) - 2 * S)
        A1(i, 1) = (Cells(9, 6) - 1) * (c(i - 1, 1) * (gamma - 1) - S)
        A2(i, 1) = 1 + (Cells(9, 6) - 1) * (c(i, 1) * (1 - 2 * gamma) + 2 * S)
        A3(i, 1) = (1 - Cells(9, 6)) * (S - c(i + 1, 1) * gamma)
        A4(i, 1) = Cells(9, 6) * (S + c(i - 1, 1) * (1 - gamma))
        A5(i, 1) = Cells(9, 6) * (S - gamma * c(i + 1, 1))

    Next i

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Create [A] Matrix
    For i = 1 To n

        For j = 1 To n
            If i = j Then
               A(i, j) = Ao(i, 1)
            ElseIf i = j + 1 Then
               A(i, j) = -A4(i, 1)
            ElseIf j = i + 1 Then
               A(i, j) = -A5(i, 1)
            Else
               A(i, j) = 0
            End If
        Next j

    Next i
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Calcualte intial DSD and volume of acid in the tube for error analysis
summm = 0
summm1 = 0
summm2 = 0

For i = 1 To n
    For j = 1 To u / 3
        ni(i, 1) = ni(i, 1) + nold(i, j)
        summm = summm + nold(i, j) * vd(i, 1)
    Next j
    For j = (u / 3) + 1 To 2 * u / 3
        ni(i, 2) = ni(i, 2) + nold(i, j)
        summm1 = summm1 + nold(i, j) * vd(i, 1)
    Next j
    For j = (2 * u / 3) + 1 To u
        ni(i, 3) = ni(i, 3) + nold(i, j)
        summm2 = summm2 + nold(i, j) * vd(i, 1)
    Next j
    nmax(i, 1) = ni(i, 1) + ni(i, 2) + ni(i, 3)
Next i

Cells(26, 13) = summm + summm1 + summm2
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Calculate {r} for first layer and intialize the ramaining layers equal to {r}
    'Boundary Conditions & Constants
    For j = 1 To u
        nold(0, j) = 0
        nnew(0, j) = 0
        nold(21, j) = 0
        nnew(21, j) = 0
    Next j
    For j = 1 To u
        For i = 1 To n
            If j = 1 Then
                r(i, j) = A1(i, 1) * nold(i - 1, j) + A2(i, 1) * nold(i, j) + A3(i, 1) * nold(i + 1, j)
            Else
                r(i, j) = r(i, 1)
            End If
        Next i
    Next j
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Calcualte Inverse Matrix
    ' generate Inverse of [A]
    For i = 1 To n
        For j = 1 To n
            If (i = j) Then
                IM(i, j) = 1
            Else
                IM(i, j) = 0
            End If
        Next j
    Next i
    'Forward elimination
    For i = 1 To n
    pivot = A(i, i)
        For j = i + 1 To n
        Fact = A(j, i) / pivot
            For k = 1 To n
                A(j, k) = A(j, k) - Fact * A(i, k)
                IM(j, k) = IM(j, k) - Fact * IM(i, k)
            Next k
        Next j
    Next i
     'Backkward elimination
    For i = n To 2 Step -1
    pivot = A(i, i)
        For j = i - 1 To 1 Step -1
        Fact = A(j, i) / pivot
            For k = 1 To n
                A(j, k) = A(j, k) - Fact * A(i, k)
                IM(j, k) = IM(j, k) - Fact * IM(i, k)
            Next k
        Next j
    Next i
    'Normalization
    For i = 1 To n
    pivot = A(i, i)
        For j = 1 To n
            IM(i, j) = IM(i, j) / pivot
            A(i, j) = A(i, j) / pivot
        Next j
    Next i
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Calculate DSD for a given time t
For T = 1 To m
    'Calculate [nnew]
    For j = 1 To u - 1 'Does not include Last layer
        For i = 1 To n
            Sum = 0
            For k = 1 To n
                Sum = Sum + IM(i, k) * r(k, j)
            Next k
            nnew(i, j) = Sum
        Next i
    Next j
    'Adjust DSD for each layer in [nnew] for Layers 2 to u
    For j = u To 2 Step -1 'Does not include layer 1
        If j = u Then
            For i = 1 To n
                nnew(i, j) = nold(i, j) + (nold(i, j - 1) - nnew(i, j - 1))
            Next i
        Else
            For i = 1 To n
                nnew(i, j) = nnew(i, j) + (nold(i, j - 1) - nnew(i, j - 1))
            Next i
        End If
    Next j
    'Recalculate [r]
    For j = 1 To u
        For i = 1 To n
           r(i, j) = A1(i, 1) * nnew(i - 1, j) + A2(i, 1) * nnew(i, j) + A3(i, 1) * nnew(i + 1, j)
        Next i
    Next j
    'Place [nold] = [nnew] for next timestep
    For j = 1 To u
        For i = 1 To n
            nold(i, j) = nnew(i, j)
        Next i
    Next j

  For j = 1 To u
    For i = 1 To n
          Cells(3 + i, 16 + j) = nnew(i, j)
    Next i
  Next j
Next T
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Divide no. of layers into three groups (Top + Next to Bottom + Bottom) & add DSD + Calculate Vacid in each layer
summ = 0
summ1 = 0
summ2 = 0
For i = 1 To n
    For j = 1 To u / 3
        nf(i, 1) = nf(i, 1) + nnew(i, j)
        summ = summ + nnew(i, j) * vd(i, 1)
    Next j
    For j = (u / 3) + 1 To 2 * u / 3
        nf(i, 2) = nf(i, 2) + nnew(i, j)
        summ1 = summ1 + nnew(i, j) * vd(i, 1)
    Next j
    For j = (2 * u / 3) + 1 To u
        nf(i, 3) = nf(i, 3) + nnew(i, j)
        summ2 = summ2 + nnew(i, j) * vd(i, 1)
    Next j
Next i
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Display Results
    'Vacid in each layer
    Cells(26, 8) = summ
    Cells(26, 9) = summ1
    Cells(26, 10) = summ2
    'DSD for Top, Next to Bottom & Bottom
    For i = 1 To n
        Cells(3 + i, 8) = nf(i, 1)
        Cells(3 + i, 9) = nf(i, 2)
        Cells(3 + i, 10) = nf(i, 3)
    Next i
End Sub
```

Lax Wenhoff Code

```
Sub Lax_Explicit()

n = InputBox("What are the number of diameters in the distribution?")
m = InputBox("How many days would you like to run the simulation?") * 80
u = (7.62 / Cells(10, 6))
Cells(11, 6) = m / 80

'Declaration
Dim D() As Double: Dim Dm() As Double: Dim vd() As Double: Dim nold() As Double: Dim ni() As Double
Dim nf() As Double: Dim nmax() As Double: Dim nnew() As Double: Dim beta() As Double: Dim alphal() As Double
Dim betal() As Double: Dim vfracT() As Double: Dim vfracNB() As Double: Dim vfracB() As Double

'redimension the arrays while preserving the contents.
ReDim Preserve D(n, 1) As Double: ReDim Preserve Dm(n, 1) As Double: ReDim Preserve vd(n, 1) As Double
ReDim Preserve nold(n + 1, u) As Double: ReDim Preserve nnew(n + 1, u) As Double: ReDim Preserve ni(n + 1, 3) As Double
ReDim Preserve nf(n + 1, 3) As Double: ReDim Preserve nmax(n, 1) As Double: ReDim Preserve beta(n + 1, 1) As Double
ReDim Preserve alphal(n + 1, 1) As Double: ReDim Preserve betal(n + 1, 1) As Double: ReDim Preserve vfracT(m, 1) As Double
ReDim Preserve vfracNB(m, 1) As Double: ReDim Preserve vfracB(m, 1) As Double

'Calculate Coefficeints Alphal, beta and betal

    'Boundary Conditions
    alphal(0, 1) = 0
    alphal(21, 1) = 0
    betal(0, 1) = 0
    betal(21, 1) = 0
    beta(0, 1) = 0
    beta(21, 1) = 0

    For i = 1 To n
        D(i, 1) = Cells(4 + i, 3)
        beta(i, 1) = (Cells(3, 6) * (Cells(5, 6) - Cells(4, 6)) * D(i, 1) ^ 2 * Cells(7, 6)) / (18 * Cells(6, 6))
        alphal(i, 1) = beta(i, 1) * (Cells(9, 6) / Cells(10, 6))
        betal(i, 1) = Cells(8, 6) * Cells(9, 6) / Cells(10, 6)
    Next i

'Generate Lognorm DSD * Calculate [nold] for any given h in the first layer and intialze all layers

    For i = 1 To n
        Dm(i, 1) = D(i, 1) * 10000
        vd(i, 1) = 4 / 3 * 3.14 * (D(i, 1)) ^ 3
        nold(i, 1) = WorksheetFunction.LogNorm_Dist(Dm(i, 1), (4), (2), False)
    Next i

    Vacid = (3.14 / 4) * (2.54 ^ 2) * Cells(10, 6) * 0.3
    vsection = (3.14 / 4) * (2.54 ^ 2) * Cells(10, 6) * u / 3
    summ = 0
    For i = 1 To n
       x = nold(i, 1) * vd(i, 1)
       summ = summ + x
    Next i

    Multiplier = Vacid / summ

    For j = 1 To u
        For i = 1 To n
            If j = 1 Then
               nold(i, j) = nold(i, j) * Multiplier
            Else
                nold(i, j) = nold(i, 1)
            End If
        Next i
   Next j



    'Boundary Conditions & Constants
    For j = 1 To u
        nold(0, j) = 0
        nnew(0, j) = 0
        nold(21, j) = 0
        nnew(21, j) = 0
    Next j
```

```
'Calcualte intial DSD and volume of acid in the tube for error analysis
    summm = 0
    summml = 0
    summm2 = 0

    For i = 1 To n

        For j = 1 To u / 3
            ni(i, 1) = ni(i, 1) + nold(i, j)
            summm = summm + nold(i, j) * vd(i, 1)
        Next j
        For j = (u / 3) + 1 To 2 * u / 3
            ni(i, 2) = ni(i, 2) + nold(i, j)
            summml = summml + nold(i, j) * vd(i, 1)
        Next j
        For j = (2 * u / 3) + 1 To u
            ni(i, 3) = ni(i, 3) + nold(i, j)
            summm2 = summm2 + nold(i, j) * vd(i, 1)
        Next j
        nmax(i, 1) = ni(i, 1) + ni(i, 2) + ni(i, 3)
        Cells(2 + i, 13) = nmax(i, 1)

    Next i

    Cells(26, 13) = summm + summml + summm2


'Calculate DSD for a given time t

    For T = 1 To m

        'Calculate [nnew]
        For j = 1 To u - 1 'Does not include Last layer
            For i = 1 To n
                nnew(i, j) = 0.5 * (alphal(i - 1, 1) + alphal(i - 1, 1) ^ 2 + 2 * betal(i - 1, 1)) * nold(i - 1, j) + (1 - 2 * betal(i, 1) - alphal(i, 1) ^ 2)
                * nold(i, j) + 0.5 * (2 * beta(i + 1, 1) - alphal(i + 1, 1) + alphal(i + 1, 1) ^ 2) * nold(i + 1, 1)
            Next i
        Next j
        'Adjust DSD for each layer in [nnew] for Layers 2 to u
        For j = u To 2 Step -1 'Does not include layer 1
            If j = u Then
                For i = 1 To n
                    nnew(i, j) = nold(i, j) + (nold(i, j - 1) - nnew(i, j - 1))
                Next i
            Else
                For i = 1 To n
                    nnew(i, j) = nnew(i, j) + (nold(i, j - 1) - nnew(i, j - 1))
                Next i
            End If
        Next j
        'Place [nold] = [nnew] for next timestep
        For j = 1 To u
            For i = 1 To n
                nold(i, j) = nnew(i, j)
            Next i
        Next j
    'Calculate Vfraction & time to create vfraction vs. time graph
        summ = 0
        summl = 0
        summ2 = 0

        For i = 1 To n
            For j = 1 To u / 3
                summ = summ + nnew(i, j) * vd(i, 1)
            Next j
            For j = (u / 3) + 1 To 2 * u / 3
                summl = summl + nnew(i, j) * vd(i, 1)
            Next j
            For j = (2 * u / 3) + 1 To u
                summ2 = summ2 + nnew(i, j) * vd(i, 1)
            Next j
        Next i
        If T Mod 10 = 0 Then
            Cells(49 + T / 10, 3) = summ / vsection
            Cells(49 + T / 10, 4) = summl / vsection
            Cells(49 + T / 10, 5) = summ2 / vsection
            Cells(49 + T / 10, 2) = (T / 18) / 24
        End If
    Next T
```

```
'Divide no. of layers into three groups (Top + Next to Bottom + Bottom) & add DSD + Calculate Vacid in each layer

    summ = 0
    summ1 = 0
    summ2 = 0

    For i = 1 To n

        For j = 1 To u / 3
            nf(i, 1) = nf(i, 1) + nnew(i, j)
            summ = summ + nnew(i, j) * vd(i, 1)
        Next j

        For j = (u / 3) + 1 To 2 * u / 3
            nf(i, 2) = nf(i, 2) + nnew(i, j)
            summ1 = summ1 + nnew(i, j) * vd(i, 1)
        Next j

        For j = (2 * u / 3) + 1 To u
            nf(i, 3) = nf(i, 3) + nnew(i, j)
            summ2 = summ2 + nnew(i, j) * vd(i, 1)
        Next j

    Next i

'Display Results

    'Vacid in each layer
    Cells(26, 8) = summ
    Cells(26, 9) = summ1
    Cells(26, 10) = summ2

    'Vfraction of acid in each layer
    Cells(28, 8) = summ / vsection
    Cells(28, 9) = summ1 / vsection
    Cells(28, 10) = summ2 / vsection

    'DSD for Top, Next to Bottom & Bottom
    For i = 1 To n
        Cells(3 + i, 8) = nf(i, 1)
        Cells(3 + i, 9) = nf(i, 2)
        Cells(3 + i, 10) = nf(i, 3)
    Next i


End Sub
```