Computational analysis of ultraviolet reactors

Xiang Li
COMPUTATIONAL ANALYSIS OF
ULTRAVIOLET REACTORS

By

Xiang Li

A Thesis Submitted
In Partial Fulfillment
of the Requirements of the Degree of
Master of Science
In
Mechanical Engineering

Approved by:
Prof. __________________________________________
Dr Ali Ogut (Thesis Advisor)

Prof. __________________________________________
Dr Panchapakesan Venkataraman (Committee Member)

Prof. __________________________________________
Dr Robert Stevens (Committee Member)

Prof. __________________________________________
Dr Wayne Walter (Department Representative)

Department of Mechanical Engineering
The Kate Gleason College of Engineering
Rochester Institute of Technology

11. 2009
COMPUTATIONAL ANALYSIS OF
ULTRAVIOLET REACTORS

By

Xiang Li

I, Xiang Li, hereby grant permission to the Wallace Memorial Library of the Rochester Institute of Technology to reproduce this document in whole or in part that any reproduction will not be for commercial use or profit.

Xiang Li

Month  Date, Year
Acknowledgment

I would like to express my sincere gratitude to my thesis advisor, Professor Ali Ogut, for his guidance, full support, and trust in me. Thanks are also expressed to Professor Steve Weinstein who walked me into the field of fluid dynamics; Professor Amitabha Ghosh who inspired me the idea of computational modeling for this study; Professor Lawrence Agbezuge who helped me build the theoretical model; Professor Panchapakesan Venkataraman and Professor Robert Stevens who served on my committe and gave me valuable comments on my thesis.

I would like to thank my colleague Elijah Richter for valuable previous work he had done with the UV calculation program and all the members in Environmental Energy Technology for their friendship and constructive discussions.

Finally, I would like to express my deepest appreciation to my wife, Yan, and my parents who have showed their complete support during my study.
Abstract

Disinfection of water treatment using ultraviolet (UV) light has received wide recognition as an important contribution to the protection of public health. To ensure a UV disinfection system that can deliver the required UV dose, analytical and numerical methods have been used to evaluate the flow field of the UV reactor. A Multiple Point Source Summation (MPSS) method was used to estimate the radiation intensity field, with the utilization of Computational Fluid Dynamics (CFD) technique. To couple the analytical model with numerical simulations, a user friendly interface software was developed to calculate the UV dosage distribution. In addition a UV lamp aging evaluation calculator was also developed for predicting the effective service life of the system. A typical UV disinfection unit has been simulated and provided flow field data to demonstrate the accuracy of dosage distribution modeling. The predicted values provided a good comparison with experimental data at 85% UV Transmittance.
# Table of Contents

Acknowledgment .................................................................................................................. ii
Abstract ................................................................................................................................... iii
Table of Contents ................................................................................................................... iv
List of Tables ........................................................................................................................... vi
List of Figures .......................................................................................................................... vii
Nomenclature ........................................................................................................................... ix

Chapter 1. Introduction .............................................................................................................. 1
  1.1 Thesis Background ........................................................................................................... 1
  1.2 Thesis Objectives and Goal ............................................................................................. 2
  1.3 Ultraviolet Disinfection Technology .............................................................................. 2
    1.3.2 Germicidal UV Technology ................................................................................. 3
  1.4 A Typical UV Disinfection System ............................................................................... 5

Chapter 2. Literature Review and Fundamentals ..................................................................... 7
  2.1 UV Disinfection Equipment ........................................................................................... 7
  2.2 UV Reactor Models ........................................................................................................ 8
  2.3 Hydrodynamics ............................................................................................................... 8

Chapter 3. UV Dosage Distribution Modeling ......................................................................... 10
  3.1 UV Light Transmission ................................................................................................... 10
  3.2 Principles of UV Radiation ............................................................................................. 15
  3.3 UV Dose Distribution Calculations ............................................................................. 16
  3.4 Reduction Equivalent Dose (RED) ............................................................................. 22
  3.5 Collimated Beam Testing: MS2 Phage ......................................................................... 22
  3.6 2D Simplified UV Dose Modeling ............................................................................. 25
  3.7 3D UV Dose Modeling Based on CFD ..................................................................... 27

Chapter 4: Computational Fluid Dynamics ............................................................................... 29
  4.1 Overview ....................................................................................................................... 29
  4.2 Fluid Flow Motion ......................................................................................................... 29
  4.3 Commercial CFD Software ......................................................................................... 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4 CFD Results and Postprocessing</td>
<td>30</td>
</tr>
<tr>
<td>4.5 CFD Solution Overview</td>
<td>30</td>
</tr>
<tr>
<td>Chapter 5: Dosage Calculation Program</td>
<td>31</td>
</tr>
<tr>
<td>5.1 Overview</td>
<td>31</td>
</tr>
<tr>
<td>5.3 UV Dose Program Interface</td>
<td>33</td>
</tr>
<tr>
<td>Chapter 6: Numerical Simulation of Commercial Model</td>
<td>36</td>
</tr>
<tr>
<td>6.1 Model Specification</td>
<td>36</td>
</tr>
<tr>
<td>6.2 Create Geometry</td>
<td>37</td>
</tr>
<tr>
<td>6.4 Specify Boundary Types</td>
<td>38</td>
</tr>
<tr>
<td>6.5 Solver Set up</td>
<td>39</td>
</tr>
<tr>
<td>6.6 Analyze Results</td>
<td>40</td>
</tr>
<tr>
<td>6.7 Dose Distribution Calculation</td>
<td>42</td>
</tr>
<tr>
<td>Chapter 7. UV Lamp Aging Evaluation</td>
<td>48</td>
</tr>
<tr>
<td>7.1 UV Lamp Overview</td>
<td>48</td>
</tr>
<tr>
<td>7.2 UV Lamp Aging</td>
<td>49</td>
</tr>
<tr>
<td>7.3 Lamp Aging Evaluation</td>
<td>50</td>
</tr>
<tr>
<td>Chapter 8: Aging Evaluation Program</td>
<td>52</td>
</tr>
<tr>
<td>8.1 Lamp Aging Evaluation Interface</td>
<td>52</td>
</tr>
<tr>
<td>Chapter 9: Summary and Conclusion</td>
<td>55</td>
</tr>
<tr>
<td>Reference</td>
<td>56</td>
</tr>
<tr>
<td>Appendix</td>
<td>63</td>
</tr>
<tr>
<td>Appendix A: Calculated Dose Distribution Comparison</td>
<td>63</td>
</tr>
<tr>
<td>Appendix B: Calculated RED Data of Commercial Model under Different Flow Rate</td>
<td>64</td>
</tr>
<tr>
<td>Appendix C: VB Dose Distribution Calculation Program Code</td>
<td>65</td>
</tr>
</tbody>
</table>
List of Tables

Table 3.1: UV dose requirements from EPA 40 CFR 141.720(d)(1)................................. 15
Table 3.2: Validation of simulation model for the experimental data................................. 26
Table 5: Main program classes.................................................................................................33
Table 6: Calculated RED value comparing with testing RED value, 70UVT.....................48
Table 7.1: Typical mercury vapor lamp characteristics.........................................................49
List of Figures

Figure 1.1: light spectrum and UVC radiation [7]................................................................. 4
Figure 1.2: DNA before ultraviolet disinfection - all bonds required for replication are intact [9].......................................................................................................................... 4
Figure 1.3: DNA after ultraviolet disinfection - broken bonds and Thymine Dimer formation prevent replication [9].......................................................................................................................... 5
Figure 1.4: a typical UV water purifiers [10]........................................................................... 6
Figure 3.1: Beer – Lambert absorption of a beam of light[50]................................................. 11
Figure 3.2: refraction of light through the air-quartz-water interface................................... 12
Figure 3.3.1: specular reflection of light.................................................................................. 12
Figure 3.3.2: diffuse reflection of light.................................................................................. 13
Figure 3.4: scattering of light.............................................................................................. 13
Figure 3.5: intensity at reception point.................................................................................. 16
Figure 3.7: intensity field of one reception point—one point source.................................... 17
Figure 3.8: sum of all the point sources contribute to the intensity at one reception point 18
Figure 3.9: intensity field of one reception point—one point source considering quartz sleeve thickness.............................................................................................................. 19
Figure 3.10: schematic representation of dosage calculation coupling with intensity field ................................................................................................................................. 20
Figure 3.11: dose distribution comparison of point source numbers' setting......................... 21
Figure 3.12: UV dose-response curve [53]............................................................................ 24
Figure 3.13: configuration of 2D simplified model............................................................... 25
Figure 3.14: UV intensity distribution, contour plot............................................................ 25
Figure 3.16.1: path line that predict the true particle trajectory........................................... 28
Figure 3.16.2: path line stops at stagnation point................................................................. 28
Figure 3.16.3: path line that falls into circulation form......................................................... 28
Figure 5.1: logo of UV calculation program......................................................................... 31
Figure 5.2: setting of the main program window............................................................... 33
Figure 5.3: apparatus setting: lamp location......................................................................... 34
Figure 5.4: apparatus setting: outlet location......................................................... 34
Figure 5.5: detailed output result from dose calculation............................................ 34
Figure 5.5: general output result from dose calculation (dose unit: mJ/cm^2).............. 34
Figure 6.1.1: simplified geometry of commercial model........................................... 35
................................................................................................................................. 36
Figure 6.1.2: dimension of commercial model......................................................... 36
Figure 6.2: geometry of commercial model created with ANSYS CFX 11.0.............. 36
Figure 6.3: mesh geometry created with CFX MESH.............................................. 37
Figure 6.4.1 path line (250 particles) generated of the commercial model............... 39
Figure 6.4.2 path line (250 particles) details of inlet................................................ 40
Figure 6.4.3 path line (250 particles) details of outlet............................................... 40
Figure 6.5.1 path line (250 particles) data example............................................... 41
Figure 6.5.2 dose calculation configuration setting.................................................. 42
Figure 6.5.3 dose calculation complete message..................................................... 42
Figure 6.5.4 example of dose calculation output, detailed report, uvt70.................... 43
Figure 6.5.5 example of dose calculation output (mJ/cm^2), general report, uvt70...... 43
Figure 6.6 bubble chart of initial position vs. dose distribution uvt70....................... 44
Figure 6.7 calculated RED value for commercial model under different flow rate and
UVTs................................................................................................................................. 45
Figure 6.8 calculated RED value comparing with testing RED value........................ 46
Figure 7.1.1: schematic representation of estimation of lamp output power apparatus.. 50
Figure 8.1: setting for age evaluation main window............................................... 51
Figure 8.2: apparatus setting for aging evaluation.................................................... 52
Figure 8.3: calculated result for aging evaluation..................................................... 53
Nomenclature

Symbols

\( A_\lambda \) absorbance of incident light passes through a water sample

\( I \) intensity of light transmitted through the sample (\( mW/cm^2 \))

\( I_0 \) intensity of light incident on the sample (\( mW/cm^2 \))

\( \varepsilon_\lambda \) the molar absorption coefficient (\( M^{-1} cm^{-1} \)) at wavelength \( \lambda \),

\( c \) the molar concentration of the absorbing species

\( l \) the path length of light passes through the sample (cm)

\( \sigma \) the absorbance coefficient of of the substance (cm\(^{-1}\))

\( T \) the transmissivity of light through the substance

\( UVT \) UV transmittance at a specified wavelength and path length

\( E_A \) total UV intensity at a reception point

\( P_i \) radiant power of one point source

\( r \) radial distance from point source to reception point

\( n \) total number of point sources within the irradiated zone

\( \sigma_w \) the absorbance coefficient of water (cm\(^{-1}\))

\( R \) radial distance from lamp axis to reception point (cm)

\( r_q \) radial distance from lamp axis to outside quartz jacket (cm)

\( \sigma_q \) the absorbance coefficient of quartz sleeve (cm\(^{-1}\))

\( t_q \) quartz jacket thickness (cm)

\( l_i \) distance from \( i \)th point source to reception point (cm)

\( \theta \) angle of \( l_i \) and \( R \)

\( N_0 \) concentration of infectious microorganisms before exposure to UV light

\( N \) concentration of infectious microorganisms after exposure to UV light

\( A, B \) coefficients of log reduction rate of the UV dose-response curve

\( \log \text{ inactivation} \) the total log inactivation

\( pnum \) number of particles in path line data

\( P_e \) estimated lamp power (W)

\( I_m \) UV intensity measured radiometrically (mWcm\(^{-2}\))
Chapter 1. Introduction

1.1 Thesis Background

UV irradiation is one of the most frequently applied physical disinfection process in water and wastewater treatment. As of the year 2000, more than 400 UV disinfection facilities worldwide were treating drinking water; these UV facilities typically treat flows of less than 1 million gallons per day. Since 2000, several large UV installations across the United States have been constructed or are currently under design, and the number of public water systems using UV disinfection is expected to increase significantly over the next decade [1]. UV light is not only effective, but also very efficient. It can disinfect water at about one-tenth the cost of other treatment methods, in part because the equipment is so small [2].

Numerous health studies have shown that Germicidal Ultraviolet light (also known as UVC, wavelength ranges from 100 nanometers (nm) to 280 nm) is very effective against allergies, asthma, mold, mildew, fungi and any DNA based viruses, bacteria and spores. UV light can deactivate the DNA of bacteria, viruses and other pathogens and thus destroys their ability to multiply and cause disease. Specifically, UVC light causes damage to the nucleic acid of microorganisms by forming covalent bonds between certain adjacent bases in the DNA. The formation of such bonds prevents the DNA from being unzipped for replication, and the organism is unable to reproduce [3].

Germicidal ultraviolet purifiers utilize lamps that are designed and calculated to produce a certain dosage of ultraviolet. To make up an effective UV dose, which is defined as the product of UV intensity and exposure time, water quality, radiation concentration, flow fare exposure time, proper wavelength, microorganism's type and source, and also distance from the light source are principles that should be considered in the design. Intensity which is defined as the rate at which photons are delivered to the target and exposure time are in turn governed by the geometry and hydrodynamics of the system. UV dosage is normally expressed in $\mu W/cm^2$, $mJ/cm^2$ or $J/m^2$ [1].
1.2 Thesis Objectives and Goal

The main objective of this thesis is to develop a program that calculates the dosage distribution of ultraviolet disinfection system for drinking water and wastewater.

In order to calculate UV dosage distributions and reduction equivalent dose (RED) values, a numerical model, the widely used point-source summation (PSS) technique is introduced to characterize the radiation intensity field. And a UV dosage distribution program will be built using Visual Basic. It can incorporate user-defined variables to calculate the resulting RED values of the system. The program will be developed in conjunction with either the particle analysis tool in FLUENT or the streamline algorithm in CFX, both of which are particle tracking tools utilized to obtain position vectors with respect to residence time.

This thesis will also try to build up a program to evaluate the UV lamp aging and fouling conditions based on UV intensity measurement. UV lamp service life can be monitored and numerical/analytical methods can be used to assist cost-effective application of appropriate levels of UV system operation.

1.3 Ultraviolet Disinfection Technology

1.3.1 Introduction

Water disinfection means the removal, deactivation, or killing of pathogenic microorganisms. Microorganisms are destroyed or deactivated, resulting in termination of growth and reproduction. When microorganisms are not removed from drinking water, drinking water usage will cause people to fall ill.

Disinfection can be attained by means of physical or chemical disinfectants. The agents also remove organic contaminants from water, which serve as nutrients or shelters for microorganisms. Disinfectants should not only kill microorganisms. Disinfectants must also have a residual effect, which means that they remain active in the water after disinfection. A disinfectant should prevent pathogenic microorganisms from growing in the plumbing after disinfection, causing the water to be recontaminated [4].

With the discovery that ultraviolet (UV) light inactivates Cryptosporidium at relatively low doses, UV disinfection has emerged as a best available technology for enhancing public health protection at drinking water treatment field [5].
The advantages of using UV, rather than chemical disinfection, include:
- It has no known toxic or significant nontoxic byproducts;
- It has as no danger of overdosing;
- It removes some organic contaminants;
- It has no volatile organic compound emissions or toxic air emissions;
- It has no onsite smell and no smell in the final water product;
- It requires very little contact time (seconds versus minutes for chemical disinfection);
- It does not require storage of hazardous material;
- It requires minimal space for equipment and contact chamber;
- It improves the taste of water because of some organic contaminants and nuisance microorganisms are destroyed;

The disadvantages of using UV include:
- UV radiation is not suitable for water with high levels of suspended solids, turbidity, color, or soluble organic matter. These materials can react with UV radiation, and reduce disinfection performance.
- UV light is not effective against any non-living contaminant, lead, asbestos, many organic chemicals, chlorine, etc.
- Tough cryptosporidia cysts are fairly resistant to UV light.
- Requires electricity to operate. In an emergency situation when the power is out, the purification will not work [6]

1.3.2 Germicidal UV Technology

Ultraviolet light is part of the light spectrum, which is classified into three wavelength ranges:
- UV-C, from 100 nanometers (nm) to 280 nm
- UV-B, from 280 nm to 315 nm
- UV-A, from 315 nm to 400 nm
Figure 1.1: light spectrum and UVC radiation [7]

UV-C light is germicidal - i.e., it deactivates the DNA of bacteria, viruses and other pathogens and thus destroys their ability to multiply and cause disease. Specifically, UV-C light causes damage to the nucleic acid of microorganisms by forming covalent bonds between certain adjacent bases in the DNA. The formation of such bonds prevents the DNA from being unzipped for replication, and the organism is unable to reproduce. In fact, when the organism tries to replicate, it dies [8].

Figure 1.2 and 1.3 shows that DNA molecules in the nucleus of the organism absorb ultraviolet light. The organism is inactivated when sufficient dosage has been absorbed to modify the molecular structure in the DNA. This results when exposure to UV light causes two thymine molecules to form an inappropriate bond, or dimer. The effect of numerous thymine dimers forming along the DNA chain inhibits replication of the organism. It may not be killed instantly, but the scrambling of the genetic code in the nucleus prevents reproduction, rendering it non-viable [9].

Figure 1.2: DNA before ultraviolet disinfection - all bonds required for replication are intact [9]
Ultraviolet technology is a non-chemical approach to disinfection. In this method of disinfection, nothing is added which makes this process simple, inexpensive and requires very low maintenance. Ultraviolet purifiers utilize germicidal lamps that are designed and calculated to produce a certain dosage of ultraviolet (usually at least 16,000 microwatt seconds per square centimeter but many units actually have a much higher dosage). The principle of design is based on a product of time and intensity.

1.4 A Typical UV Disinfection System

The purpose of a UV disinfection system is to reduce the number of viable pathogen in a fluid stream to an acceptable level. The design of a typical UV system is shown in Figure 1.4. A UV bulb or lamp, housed in clear quartz tube, is placed at the center of the chamber. While water is flowing inside the reactor system, the UV irradiates the organisms in the flow water [10].

Figure 1.3: DNA after ultraviolet disinfection - broken bonds and Thymine Dimer formation prevent replication [9]
As the goal in designing UV reactors for drinking water disinfection is to efficiently deliver the dose necessary to inactivate pathogenic microorganisms, within the closed channel stainless steel chamber, UV lamps typically are housed within the lamp sleeves, which protect and insulate the lamps. Reactors also include automatic cleaning mechanisms to keep the lamp sleeves free of deposits. UV sensors, flow meters, and, in some cases, UVT analyzers, are used to monitor dose delivery by the reactor.
Chapter 2. Literature Review and Fundamentals

2.1 UV Disinfection Equipment

Ultraviolet disinfection technology has been established and supported by fundamental and applied research since Downes and Blunt (1877) [11] discovered the germicidal properties of sunlight. The first mercury lamp used as artificial UV light sources in 1901 and quartz as UV transmitting material in 1906 were applied into drinking water disinfection in Marseilles, France, in 1910 [12]. In 1929, a link between UV disinfection and absorption of UV light by nucleic acid identified (Gates 1929) [13]. And during the 1950s considerable research on this mechanism was done (Dulbecco 1950, Kelner 1950, Brandt and Giese 1956, Powell 1959) [14][15][2][16].

The first reliable applications of UV light for disinfecting municipal drinking water occurred in Switzerland and Austria in 1955 (Kruithof and van der Leer 1990) [17]. And the application of UV radiation is growing in importance and frequency through the recent years (Pruden and Ollis, 1983; Kormann et al., 1991) [18][19]. The reason for UV radiation gaining more attention was its low hazardous oxidation and extremely limited by-products (Hoyer 2004) [20]. Since the 1980s Europe has been using UV disinfection widely and after the 1990s UV application was used as a primary high efficiency treatment against Cryptosporidium and Giardia (Kruithof et al., 1992; Bryant et al., 1992; Clancy et al., 1998) [21][22][23]. Since 2000, several large UV installations across the United States have been constructed or are currently under design. The number of public water systems using UV disinfection is expected to increase significantly over the next decade (US EPA 2000) [10].

The quantitative microbiological risk assessment (QMRA) has been used to define the microbiological safety of drinking water and become a growing interest. Dose-response assessment offers a promising way to analyse the impact of water pathogen concentration in terms of health risks in a quantitative manner (Haas et al., 1983; Gerba and Rose 1993; Maya et al., 2003) [24][25][26].
2.2 UV Reactor Models

The study of UV disinfection system design involves many fundamental principles. In order to get the details of how UV radiation enters the reactor and get attenuated by absorption, reflection in the reactor, optics to the specific geometry and composition is introduced. A Multiple Point Source Summation (MPSS) method has been proved to be efficient (Jacob and Dranoff, 1970; Scheible et al., 1985; Suidan and Severin, 1986)[27][28][29]. Blatchley(1997) has extended this model by to a Line Source Integration (LSI) method to deal with infinite number of point sources[30]. Irazoqui (1973) expanded MPSS model to consider a three dimensional source with volumetric emission[31], and Foraboschi (1959) considered a three dimensional source with superficial emission[32]. Irazoqui's study (1973) showed that there is only a little difference between radiation fields for MPSS model with spherical emission and three dimensional cylindrical source with volumetric emission (Bolton 1999; Ducoste et al., 2005)[33][34].

2.3 Hydrodynamics

Hydrodynamics plays an important role in UV systems. Not only does it affect disinfection efficiency by dictating the dose, but also governs fouling behavior as a result of mass, energy and momentum transport of the quartz-water interface (Blatchley and Hunt 1994; Ernest et al., 1997; Kowalski et al., 2001)[35][36][37].

Although the hydrodynamic behavior is critical to the prediction of UV system performance it has not been fully investigated yet (Severing et al., 1983; Blatchley and Hunt 1994)[28][35]. However, CFD UV dose modeling of the impact of UV reactor inlet and outlet conditions on Reduction Equivalent Dose (RED) could be used in conjunction with other experimental approaches to optimize UV dose delivery design (Bolton 1999; Wright and Reddy 2003; Dzurny et al., 2003; Chetverushkin et al., 2003; Wright et al., 2006)[38][39][40][41][42].

Because of nonuniformities in the UV radiation intensity and flow fields, microorganisms traveling through a UV system will experience a broad distribution of doses (Chiu et al., 1999)[43]. Only limited information exists regarding the fundamental hydrodynamics of UV systems. To combine complex hydrodynamics and irregular UV
intensity that results in a broad distribution of UV dose, certain turbulence models have been introduced (Chetverushkin 2003; Taghipiour 2003; Cohn 2002; Islek 2004; Moran 2004)[44][45][46][47][48].

Among the studies of hydrodynamics of UV water treatment systems, Schoenen et al. 1993 demonstrated the effect of the reactor flow field on the performance of water disinfection systems[49]. Chiu et al. 1999 used laser Doppler velocimetry to obtain the velocity field of a cross flow UV reactor experimentally[43]. Recently, in order to verify CFD simulation experimentally, certain advanced experimental flow visualization methods, providing the main characteristics of a given flow configuration for the evaluation of CFD results, have been introduced into this field (particle image velocimetry (PIV), Taghipiour, F. 2003)[45].
Chapter 3. UV Dosage Distribution Modeling

3.1 UV Light Transmission

As UV light is generated from the lamp source, it interacts with the materials it encounters through absorption, reflection, refraction, and scattering. In disinfection applications, these phenomena result from interactions between the emitted UV light and UV reactor components (e.g., lamp sleeves, reactor walls) and also the water being treated.

When assessing water quality, UV absorbance or UV transmittance (UVT) is the parameter that incorporates the effect of absorption and scattering.

3.1.1 Absorption and UV Absorbance (A)

Absorption is the transformation of light to other forms of energy as it passes through a substance. UV absorbance of a substance varies with the wavelength of the light. The components of a UV reactor and the water passing through the reactor all absorb UV light to varying degrees, depending on their material composition. When UV light is absorbed, it is no longer available to disinfect microorganisms.

UV absorbance quantifies the decrease in the amount of incident light as it passes through a water sample over a specified distance or pathlength. In spectroscopy, the absorbance \( A_\lambda \) is defined as:

\[
A_\lambda = -\ln \left( \frac{I_1}{I_0} \right)
\]  

(1)

Where

\( I_1 \)  the intensity of light at a specified wavelength \( \lambda \) that has passed through a sample (transmitted light intensity)

\( I_0 \)  the intensity of the light before it enters the sample or incident light intensity

Also, from the empirical Beer–Lambert law, the relationship between the absorption of light and the properties of the material through which the light is travelling can be
\[ A_\lambda = \varepsilon_\lambda l c = \sigma l \] (2)

where

- \( \varepsilon_\lambda \): the molar absorption coefficient (M\(^{-1}\) cm\(^{-1}\)) at wavelength \( \lambda \),
- \( c \): the molar concentration of the absorbing species (M)
- \( l \): the path length (cm)
- \( \sigma \): the absorbance coefficient of the substance (cm\(^{-1}\)) which equals to \( \varepsilon c \)

Figure 3.1 is the diagram of Beer–Lambert absorption of a beam of light as it travels through a cuvette of width \( l \) [41].

This equation could also be written as:

\[ T = e^{-\varepsilon lc} = e^{-\sigma l} \] (3)

Where \( T \) is the transmissivity of light through the substance, which is defined as

\[ T = e^{-A_\lambda} = \frac{I_1}{I_0} \] (4)

Figure 3.1: Beer–Lambert absorption of a beam of light [50]
In UV disinfection applications, $A_{254}$ is used to measure the amount of UV light passing through the water and reaching the target organisms. $A_{254}$ is measured using a spectrophotometer with 254 nm incident light[1].

### 3.1.2 Refraction

Refraction (Figure 3.2) is the change in the direction of light propagation as it passes through the interface between one medium and another. In UV reactors, refraction occurs when light passes from the UV lamp into an air gap, from the air gap into the lamp sleeve, and from the lamp sleeve into the water. Refraction changes the angle that UV light strikes target pathogens, but how this ultimately affects the UV disinfection process is unknown.

![Figure 3.2: refraction of light through the air-quartz-water interface](image)

### 3.1.3 Reflection

Reflection is the change in direction of light propagation when it is deflected by a surface. Reflection may be classified as specular(Figure 3.3.1) or diffuse(Figure 3.3.2). Specular reflection occurs from smooth polished surfaces and follows the Law of Reflection (the angle of incidence is equal to the angle of reflection). Diffuse reflection occurs from rough surfaces and scatters light in all directions with little dependence on the incident angle[51].
In UV reactors, reflection will take place at interfaces that do not transmit UV light (e.g., the reactor wall) and also at UV transmitting interfaces (e.g., the inside of a lamp sleeve). The type of reflection and intensity of light reflected from a surface depends on the material of the surface.

### 3.1.4 Scattering and UV Transmittance (UVT)

Scattering of light is the change in direction of light propagation caused by interaction with a particle (Figure 3.4). Particles can cause scattering in all directions, including toward the incident light source (back-scattering).
Figure 3.4: scattering of light

UV Transmittance (UVT) has also been used extensively in the literature when describing the behavior of UV light. UVT is the percentage of light passing through material (e.g., water or quartz) over a specified distance. The UVT can be calculated using Beer’s law:

\[
\%UVT = \frac{I}{I_0} \times 100 \tag{5}
\]

Where

- \( UVT \)  UV transmittance at a specified wavelength and path length
- \( I \)    Intensity of light transmitted through the sample (\( mW/cm^2 \))
- \( I_0 \)  Intensity of light incident on the sample (\( mW/cm^2 \))

UVT can also be calculated by relating it to UV absorbance using Equation(2)

\[
\%UVT = 100 \times e^{-A} \tag{6}
\]

Where

- \( UVT \)  UV transmittance at a specified wavelength and path length
- \( A \)    UV absorbance at a specified wavelength and path length
In Point Source Summation (PSS) model, the refraction, reflection, and scattering issues of UV light were not concerned within the numerical model. The program based on the numerical model simulated the CFD model under three different UV transmittance (90%, 85% and 70%) to help get a better understanding of the system performance in different UV light transmission situation.

### 3.2 Principles of UV Radiation

#### 3.2.1 UV Dose

UV dose is the integral of UV intensity during the exposure period (i.e., the area under intensity versus time curve). If the UV intensity is constant over the exposure time, UV dose is defined as the product of the intensity and the exposure time.

\[ Dosage = I \times t \quad (7) \]

Table 3.1 is a UV dose requirements developed by EPA for public water system to receive credit for inactivation of Cryptosporidium, Giardia, and viruses [1].

**Table 3.1: UV dose requirements from EPA 40 CFR 141.720(d)(1)**

<table>
<thead>
<tr>
<th>Target Pathogens</th>
<th>Log Inactivation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5</td>
</tr>
<tr>
<td>Cryptosporidium</td>
<td>1.6</td>
</tr>
<tr>
<td>Giardia</td>
<td>1.5</td>
</tr>
<tr>
<td>Virus</td>
<td>39</td>
</tr>
</tbody>
</table>

\(40\) CFR 141.720(d)(1)

Unlike chemical disinfectants, UV leaves no residual that can be monitored to determine UV dose and inactivation credit. The UV dose depends on the UV intensity (measured by UV sensors), the flow rate, and the UV transmittance (UVT). A relationship between the required UV dose and these parameters must be established and then monitored at the water treatment plant to ensure sufficient disinfections of microbial pathogens.
3.2.2 UV Intensity

UV intensity is a important property of UV light for water treatment and has the units of watts per meter squared (W/m²) [52].

The total UV intensity at a reception point $E_A$ is the total radiant power of all wave lengths passing from all incident directions onto its area, divided by the area (shown in figure 3.5)

$$E_A = \frac{Total \ Radiant \ Power}{dA}$$  \hspace{1cm} (8)

![Figure 3.5: intensity at reception point $E_A$](image)

3.3 UV Dose Distribution Calculations

In this paper, a numerical method is introduced to calculate the UV dose distribution. In this calculated dose approach, UV dose is defined as the product of UV intensity and exposure time and is expressed in SI units e.g. $J/m^2$ or $mJ/cm^2$.

Basically, the intensity of a UV lamp can be calculated by dividing the UV bulb output power (P) by the surface area. In order to produce more accurate representation of the UV dose equation, a method of Point Source Summation (PSS) was introduced for estimation of the intensity field characteristics in conventional UV disinfection systems[5][33].

In the PSS method, a lamp is simulated as a series of co-linear point sources of radiation(Figure 3.6).
The equation 9 can be derived to get intensity field of one reception point(dA) gained form one point source, as in Figure 3.7

$$E_A = \frac{P_i}{4\pi r^2} T \quad (9)$$

Where

- $P_i$ radiant power of one point source
- $T$ the transmissivity of the radiant light through all substance
- $r$ radial distance from point source to reception point
Substitute equation 3 into 9, get

\[ E_A = \frac{P_i}{4\pi r^2} e^{-\sigma r} \]  \hspace{1cm} (10)

Where

- \( P_i \) radiant power of one point source (W)
- \( r \) radial distance from point source to reception point (cm)
- \( \sigma \) the absorbance coefficient of of the substance (cm\(^{-1}\))

From equation (10) we can get that the intensity field around a point source can be accurately quantified by accounting for dissipation (divergence) and absorbance, both of which can be accurately described as a function of distance from the point source.

Shown in Figure 3.8, the intensity at any given receptor within an irradiated zone is then estimated as the sum of all intensity contributions from point sources within the system

\[ I_A = \sum_{i=1}^{n} \frac{P_i}{4\pi r_i^2} e^{-\sigma_i r_i} \]  \hspace{1cm} (11)

Where

- \( P_i \) radiant power of one point source (W)
- \( r_i \) radial distance from point source to reception point (cm)
- \( \sigma_i \) the absorbance coefficient of of the substance (cm\(^{-1}\))
- \( n \) total number of point sources within the irradiated zone

Figure 3.8: sum of all the point sources contribute to the intensity at one reception point
For a typical UV disinfection system (Figure 3.9), UV lamp is housed in a quartz sleeve. That is to say UV light go through two substances, the total distance and the total absorbance coefficient can be derived as:

$$
\sigma_i l_i = \frac{[\sigma_w (R - r_q) + \sigma_q t_q]}{\cos \theta} = \frac{[\sigma_w (R - r_q) + \sigma_q t_q]}{R} l_i
$$

(12)

Where

- $\sigma_i$ the absorbance coefficient of of the substance (cm$^{-1}$)
- $r_i$ radial distance from point source to reception point (cm)
- $\sigma_w$ the absorbance coefficient of water (cm$^{-1}$)
- $R$ radial distance from lamp axis to reception point (cm)
- $r_q$ radial distance from lamp axis to outside quartz jacket (cm)
- $\sigma_q$ the absorbance coefficient of quartz sleeve (cm$^{-1}$)
- $t_q$ quartz jacket thickness (cm)
- $l_i$ distance from $i$th point source to reception point (cm)
- $\theta$ angle of $l_i$ and $R$

![Figure 3.9: intensity field of one reception point—one point source considering quartz sleeve thickness](image)
Substitute equation (12) into (11), we can get:

\[ I_d = \sum_{i=1}^{n} \frac{P}{4\pi l_i^2} \exp \left[ -\left( \sigma_n (R - r_q) + \sigma_q r_q \right) \frac{l_i}{R} \right] \]  

(13)

Where

- \( P \) lamp output power (W)
- \( n \) number of point source
- \( \sigma_n \) the absorbance coefficient of water (cm\(^{-1}\))
- \( R \) radial distance from lamp axis to reception point (cm)
- \( r_q \) radial distance from lamp axis to outside quartz jacket (cm)
- \( \sigma_q \) the absorbance coefficient of quartz sleeve (cm\(^{-1}\))
- \( l_q \) quartz jacket thickness (cm)
- \( l_i \) distance from \( i \)th point source to reception point (cm)

Now apply a pathline S (Figure 3.10), which the reception point dA has followed throughout the time t. From the definition of dosage(equation 7), we can get:

\[ Dosage_d = \left[ \sum I_d(l, R) \right] \times t \]  

(14)

Figure 3.10: schematic representation of dosage calculation coupling with intensity field
Figure 3.10 provides a schematic representation of the terms in this equation for a system containing a single lamp. The accuracy of the predictions from the PSS model asymptotically approach a limit as the number of point sources ($n$) increases and the computational time required completing the PSS simulation increases with $n$. A limitation in the application of the discreet form of the PSS model is uncertainty regarding the number of point sources to use [36].

![Dosage distribution of one path line based on different number of point sources](image)

**Figure 3.11:** Dose distribution comparison of point source numbers' setting

Figure 3.11 could be used to show the accuracy of the PSS model as the number of point sources ($n$) increased. The dose distribution of a single pathline from a CFD model (the simulation detail was in chapter 7) was given based on different point source numbers (1~1000). When the point source number was low, the calculated dosage value fluctuated due to the "rough" cutting of the lamp. As the lamp being divided into 50~1000 parts, the dose could be maintained in a comparatively steady range, the average deviation of which was from 0.3% to 0.6%.
3.4 Reduction Equivalent Dose (RED)

Reduction Equivalent Dose (RED) is a dose quantity used for providing an indication of the potential for biological effects of the UV dose delivery by UV disinfection system.

RED is currently measured using biodosimetry, a procedure involves measuring the inactivation of a challenge microorganism after exposure to UV light in the UV reactor and comparing the results to the known UV dose-response curve of the challenge microorganism [52].

Reduction Equivalent Dose (RED) value can be derived by entering the log inactivation measured during full-scale reactor testing into the UV dose-response curve through collimated beam testing.

RED values are always specific to the following:

• the challenge microorganism used during experimental testing

• validation test conditions during full-scale reactor tests (flow rate, UVT, lamp status, and UV intensity as measured by the UV sensor)

In this paper, UV dose delivery is calculated in a semi-empirical way, as related to UV intensity as a function of certain conditions, for example, flow rate and UVT, etc. By using advanced numerical modeling approaches, a quadratic UV dose-response equation is provided below:

\[ \text{RED} = A \times (\log \text{inactivation})^2 + B \times (\log \text{inactivation}) \]  \hspace{1cm} (15)

The coefficients A and B are solved for using the previous collimated beam testing data (Atlantic UV water purifiers Mighty Pure model MP36B, MS2 phage (American Type Culture Collection [ATCC] 15597-B1), Richter 2006)[53], as the coefficients for log reduction rate of the UV dose-response curve.

3.5 Collimated Beam Testing: MS2 Phage

UV dose response data for using Male-Specific-2 Bacteriophage (MS2 phage) as challenge microorganism were prepared in order to correlate the measured log inactivation of phage with UV dosage. This was done with a collimated beam apparatus using the test water done by Environmental Energy Technologies, Inc, 2006.
The collimated beam testing procedure involves placing sample water with the challenge microorganism in an open cylindrical container and exposing the sample to collimated UV light for a predetermined amount of time. The UV dose is calculated using the measured intensity of the UV light, UV absorbance of the water, and exposure time[54]. The measured concentration of microorganisms before and after exposure provides the “response,” or log inactivation of the microorganisms from exposure to UV light, following equation:

\[ \text{Log inactivation} = \log_{10} \frac{N_0}{N} \quad (16) \]

where

- \( N_0 \) Concentration of infectious microorganisms before exposure to UV light
- \( N \) Concentration of infectious microorganisms after exposure to UV light

The recommended testing procedure for collimated beam testing of a water sample containing challenge microorganisms is listed as follows[1]:

1. Measure the A254 of the sample using a spectrophotometer that has a measurement uncertainty of 10 percent of less.
2. Place a known volume from the water sample into a petri dish and add a stir bar. Measure the water depth in the petri dish.
3. Measure the UV intensity delivered by the collimated beam with no sample present using a calibrated radiometer.
4. Calculate the required exposure time to deliver the target UV dose.
5. Block the light from the collimating tube using a shutter or equivalent.
6. Center the petri dish with the water sample under the collimating tube.
7. Unblock the light from the collimating tube and start the timer.
8. When the target exposure time has elapsed, block the light from the collimating tube.
9. Remove the petri dish and collect the sample for measurement of the challenge microorganism concentration. If the sample is not assayed immediately, store in the dark at 4 °C. Each sample should be plated in triplicate and the average microbial value for the sample calculated from the three plate replicates.
10. Re-measure the UV intensity and calculate the average of this measurement and the measurement taken in Step 3. The value should be within 5 percent of the value
measured in Step 3.

11. Calculate the UV dose applied to the sample based on experimental conditions (this should be similar to the target dose)

12. Repeat steps 1 through 11 for each replicate and target UV dose value. Repeat all steps for each water test condition replicate.

As collimated beam tests produce the following types of experimental data:

- UV Dose in units of mJ/cm²,
- Concentration of microorganisms in the petri dish prior to UV exposure (N₀) in units of pfu/mL
- Concentration of microorganisms in the petri dish after UV exposure (N) in units of pfu/mL

The UV dose-response curve can be developed for each UVT condition tested, as in Figure 3.12, which provides the coefficients A and B in equation 15.

![UV Dose Response Curve](image)

Figure 3.12: UV dose-response curve [53]
3.6 2D Simplified UV Dose Modeling

A two dimensional simplified model has been used to check the UV intensity distribution and the agreement between computed data and measured data. In this model the velocity field was assumed to be a constant value and flowed through the chamber in one direction. As shown in figure 3.13.

![Figure 3.13: configuration of 2D simplified model](image)

According to equation 13 and our simplified chamber the UV distribution was uniform along the lamp except for end effects. The contour plot in figure 3.14 indicated that the intensity field decayed rapidly in exponential fashion (shown in figure 3.15) with respect to the radial distance from lamp.

![Figure 3.14: UV intensity distribution, contour plot](image)
Figure 3.15: UV intensity distribution at specified location along the lamp, L is the length of chamber.

A series of models have been tested against the experimental values. The agreement was considerably good for 85UVT, as the average error is 2% for 12 gpm case, and 3% for 6 gpm case. Table 3.2 shows the detailed setting of this comparison. As UVT went down to 70%, scattering become an important issue for UV dose, and the neglecting of that affect in the theoretical calculation provided poor agreement with the testing data.

Table 3.2: Validation of simulation model for the experimental data

<table>
<thead>
<tr>
<th>Nominal Flow and %T</th>
<th>Sample</th>
<th>Flow (gpm)</th>
<th>%T</th>
<th>Measured RED (mJ/cm²)</th>
<th>Calculated RED (mJ/cm²)</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>85UVT</td>
<td>A</td>
<td>12.10</td>
<td>84.2</td>
<td>47.9</td>
<td>42.9</td>
<td>10.3</td>
</tr>
<tr>
<td>12 gpm</td>
<td>B</td>
<td>12.11</td>
<td>84.1</td>
<td>42.3</td>
<td>42.9</td>
<td>1.3</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>12.06</td>
<td>84.1</td>
<td>41.7</td>
<td>43.0</td>
<td>3.2</td>
</tr>
<tr>
<td>Average:</td>
<td></td>
<td>12.09</td>
<td>84.1</td>
<td>44.0</td>
<td>42.9</td>
<td>2.4</td>
</tr>
<tr>
<td>85UVT</td>
<td>A</td>
<td>5.91</td>
<td>83.9</td>
<td>84.9</td>
<td>87.6</td>
<td>3.2</td>
</tr>
<tr>
<td>6 gpm</td>
<td>B</td>
<td>5.88</td>
<td>84.2</td>
<td>85.4</td>
<td>88.4</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>5.90</td>
<td>84.2</td>
<td>86.1</td>
<td>88.1</td>
<td>2.3</td>
</tr>
<tr>
<td>Average:</td>
<td></td>
<td>5.90</td>
<td>84.1</td>
<td>85.4</td>
<td>88.0</td>
<td>3.1</td>
</tr>
</tbody>
</table>
### 3.7 3D UV Dose Modeling Based on CFD

A three dimensional computational model of the UV system has been constructed, including all major components that influence the flow patterns in the reactor. Following a steady-state CFD simulation by solving governing flow equations (Navier-Stokes and turbulence equations), this simulation model resulted in a prediction of point velocities across the interior of the UV system for the specified inlet flow rate. The CFD simulation methods used for UV disinfections system were presented in Chapter 5.

To calculate the estimated UV dose with the numerical method in section 3.2, trajectory prediction of certain numbers of microbes (in this project, the number was usually from 1500 to 5000, depending on the complexity of the UV system) through the UV reactors was required. In the post-processing section of the CFD simulation, particle tracking method was used to provide the path line data for each particle as x, y, z coordinates were predicted as a function of time.

When obtaining the path line data from CFD software, although most path lines indicated a reasonable random microbial path (Figure 3.5.1) and could be used to summon up the \( I/\theta \) data and get that particle intensity based on intensity model (equation 13, 14). There were always certain path lines could not predict the true trajectories of the microbes. For example, some stoped at stagnation points (Figure 3.5.2) and some was following a "circulation form" that resulted in a very high dose if it was to be used to calculate the particle intensity.

<table>
<thead>
<tr>
<th>Nominal Flow and %T</th>
<th>Sample</th>
<th>Flow (gpm)</th>
<th>%T</th>
<th>Measured RED (mJ/cm²)</th>
<th>Calculated RED (mJ/cm²)</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>70UVT</td>
<td>A</td>
<td>11.58</td>
<td>70.4</td>
<td>25.3</td>
<td>40.0</td>
<td>58.1</td>
</tr>
<tr>
<td>12 gpm</td>
<td>B</td>
<td>11.79</td>
<td>70.4</td>
<td>37.3</td>
<td>39.0</td>
<td>4.6</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>11.69</td>
<td>70.4</td>
<td>31.3</td>
<td>39.5</td>
<td>31.4</td>
</tr>
</tbody>
</table>
Figure 3.16.1: path line that predict the true particle trajectory
Figure 3.16.2: path line stops at stagnation point
Figure 3.16.3: path line that falls into circulation form

The "stagnation path line" usually provided a very small dose. However, concerning those particles had lost all the velocity and been trapped within the chamber, never the less stagnation is due to the numerical model feature(mesh size limitation, algorithm flaw), the program only counted pathlines that reaches the outlet of the chamber to avoid counting "stagnation pathline".

An ideal log reduction, $N_0$ was set to avoid "circulation path line" which in the other hand, had a tremendous high dose. As any log inactivation, $\log I$ of each path line that was larger than $N_0$ would be considered as a circulation form, and its $\log I$ would be set to $N_0$.

The number of $N_0$ was set to $10^6$ which meant the reduction of bacteria reaches 99.9999% [55].

28
Then we could get the average \((\log \text{inactivation})_{\text{avg}}\):

\[
(\log \text{inactivation})_{\text{avg}} = \frac{\sum_{i=1}^{\text{pnum}} \log \text{inactivation}}{\text{pnum}}
\]  

(17)

Where

\text{pnum} \quad \text{Number of particles in path line data}

Substitute into equation 15, the RED value becomes:

\[
\text{Dosage} = A \times (\log \text{inactivation})_{\text{avg}}^2 + B \times (\log \text{inactivation})_{\text{avg}}
\]  

(18)
Chapter 4: Computational Fluid Dynamics

4.1 Overview

Computational fluid dynamics (CFD) is one of the branches of fluid mechanics that uses numerical methods and algorithms to solve and analyse problems that involve fluid flows. Computers are used to perform the millions of calculations required to simulate the interaction of fluids and gases with the complex surfaces used in engineering[56][57].

4.2 Fluid Flow Motion

In order to formulate a mathematically tractable problem, we considered three dimensional flows passing inside the chamber. The governing equations are the mass, momentum and energy equations.

Continuity Equation

The conservation of mass law applied to a fluid passing through an infinitesimal, fixed control volume yield the following equation of continuity:

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \quad (19)
\]

Momentum Equation

Newton’s Second Law applied to a fluid passing through an infinitesimal, fixed control volume yield the following momentum equation:

\[
\frac{\partial}{\partial t}(\rho \vec{v}) + \nabla \cdot \rho \vec{v} \vec{v} = \rho \vec{f} + \nabla \cdot \Pi \vec{g} \quad (20)
\]

Energy Equation

The First Law of Thermodynamics applied to a fluid passing through an infinitesimal, fixed control volume yield the following energy equation:

\[
\frac{\partial E}{\partial t} + \nabla \cdot E \vec{v} = \frac{\partial Q}{\partial t} - \nabla \cdot q + \rho \vec{f} \cdot \vec{v} + \nabla \cdot \left( \Pi \vec{g} \cdot \vec{v} \right) \quad (21)
\]

4.3 Commercial CFD Software

In this project two commercial CFD software was used to get the numerical flow field data and two individual ports were set up to read the data generated from these two software.
FLUENT is the primary product of Fluent, Inc. The software code is based on the finite volume method on a collocated grid. FLUENT offers a wide array of physical models that can be applied to a wide array of industries.

CFX is a commercial CFD program used to simulate fluid flow in a wide variety of applications. CFX allows engineers to test systems in a virtual environment. It is highly scalable and has been shown to maintain nearly linear scalability to as many as 500 processors.

4.4 CFD Results and Postprocessing

4.4.1 Hydraulic Design Considerations

The major elements that should be considered in the hydraulic design of a UV reactor are: dispersion, turbulence, effective volume, residence time distribution, and flow rate.

4.4.2 Turbulence

In addition to plug flow characteristics, the ideal UV reactor has a flow that is turbulent radially from the direction of flow, to eliminate dead zones. This radially turbulent flow pattern promotes uniform application of UV radiation. A negative of having a radially turbulent flow pattern is that some axial dispersion results, thus disrupting the plug flow characteristics. Techniques such as misaligning the inlet and outlet, and using perforated stilling plates, have been used to accommodate the contradicting characteristics of plug flow and turbulence.

4.5 CFD Solution Overview

Based on the ideal UV reactor model setting, hydrodynamics and UV dose characteristics can be captured through particle tracking or streamline data.

Steady-state, 3D standard $k–\varepsilon$ solver was used to solve the designed systems. After continuous phase flow field was solved, particle tracking or streamline could be utilized to obtain position vectors with respect to residence time. A typical UV disinfection system was simulated and discussed at chapter 7.
Chapter 5: Dosage Calculation Program

5.1 Overview

Figure 5.1: logo of UV calculation program
The software program for UV reactors was designed to calculate 3D fluent rate distributions, UV dosage distribution and relevant Reduction Equivalent Dosage (RED) value based on multiple point source summation method.

The program had advanced Human–computer interaction (HCI) module including different reactor design choosing, UV power input, boundary conditions and other essential variables input for calculation. By reading the designated particle tracking file, it would generate resultant files containing properties of injection particles, UV intensity, and UV dose, etc. The final program was an installable PC-based package and available for future simulation of other single lamp UV disinfection models.

5.2 Visual Basic Programming Language

Visual Basic (VB) is the third-generation event-driven programming language and integrated development environment (IDE). The version Visual Basic 2005 (VB 8.0) is used in this thesis to develop the graphical user interface (GUI).

By using the powerful IDE provided by VB, the whole program was constructed by various classes to serve the object-oriented applications, in our case, a button clicked event to read the input data and perform calculation.
Table 5 shows the list of classes (or functions) that has been used in the main program code.

<table>
<thead>
<tr>
<th>Table 5: Main program classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>.AtEndOfStream</td>
</tr>
<tr>
<td>.Checked</td>
</tr>
<tr>
<td>.Close</td>
</tr>
<tr>
<td>.CombinePath</td>
</tr>
<tr>
<td>.Enabled</td>
</tr>
<tr>
<td>.FileName</td>
</tr>
<tr>
<td>.FileSystemObject</td>
</tr>
<tr>
<td>.Filter</td>
</tr>
<tr>
<td>.FilterIndex</td>
</tr>
<tr>
<td>.Hide</td>
</tr>
<tr>
<td>.InitialDirectory</td>
</tr>
<tr>
<td>.Left</td>
</tr>
<tr>
<td>.Length</td>
</tr>
<tr>
<td>.Maximum</td>
</tr>
<tr>
<td>.Minimum</td>
</tr>
<tr>
<td>.MsgBox</td>
</tr>
</tbody>
</table>
5.3 UV Dose Program Interface

The dosage calculation program had a graphical user interface (GUI), and its main console window had four components: Configuration setting panel, Output option panel, I/O—Calculation setting, and config-graphic window (Figure 5.2).

![Configuration setting panel, Output option panel, I/O—Calculation setting, and config-graphic window](image)

Figure 5.2: setting of the main program window

To calculate the dose distribution, first thing was to input dimension information of disinfection system through Configuration setting. The lamp location, length, and power, the outlet location was required(Figure 5.3, 5.4). During the setting of each apparatus, user could see more information from the config-graphic window.
Figure 5.3: apparatus setting: lamp location  
Figure 5.4: apparatus setting: outlet location

In Output option panel, user could chose whether to have detailed calculation report saved, or just a general report. Figure 5.5 is an example of a detailed report, which includes initial and final positions of all particles, time spending, dosage and RED values, etc.

<table>
<thead>
<tr>
<th>Particle</th>
<th>Initial x</th>
<th>Initial y</th>
<th>Initial z</th>
<th>Final x</th>
<th>Final y</th>
<th>Final z</th>
<th>Final time</th>
<th>Dosage</th>
<th>Kred</th>
<th>Nd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.31824398</td>
<td>2.91484166</td>
<td>1.27751263</td>
<td>-0.00572377</td>
<td>0.00601186</td>
<td>-0.12192</td>
<td>7.48237085</td>
<td>101.954815</td>
<td>9.78863847</td>
<td>16.380897</td>
</tr>
<tr>
<td>1</td>
<td>1.155718</td>
<td>2.80797606</td>
<td>1.07797649</td>
<td>-1.634-66</td>
<td>-0.02139285</td>
<td>0.03632698</td>
<td>5.78570771</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.4420519</td>
<td>2.87489441</td>
<td>0.78587224</td>
<td>0.637805223</td>
<td>0.0436355</td>
<td>-0.12192</td>
<td>5.55448374</td>
<td>71.6125455</td>
<td>3.62169738</td>
<td>233.5962</td>
</tr>
<tr>
<td>3</td>
<td>1.67174412</td>
<td>2.88564935</td>
<td>0.60776122</td>
<td>-3.01239269</td>
<td>0.00000000</td>
<td>-0.12192</td>
<td>6.60612271</td>
<td>54.9017152</td>
<td>1.42971219</td>
<td>75.641079</td>
</tr>
<tr>
<td>4</td>
<td>1.51166003</td>
<td>2.60794232</td>
<td>1.11966906</td>
<td>-4.00701241</td>
<td>0.00036164</td>
<td>-0.12192</td>
<td>7.39055859</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Kred_T | 3.96450029 | Bolt | 325.54793683 | RED: | 80.0592896 | m: | 5 |

Figure 5.5: detailed output result from dose calculation

As for general report, the output file only provides total RED and particle reading and calculation information (Figure 5.6).

<table>
<thead>
<tr>
<th>export00t</th>
</tr>
</thead>
</table>

RED value:
UVT70: 23.61
UVT85: 40.4
UVT90: 72.86

Total particle number: 250
Valid particle number: 197
Figure 5.5: general output result from dose calculation(dose unit: mJ/cm^2)

At I/O—calculation panel, after designating path line data, program would start calculation after user clicked the "Dosage Calc" button.
Chapter 6: Numerical Simulation of Commercial Model

6.1 Model Specification

Although various design (Lamp placement, baffles, and inlet and outlet conditions) would affect dose delivery, a 3 dimensional computational model of the UV system including major components that influence the flow patterns in the reactor was not very complicated in this commercial model (Atlantic UV water purifiers Mighty Pure model MP36B) case.

Figure 6.1.1: simplified geometry of commercial model
Figure 6.1.2: dimension of commercial model
As shown in Figure 6.1, the computational domain of this model was a long cylinder region, with inlet and outlet at the top, and the quartz tube was placed in the center of the chamber.

6.2 Create Geometry
The creation of geometry was done by ANSYS CFX 11.0 as shown in Figure 6.2.

Figure 6.2: geometry of commercial model created with ANSYS CFX 11.0
6.3 Mesh Geometry

Shown in figure 6.3, a spacing of .125 was used to get the mesh model. As a normal PC was used to run this simulation, one million elements was a decent quantity considering both the accuracy and the efficiency.

Figure 6.3: mesh geometry created with CFX MESH

Total Number of Nodes = 423650
Total Number of Elements = 1784387
Total Number of Tetrahedrons = 1473977
Total Number of Prisms = 310410
Total Number of Faces = 62590
Minimum Orthogonal Angle [degrees] = 33.6
Maximum Aspect Ratio = 22.4
Maximum Mesh Expansion Factor = 34.5

6.4 Specify Boundary Types

6.4.1 Inlet

FLOW DIRECTION:
Option = Normal to Boundary Condition

FLOW REGIME:
Option = Subsonic

MASS AND MOMENTUM:
Mass Flow Rate = 0.757 [kg s⁻¹]
Option = Mass Flow Rate

6.4.2 Outlet

FLOW REGIME:
Option = Subsonic

MASS AND MOMENTUM:
Option = Average Static Pressure
Relative Pressure = 0 [Pa]
PRESSURE AVERAGING:
Option = Average Over Whole Outlet

6.4.3 Wall

WALL INFLUENCE ON FLOW:
Option = No Slip

WALL ROUGHNESS:
Option = Smooth Wall

6.5 Solver Set up

The Equations Solved in This Calculation
Subsystem: Momentum and Mass
   U-Mom
   V-Mom
   W-Mom
   P-Mass
Subsystem: TurbKE and Diss.K
   K-TurbKE
   E-Diss.K
6.6 Analyze Results

The simulation model converged after about 1800 numbers of iteration, which took about four hours. One basic strategy was to check the mass flow balance. The outlet mass flow was -0.757 kg/s, which was equal to the inlet (0.757 kg/s). The minus sign meant the flow was flowing out of the outlet plane.

Figure 6.4 shows the visualized path line data generated from the simulation. 250 particles has been injected into the chamber through inlet. The reason to pick that number was to give a more clear vision of the flow field. For the 1500 particle would almost fill out the entirely domain.

![Figure 6.4.1 path line (250 particles) generated of the commercial model](image)

From the enlarged image of path line(Figure 6.4.2), we could see that there were certain circulation near the inlet. As the flow moving forward to the outlet, the flow field turned out to be more uniform(Figure 6.4.3). One optimization of the UV chamber was to manipulate the inlet/outlet, to get a more complicated flow (for example, more circulation around the lamp). With the extension of particle residence time, the higher and more uniform dosage could be achieved [3].
Figure 6.4.2 path line (250 particles) details of inlet

Figure 6.4.3 path line (250 particles) details of outlet
6.7 Dose Distribution Calculation

Figure 6.5.1 shows the example of path line data generated from the CFD simulation of commercial model. Each column represented the particle's X Y Z location, the residence time, and the particle ID (in figure 6.5.1 it was particle number 0). Then the data was put into the calculation program, with the correct setting of configuration information (Figure 6.5.2), and the dosage distribution results were generated(Figure 6.5.3, 6.5.4, 6.5.5).
Figure 6.5.2 dose calculation configuration setting

Figure 6.5.3 dose calculation complete message
<table>
<thead>
<tr>
<th>Particle</th>
<th>Initial $x$</th>
<th>Initial $y$</th>
<th>Final $x$</th>
<th>Final $y$</th>
<th>Final $z$</th>
<th>Final Z</th>
<th>Dosage</th>
<th>Xrd</th>
<th>Nl</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00292319</td>
<td>32, 9748804</td>
<td>9.044859</td>
<td>0.106</td>
<td>0.0203194</td>
<td>0.695911</td>
<td>29321026</td>
<td>6.713564</td>
<td>0.8862</td>
<td></td>
</tr>
<tr>
<td>1.04074513</td>
<td>32, 98098838</td>
<td>0.038732</td>
<td>0.106</td>
<td>0.0201275</td>
<td>7.903443</td>
<td>40410024</td>
<td>9.28210305</td>
<td>0.6521</td>
<td></td>
</tr>
<tr>
<td>2.05215634</td>
<td>32, 99689213</td>
<td>0.036795</td>
<td>0.106</td>
<td>0.0462061</td>
<td>8.618942</td>
<td>123730710</td>
<td>11.73821034</td>
<td>0.6561</td>
<td></td>
</tr>
<tr>
<td>3.01116135</td>
<td>32, 99699213</td>
<td>0.036995</td>
<td>0.106</td>
<td>0.0582047</td>
<td>7.396345</td>
<td>203972658</td>
<td>11.24277933</td>
<td>0.6591</td>
<td></td>
</tr>
<tr>
<td>4.00693648</td>
<td>32, 39309294</td>
<td>-0.036414</td>
<td>0.106</td>
<td>0.036334</td>
<td>5.470706</td>
<td>10309705</td>
<td>4.77498314</td>
<td>17.39321061</td>
<td></td>
</tr>
<tr>
<td>5.0-1736725</td>
<td>32, 94910495</td>
<td>0.061696</td>
<td>0.106</td>
<td>0.053366</td>
<td>7.48414</td>
<td>29341067</td>
<td>1.71115331</td>
<td>19.44673518</td>
<td></td>
</tr>
<tr>
<td>6.0-20609229</td>
<td>32, 39607905</td>
<td>0.0999952</td>
<td>0.106</td>
<td>0.059696</td>
<td>7.43241</td>
<td>22360772</td>
<td>1.37526259</td>
<td>439660.5962</td>
<td></td>
</tr>
<tr>
<td>7.0-29323837</td>
<td>32, 38799276</td>
<td>0.048962</td>
<td>0.106</td>
<td>0.047628</td>
<td>9.408894</td>
<td>34778582</td>
<td>1.36528202</td>
<td>103943.9822</td>
<td></td>
</tr>
<tr>
<td>8.0-11570359</td>
<td>32, 39840877</td>
<td>-0.03527</td>
<td>0.106</td>
<td>0.034171</td>
<td>8.64894</td>
<td>27379897</td>
<td>1.63941882</td>
<td>25407.0693</td>
<td></td>
</tr>
<tr>
<td>9.0-08269329</td>
<td>32, 93748003</td>
<td>-0.082283</td>
<td>0.106</td>
<td>0.047724</td>
<td>6.900495</td>
<td>27369776</td>
<td>1.64109064</td>
<td>22836.0553</td>
<td></td>
</tr>
<tr>
<td>10.0-11592883</td>
<td>32, 93530708</td>
<td>0.0532918</td>
<td>0.106</td>
<td>0.0481618</td>
<td>7.83239</td>
<td>159301559</td>
<td>1.12070108</td>
<td>74446.6849</td>
<td></td>
</tr>
<tr>
<td>110.0-14291111</td>
<td>32, 26311029</td>
<td>-0.0276615</td>
<td>0.106</td>
<td>0.054238</td>
<td>7.42668</td>
<td>396110131</td>
<td>4.35144905</td>
<td>46.65641594</td>
<td></td>
</tr>
<tr>
<td>120.0-17481323</td>
<td>32, 32849393</td>
<td>-0.0482312</td>
<td>0.106</td>
<td>0.0491929</td>
<td>11.13421</td>
<td>652294075</td>
<td>3.2990971</td>
<td>403599317</td>
<td></td>
</tr>
<tr>
<td>130.0-14202693</td>
<td>32, 30404095</td>
<td>0.0377294</td>
<td>0.106</td>
<td>0.0611707</td>
<td>6.111481</td>
<td>141049326</td>
<td>0.88050341</td>
<td>131855.5326</td>
<td></td>
</tr>
<tr>
<td>130.0-13316111</td>
<td>32, 40451949</td>
<td>0.0856306</td>
<td>0.106</td>
<td>0.0547473</td>
<td>7.238122</td>
<td>21480362</td>
<td>1.27309264</td>
<td>53542.29428</td>
<td></td>
</tr>
<tr>
<td>140.0-96995449</td>
<td>32, 44599904</td>
<td>0.0420907</td>
<td>0.106</td>
<td>0.0587620</td>
<td>7.801232</td>
<td>526950767</td>
<td>2.02699951</td>
<td>1497.2783</td>
<td></td>
</tr>
<tr>
<td>150.0-04776299</td>
<td>32, 42972341</td>
<td>-0.0031753</td>
<td>0.106</td>
<td>0.0543977</td>
<td>7.5093155</td>
<td>66379202</td>
<td>3.40835004</td>
<td>5433785311</td>
<td></td>
</tr>
<tr>
<td>160.0-0.7180066</td>
<td>32, 3698819</td>
<td>-0.0024513</td>
<td>0.106</td>
<td>0.062431</td>
<td>6.576812</td>
<td>20323776</td>
<td>6.02035693</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 6.5.4 example of dose calculation output, detailed report, uvt70**

<table>
<thead>
<tr>
<th><strong>export00t</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RED value:</strong></td>
</tr>
<tr>
<td>UVT70:</td>
</tr>
<tr>
<td>UVT85:</td>
</tr>
<tr>
<td>UVT90:</td>
</tr>
</tbody>
</table>

|**Total particle number:** | 250 |
|**Valid particle number:** | 197 |

**Figure 6.5.5 example of dose calculation output(mJ/cm^2), general report, uvt70**
Based on output data, it was possible to validate energy efficiency of the unit with specific design dimensions, etc.

Figure 6.6 bubble chart of initial position vs. dose distribution uvt70

Figure 6.6 is the bubble chart of the dose distribution of each particles showed at their initial input location. The size of the bubble was corresponding to the dose of each particle. It could be shown that the regions near the sidewall are critical for its effect on turbulent mixing [43].
Figure 6.7 calculated RED value for commercial model under different flow rate and UVTs.

Figure 6.7 is the RED value for this commercial model under different flow rate. It was clear that the slower the flow flows, better dose the system get. But to get the best energy efficiency results, an optimized setting could be found between the chamber's dimensions and flow rate.
Figure 6.8 calculated RED value comparing with testing RED value, 85UVT

Figure 6.8 is the chart of numerical data versus experimental data at 85UVT[53]. Most testing data (except 7-26,6gpm) was within the 5% error range.

Table 6 shows the error between numerical and testing results at 70UVT. Similar to the theoretical model, result was not very good as the mathematical model was affected by the light scattering issue.

<table>
<thead>
<tr>
<th>Flow (gpm)</th>
<th>UVT70(sim) (mJ/cm²)</th>
<th>UVT70(6-21-06) (mJ/cm²)</th>
<th>%Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>23.61</td>
<td>25.3</td>
<td>7.2</td>
</tr>
<tr>
<td>6</td>
<td>40.1</td>
<td>56.5</td>
<td>40.9</td>
</tr>
</tbody>
</table>
Chapter 7. UV Lamp Aging Evaluation

7.1 UV Lamp Overview

UV light can be produced by the following variety of lamps:

- LP mercury vapor lamps
- Low-pressure high-output (LPHO) mercury vapor lamps
- MP mercury vapor lamps
- Electrode-less mercury vapor lamps
- Metal halide lamps
- Xenon lamps (pulsed UV)
- Eximer lamps
- UV lasers
- Light emitting diodes (LEDs)

The commercial models being discussed generally use LP, LPHO mercury vapor lamps. Therefore, this paper focused on these lamp technologies. Table 4.1 shows the characteristics of typical mercury vapor lamp.

Table 7.1: Typical mercury vapor lamp characteristics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Low-pressure</th>
<th>Low-pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High-output</td>
<td></td>
</tr>
<tr>
<td>Germicidal UV Light</td>
<td>Monochromatic at 254 nm</td>
<td>Monochromatic at 254 nm</td>
</tr>
<tr>
<td>Mercury vapor Pressure (Pa)</td>
<td>Approximately 0.93 (1.35x10^-4 psi)</td>
<td>0.18 – 1.6 (2.6x10^-5 – 2.3x10^-4 psi)</td>
</tr>
<tr>
<td>Operating Temperature (° C)</td>
<td>Approximately 40</td>
<td>60 – 100</td>
</tr>
<tr>
<td>Electrical Input [watts per centimeter (W/cm)]</td>
<td>0.5</td>
<td>1.5 – 10</td>
</tr>
<tr>
<td>Germicidal UV Output (W/cm)</td>
<td>0.2</td>
<td>0.5 – 3.5</td>
</tr>
<tr>
<td>Electrical to Germicidal UV Conversion Efficiency (%)</td>
<td>35 – 38</td>
<td>30 – 35</td>
</tr>
<tr>
<td>Arc Length (cm)</td>
<td>10 – 150</td>
<td>10 – 150</td>
</tr>
<tr>
<td>Lifetime [hour (hr)]</td>
<td>8,000 – 10,000</td>
<td>8,000 – 12,000</td>
</tr>
</tbody>
</table>
Low-pressure and Low-pressure High-output operational advantages:

- Higher germicidal efficiency; Nearly all output at 254 nm
- Smaller power draw per lamp (less reduction in dose if lamp fails)
- Longer lamp life

### 7.2 UV Lamp Aging

Lamp degradation is a function of lamp powers in operation, number of on/off cycles, power applied per unit (lamp) length, water temperature, and heat transfer from lamps.

Previous findings from ongoing research into lamp aging at water and wastewater UV facilities shows that LPHO and MP lamp aging is non-uniform with respect to axial and horizontal output and varies greatly from lamp to lamp \[58][59]. The lamp aging study by Mackey et al. is still ongoing, and any future findings from this or other studies should be evaluated and considered once results are available.

As suggested by EPA \[10\], lamp aging can be accounted for with the fouling/aging factor in the design of the UV facility through pilot- or demonstration-scale testing.

During design, the engineer and UV manufacturer will typically estimate a “fouling factor” and an “aging factor” for the reactor. The fouling factor is defined as the fraction of UV light passing through a fouled sleeve as compared to a new sleeve. The aging factor is the fraction of UV light emitted from aged lamps and sleeves at the end of the specified useful life compared to UV light emitted from new lamps and sleeves. The “fouling/aging factor” is equal to the fouling factor multiplied by the aging factor and typically ranges from 0.4 to 0.9 \[60\], then it can be used in validation testing to ensure the system meets the required dose in a fouled and/or aged condition as:

\[
\text{UV dose with clean lamps} \times \text{fouling factor} \times \text{aging factor} = \text{required UV dose} \quad (22)
\]
7.3 Lamp Aging Evaluation

Although specific fouling rate and optimal cleaning protocol for any given application couldn't be predicted with existing empirically-proven, mathematical equations, the numerical modeling of UV intensity (equation 13) could be derived to estimate the lamp output power. [29]

\[
P_e = \frac{I_m}{\sum_{i=1}^{n} \exp\left(-\frac{\left[\sigma_w (R - r_q) + \sigma_q \tau_q\right] I_i}{4\pi I_f^2}\right)}
\]

(23)

where

- \(P_e\) estimated lamp power (W)
- \(I_m\) UV intensity measured radiometrically (mWcm\(^{-2}\))
- \(n\) number of point source
- \(\sigma_w\) the absorbance coefficient of water (cm\(^{-1}\))
- \(R\) radial distance from lamp axis to reception point (collimator) (cm)
- \(r_q\) radial distance from lamp axis to outside quartz jacket (cm)
- \(\sigma_q\) the absorbance coefficient of quartz sleeve (cm\(^{-1}\))
- \(I_q\) quartz jacket thickness (cm)
- \(I_i\) distance from \(^i\) th point source to reception point (collimator) (cm)
Figure 7.1.1: schematic representation of estimation of lamp output power apparatus

By comparing the predicted power with the lamp full power, it was possible to get more accurate aging/fouling factor using the combination of numerical modeling and carefully designed testing, using:

\[ \text{UV dose with clean lamps} \times \text{fouling/aging factor} \times \frac{P_c}{P} = \text{required UV dose} \quad (24) \]
Chapter 8: Aging Evaluation Program

8.1 Lamp Aging Evaluation Interface

![Diagram of lamp aging evaluation interface](image)

Figure 8.1: setting for age evaluation main window

The aging evaluation calculator was pretty straight forward, for it was a reversed intensity calculation solver (Figure 8.1). From the system config panel user could input UV disinfection system information like chamber diameter, collimator location, lamp full power and the reading from the intensity meter. Then in the lamp location panel, dimension and location of lamp was required(Figure 8.2).
Although the only way to prove the accuracy of the numerical model was comparing it with the experimental data, which was not part of this thesis. By using data provided by the previous study (measured intensity [30]) and manufacturer (lamp dimensions and full power, Atlantic Ultraviolet Corporation), this calculator provided a reasonable result.
Figure 8.3: calculated result for aging evaluation

From the given intensity and full power, the modeled UV disinfection system gave out a 78% germicidal output. Considering validation testing usually chose 0.75 as a reduced factor of lamp output [60]. The 0.78 shows our fictional unit was under normal working condition.
Chapter 9: Summary and Conclusion

Ultraviolet disinfection system is the most frequently applied physical water treatment process. The Multiple Point Source Summation (MPSS) method has been developed to estimate the radiation intensity field, which could be treated as a summation of projected particles' received radiation from a series of co-linear point source. Computational Fluid Dynamics (CFD) has been used to predict the flow field of the whole system and provide path line data for UV dose distribution calculation, done by the graphical user interface program.

A lamp aging factor calculator has also been developed from the reversing intensity model, to help checking the resident lamp output power during the validation testing or for operation maintenance.

A typical UV disinfection unit model has been built up using commercial CFD software. The combination of hydrodynamics simulation with radiation distribution modeling showed a good results comparing to the experimental testing at 85% UV Transmittance.

For the future study of reactor optimal design, following further work can be done: a new point sources summation method can be developed considering the effects of light scattering, reflection, and deflection, etc to better estimate the unit performance under low UV Transmittance (75%). A multiple lamps UV distribution calculator can be developed based on the single bulb model with an efficient algorithm to incorporate the multiple effects of lamp radiation. More ports can be add up to accept numerical data from other commercial CFD software. And experimental testing of lamp output power need to be done to help validate the aging evaluation program to minimize the disagreement between physical measurement and numerical prediction.
Reference


[38]Bolton, J.R., Calculation of Ultraviolet Fluency Rate Distributions in an Annular Reactor: Significance of Refraction and Reflection, Water Research 34:3315–3324, 1999


[58] Stubley, G.D., Mysteries of Engineering Fluid Mechanics, 2003


Appendix

Appendix A: Calculated Dose Distribution Comparison

![Graph showing dosage distribution of one path line based on different number of point sources]

<table>
<thead>
<tr>
<th>Number of Point Source</th>
<th>UVT 70</th>
<th>UVT 80</th>
<th>UVT 90</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26.55</td>
<td>49.88</td>
<td>88.79</td>
</tr>
<tr>
<td>2</td>
<td>23.41</td>
<td>48.87</td>
<td>81.86</td>
</tr>
<tr>
<td>5</td>
<td>28.58</td>
<td>50.45</td>
<td>82.33</td>
</tr>
<tr>
<td>10</td>
<td>30.35</td>
<td>50.32</td>
<td>83.61</td>
</tr>
<tr>
<td>20</td>
<td>30.3</td>
<td>49.72</td>
<td>83.73</td>
</tr>
<tr>
<td>30</td>
<td>29.74</td>
<td>49.71</td>
<td>82.85</td>
</tr>
<tr>
<td>50</td>
<td>29.73</td>
<td>49.70</td>
<td>82.73</td>
</tr>
<tr>
<td>70</td>
<td>29.73</td>
<td>49.70</td>
<td>82.72</td>
</tr>
<tr>
<td>100</td>
<td>29.73</td>
<td>49.70</td>
<td>82.72</td>
</tr>
<tr>
<td>150</td>
<td>29.73</td>
<td>49.70</td>
<td>82.72</td>
</tr>
<tr>
<td>200</td>
<td>29.73</td>
<td>49.70</td>
<td>82.72</td>
</tr>
<tr>
<td>300</td>
<td>29.73</td>
<td>49.70</td>
<td>82.72</td>
</tr>
<tr>
<td>500</td>
<td>29.73</td>
<td>49.70</td>
<td>82.72</td>
</tr>
<tr>
<td>1000</td>
<td>29.73</td>
<td>49.70</td>
<td>82.72</td>
</tr>
</tbody>
</table>
Appendix B: Calculated RED Data of Commercial Model under Different Flow Rate

![Graph showing RED value for commercial model]

<table>
<thead>
<tr>
<th>Mass Flow Rate</th>
<th>UV790</th>
<th>UV785</th>
<th>UV770</th>
</tr>
</thead>
<tbody>
<tr>
<td>12GPM</td>
<td>72.36</td>
<td>40.4</td>
<td>23.61</td>
</tr>
<tr>
<td>9GPM</td>
<td>120.11</td>
<td>65.65</td>
<td>36.43</td>
</tr>
<tr>
<td>6GPM</td>
<td>126.88</td>
<td>69.7</td>
<td>40.1</td>
</tr>
</tbody>
</table>
Appendix C: VB Dose Distribution Calculation Program Code

Option Explicit On
Option Strict On
Imports System
Imports System.IO
Imports System.IO.Stream
Imports System.Math
Imports System.Reflection
Imports System.Runtime.Serialization

Public Class UVcalcMain

'********************program control/particle location variables********************
Dim pi As Double = 3.14159265 'define pi
Dim aa As Double = 1.352 'coefficients for ax^2+bx+c=0 (log reduction rate)
Dim bb As Double = 14.834 ' 

Dim maxlength As Integer = 53
Dim No As Integer = 1000000

Dim px, py, pz, time, period As Double 'particle streamline data input
Dim px_last As Double = 0 'save the last data for calculation and output
Dim py_last As Double = 0 ' 
Dim pz_last As Double = 0 ' 
Dim time_last As Double = 10000 ' 

Dim p_bound As Double = 0 'the outlet location
Dim lcenter_h As Double = 0 'Lamp center horizontal distance
Dim lcenter_v As Double = 0 'Lamp center vertical distance
Dim laxis_low, laxis_high As Double 'total lamp length = from low to high

Dim RRz As Double
Dim Rx, Ry, Rz As Double 'designate streamline data to different output set
Dim Rx_last As Double = 0 'save the last data for calculation and output
Dim Ry_last As Double = 0 ' 
Dim Rz_last As Double = 0 ' 

Dim inputData As String
Dim c2fuent As Boolean

'****************************dosage/intensity variables****************************

Dim pnum As Integer = 0
Dim pnum_last As Integer = 0
Dim pnum_T As Integer = 0
Dim intensity99, intensity85, intensity70 As Double
Dim dosage99, dosage85, dosage70 As Double
Dim Tintensity99, Tintensity85, Tintensity70 As Double
Dim Xred99, Xred85, Xred70 As Double
Dim Xred_T99, Xred_T85, Xred_T70 As Double
Dim red99, red85, red70 As Double
Dim Ni99, Ni85, Ni70 As Double
Dim Ni_T99 As Double = 0
Dim Ni_T85 As Double = 0
Dim Ni_T70 As Double = 0
```plaintext
Dim Lamp_axis As Char  'the axis lamp was paralleled to
Dim out_plane As Char  'out_plane indicates the axis normal to the outlet
Dim out_bound As Double 'outlet location boundary value
Dim S lamp As Double   'total lamp power

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles InputVelocity.Click
    Dim openFileDialog1 As New OpenFileDialog()
    openFileDialog1.InitialDirectory = "c:"
    openFileDialog1.Filter = ".csv files*.csv files*.fvp files*.fvp files*.All files*.*"
    openFileDialog1.FilterIndex = 2
```
openFileDialog1.RestoreDirectory = True

    Try
        myStream = openFileDialog1.OpenFile()
        If (myStream IsNot Nothing) Then
            nameOfFile = openFileDialog1.FileName
            MsgBox("you have chosen to open: ", & nameOfFile.ToString())
            Me.InputVelocity.Text = "You have chosen to save the input data at:" & vbCrLf & vbCrLf & nameOfFile
        End If
    Catch Ex As Exception
        MessageBox.Show("Cannot read file from disk. Original error: ", & Ex.Message)
    Finally
    ' Check this again, since we need to make sure we didn't throw an exception on open.
    If (myStream IsNot Nothing) Then
        '**************************get the file name of the input velocity data(store the name in FileName)**************************

        Dim I As Integer
        For I = Len(nameOfFile) To 1 Step -1
            If CBool(InStr("", Mid$(nameOfFile, I, 1))) Then Exit For
        Next
        FileName = Mid$(nameOfFile, I + 1, Len(nameOfFile) - I)

        Dim P As Integer
        For P = Len(FileName) To 1 Step -1
            If CBool(InStr(".", Mid$(FileName, P, 1))) Then Exit For
        Next
        FileName = Strings.Left(FileName, P - 1)
        'MsgBox(FileName.ToString)
        'get the extension of file name
        GetFileExt = Mid$(nameOfFile, InStrRev(nameOfFile, ",") + 1, Len(nameOfFile) -
        InStrRev(nameOfFile, ",") + 1, Len(nameOfFile) -
        InStrRev(nameOfFile, ",") - 1)
        If GetFileExt = "csv" Then
            CheckBox1.Enabled = True
        End If
        'End If
        'MsgBox(GetFileExt)
    End If

    '**************************

End Try

End If

End Sub

Private Sub Button1_Click_2(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
'*******************import configuration data*******************

out_bound = Convert.ToDouble(TextBox2.Text)  'import output location and convert from inch
to cm
out_bound *= 2.54
'MsgBox("out_bound = " & out_bound)
'out_radius = Convert.ToDouble(TextBox4.Text)  'import outlet radius and convert from inch to
cm
' out_radius *= 2.54
lcenter_h = Convert.ToDouble(TextBox5.Text)  'import Lamp center horizontal distance and
convert from inch to cm
lcenter_h *= 2.54
lcenter_v = Convert.ToDouble(TextBox6.Text)  'import Lamp center vertical distance and
convert from inch to cm
lcenter_v *= 2.54
laxis_low = Convert.ToDouble(TextBox7.Text)  'import Lamp length lower end and convert
from inch to cm
laxis_low *= 2.54
laxis_high = Convert.ToDouble(TextBox1.Text)  'import Lamp length upper end and convert
from inch to cm
laxis_high *= 2.54

If laxis_high <= laxis_low Then  'make sure the lamp
length is positive value
    MessageBox.Show("<lamp axis upper> should be greater than <lamp axis lower>!")
    Return
End If

If out_plane = Nothing Then
    MsgBox("Please select the axis Outlet plane is normal to!")
    Return
End If

If Lamp_axis = Nothing Then
    MsgBox("Please select the axis Lmap is parallel to!")
    Return
End If

S_lamp = Convert.ToDouble(TextBox8.Text)  'lamp power per cm converted to total lamp
power
S_lamp *= (laxis_high - laxis_low)
get_L_pum = CInt(TextBox3.Text)

'***************************

dz = (laxis_high - laxis_low) / n  'thickness of the light section
'MsgBox(nameOfFile)
If (nameOfFile IsNot Nothing) Then  'Read the input stream line data as stream
    Dim strLine As String
    Dim intLine As Integer

    '**************************Set up the path and name of output data, with suffixes
added**************
out_file70_path = My.Computer.FileSystem.CombinePath(out_file70_path, FileName & ".-UVT70.xls")
'**************create the output file and write the the header of the particle detail information*****
' to use FSO, should add reference "MicroSoft Scripting Runtime" into the project
Dim fs099, fs085, fs070 As New Scripting.FileSystemObject
Dim txt99, txt85, txt70 As Scripting.TextStream

    txt99 = fs099.OpenTextFile(out_file99_path, Scripting.IOMode.ForWriting, True)
    txt99.WriteLine("Particle ID" & vbTab & "Initial x" & vbTab & "Initial y" & vbTab & "Initial z" & vbTab & " Final x" & vbTab & " Final y" & vbTab & " Final z" & vbTab & " Final time" & vbTab & "Dosage" & vbTab & "Xred" & vbTab & "Ni")
    txt99.WriteLine()
    txt99.Close()

    txt85 = fs085.OpenTextFile(out_file85_path, Scripting.IOMode.ForWriting, True)
    txt85.WriteLine("Particle ID" & vbTab & "Initial x" & vbTab & "Initial y" & vbTab & "Initial z" & vbTab & " Final x" & vbTab & " Final y" & vbTab & " Final z" & vbTab & " Final time" & vbTab & "Dosage" & vbTab & "Xred" & vbTab & "Ni")
    txt85.WriteLine()
    txt85.Close()

    txt70 = fs070.OpenTextFile(out_file70_path, Scripting.IOMode.ForWriting, True)
    txt70.WriteLine("Particle ID" & vbTab & "Initial x" & vbTab & "Initial y" & vbTab & "Initial z" & vbTab & " Final x" & vbTab & " Final y" & vbTab & " Final z" & vbTab & " Final time" & vbTab & "Dosage" & vbTab & "Xred" & vbTab & "Ni")
    txt70.WriteLine()
    txt70.Close()

'******************************************************************************************

'******************************************************************************************Main program fvp execution
block******************************************************************************************

' ****Disable the button for the duration of the calculation****
    Me.Button1.Enabled = False
    Me.Button2.Enabled = False
    Me.InputVelocity.Enabled = False
    Me.TextBox2.Enabled = False
    Me.TextBox1.Enabled = False
    ' Me.TextBox4.Enabled = False
    Me.TextBox5.Enabled = False
    Me.TextBox6.Enabled = False
    Me.TextBox7.Enabled = False
    Me.TextBox8.Enabled = False
    Me.TextBox3.Enabled = False
    Me.OpenDirec.Enabled = False
    Me.OutletAxis.Enabled = False
    Me.OutPlaneAxis.Enabled = False
    Me.ProgressBar.Visible = True

'******************************************************************************************
If GetFileExt = "fvp" Then

'******************************************************************************
'************particle data excution************
'******************************************************************************
Dim objFSO As New Scripting.FileSystemObject
Dim objText As Scripting.TextStream

objText = objFSO.OpenTextFile(nameOfFile, Scripting.IOMode.ForReading, False)
'Dim objSize As Long = objFSO.Size

'skip the first 8 line to avoid the header
Dim Sm As Integer
For Sm = 0 To 7 Step 1
   objText.SkipLine()
Next

Do Until objText.AtEndOfStream 'can also use Do Until myTextStream.AtEndOfLine
   strLine = objText.ReadLine 'read each line of input data
   intLine = strLine.Length 'define intLine as the length of each line
   'MsgBox(strLine)
   If intLine > 20 Then 'only dealing with the useful data while reading
      strLine = LTrim(strLine) 'Read the stream line data line by line, cut the valid data into
      string array

      dataArray = Split(strLine)
      Dim LastNonEmpty As Integer = -1
      For i As Integer = 0 To dataArray.Length - 1
         If dataArray(i) <> "" Then
            LastNonEmpty += 1
            dataArray(LastNonEmpty) = dataArray(i)
         End If
      Next
      ReDim Preserve dataArray(LastNonEmpty)

      'Get the single location at each time of each point
      px = CDbl(dataArray(0))
      py = CDbl(dataArray(1))
      pz = CDbl(dataArray(2))
      time = CDbl(dataArray(3))
      pnum = CInt(dataArray(4))
      'MsgBox("px =" & px)
      'MsgBox("py =" & py)
      'MsgBox("pz =" & pz)
      'MsgBox("time =" & time)
      'MsgBox("pnum =" & pnum)

      '****based on the lamp parallel axis, define the Rz Rx Ry, convert from m to cm****
      '****based on the outlet plane normal to which axis, define the boudary value should

      use*****
      If Lamp_axis = "x" Then
         Rz = px * 100
         Rx = py * 100
         Ry = pz * 100
      If out_plane = "x" Then
         p_bound = Rz_last
      ElseIf out_plane = "y" Then
         p_bound = Rx_last
      End If
ElseIf out_plane = "z" Then
    p_bound = Rz_last
End If

ElseIf Lamp_axis = "y" Then
    Rz = py * 100
    Rx = px * 100
    Ry = pz * 100

    If out_plane = "x" Then
        p_bound = Rx_last
        ElseIf out_plane = "y" Then
            p_bound = Ry_last
            ElseIf out_plane = "z" Then
                p_bound = Rz_last
        End If
    End If

ElseIf Lamp_axis = "z" Then
    Rz = pz * 100
    Rx = px * 100
    Ry = py * 100

    If out_plane = "x" Then
        p_bound = Rx_last
        ElseIf out_plane = "y" Then
            p_bound = Ry_last
            ElseIf out_plane = "z" Then
                p_bound = Rz_last
        End If
    End If

End If

******************************************************************************
'MsgBox(Rx & Ry & Rz)
For I_pum = 0 To get_I_pum Step 1  'deal with particle data one by one
    If I_pum - pnum = 0 Then
        ProgressBar.Value = pnum

        If pnum <> pnum_last Then  'check if reach the end of each particle data
            ' MsgBox(pnum & vbTab & dosage99)

            ' MsgBox(px)
            'write the final position into the output file****
            Dim outso99FF As New Scripting.FileSystemObject
            Dim outtxt99FF As Scripting.TextStream
            outtxt99FF = outso99FF.OpenTextFile(out_file99_path, Scripting.IOMode.ForAppending, True, Scripting.Tristate.TristateMixed)
            outtxt99FF.Write(px_last & vbTab & py_last & vbTab & pz_last & vbTab & time_last & vbTab)
            outtxt99FF.Close()
            Dim outso85FF As New Scripting.FileSystemObject
            Dim outtxt85FF As Scripting.TextStream
            outtxt85FF = outso85FF.OpenTextFile(out_file85_path, Scripting.IOMode.ForAppending, True, Scripting.Tristate.TristateMixed)
            outtxt85FF.Write(px_last & vbTab & py_last & vbTab & pz_last & vbTab & time_last & vbTab)
            outtxt85FF.Close()
            Dim outso70FF As New Scripting.FileSystemObject
            Dim outtxt70FF As Scripting.TextStream
If the particle reach the outlet, add up the Xred into total dosage, other wise it will be neglected

Then 'And R_last < out_radius) Then

'calculate Xred value when reach the end of each particle data

Xred99 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage99))) / (2 * aa)
Xred85 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage85))) / (2 * aa)
Xred70 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage70))) / (2 * aa)

' MsgBox(Xred99)
'***calculates Ni and sets to a value of 1 if the reduction value indicates full microbe reduction***

If (Xred99 < 6) Then
    Dim temp As Double
    temp = No / Pow(10, Xred99)
    Ni99 = temp
Else
    Ni99 = 1
End If

If (Xred85 < 6) Then
    Dim temp As Double
    temp = No / Pow(10, Xred85)
    Ni85 = temp
Else
    Ni85 = 1
End If

If (Xred70 < 6) Then
    Dim temp As Double
    temp = No / Pow(10, Xred70)
    Ni70 = temp
Else
    Ni70 = 1
End If

'write the dosage, Xred and Ni value into the output file
Dim outso99FFR As New Scripting.FileSystemObject
Dim outtxt99FFR As Scripting.TextStream
outtxt99FFR = outso99FFR.OpenTextFile(out_file99_path,
Scripting.IOMode.ForAppending, True, ScriptingTriState._
TristateMixed)
outtxt99FFR.WriteLine(dosage99 & vbTab & Xred99 & vbTab &
Ni99)
outtxt99FFR.Close()
Dim outso85FFR As New Scripting.FileSystemObject
Dim outtxt85FFR As Scripting.TextStream
outtxt85FFR = outso85FFR.OpenTextFile(out_file85_path,
Scripting.IOMode.ForAppending, True, ScriptingTriState._
TristateMixed)
outtxt85FFR.WriteLine(dosage85 & vbTab & Xred85 & vbTab & Nb85)

outtxt85FFR.Close()
Dim outfs070FFR As New Scripting.FileSystemObject
Dim outtxt70FFR As Scripting.TextStream
outtxt70FFR = outfs070FFR.OpenTextFile(out_file70_path,
Scripting.IOMode.ForAppending, True, ScriptingTriState.

TristateMixed)
outtxt70FFR.WriteLine(dosage70 & vbTab & Xred70 & vbTab & Nb70)

outtxt70FFR.Close()

'MsgBox(pnum & vbTab & pnum_T & vbTab & Ni_T99)

pnum_T = pnum_T + 1
Ni_T99 += Ni99
Ni_T85 += Ni85
Ni_T70 += Ni70

Else

'If the particle is out of boundary, just go to the next line
Dim outfs099FFRK As New Scripting.FileSystemObject
Dim outtxt99FFRK As Scripting.TextStream
outtxt99FFRK = outfs099FFRK.OpenTextFile(out_file99_path,
Scripting.IOMode.ForAppending, True, Scripting__

Tristate.TristateMixed)
outtxt99FFRK.WriteLine()
outtxt99FFRK.Close()
Dim outfs085FFRK As New Scripting.FileSystemObject
Dim outtxt85FFRK As Scripting.TextStream
outtxt85FFRK = outfs085FFRK.OpenTextFile(out_file85_path,
Scripting.IOMode.ForAppending, True, Scripting__

Tristate.TristateMixed)
outtxt85FFRK.WriteLine()
outtxt85FFRK.Close()
Dim outfs070FFRK As New Scripting.FileSystemObject
Dim outtxt70FFRK As Scripting.TextStream
outtxt70FFRK = outfs070FFRK.OpenTextFile(out_file70_path,
Scripting.IOMode.ForAppending, True, Scripting__

Tristate.TristateMixed)
outtxt70FFRK.WriteLine()
outtxt70FFRK.Close()

End If

End If

'Write the initial data into the output file
If time = 0 Then

Dim outfs099 As New Scripting.FileSystemObject
Dim outtxt99 As Scripting.TextStream
outtxt99 = outfs099.OpenTextFile(out_file99_path,
Scripting.IOMode.ForAppending, True)

'outtxt99.WriteLine(pnum & vbTab & px & vbTab & py & vbTab & pz &
vbTab & pz / 0.0254 & vbTab)
outtxt99.WriteLine(pnum & vbTab & px & 0.0254 & vbTab & py / 0.0254 &
vbTab & pz / 0.0254 & vbTab)
outtxt99.Close()
Dim outfs085 As New Scripting.FileSystemObject
Dim outtxt85 As Scripting.TextStream
outtxt85 = outfs85.OpenTextFile(out_file85_path, 

vbTab)

vbTab & pz / 0.0254 & vbTab)

Scripting.OMode.ForAppending, True)

vbTab)

vbTab & pz / 0.0254 & vbTab)

outtxt85.Write(pnum & vbTab & px & vbTab & py & vbTab & pz & 

outtxt85.Write(pnum & vbTab & px & 0.0254 & vbTab & py / 0.0254 & 

outtxt85.Close()

Dim outfs70 As New Scripting.FileSystemObject

Dim outtxt70 As Scripting.TextStream

outtxt70 = outfs70.OpenTextFile(out_file70_path, 

'outtxt70.Write(pnum & vbTab & px & vbTab & py & vbTab & pz & 

outtxt70.Write(pnum & vbTab & px / 0.0254 & vbTab & py / 0.0254 & 

outtxt70.Close()

'******set/reset initial conditions******

' in_bounds = True

dosage99 = 0

dosage85 = 0

dosage70 = 0

' dset_end = False

R = 99999

Rz = laxis_low

'*********************************************************

period = 0

Else

period = time - time_last 

'calculate the time period of the step

End If

R = Sqr((Rx - lcenter_h), 2) + Pow((Ry - lcenter_v), 2))

' MsgBox("R=" & R)

RRz = (dz / 2)

Tintensity99 = 0

Tintensity85 = 0

Tintensity70 = 0

Dim m As Integer

'calculate intensity(R,Z)

For m = 1 To n Step 1

ro_i = Sqr((Rz - RRz, 2) + Pow(R, 2))

(R - r_q) + alpha_q * t_q) * (ro_i / R))

Tintensity99 += ((S_lamp / n) / (4 * pi * Pow(ro_i, 2))) * Exp(-uvt99 * 

(R - r_q) + alpha_q * t_q) * (ro_i / R))

Tintensity85 += ((S_lamp / n) / (4 * pi * Pow(ro_i, 2))) * Exp(-uvt85 * 

(R - r_q) + alpha_q * t_q) * (ro_i / R))

Tintensity70 += ((S_lamp / n) / (4 * pi * Pow(ro_i, 2))) * Exp(-uvt70 * 

RRz += dz

'MsgBox(pnum & vbTab & ro_i & vbTab & Tintensity99)

Next

'MsgBox(Tintensity99)

dosage99 += (Tintensity99 * period)
dosage85 += (Tintensity85 * period)
dosage70 += (Tintensity70 * period)
'MsgBox(pnum & vbTab & dosage99 & vbTab & Tintensity99 & vbTab & period)

time_last = time
pnum_last = pnum
px_last = px
py_last = py
pz_last = pz
Rx_last = Rx
Ry_last = Ry
Rz_last = Rz
R_last = R

End If ' }
Next 'end of deal with particle data one by one

End If 'End of dealing with the useful data while reading (strLength>20)
Loop 'end of read each line of input data
'MsgBox(px)

'write the final position of the final particle into the output file
Dim outfs099F As New Scripting.FileSystemObject
Dim outtxtx99F As Scripting.TextStream
outtxtx99F = outfs099F.OpenTextFile(out_file99_path, Scripting.IOMode.ForAppending, True,
ScriptingTriStateTriStateMixed)
outtxtx99F.Write(px_last & vbTab & py_last & vbTab & pz_last & vbTab & time_last & vbTab)
outtxtx99F.Close()
Dim outfs085F As New Scripting.FileSystemObject
Dim outtxt85F As Scripting.TextStream
outtxt85F = outfs085F.OpenTextFile(out_file85_path, Scripting.IOMode.ForAppending, True,
ScriptingTriStateTriStateMixed)
outtxt85F.Write(px_last & vbTab & py_last & vbTab & pz_last & vbTab & time_last & vbTab)
outtxt85F.Close()

'MsgBox(p_bound)
'If the final particle reach the outlet, add up the Xred into total dosage, other wise it will be neglected
If (p_bound <= out_bound + 0.02 And p_bound >= out_bound - 0.02) Then 'And R_last < out_radius) Then

'calculate Xred value when reach the end of the final particle data
Xred99 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage99))) / (2 * aa)
Xred85 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage85))) / (2 * aa)
Xred70 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage70))) / (2 * aa)

***calculates Ni and sets to a value of 1 if the reduction value indicates full microbe reduction***
If (Xred99 < 6) Then
Dim temp As Double
temp = No / Pow(10, Xred99)
Ni99 = temp

76
Else
  Ni99 = 1
End If

If (Xred85 < 6) Then
  Dim temp As Double
  temp = No / Pow(10, Xred85)
  Ni85 = temp
Else
  Ni85 = 1
End If

If (Xred70 < 6) Then
  Dim temp As Double
  temp = No / Pow(10, Xred70)
  Ni70 = temp
Else
  Ni70 = 1
End If

'****************************************************************************************************************

*****
'write the final particle's dosage Xred Ni to the output file
Dim outso99FR As Scripting.FileSystemObject
Dim outtxt99FR As Scripting.TextStream
outtxt99FR.WriteLine(dosage99 & vbTab & Xred99 & vbTab & Ni99)
outtxt99FR.Close()

Dim outso85FR As New Scripting.FileSystemObject
Dim outtxt85FR As Scripting.TextStream
outtxt85FR = outso85FR.OpenTextFile(out_file85_path, Scripting.IOMode.ForAppending, True, Scripting.Tristate.TristateMixed)
outtxt85FR.WriteLine(dosage85 & vbTab & Xred85 & vbTab & Ni85)
outtxt85FR.Close()

Dim outso70FR As New Scripting.FileSystemObject
Dim outtxt70FR As Scripting.TextStream
outtxt70FR = outso70FR.OpenTextFile(out_file70_path, Scripting.IOMode.ForAppending, True, Scripting.Tristate.TristateMixed)
outtxt70FR.WriteLine(dosage70 & vbTab & Xred70 & vbTab & Ni70)
outtxt70FR.Close()

pnum_T = pnum_T + 1
Ni_T99 += Ni99
Ni_T85 += Ni85
Ni_T70 += Ni70

' MsgBox(Ni_T99)

Else
  'if the final particle is out of boundary, just go to the next line
  Dim outso99FRK As New Scripting.FileSystemObject
  Dim outtxt99FRK As Scripting.TextStream
  outtxt99FRK = outso99FRK.OpenTextFile(out_file99_path, Scripting.IOMode.ForAppending, True, Scripting.Tristate.TristateMixed)
  outtxt99FRK.WriteLineBlankLines(1)
  outtxt99FRK.Close()

  Dim outso85FRK As New Scripting.FileSystemObject
  Dim outtxt85FRK As Scripting.TextStream
  outtxt85FRK = outso85FRK.OpenTextFile(out_file85_path, Scripting.IOMode.ForAppending, True, Scripting.Tristate.TristateMixed)
  outtxt85FRK.WriteLineBlankLines(1)
  outtxt85FRK.Close()
Dim outfo70FRK As New Scripting.FileSystemObject
Dim outtxt70FRK As Scripting.TextStream
outtxt70FRK = outfo70FRK.OpenTextFile(out_file70_path, Scripting.IOMode.ForAppending, True, ScriptingTriStateTriStateMixed)
outtxt70FRK.WriteString("Hello World")
outtxt70FRK.WriteBlankLines(1)
outtxt70FRK.Close()

End If

******************************************************************************

objText.Close()

End If  'End of if extension if fyp
If GetFileExt = "csv" Then

******************************************************************************

Dim objFSO As New Scripting.FileSystemObject
Dim objText As Scripting.TextStream

objText = objFSO.OpenTextFile(nameOfFile, Scripting.IOMode.ForReading, False)
'dim objSize As Long = objFSO.Size

'skip the first 8 line to avoid the header

Try
  Dim Sm As Integer
  For Sm = 0 To 5 + get_I_pum Step 1
    objText.SkipLine()
  Next
Catch ex As EndOfStreamException
  MsgBox("Error writing data: check the input particle number.")
  Return
End Try

Do Until objText.AtEndOfStream  'can also use  Do Until myTextStream.AtEndOfLine
  strLine = objText.ReadLine  'read each line of input data
  intLine = strLine.Length  'define intLine as the length of each line

  'MsgBox(strLine)
  If intLine > 20 Then  'only dealing with the useful data while reading
    strLine = LTrim(strLine)  'Read the stream line data line by line, cut the valid data into string array
    dataArray = Split(strLine, ",")  'csv file's data is seperated by ","
    Dim LastNonEmpty As Integer = -1
    For i As Integer = 0 To dataArray.Length - 1
      If dataArray(i) <> "" Then
        LastNonEmpty += 1
        dataArray(LastNonEmpty) = dataArray(i)
      End If
    Next
    ReDim Preserve dataArray(LastNonEmpty)
  End If
Next

78
'Get the single location at each time of each point
px = CDbl(dataArray(0))
py = CDbl(dataArray(1))
pz = CDbl(dataArray(2))
'time = CDbl(dataArray(3))
pnum = CInt(dataArray(3))
time = CDbl(dataArray(4))
'MsgBox("px =" & px)
'MsgBox("py =" & py)
'MsgBox("pz =" & pz)
'MsgBox("time =" & time)
'MsgBox("pnum =" & pnum)

'****based on the lamp parallel axis, define the Rz Rx Ry, convert from m to cm****
'****based on the outlet plane normal to which axis, define the boundary value should

use*****

If Lamp_axis = "x" Then
    Rz = px * 100
    Rx = py * 100
    Ry = pz * 100

    If out_plane = "x" Then
        p_bound = Rz_last
    ElseIf out_plane = "y" Then
        p_bound = Rx_last
    ElseIf out_plane = "z" Then
        p_bound = Ry_last
    End If

ElseIf Lamp_axis = "y" Then
    Rz = py * 100
    Rx = px * 100
    Ry = pz * 100

    If out_plane = "x" Then
        p_bound = Rx_last
    ElseIf out_plane = "y" Then
        p_bound = Rz_last
    ElseIf out_plane = "z" Then
        p_bound = Ry_last
    End If

ElseIf Lamp_axis = "z" Then
    Rz = pz * 100
    Rx = px * 100
    Ry = py * 100

    If out_plane = "x" Then
        p_bound = Rx_last
    ElseIf out_plane = "y" Then
        p_bound = Ry_last
    ElseIf out_plane = "z" Then
        p_bound = Rz_last
    End If

End If

'******************************************************************************

'MsgBox(Rx & " & Ry & Rz)
For l_pum = 0 To get_l_pum Step 1  'deal with particle data one by one
    If l_pum - pnum = 0 Then
        ',

End If

79
ProgressBar.Value = pnum

If pnum <> pnum_last Then
    'check if reach the end of each particle
    data

    ' MsgBox(px)
    ' write the final position into the output file
    Dim outxso99FF As New Scripting.FileSystemObject
    Dim outtxt99FF As Scripting.TextStream
    outtxt99FF = outxso99FF.OpenTextFile(out_file99_path, Scripting.IOMode.ForAppending, True, Scripting.Tristate.TristateMixed)
    outtxt99FF.Write(px_last & vbTab & py_last & vbTab & pz_last & vbTab & time_last & vbTab)
    outtxt99FF.Close()
    Dim outxso85FF As New Scripting.FileSystemObject
    Dim outtxt85FF As Scripting.TextStream
    outtxt85FF = outxso85FF.OpenTextFile(out_file85_path, Scripting.IOMode.ForAppending, True, Scripting.Tristate.TristateMixed)
    outtxt85FF.Write(px_last & vbTab & py_last & vbTab & pz_last & vbTab & time_last & vbTab)
    outtxt85FF.Close()
    Dim outxso70FF As New Scripting.FileSystemObject
    Dim outtxt70FF As Scripting.TextStream
    outtxt70FF = outxso70FF.OpenTextFile(out_file70_path, Scripting.IOMode.ForAppending, True, Scripting.Tristate.TristateMixed)
    outtxt70FF.Write(px_last & vbTab & py_last & vbTab & pz_last & vbTab & time_last & vbTab)
    outtxt70FF.Close()

    ' If the particle reach the outlet, add up the Xred into total dosage, other wise it will be neglected
    If (p_bound <= out_bound + 0.02 And p_bound >= out_bound - 0.02) Then
        ' calculate Xred value when reach the end of each particle data
        Xred99 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage99))) / (2 * aa)
        Xred85 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage85))) / (2 * aa)
        Xred70 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage70))) / (2 * aa)

        ' MsgBox(Xred99)
        ***calculates Ni and sets to a value of 1 if the reduction value indicates full microbe reduction
        If (Xred99 < 6) Then
            Dim temp As Double
            temp = No / Pow(10, Xred99)
            Ni99 = temp
        Else
            Ni99 = 1
        End If

        If (Xred85 < 6) Then
            Dim temp As Double
            temp = No / Pow(10, Xred85)
            Ni85 = temp
        Else
            Ni85 = 1
        End If

    End If
End If

If (Xred70 < 6) Then
    Dim temp As Double
    temp = No / Pow(10, Xred70)
    Ni70 = temp
Else
    Ni70 = 1
End If

'********************************************************************************

*****

'write the dosage, Xred and Ni value into the output file
Dim outso99FFR As New Scripting.FileSystemObject
Dim outtxt99FFR As Scripting.TextStream
outtxt99FFR = outso99FFR.OpenTextFile(out_file99_path,
Scripting.IOMode.ForAppending, True, Scripting._
    Tristate.TristateMixed)
outtxt99FFR.WriteLine(dosage99 & vbTab & Xred99 & vbTab &
Ni99)
outtxt99FFR.Close()
Dim outso85FFR As New Scripting.FileSystemObject
Dim outtxt85FFR As Scripting.TextStream
outtxt85FFR = outso99FFR.OpenTextFile(out_file85_path,
Scripting.IOMode.ForAppending, True, Scripting._
    Tristate.TristateMixed)
outtxt85FFR.WriteLine(dosage85 & vbTab & Xred85 & vbTab &
Ni85)
outtxt85FFR.Close()
Dim outso70FFR As New Scripting.FileSystemObject
Dim outtxt70FFR As Scripting.TextStream
outtxt70FFR = outso70FFR.OpenTextFile(out_file70_path,
Scripting.IOMode.ForAppending, True, Scripting._
    Tristate.TristateMixed)
outtxt70FFR.WriteLine(dosage70 & vbTab & Xred70 & vbTab &
Ni70)
outtxt70FFR.Close()
'MsgBox(pnum & vbTab & pnum_T & vbTab & Ni_T99)

pnum_T = pnum_T + 1
Ni_T99 += Ni99
Ni_T85 += Ni85
Ni_T70 += Ni70

Else

'if the particle is out of boundary, just go to the next line
Dim outso99FFRK As New Scripting.FileSystemObject
Dim outtxt99FFRK As Scripting.TextStream
outtxt99FFRK = outso99FFRK.OpenTextFile(out_file99_path,
Scripting.IOMode.ForAppending, True, Scripting._
    Tristate.TristateMixed)
outtxt99FFRK.WriteLine()
outtxt99FFRK.Close()
Dim outso85FFRK As New Scripting.FileSystemObject
Dim outtxt85FFRK As Scripting.TextStream
outtxt85FFRK = outso85FFRK.OpenTextFile(out_file85_path,
Scripting.IOMode.ForAppending, True, Scripting._
    Tristate.TristateMixed)
outtxt85FFRK.WriteLine()
outtxt85FFRK.Close()

End If
Dim outfs070FFRK As New Scripting.FileSystemObject
Dim outtxt70FFRK As Scripting.TextStream
outtxt70FFRK = outfs070FFRK.OpenTextFile(out_file70_path,
Scripting.IOMode.ForAppending, True, Scripting._Tristate.TristateMixed)
outtxt70FFRK.WriteLine()
outtxt70FFRK.Close()
End If
End If

' write the initial data into the output file
If time < time_last Then
  Dim outfs99 As New Scripting.FileSystemObject
  Dim outtxt99 As Scripting.TextStream
  outtxt99 = outfs99.OpenTextFile(out_file99_path,
Scripting.IOMode.ForAppending, True)
  outtxt99.WriteLine(pnum & vbTab & px & vbTab & py & vbTab & pz &
  vbTab & pz / 0.0254 & vbTab)
  outtxt99.WriteLine(pnum & vbTab & px / 0.0254 & vbTab & py / 0.0254 &
  vbTab & pz / 0.0254 & vbTab)
  outtxt99.Close()
  Dim outfs85 As New Scripting.FileSystemObject
  Dim outtxt85 As Scripting.TextStream
  outtxt85 = outfs85.OpenTextFile(out_file85_path,
Scripting.IOMode.ForAppending, True)
  outtxt85.WriteLine(pnum & vbTab & px & vbTab & py & vbTab & pz &
  vbTab & pz / 0.0254 & vbTab)
  outtxt85.WriteLine(pnum & vbTab & px / 0.0254 & vbTab & py / 0.0254 &
  vbTab & pz / 0.0254 & vbTab)
  outtxt85.Close()
  Dim outfs70 As New Scripting.FileSystemObject
  Dim outtxt70 As Scripting.TextStream
  outtxt70 = outfs70.OpenTextFile(out_file70_path,
Scripting.IOMode.ForAppending, True)
  outtxt70.WriteLine(pnum & vbTab & px & vbTab & py & vbTab & pz &
  vbTab & pz / 0.0254 & vbTab)
  outtxt70.WriteLine(pnum & vbTab & px / 0.0254 & vbTab & py / 0.0254 &
  vbTab & pz / 0.0254 & vbTab)
  outtxt70.Close()

  '**********set/reset initial conditions******
  '  in_bounds = True
dosage99 = 0
dosage85 = 0
dosage70 = 0
  '  dset_end = False
R = 99999
Rz = laxis_low

  '****************************
  period = 0
Else
  period = time - time_last  ' calculate the time period of the step
End If

82
R = Sqr(Pow((Rx - lcenter_h), 2) + Pow((Ry - lcenter_v), 2))
  ' MsgBox("R=", & R)
RRz = (dz / 2)
Tintensity99 = 0
Tintensity85 = 0
Tintensity70 = 0
Dim m As Integer  'calculate intensity(R,Z)
For m = 1 To n Step 1
  ro_i = Sqr(Pow((Rz - RRz), 2) + Pow(R, 2))
  (R - r_q) + alpha_q * t_q) * (ro_i / R))
  Tintensity99 += ((S_lamp / n) / (4 * pi * Pow(ro_i, 2))) * Exp(-(uvt99 *
  (R - r_q) + alpha_q * t_q) * (ro_i / R))
  Tintensity85 += ((S_lamp / n) / (4 * pi * Pow(ro_i, 2))) * Exp(-(uvt85 *
  (R - r_q) + alpha_q * t_q) * (ro_i / R))
  Tintensity70 += ((S_lamp / n) / (4 * pi * Pow(ro_i, 2))) * Exp(-(uvt70 *
    RRz += dz
  'MsgBox(pnum & vbTab & ro_i & vbTab & Tintensity99)
Next
  'MsgBox(Tintensity99)
  dosage99 += (Tintensity99 * period)
  dosage85 += (Tintensity85 * period)
  dosage70 += (Tintensity70 * period)
  'MsgBox(pnum & vbTab & dosage99 & vbTab & Tintensity99 & vbTab &
  period)
  time_last = time
  pnum_last = pnum
  px_last = px
  py_last = py
  pz_last = pz
  Rx_last = Rx
  Ry_last = Ry
  Rz_last = Rz
  R_last = R

End If  ' 
  Next   'end of deal with particle data one by one
End If    'End of dealing with the useful data while reading (strLength>20)

Loop    'end of read each line of input data
  'MsgBox(px)

'write the final position of the final particle into the output
file*************************************************************
Dim outso99F As New Scripting.FileSystemObject
Dim outtxt99F As Scripting.TextStream
outtxt99F = outso99F.OpenTextFile(out_file99_path, Scripting.IOMode.ForAppending, True,
ScriptingTriStateTriStateMixed)
outtxt99F.Write(px_last & vbTab & py_last & vbTab & pz_last & vbTab & time_last & vbTab)
outtxt99F.Close()
Dim outfs085F As New Scripting.FileSystemObject
Dim outtxt85F As Scripting.TextStream
outtxt85F = outfs085F.OpenTextFile(out_file85_path, Scripting.IOMode.ForAppending, True,
ScriptingTriStateTriStateMixed)
outtxt85F.Write(px_last & vbTab & py_last & vbTab & pz_last & vbTab & time_last & vbTab)
outtxt85F.Close()
Dim outfs070F As New Scripting.FileSystemObject
Dim outtxt70F As Scripting.TextStream
outtxt70F = outfs070F.OpenTextFile(out_file70_path, Scripting.IOMode.ForAppending, True,
ScriptingTriStateTriStateMixed)
outtxt70F.Write(px_last & vbTab & py_last & vbTab & pz_last & vbTab & time_last & vbTab)
outtxt70F.Close()

'MsgBox(p_bound)
'If the final particle reach the outlet, add up the Xred into total dosage, other wise it will be
neglected
If (p_bound <= out_bound + 0.02 And p_bound >= out_bound - 0.02) Then 'And R_last <
out_radius) Then

'calculate Xred value when reach the end of the final particle data
Xred99 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage99))) / (2 * aa)
Xred85 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage85))) / (2 * aa)
Xred70 = ((-bb) + Sqrt(Pow(bb, 2) + (4 * aa * dosage70))) / (2 * aa)

***calculates Ni and sets to a value of 1 if the reduction value indicates full microbe
reduction***
If (Xred99 < 6) Then
  Dim temp As Double
  temp = No / Pow(10, Xred99)
  Ni99 = temp
Else
  Ni99 = 1
End If

If (Xred85 < 6) Then
  Dim temp As Double
  temp = No / Pow(10, Xred85)
  Ni85 = temp
Else
  Ni85 = 1
End If

If (Xred70 < 6) Then
  Dim temp As Double
  temp = No / Pow(10, Xred70)
  Ni70 = temp
Else
  Ni70 = 1
End If

'************************************************************
*****
'write the final particle's dosage Xred Ni to the output file
Dim outfs099FR As New Scripting.FileSystemObject
Dim outtxt99FR As Scripting.TextStream
outtxt99FR = outfs099FR.OpenTextFile(out_file99_path, Scripting.IOMode.ForAppending, True,
ScriptingTriStateTriStateMixed)
outtxt99FR.WriteLine(dosage99 & vbTab & Xred99 & vbTab & Ni99)
outtxt99FR.Close()
Dim outfs085FR As New Scripting.FileSystemObject

84
Dim outtxt85FR As Scripting.TextStream
outtxt85FR = outso85FR.OpenTextFile(out_file85_path, Scripting.IOMode.ForAppending, True, Scriptинг.Tristate.TristateMixed)
outtxt85FR.WriteLine(dosage85 & vbTab & Xred85 & vbTab & Ni85)
outtxt85FR.Close()
Dim outso70FR As New Scripting.FileSystemObject
Dim outtxt70FR As Scripting.TextStream
outtxt70FR = outso70FR.OpenTextFile(out_file70_path, Scripting.IOMode.ForAppending, True, Scriptинг.Tristate.TristateMixed)
outtxt70FR.WriteLine(dosage70 & vbTab & Xred70 & vbTab & Ni70)
outtxt70FR.Close()

nnum_T = nnum_T + 1
Ni_T99 += Ni99
Ni_T85 += Ni85
Ni_T70 += Ni70

' MsgBox(Ni_T99)
Else
' if the final particle is out of boundary, just go to the next line
Dim outtxt99FRK As New Scripting.FileObject
Dim outtxt99FRK As Scripting.TextStream
outtxt99FRK = outso99FRK.OpenTextFile(out_file99_path, Scripting.IOMode.ForAppending, True, Scriptинг.Tristate.TristateMixed)
outtxt99FRK.WriteLine(1)
outtxt99FRK.Close()
Dim outso85FRK As New Scripting.FileSystemObject
Dim outtxt85FRK As Scripting.TextStream
outtxt85FRK = outso99FRK.OpenTextFile(out_file85_path, Scripting.IOMode.ForAppending, True, Scriptинг.Tristate.TristateMixed)
outtxt85FRK.WriteLine(1)
outtxt85FRK.Close()
Dim outso70FRK As New Scripting.FileSystemObject
Dim outtxt70FRK As Scripting.TextStream
outtxt70FRK = outso70FRK.OpenTextFile(out_file70_path, Scripting.IOMode.ForAppending, True, Scriptинг.Tristate.TristateMixed)
outtxt70FRK.WriteLine(1)
outtxt70FRK.Close()

End If

'**************************************************************************
**************************************************************************
**************************************************************************

objText.Close()

End If

**************************************************************************End of main program fvp execution block**************************************************************************

'**************************************************************************RED calculation**************************************************************************
Xred_T99 = Log10((nnum_T * No) / Ni_T99)
Xred_T85 = Log10((nnum_T * No) / Ni_T85)
Xred_T70 = Log10((nnum_T * No) / Ni_T70)
red99 = (aa * Pow(Xred_T99, 2)) + (bb * Xred_T99)

85
\[
\text{red85} = (aa \times \text{Pow}(Xred\_T85, 2)) + (bb \times Xred\_T85) \\
\text{red70} = (aa \times \text{Pow}(Xred\_T70, 2)) + (bb \times Xred\_T70)
\]

'write final data to files
Dim outfs99Final As New Scripting.FileSystemObject
Dim outtxt99Final As Scripting.TextStream
outtxt99Final = outfs99Final.OpenTextFile(out_file99_path, Scripting.IOMode.ForAppending, True, ScriptingTriStateMixed)
outtxt99Final.WriteLine("Xred\_T:" & vbTab & Xred\_T99 & vbTab & "Ni\_T:" & vbTab & & _
"n:" & vbTab & pnum\_T)
outtxt99Final.Close()
Dim outfs85Final As New Scripting.FileSystemObject
Dim outtxt85Final As Scripting.TextStream
outtxt85Final = outfs85Final.OpenTextFile(out_file85_path, Scripting.IOMode.ForAppending, True, ScriptingTriStateMixed)
outtxt85Final.WriteLine("Xred\_T:" & vbTab & Xred\_T85 & vbTab & "Ni\_T:" & vbTab & & _
"n:" & vbTab & pnum\_T)
outtxt85Final.Close()
Dim outfs70Final As New Scripting.FileSystemObject
Dim outtxt70Final As Scripting.TextStream
outtxt70Final = outfs70Final.OpenTextFile(out_file70_path, Scripting.IOMode.ForAppending, True, ScriptingTriStateMixed)
outtxt70Final.WriteLine("Xred\_T:" & vbTab & Xred\_T70 & vbTab & "Ni\_T:" & vbTab & & _
"n:" & vbTab & pnum\_T)
outtxt70Final.Close()

' The calculation is done, so enable the button.
ProgressBar.Visible = False
Me.Button1.Enabled = True
Me.Button2.Enabled = True
Me.InputVelocity.Enabled = True
Me.OpenDirec.Enabled = True
Me.OutAxis.Enabled = True
Me.TextBox2.Enabled = True
Me.TextBox1.Enabled = True
' Me.TextBox4.Enabled = True
Me.TextBox5.Enabled = True
Me.TextBox6.Enabled = True
Me.TextBox7.Enabled = True
Me.TextBox8.Enabled = True
Me.TextBox3.Enabled = True
Me.OutPlaneAxis.Enabled = True
Me.ProgressBar.Visible = False

'Dim Delefile As New Scripting.FileSystemObject
'Delefile.GetFileName("C:\FSOuvTemp")
'Delefile.DeleteFile("C:\FSOuvTemp")

MsgBox("The calculation is complete." & vbNewLine & vbNewLine & vbTab & "RED value:" & _
vbNewLine & "\"UVT0\":" & red70 & vbNewLine _
& "\"UVT85\":" & red85 & vbNewLine & "\"UVT90\":" & red99 & vbNewLine & vbNewLine & "Total input particle number:" & _
& get_I_pum & vbNewLine _
& "Valid particle number:" & pnum_T)

Else
    MessageBox.Show("Please set the direction of input velocity profile")
    Return
End If
'out_file99 = Nothing
'out_file85 = Nothing
'out_file70 = Nothing
nameofFile = Nothing

out_file99_path = Nothing
out_file85_path = Nothing
out_file70_path = Nothing

Me.OpenDir.Text = "&Output Directory"

Me.InputVelocity.Text = "&Input Velocity Profile"

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

Dim Result_Cancel01 As DialogResult

Result_Cancel01 = MessageBox.Show("Are you sure to exit the Dosage Calc program?", "Exit", MessageBoxButtons.YesNo)
'Displays a MessageBox using the Question icon and specifying the No button as the default.

If Result_Cancel01 = DialogResult.Yes Then
    ' Gets the result of the MessageBox display.
    Me.Close()
    ' Closes the parent form.
    px = 0
    py = 0
    pz = 0
    Xred99 = 0
    Xred85 = 0
    Xred70 = 0
    Ni99 = 0
    Ni85 = 0
    Ni70 = 0
End If

End Sub

Private Sub OutletAxis_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OutletAxis.SelectedIndexChanged
    Lamp_axis = CChar(OutletAxis.SelectedItem)
    'MsgBox(out_plane)

End Sub

Private Sub OutletDirection_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OutletDirection.SelectedIndexChanged
    out_plane = CChar(OutletPlaneAxis.SelectedItem)
    'MsgBox(out_dir)

End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenDirec.Click
    ' Declare a variable named theFolderBrowser of type FolderBrowserDialog.
    Dim theFolderBrowser As New FolderBrowserDialog
    ' Set theFolderBrowser object's Description property to give the user instructions.
    theFolderBrowser.Description = "Please select a folder for the output RED value data."
    ' Set theFolderBrowser object's ShowNewFolder property to false when the a FolderBrowserDialog is to be used only for selecting an existing folder.
    theFolderBrowser.ShowNewFolderButton = True
    ' Optionally set the RootFolder and SelectedPath properties to control which folder will be selected when browsing begging and to 'make it the selected folder.
    ' For this example start browsing in the Desktop folder.
    theFolderBrowser.RootFolder = System.Environment.SpecialFolder.Desktop
    ' Default theFolderBrowserDialog object's SelectedPath property to the path to the Desktop folder.
    ' If the user clicks theFolderBrowser's OK button..
    If theFolderBrowser.ShowDialog = Windows.Forms.DialogResult.OK Then
        ' Set the FolderChoiceTextBox's Text to theFolderBrowserDialog's
        ' SelectedPath property.
        Me.OpenDirec.Text = "You have chosen to save the output data at:" & vbCrLf & vbCrLf & theFolderBrowser.SelectedPath
        out_file99_path = theFolderBrowser.SelectedPath
        out_file85_path = theFolderBrowser.SelectedPath
        out_file70_path = theFolderBrowser.SelectedPath
        If (out_file99_path = Nothing) Or (out_file85_path = Nothing) Or (out_file70_path = Nothing) Then
            MsgBox("Can not save to the output data to the designated direction, please check again.")
            Return
        End If
    End If
End Sub

Private Sub TextBox5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.Click
    PictureBox2.Hide()
lampCenterPic.Show()
End Sub
Private Sub TextBox5_Leave(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.Leave
    PictureBox2.Show()
lampCenterPic.Hide()
End Sub
Private Sub TextBox6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.Click
    PictureBox2.Hide()
    lampCenterPic.Show()
End Sub

Private Sub TextBox6_Leave(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.Leave
    PictureBox2.Show()
    lampCenterPic.Hide()
End Sub

Private Sub TextBox2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.Click
    PictureBox2.Hide()
    PictureBox4.Show()
End Sub

Private Sub TextBox2_Leave(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.Leave
    PictureBox2.Show()
    PictureBox4.Hide()
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    Dim Result_Cancel01 As DialogResult

    Result_Cancel01 = MessageBox.Show("Are you sure to exit the Dosage Calc program?", "Exit", MessageBoxButtons.YesNo) 'Displays a MessageBox using the Question icon and specifying the No button as the default.

    If Result_Cancel01 = DialogResult.Yes Then ' Gets the result of the MessageBox display.
        Me.Close() ' Closes the parent form.
        PictureBox3.Show() ' Closes the parent form.
        TextBox16.Text = Nothing
        TextBox16.Hide()
    End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

    lamp_low = Convert.ToDouble(TextBox10.Text) 'import lamp location and convert from inch to cm
    lamp_low *= 2.54

    lamp_high = Convert.ToDouble(TextBox11.Text)
    lamp_high *= 2.54

    len_lamp = lamp_high - lamp_low
lamp_h = Convert.ToDouble(TextBox12.Text)
lamp_h *= 2.54

lamp_v = Convert.ToDouble(TextBox13.Text)
lamp_v *= 2.54

d_chamber = Convert.ToDouble(TextBox4.Text)
d_chamber *= 2.54

distance_colli = Convert.ToDouble(TextBox9.Text)
distance_colli *= 2.54

p_lamp = Convert.ToDouble(TextBox14.Text)
p_lamp *= Eg

meas_I = Convert.ToDouble(TextBox15.Text)

If meas_I = Nothing Then 'make sure the lamp length is positivie value
  MessageBox.Show("please enter the mesured intensity value!")
  Return
End If

If p_lamp = Nothing Then 'make sure the lamp length is positivie value
  MessageBox.Show("please enter the lamp full power value!")
  Return
End If

Dim age lz As Double
Dim age_dz As Double
Dim age_n As Integer = 100
Dim age_R, age_ro_i As Double
Dim age_Rx, age_Ry, age_Rz As Double
Dim age_pow As Double
Dim E_lamp As Double

age_Rx = 0
age_Ry = d_chamber / 2
age_Rz = distance_colli
age_dz = len_lamp / age_n

'****calculate the power from the meas_I*****
age_R = Sqr(Pow((age_Rx - lcenter_h), 2) + Pow((age_Ry - lcenter_v), 2))
'Dim = d_chamber - dep
'  MsgBox("R=" & R)
age_lz = (age_dz / 2)

Dim age_Int As Double

age_Int = 0

Dim m As Integer
  'calculate intensity(R,Z)
For m = 1 To age_n Step 1
  age_ro_i = Sqr(Pow((age_Rz - age_lz), 2) + Pow(age_R, 2))

  age_Int = Exp(-(uvt99 * (age_R - r_q) + alpha_q * t_q) * (age_ro_i / age_R)) / (4 * pi * age_n * Pow(age_ro_i, 2))

  age_lz += age_dz
MsgBox(pnum & vbCrLf & ro_i & vbCrLf & Tintensity99)
Next

age_pow = (meas_I / age_Int)
E_lamp = (age_pow / p_lamp) * 100

TextBox16.Text = ("the calculation result:" & vbCrLf & vbCrLf & "Lamp operating power: " & vbCrLf & vbCrLf & "W" & vbCrLf & vbCrLf & "the germicidal output: " & vbCrLf & vbCrLf & ")

End Sub

Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
' If CheckBox1.CheckState = 1 Then
'    c2fluent = True
'    MsgBox("converting input data will take more computational time")
' Else
'    c2fluent = False
'    End If

End Sub

End Class