To the Graduate Council:

I am submitting herewith a thesis written by Kaushik Deb titled "A Visual Basic Application For Design of An Ionizing Wet Scrubber System." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Environmental Engineering.

Wayne T Davis

Wayne T. Davis, Major Professor

We have read this thesis
and recommend its acceptance:

Terry Miller

Chris D Cox

Accepted for the Council:

Cecil Minkel

Associate Vice Chancellor and
Dean of The Graduate School

A VISUAL BASIC APPLICATION FOR DESIGN OF AN IONIZING WET SCRUBBER SYSTEM

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Kaushik Deb
December 1997

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude and appreciation to Dr. Wayne T. Davis, my major professor for allowing me the opportunity to work on this project and sharing his insights on air pollution control design. I will always be grateful for his guidance, interminable patience, perseverance, pursuit of perfection and reinforcing my love for air pollution control. I would also like to thank Dr. Terry Miller and Dr. Chris Cox for serving on my committee and providing their reviews towards improving the model. Funding for the research was provided by Lockheed Martin Energy Systems Inc.

ABSTRACT

A comprehensive user-friendly Air Pollution Control (APC) linked model simulating the TSCA hazardous waste incinerator air pollution control system configuration was developed in Visual Basic (VB). VB is a very effective platform for aesthetically appealing GUI (Graphical User Interface) applications. The air pollution control system consists of a series of devices (or operations) including a quench tower, venturi scrubber, a packed bed and two ionizing wet scrubbers. Based on user-defined inlet gas and particle characteristics, the VB application utilizes a mix of classical efficiency predictor equations along with the various empirical equations developed at the University of Tennessee, Knoxville to predict both device-specific and overall system particle collection efficiencies for the TSCA facility APC system configuration. In addition, the application can be molded to define other system configurations containing any of the device elements included in the application package to predict the particle collection efficiency for a variety of inlet gas-particle scenarios.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

Figure **Page**

NOMENCLATURE

acfm	actual cubic feet per minute
APC	Air Pollution Control
BASIC	Beginners All-Purpose Symbolic Instruction Code
d_a	aerodynamic diameter, *meters or microns*
DOE	Department of Energy
d_p	physical diameter of particle, *meters or microns*
dscf	dry standard cubic feet
ESP	electrostatic precipitator
ft	feet
g	gram
gpm	gallons per minute
gr	grains
GUI	Graphical User Interface
IWS	Ionizing Wet Scrubber
lb	pound
MDI	Multiple Document Interface
MMD	Mass Mean Diameter, *meters or microns*
ORNL	Oak Ridge National Laboratory
P_t	fractional penetration
PC	personal computer
Q	gas flowrate, ft^3/min
s	second
SCA	Specific Collection Area, *s/m or plate area* (ft^2) *per 1000 acfm gas treated*
T	absolute temperature, *K*
TSCA	Toxic Substances Control Act
USEPA	United States Environmental Protection Agency
VB	Visual Basic

1. INTRODUCTION

The Department of Energy (DOE) owns and operates a Toxic Substances Control Act (TSCA) Incinerator located at Oak Ridge, TN, as a means of thermally destructing both hazardous and low-level radioactive waste. The permitted incinerator operates under an air permit issued by the Tennessee Department of Environment and Conservation (TDEC) and a RCRA (Resource Conservation and Recovery Act) Part B permit.

The TSCA facility air pollution control system consists of a venturi scrubber, followed by a packed bed and two ionizing wet scrubbers (IWSs) in series as a means of controlling emissions of acid gases and particulates. Owing to some concern that the facility might have been unable to meet proposed USEPA (United States Environmental Protection Agency) particulate emission standards for hazardous waste incineration facilities, a test program was conducted on a pilot plant to characterize the performance of the APC system. A pilot plant simulating the system was constructed at the University of Tennessee, Knoxville and performance tests were conducted in order to develop an empirical particle collection efficiency prediction model. These studies helped generate relationships for prediction of particle collection efficiencies as a function of particle diameter and gas flowrate through the system. Additionally, these studies also quantified the effect of electrostatic collection of particles on penetration which were later incorporated into a complete predictive model discussed in this document. The model can be utilized to estimate individual device collection efficiencies and overall system collection efficiency as a function of particle size, gas flowrate, depth of packing and other relevant model inputs. The model also provides some insight into the various operative particle collection mechanisms.

A comprehensive user-friendly Air Pollution Control (APC) linked application simulating the TSCA facility system configuration was developed in Visual Basic (VB) which is a very effective platform for aesthetically appealing GUI (Graphical User Interface) applications. Although, it is uncommon for a traditional textual procedural-based linked program to be coded in Visual Basic, controls were incorporated into the VB application such that both user-friendliness and calculation integrity were preserved. Dr. Wayne T. Davis had originally written a lot of the equations in BASIC (Beginners All-Purpose Symbolic Instruction Code) for some of the individual control device components (venturi, electrostatic precipitator). Based on user-defined inlet gas and particle characteristics, the VB application utilizes a mix of classical efficiency predictor equations along with the various empirical equations developed under this project to predict both device-specific and overall system particle collection efficiencies for the TSCA facility APC system configuration. In addition, the application can be molded to define other system configurations (e.g., user can bypass any control device by entering appropriate inputs discussed later) containing any of the device elements included in the application package to predict the particle collection efficiency for any number of inlet gas-particle scenarios.

2. VB APPLICATION LAYOUT

The VB application is organized into discrete modules for performing specific tasks. After the entire application was coded in Visual Basic, it was compiled as an executable .exe file. The executable file does not require a given PC (personal computer) to have a resident Visual Basic program, i.e., the .exe file can run on its own. This provides the application with ease of transport across PCs and of course, the flexibility to be able to run the application on any PC. However, the PC should either have a Microsoft Windows 3.1 GUI (graphical user interface), Windows 95 or Windows NT operating systems. Based on trial and error, it was ascertained that in order to run the application successfully, the PC should have at least 16 MBytes of RAM (Random Access Memory). The reference material for the VB language was obtained from Cunningham (1994) and Potter (1993).

As soon as the user starts up the application, the first screen is a startup screen shown in Figure 2-1. This screen essentially contains information about the application title ("A Visual Basic Program For Design of Ionizing Wet Scrubber System") and the names of the persons involved with the project (Dr. Wayne T. Davis and Kaushik Deb) and their institutional affiliation (The University of Tennessee, Knoxville). The next screen is a MDI (Multiple Document Interface) form which constitutes the basic underlying fabric for the entire application. This screen has been shown as Figures 2-2 through 2-4 with different variations. Shortcut access keys (Alt+key) were provided for all menu elements and TabIndex hierarchy was developed for all the screens such that the user can have the

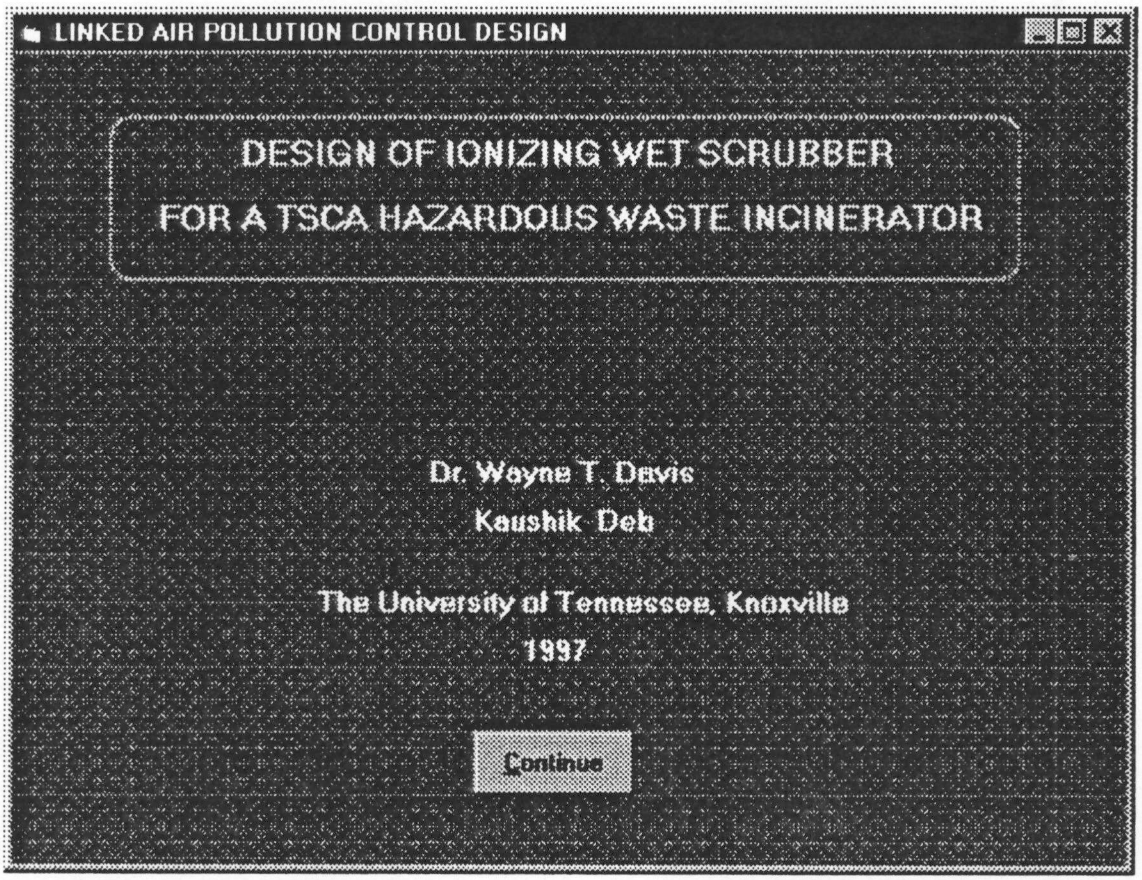

Figure 2-1. Startup screen for VB application.

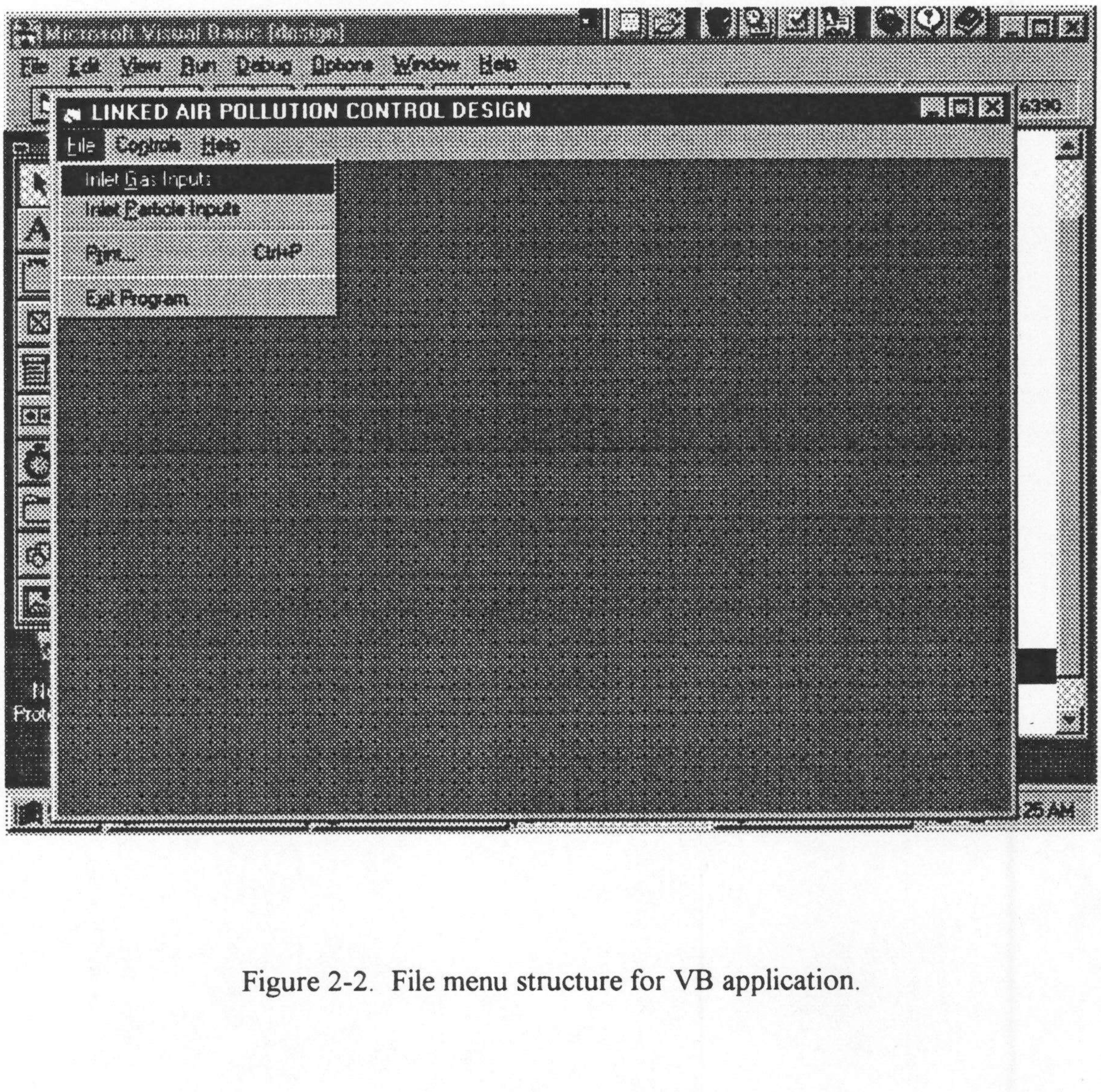

Figure 2-2. File menu structure for VB application.

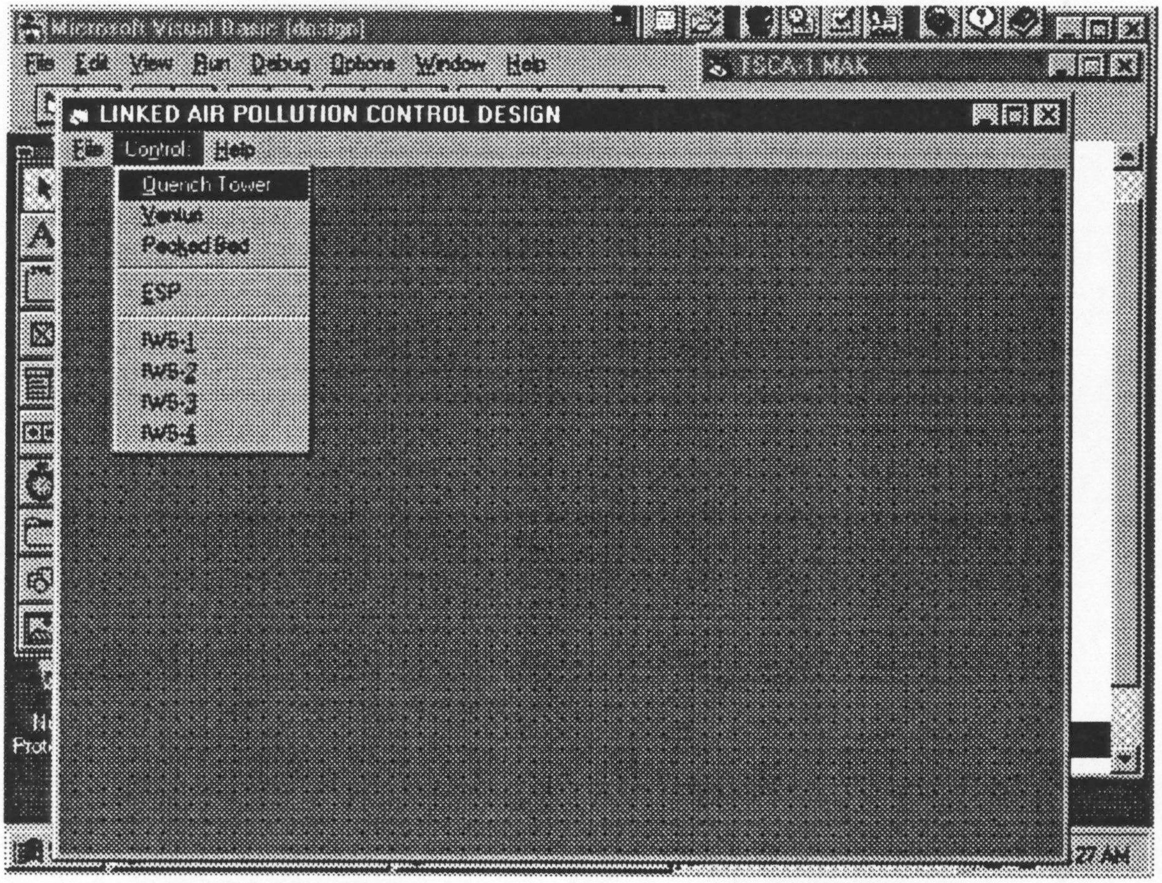

Figure 2-3. Control devices menu structure for VB application.

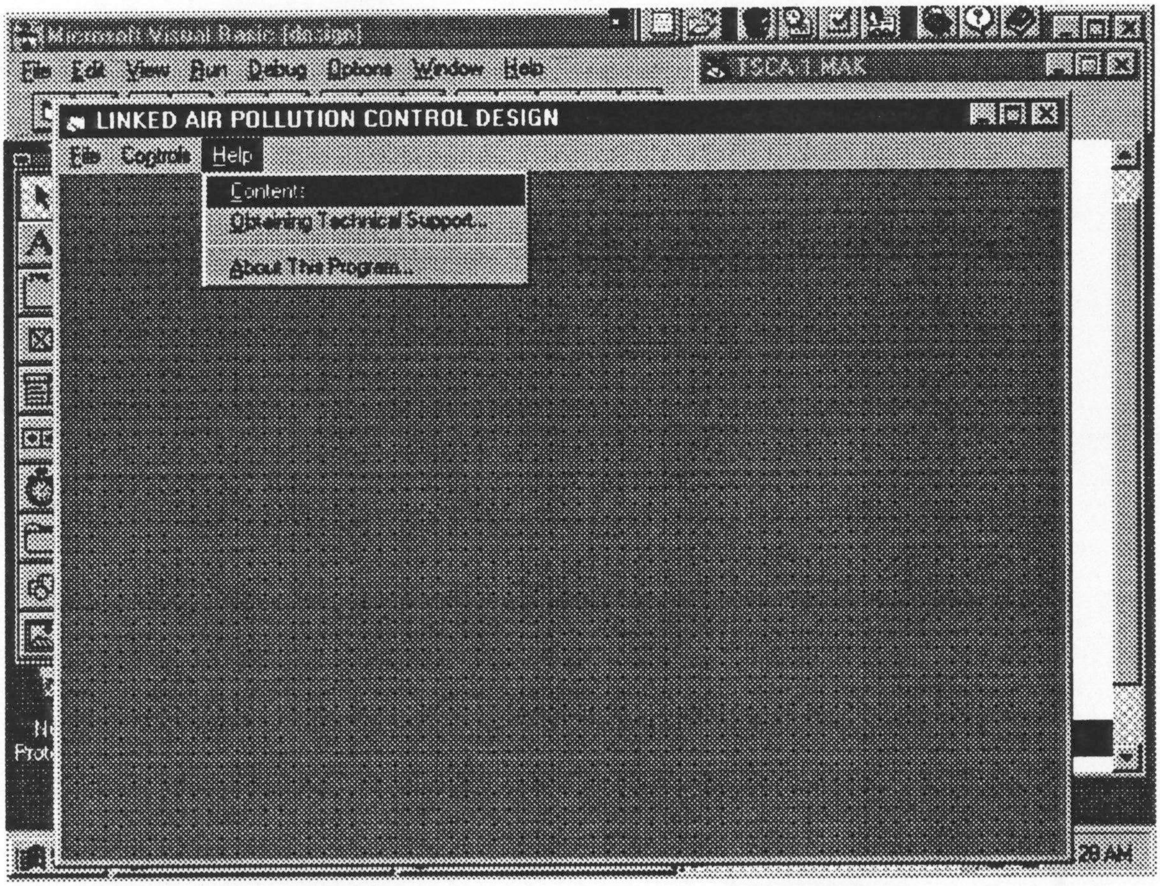

Figure 2-4. Help menu structure for VB application.

ease of tabbing through input boxes and command buttons. The command line on the top of the MDI form contains a designed menu structure (File, Controls and Help) which provides the gateway and access to the different modules in the application. The dropdown File menu structure elements (Inlet Gas Inputs, Inlet Particle Inputs, Print, Exit Program..) have been shown in Figure 2-2. Once the MDI form is loaded by the application and the user clicks on the File menu, the menu elements are displayed as a dropdown. The user has the option of starting the first module of the application - Inlet Gas Inputs or viewing the message box under Print or Exit the application. Unless the user has completed specifying the inlet gas characteristics, the application will not allow access to Inlet Particle Inputs on the menu. The schematic for the VB application is shown in Figure 2-5.

The application's front-end consists of user specified inputs for both the inlet gas and particle characteristics which are discussed in more detail later in this document. After securing these inputs, the application traverses sequentially through a quench tower, a venturi scrubber, a packed bed and finally either an ESP (conventional electrostatic precipitator) or a series of ionizing wet scrubbers (IWSs). The control devices menu structure is shown in Figure 2-3. The elements in this menu are - Quench Tower, Venturi, Packed Bed, ESP and IWS-1 through IWS-4. These elements can only be accessed sequentially after successfully completing the inputs/design for the previous element. Beyond the packed bed, the application provides for a choice of either accessing a traditional ESP treatment (multiple ESP sections available) or a series of up to four (4) IWSs. If the ESP option is exercised, the IWSs are no longer highlighted (or available)

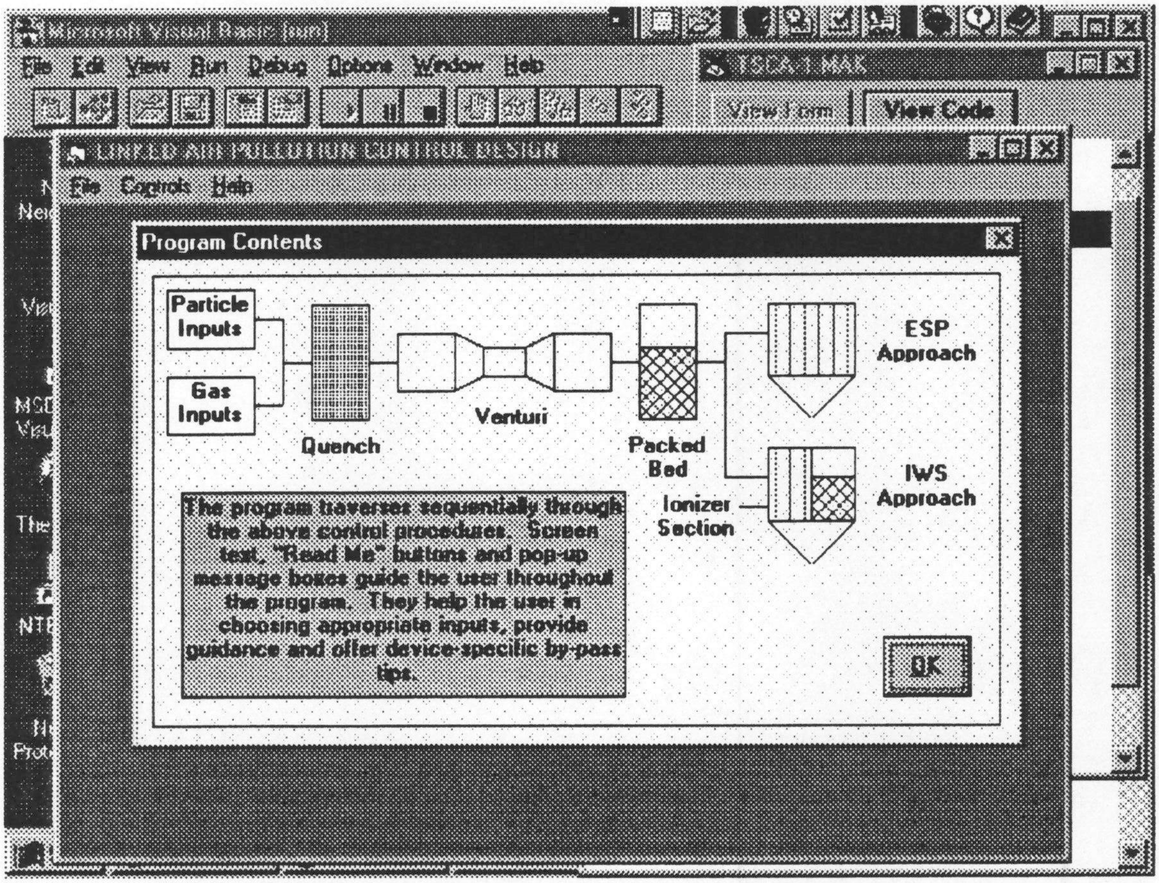

Figure 2-5. Program schematic screen for VB application.

and vice versa. Additionally, the user can reserve the option to bypass a particular control device by entering appropriate inputs discussed later in this document.

The Help menu structure elements (Contents, Obtaining Technical Support, About This Program) have been shown in Figure 2-4. These are typically either descriptions about the application or message boxes indicating whom to contact for additional technical support. In keeping with the unwritten industry conformity standard for Windows applications, the appearance of the application was designed to be devoid of distracting background palette colors, bright highlights and redundant object clutter.

The complete VB application flow schematic has been shown in Figure 2-6. After application startup, the user sequentially specifies the inlet gas characteristics and the particle characteristics. Then the application traverses through a sequence of units (indicated on the control devices menu structure: Quench Tower, Venturi, Packed Bed, ESP and IWS-1 through IWS-4) with options for by-pass with appropriate user inputs (e.g., if liquid addition and pressure drop are both specified as zero in the venturi scrubber module, then the venturi calculations result in no impact on the inlet gas/particle stream). Beyond the venturi scrubber, the user has the option of specifying either ESP calculations utilizing the Deutsch-Andersen equation or proceeding to the IWS calculations utilizing empirical efficiency prediction formulations. The ESP calculations can handle design of multiple sections. The application allows for up to four (4) IWSs in series with the option of ending the application after any single IWS.

The output printing option for the application was developed to encourage printing to a user-specified file (*c:\tsca.out* by default) in a chosen drive (e.g., c:\, d:\, a:\,

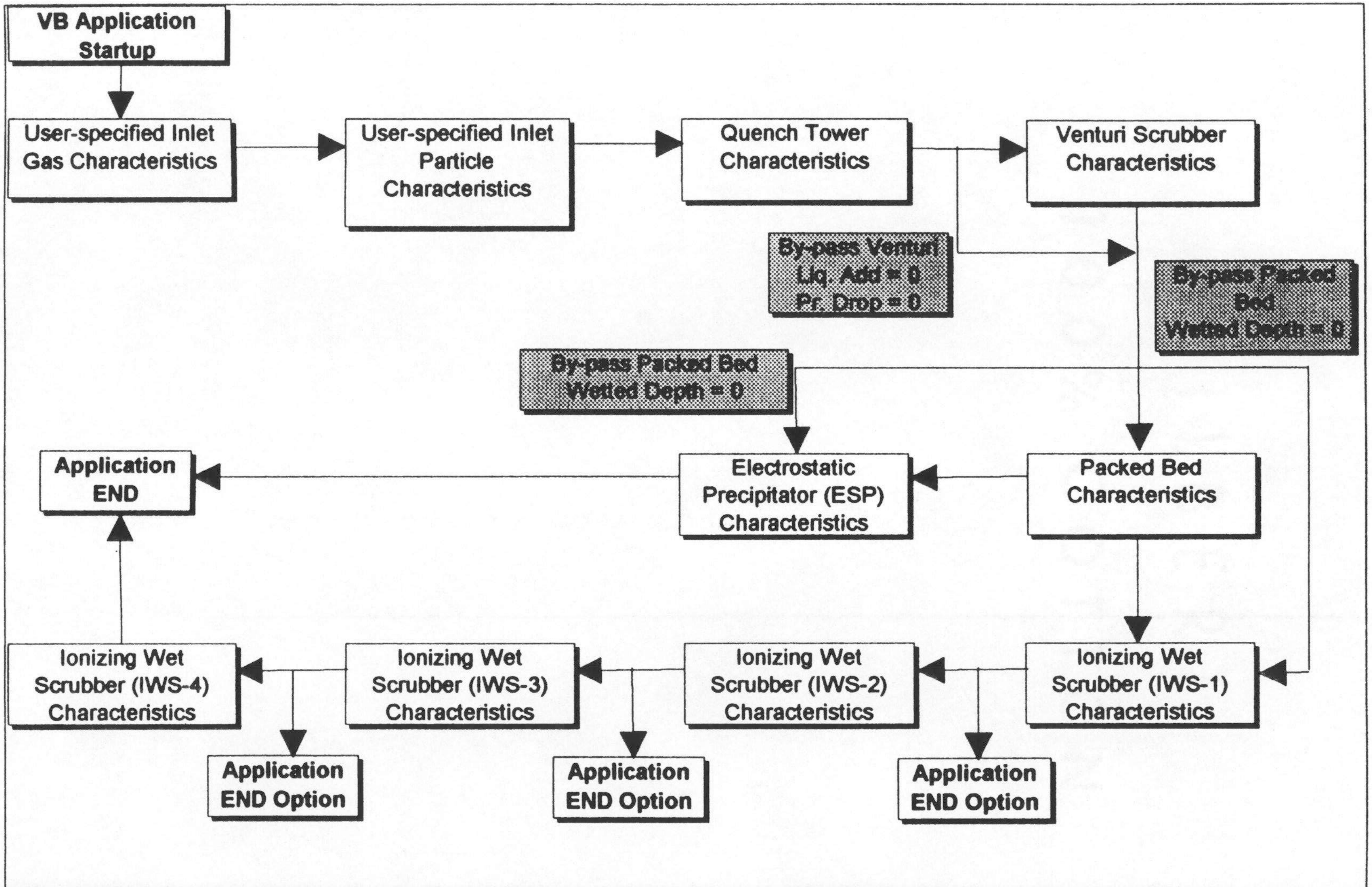

Figure 2-6. Visual Basic Application Flow Schematic for Design of Ionizing Wet Scrubber with Option for ESP Calculation.

b:\ ) at every procedure output screen. This option enables the user to save the output from a session in a comma-delimited text file which may be imported into a spreadsheet program (e.g., MS Excel) to generate an appropriately formatted program output.

3. VB APPLICATION EQUATIONS

The VB application utilizes a set of classical equations for initializing the inlet gas and particle characteristics based on physical attributes of the inlet stream. Next the application adopts a mix of widely accepted efficiency predictor equations along with the various empirical equations developed under this project to predict both device-specific and overall system particle collection efficiencies for a given APC system configuration. The application can be molded to define any system configuration (with the expedient of by-passing a redundant control device) containing any of the device elements included in the application package to predict the particle collection efficiency for any number of inlet gas-particle scenarios. This chapter is devoted to a detailed description of the various equations employed in the respective calculation modules of the application. The following chapter illustrates a sample IWS design case with a set of test case data.

3.1 INLET GAS CHARACTERISTICS

Inlet gas characteristics are specified by the user with the expedient of the following inputs:

(i) Inlet gas temperature (°F or °C)

(ii) Inlet gas molecular weight

(iii) Inlet gas static pressure (inches of H_2O)

(iv) Barometric Pressure (inches of Hg)

(v) Inlet gas flowrate (acfm or m^3/sec)

(vi) Inlet gas moisture content (vol%)

Depending upon the choice of user input, the application converts the inlet gas temperature (°F or °C) to Kelvin (K):

$$TEMPK(K) = TEMPC(°C) + 273°$$

$$TEMPK(K) = ((TEMPF(°F) - 32)/1.8) + 273°$$

Inlet gas molecular weight (g/gmole) can either be directly specified by the user (default value for air: 28.96) or the user can opt to specify component volume fractions of CO_2, O_2, N_2 and H_2O respectively. The user needs to ensure that the sum of the above volume fractions add to 100%. The following equation computes the inlet gas molecular weight based on the component volume fractions input:

$$MW = ((CO_2 * 44) + (O_2 * 32) + (N_2 * 28) + (H_2O * 18))/100$$

where MW is the inlet gas molecular weight, CO_2 is the volume fraction of carbon dioxide, O_2 is the volume fraction of oxygen, N_2 is the volume fraction of nitrogen and H_2O is the volume fraction of water in the inlet gas stream.

The absolute pressure of the inlet gas stream is computed based on its static pressure and the barometric pressure:

$$ABSPRESS = 29.92 + SPRESS/13.6$$

where ABSPRESS is the absolute pressure in inches of Hg and SPRESS is the inlet gas static pressure in inches of water. The user specified inlet gas flowrate (in acfm) is converted to m^3/sec which is utilized by the remainder of the application calculations. If the user opted to specify the inlet gas molecular weight by inputting component volume fractions, the application utilizes the H_2O volume fraction as the moisture content. Next, based on the user inputs the application computes the inlet gas density and gas viscosity with the following equations:

$$ROG = (MW/22.4) * (273/TEMPK) * (ABSPRESS/29.92)$$

$$VIS = (EXP(((Log(TEMPK) - 18.0075)/1.431))) * 0.1$$

where ROG is the inlet gas density in kg/m^3 (from Ideal Gas Law), MW is the inlet gas molecular weight, TEMPK is the inlet gas temperature in Kelvin, ABSPRESS is the absolute pressure in inches of Hg, and VIS is the dry gas viscosity in kg/m-sec (fit of the data available in Crawford, 1986; refer Section 3.3).

Finally, the application calculates the wet bulb temperature and the mass ratio of water to dry air with the help of the following equations:

$$MWD = (MW - 18 * (H2O/100))/ (1 - (H2O/100))$$

$$HV = H2O /(100 - H2O)$$

$$HM = HV * 18/ MWD$$

$$PW = ABSPRESS * H2O/100$$

where MWD is the inlet gas dry molecular weight, H2O is the inlet gas moisture content in volume %, HV is the molar ratio of lbmol water to lbmol dry gas, HM is the mass ratio (in lbs) of water to dry air, PW is the vapor pressure of water vapor in the inlet gas in inches of Hg. In order to calculate the wet bulb temperature of the inlet gas stream (TWB in K), the application utilizes an iterative solution of the Carrier equation (Cooper and Rossano, 1971) over the temperature range $273K \leq TWB \leq 473K$:

$$PWB = EXP\ ((-5800.22/TWB) + 1.39149 - 0.048602 * TWB + (0.417647 * 10^{-4}) * (TWB^2) - (0.1445 * 10^{-7}) * (TWB^3) + 6.5459 * Ln(TWB))\ [Pa]$$

$$PWB = PWB * 2.95 * 10^{-4} \quad \text{[Conversion of PWB to inches of Hg]}$$

$$TWBF = 1.8 * (TWB - 273) + 32$$

$$CK = PWB - ((ABSPRESS - PWB) * (TEMPF-TWBF)/(2830 - 1.3 * TWBF))$$

where PWB is the vapor pressure of water in Pascals (Keenan et al. 1969), TWBF is the inlet gas temperature in degrees Fahrenheit. When CK exceeds PW, the solution closure is provided and the last TWB value (K) is utilized by the application as the wet bulb temperature of the inlet gas stream.

3.2 INLET PARTICLE CHARACTERISTICS

The user has the option of either specifying the inlet particle characteristics as a log-normal particle size distribution or as actual mass fractions for a prescribed number of particle size increments. In the event, the user chooses to specify the inlet particle characteristics as a log-normal distribution, the following inputs are required:

(i) Smallest particle size, D_{min} (in μm, microns)

(ii) Largest particle size, D_{max} (in μm, microns)

(iii) Width of particle size increments (in μm, microns)

(iv) Mass Mean Diameter, MMD (in μm, microns)

(v) Geometric Standard Deviation, σ_g

(vi) Inlet dust loading (g/m^3)

(vii) Inlet particle density (g/cm^3)

Since the log-normal distribution is a continuous distribution, the program only utilizes finite end parts (i.e., D_{min} and D_{max}) as the boundaries within which the distribution is defined. By default, the application places any mass fraction for particle sizes less than D_{min} in the first (smallest) particle size increment and any mass fraction for particle sizes greater than D_{max} in the last (largest) particle size increment. It is recommended that the user specify D_{min} and D_{max} according to the following relationships which ensure that most of the particle sizes within the distribution will be represented appropriately by a log-normal distribution:

$$D_{min} = MMD / (2 * \sigma_g)$$

$$D_{max} = MMD * (2 * \sigma_g)$$

where MMD is the mass mean diameter and σ_g is the geometric standard deviation. Due to the manner in which the application allocates mass fractions to respective particle sizes, the above recommendation ensures that a wider range of particle sizes i.e., $\pm 2\sigma_g$ from the mean particle size (especially, the outlying lower and higher particle sizes) are appropriately represented and undue mass fraction weightage is not apportioned to an inappropriate particle size which can have adverse impacts on the particle collection prediction confidence. Note that for a log-normal particle size distribution, 68% of the mass of particles lies within $\pm 1\sigma_g$ from the mean particle size whereas 95% lies within $\pm 2\sigma_g$. Additionally, the application warns the user to check that based on the choice of the width of the size increments (WI), the number of size increments (Z) is an integer value. Since $Z = (D_{max} - D_{min})/WI$, depending on the choice of the variables, there can be instances when (Z) is a fractional value which can result in an erroneous output in the application. The manner in which the log-normal particle size distribution is coded in the application can be found in the Appendix.

In the event the user chooses to specify the inlet particle characteristics as actual mass fractions for a prescribed number of particle size increments, the following inputs are required:

(i) Smallest particle size, D_{min} (in μm, microns)

(ii) Largest particle size, D_{max} (in μm, microns)

(iii) Width of particle size increments (in μm, microns)

(iv) Inlet gas dust loading (g/m^3)

(v) Inlet gas particle density (g/cm^3)

(vi) Mass fraction input for number of size increments

The user needs to exercize caution in inputting mass fraction values for the respective particle sizes and ensure that the sum of the input mass fractions equal one (1.0). In the event that the sum of the mass fractions are not equal to 1.0, the application prompts the user to re-enter the mass fractions. Note that in this option for specifying particle size characteristics, MMD and σ_g are not required to be specified since the particle size distribution has not been prescribed to be log-normal. Likewise, this option avoids the need for the model to place mass less than D_{min} in the first size increment and mass greater than D_{max} in the last size increment. The user makes that decision rather than using the default.

After the inlet gas and particle characteristics have been user-specified, the user is ready to sequentially invoke the respective devices in the APC configuration for the application package.

3.3 QUENCH TOWER CHARACTERISTICS

The quench tower is utilized in the APC configuration to cool the inlet gas stream. Based on the inlet gas characteristics, the application computes the amount of water required to reach saturation conditions (WIT, kg/sec) at gas saturation temperature (i.e., TWBF, inlet gas wet bulb temperature in degrees Fahrenheit) as follows:

MCSAT = (PWB/ABSPRESS) * 100	Moisture % at saturation
HVSAT = MCSAT / (100 - MCSAT)	Mole fraction of water/dry air at sat.
HMSAT = HVSAT * 18/ MWD	Mass fraction of water/dry air at sat.
MGAS = FLOW * ROG	Mass of gas flow, kg/sec
MFRACH2O = HM / (HM + 1)	Mass fraction water/gas before inject
MH2O = MGAS * MFRACH2O	Mass of inlet water, kg/sec
MAIR = MGAS * (1 - MFRACH2O)	Mass of dry air, kg/sec
WIT = (HMSAT * MAIR) - MH2O	

where PWB is the vapor pressure of water in inches of Hg, ABSPRESS is the absolute pressure in inches of Hg, MWD is the inlet gas dry molecular weight, FLOW is the inlet gas volumetric flow in m^3/sec, ROG is the inlet gas density in kg/m^3 and HM is the mass ratio of water to dry air.

For the quench tower, the application prompts the user to specify an actual water injection rate (WI, kg/sec) greater than WIT, kg/sec and a device pressure drop in inches of H_2O (DELPQ). Based on these two user inputs, the application calculates the following parameters:

(i) Outlet gas temperature, TEMPKQ (K)

(ii) Outlet static pressure, SPRESSQ (inches of H_2O)

(iii) Absolute pressure, ABSPRESSQ (inches of Hg)

(iv) Outlet gas molecular weight, MWQ

(v) Outlet gas density, ROGQ (kg/m^3)

(vi) Outlet gas viscosity, VISQ (kg/m-sec)

(vii) Outlet gas flowrate, FLOWQ (m^3/sec)

(viii) Outlet gas moisture content, H2OQ (volume %)

(ix) Mass of dry air, MAIR (kg/sec)

(x) Mass of outlet water vapor, MH2OQ (kg/sec)

(xi) Water remaining as liquid, WNETQ (kg/sec)

The above parameters are calculated with the help of the following equations:

MH2OQ = MH2O + WIT Mass of water entering quench, kg/sec

WNETQ = WI - WIT Mass of water leaving quench as carryover

TEMPKQ = TWB Outlet gas temperature, K

TEMPFQ = 1.8 * (TEMPKQ - 273°) + 32 Outlet gas temperature, °F

MWQ = MWD * (1 - (MCSAT/100)) + 18 * (MCSAT/100) Wet Mol. Wt.

SPRESSQ = SPRESS - DELPQ Outlet static pressure, inches of H_2O

ABSPRESSQ = ABSPRESS + SPRESSQ/ 13.6 Absolute pressure, in Hg

ROGQ = (MWQ/22.4) * (273/TEMPKQ) * (ABSPRESSQ/ 29.92)

H2OQ = MCSAT Outlet moisture content, volume %

FLOWQ = (MAIR + MH2O + WIT)/ ROGQ Outlet gas flowrate, m^3/sec where MH2O is the mass of inlet water in kg/sec, TWB is the inlet gas wet bulb temperature in Kelvin, MWD is the inlet gas dry molecular weight, MCSAT is the volumetric moisture percent at saturation, SPRESS is the inlet gas static pressure in inches of H_2O and ABSPRESS is the inlet gas absolute pressure in inches of Hg.

For dry air the gas viscosity in kg/m-sec)can be calculated with the following equation (fit of the data in Crawford, 1986):

VISQ = (EXP(((Log(TEMPKQ) - 18.0075) / 1.431))) * 0.1

However, for a moist gas stream, the above equation cannot be applied. The viscosity of a low-pressure gas mixture can be calculated from kinetic theory but the computations are complicated (Perry's Chemical Engineers' Handbook, 1984). Based on the rigorous solution derived from kinetic theory of gases, a simpler form of the solution given in Perry's Chemical Engineers' Handbook (1984) was adopted:

$$\mu_{mix} = \sum_{i=1}^{n}\left(y_i \mu_i \sqrt{M_i}\right) \Big/ \sum_{i=1}^{n}\left(y_i \sqrt{M_i}\right)$$

where μ_{mix} is the viscosity of a low-pressure gas mixture containing n number of components with gas-phase mole fractions y_i, μ_i is the viscosity of a pure component at the same temperature and at a low pressure and M_i is the dry molecular weight of the individual components. Based on gas viscosity data available in the CRC Handbook of Chemistry and Physics (1993), linear regression models for air and water vapor were developed in the temperature range of 300-600K and pressure of 1 atmosphere which is appropriate for typical quench tower outlet conditions:

$$\mu_{air} = 4.06 \times 10^{-8} \text{ (T in Kelvin)} + 6.63 \times 10^{-6} \quad \text{(kg/m-sec)}$$

$$\mu_{water} = 3.82 \times 10^{-8} \text{ (T in Kelvin)} - 1.76 \times 10^{-6} \quad \text{(kg/m-sec)}$$

where μ_{air} is the viscosity of air, μ_{water} is the viscosity of water vapor and (T in Kelvin) is the reference temperature in Kelvin. At T=350K, the variation of gas viscosity adjusted for a range of moisture contents shows that μ_{mix} can be significantly lower than μ_{air} by as much as 29%. The sample calculation is shown in Table 3-1 below.

Table 3-1. Sample Calculation Showing the Variation of Gas Viscosity Adjusted for Moisture Content at T=350K.

MWD	Air %	H2O %	Visc-mix kg/m-sec	Visc-air kg/m-sec	%Diff
28.96	100%	0%	2.08E-05	2.08E-05	0.00%
29.54	95%	5%	2.04E-05	2.08E-05	1.75%
30.18	90%	10%	2.01E-05	2.08E-05	3.50%
30.89	85%	15%	1.97E-05	2.08E-05	5.25%
31.70	80%	20%	1.93E-05	2.08E-05	7.01%
32.61	75%	25%	1.90E-05	2.08E-05	8.78%
33.66	70%	30%	1.86E-05	2.08E-05	10.55%
34.86	65%	35%	1.82E-05	2.08E-05	12.34%
36.27	60%	40%	1.79E-05	2.08E-05	14.14%
37.93	55%	45%	1.75E-05	2.08E-05	15.94%
39.92	50%	50%	1.71E-05	2.08E-05	17.77%
42.36	45%	55%	1.67E-05	2.08E-05	19.61%
45.40	40%	60%	1.63E-05	2.08E-05	21.48%
49.31	35%	65%	1.59E-05	2.08E-05	23.39%
54.53	30%	70%	1.55E-05	2.08E-05	25.33%
61.84	25%	75%	1.51E-05	2.08E-05	27.34%
72.80	20%	80%	1.47E-05	2.08E-05	29.43%

The above data is graphically represented in Figure 3-1. For the application, in order to develop the outlet gas viscosity from the quench tower, the above relationships were incorporated in a gas viscosity calculation module and which can be found in the Appendix.

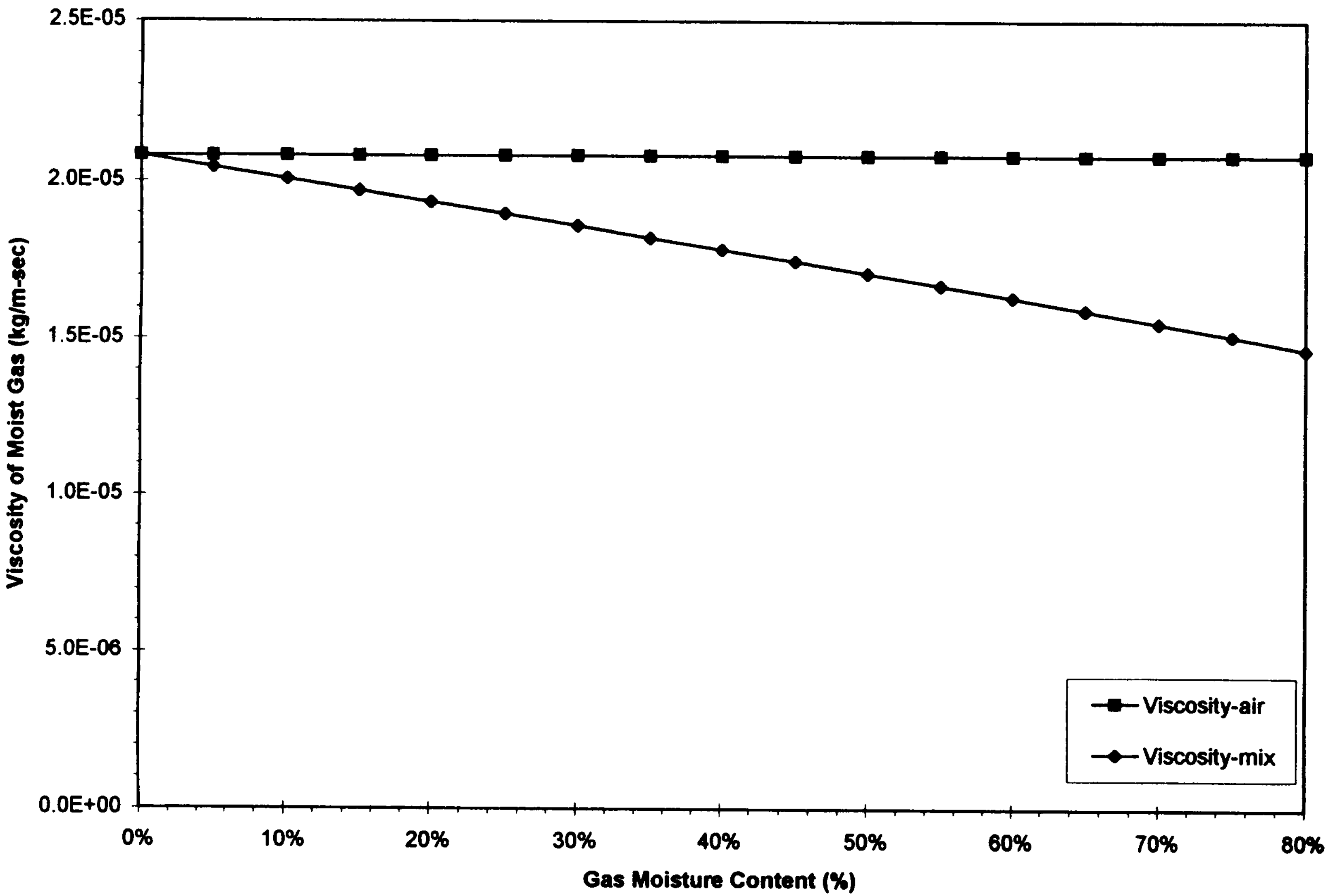

Figure 3-1. Effect of Moisture Content on Gas Viscosity at T=350K

3.4 VENTURI SCRUBBER CHARACTERISTICS

The venturi scrubber is the first control device in the APC configuration which is intended to serve as a pre-collector for the coarser particle fractions. The inlet gas stream first enters the flared section of the device and then proceeds to travel through a tapered section to the throat of the venturi before exiting through another tapered section followed by a flared section to the next control device in the system configuration. As the gas stream enters the throat of the venturi, it encounters a lot of turbulence. Due to Bernoulli's principle of conservation of total head for incompressible flows, the gas stream velocity is significantly enhanced in the throat section of the venturi (around 6,000-30,000 fpm depending on design). Liquid (usually water) is introduced as a spray into the throat section of the venturi at the rate of 10-20 gal/1000 acfm of gas to provide sufficient gas-particle contact. The liquid droplets are atomized almost instantaneously to around 80-100 μm and suspended particles in the gas stream are collected by inertial impaction which is very efficient for coarser particle sizes and increases with particle size (Cooper and Alley, 1994).

The application informs the user that for venturi calculations, either pressure drop (PR, in H_2O) or throat velocity (VG, ft/min) has to be specified since these are dependent on each other. Once one of the above parameters are specified, the other will be calculated. To facilitate this within the program, if pressure drop has been specified, then throat velocity needs to be input as zero and vice versa. If it is desired to simply by-pass the venturi (i.e., no venturi in the control system), the user needs to enter zero liquid addition (QL=0 gal/min) and zero pressure drop (PR=0 in H_2O). In this event, the user has to enter a non-zero value for throat velocity which anyway becomes redundant in the calculations. If the user enters a zero value for throat velocity, then a message box stops the execution of the program stating that "Both pressure drop and throat velocity cannot be zero". Then the user is urged to re-enter the venturi inputs. For specifying the venturi scrubber characteristics, the application prompts the user for the following combination of inputs:

(i) Liquid Injection Rate, QL (gal/min)

(ii) Venturi Pressure Drop, PR (inches of H_2O)

(iii) Gas Velocity in Throat, VG (ft/min)

(iv) F-factor (Default F=0.25)

(v) Liquid Density, RL (for water 1 g/cm^3)

The application then converts the user inputs into c.g.s. units utilized in the calculations:

PR = PR * 2.54	[cm of H_2O]
QL = QL * (3783/ 60)	[g/sec or cm^3/sec for water]
FLOWV = FLOWQ * 10^6	[cm^3/sec] FLOWQ was the flow through quench.
TEMPKV = TEMPKQ	[K] TEMPKQ was gas outlet temp from quench.
H2OV = H2OQ	[volume %] Unchanged moisture content of gas.
VG = VG * (30.48/ 60)	[cm/sec]

It is assumed that the gas viscosity is unchanged, i.e.,

VISV = VISQ * 10	[g/cm-sec]

When PR=0 and VG has been specified, the calculation of the pressure drop is made using the pressure drop equation as reported by Calvert in Wark and Warner (1981):

$$PR = 1.03\times10^{-3} * (VG)^2 * QL/ FLOWV \qquad [\text{cm of } H_2O]$$

Similarly, when VG=0 and PR has been specified, the throat velocity is calculated as follows:

$$VG = ((PR * FLOWV)/ (1.03\times10^{-3} * QL))^{0.5} \qquad [\text{cm/sec}]$$

Thus, the application has the flexibility to utilize both forms of user input and calculate the parameters necessary to specify the remainder of the venturi scrubber characteristics as follows:

DELPV = PR / 2.54 [inches of H_2O] Venturi pressure drop.

SPRESSV = SPRESSQ - DELPV [inches H_2O] Static pressure.

ABSPRESSV = ABSPRESSQ + SPRESSV /13.6 [inches Hg]

where ABSPRESSV is the absolute pressure at the exit of the venturi. The next calculation module involves the calculation of the droplet size (DD, in cm) using the Nukiyama-Tanasawa (1938) equation whose full form is expressed as follows (Cooper and Alley, 1994):

$$DD = 58{,}600/V_G * (\sigma/\rho_L)^{0.5} + 597 * (\mu_L/(\sigma/\rho_L)^{0.5})^{0.45} * (1000 * QL/FLOWV)^{1.5} \quad [\text{in } \mu\text{m}]$$

where ρ_L is the density of the liquid in g/cm^3 (usually, 1.0 for water which is used in most venturi scrubbers), σ is the liquid surface tension in dynes/cm, and μ_L is the liquid viscosity in poise. Based on the value of TEMPKV (venturi gas temperature in Kelvin), the application calculates the values for liquid surface tension (σ) and liquid viscosity (μ_L) from fits of the data available in the CRC Handbook of Chemistry and Physics (1993).

The Handbook presents surface tension and viscosity data for liquid water in the temperature range of 0-100 °C (or 273-373K) which contains the domain of expected gas temperatures in the venturi scrubber (typically around 300-350K for the present application). The data was fitted to the following equations which are utilized by the application:

$$\sigma = -0.1676 * \text{TEMPKV} + 121.81 \qquad \text{[in dynes/cm]}$$

$$\mu_L = 2e{+}12 * \text{TEMPKV}^{-5.8009} \qquad \text{[in poise]}$$

After calculation of DD in microns (μm) as given by the Nukiyama-Tanasawa equation, the application converts the value of DD to cm for the venturi calculations.

The application next calculates the Cunningham correction factor (KC) for the various mean particle sizes DP(I) for each of the (Z) particle size increments, i.e., I = 1 to Z. As particle sizes become smaller e.g., DP(I) $\leq$ 5 μm, they approach the mean free path of gas molecules and tend to slip past the gas molecules resulting in higher velocities under an applied external force than that predicted by Stokes law (Wark and Warner, 1981). The Cunningham correction factor adjusts for this anomaly and becomes very significant at smaller particle sizes. Wark and Warner (1981) presented a simplified form for dimensionless KC for particle sizes greater than 1 μm and which works very well for higher particle sizes:

$$\text{KC(I)} = 1 + (9.73\text{x}10^{-3} * \text{TEMPKV}^{0.5}) / \text{DP(I)}$$

where DP(I) is the particle size in μm, microns and TEMPKV is the gas temperature in Kelvin. For the application, the full-blown version of the KC relationship was utilized which is as follows (Wark and Warner, 1981):

$RG = 8314.3$ [J/kgmol-°K] Universal gas constant

$TV = ((8 * RG * TEMPKV / (\pi * MWQ)))^{0.5}$ [m/sec]

$LAMBDA = (VISV / 10) / (0.499 * ROGQ * TV)$ [m]

$SIM1(I) = 1.257 + 0.4 * EXP(-0.55 * DP(I) * 0.000001) / LAMBDA)$

$KC(I) = 1 + 2 * LAMBDA * SIM1(I) / (DP(I) * 0.000001)$

where TV is the mean thermal velocity of gas molecules, TEMPKV is the gas temperature in the venturi in Kelvin, MWQ is the gas molecular weight from the quench, VISV is the gas viscosity in the venturi which has been adjusted for moisture in the quench tower in kg/m-sec, LAMBDA is the mean free path of gas molecules in meters, SIM1(I) is a part expression of KC(I), DP(I) is the mean particle size for a given size interval in μm and KC(I) is the dimensionless Cunningham correction factor.

After computation of the Cunningham correction factor, the application utilizes the particle penetration equations developed by Calvert et al. (1972) to predict the particle collection efficiency of the venturi scrubber:

$SI(I) = VG * ROP(1) * KC(I) * ((DP(I) * 10^{-4})^2 / (9 * VISV * DD)$

$FK(I) = (-0.7 - SI(I) * F + 1.4 * LOG((SI(I) * F + 0.7)/ 0.7) + (0.49/ (0.7 + SI(I) * F)))/ SI(I)$

$PT(I) = EXP((2 * QL * VG * RL * DD * FK(I))/ (55 * FLOWV * VISV))$

$EFF(I) = 1 - PT(I)$

where SI(I) is the dimensionless inertial impaction parameter, F is the dimensionless empirical F-factor which is 0.25 for hydrophobic particles (by default) and 0.50 for hydrophilic particles, PT(I) is the particle penetration through the venturi scrubber and EFF(I) is the particle collection efficiency of the venturi scrubber.

The application keeps up with all the individual DP(I) (the mean particle size for every size increment up to Z increments) sizes, computes the fractional efficiency for every size increment and the overall device efficiency. Due to the particle collection efficiency of the venturi scrubber, the sum of the particle mass fractions penetrating the venturi scrubber Σ MPV(I) is no longer equal to unity (1.0). Therefore, the application normalizes the particle mass fractions exiting the venturi scrubber such that the sum of the mass fractions is unity (1.0) before they enter the next control device. This procedure is repeated for every control device selected by the user. The procedure ensures that in addition to an overall system efficiency, individual device efficiencies can also be predicted accurately.

3.5 PACKED BED CHARACTERISTICS

The packed bed is utilized in the post-venturi APC configuration to provide an additional device for collection of coarser particle fractions for particle size distributions with few fines and thus, prevent loading of the downstream control devices. The packed bed calculations follow the empirical equation developed at UTK under this contract (please refer Morse, 1996 for details). The empirical equation predicts the efficiency of a cross-flow packed bed containing 2-inch type-R tellerette packing. The model was developed to describe the penetration of the packed bed as a function of particle aerodynamic diameters DAERO(I). Since the prediction model is based on actual experimental data, it is limited by particle sizes in the range $0.12 \leq$ DAERO(I) ≤ 5 μm (Morse, 1996). Since, the only user input for this calculation module is the depth of wetted packing DPB in feet, should the user opt to by-pass the packed bed, then DPB can be set to zero. The form of the empirical prediction equation is given by:

$$\mathrm{DAERO(I)} = \mathrm{DP(I)} * \mathrm{ROP(1)}^{0.5} \quad [\mu\mathrm{m}]$$

$$\mathrm{PBEFF(I)} = 1 - (((0.006199 * \mathrm{DAERO(I)}^3) - (0.060103 * \mathrm{DAERO(I)}^2) + (0.011302 * \mathrm{DAERO(I)}) + 0.77037)^{(\mathrm{DPB}/8)}$$

where PBEFF(I) is the fractional efficiency of the packed bed and DPB is the depth of the wetted packing in feet. Figure 3-2 shows a graphical representation of the above model.

Due to the limitations imposed by the above empirical equation, the user should be cognizant of the DAERO(I) limits, especially for packed bed efficiency predictions outside of the valid DAERO(I) range. The application imposes an artificial efficiency prediction curb on particle sizes greater than DAERO(I) = 5 μm by reporting a constant fractional efficiency value equal to the efficiency at 5 μm for aerodynamic particle diameters greater than 5 μm which, in effect, translates to a more conservative packed bed efficiency estimate. Additionally, the application also reports an overall system efficiency (i.e., 1- product of penetrations for the individual control devices) for the APC configuration selected thus far. At this juncture, the user can terminate the design or continue selecting <u>either</u> the design of IWSs (up to four in series) or the option for multiple-section ESP calculations utilizing the Deutsch-Andersen equation. The ESP calculations are discussed in the following section followed by a discussion on the IWS calculations.

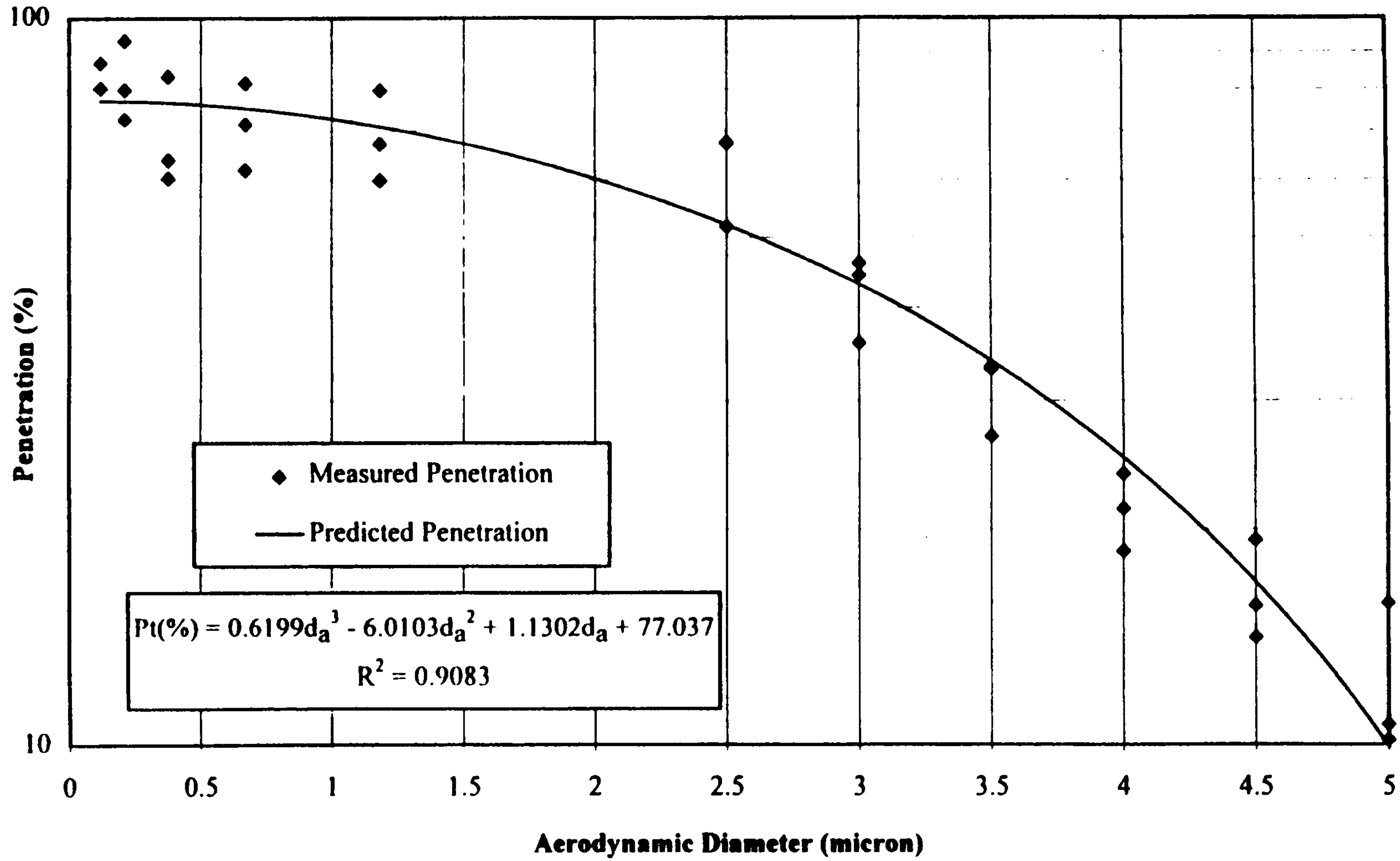

Figure 3-2. Measured and Predicted Penetrations for Packed Bed.
Source: Morse, Thomas C. "Characterization of Particle Collection Efficiency in an Ionizing Wet Scrubber." M.S. Thesis, University of Tennessee, Knoxville, TN (1996).

3.6 ESP CHARACTERISTICS

The electrostatic precipitator (ESP) is one of the two choices for a final particle control device utilized in the APC configuration for collection of the finer particle fractions. In a given APC system configuration, it is the ESP performance which will determine whether or not a particular facility will meet the regulatory particulate emission standards. Therefore, specifying the ESP characteristics appropriately is of paramount importance. The intent behind providing an ESP calculation option in this application package was to encourage the user to run comparative calculations with both the ESP and the IWS option and then ascertain which APC system configuration afforded a better overall system efficiency.

The fundamental mechanisms behind an ESP operation involve the following (Cooper and Alley, 1994):

(a) Ionization of the inlet gas stream passing between a set of collecting plates and an array of charging wires.

(b) Charging, migration and collection of the particles on oppositely charged collection plates.

(c) Removal of the collected particles from the collection plates by either rapping or wetting agents such as water (i.e., wet ESP).

Based on user-defined inputs, the application employs the Deutsch-Andersen equation to predict the efficiency of particle collection for an ESP with either Field Charging or both Field and Diffusion Charging. The ESP user input screen requests the following input parameters:

(i) Number of ESP sections in series, NS (design of multiple sections possible)

(ii) Depth of ESP sections, DSECT (meters)

(iii) Active Height of ESP plates, HPLATE (meters)

(iv) Wire-to-plate spacing, L (usually, 10-15 cm)

(v) Inlet gas velocity, VGESP (usually, less than 3 m/sec)

(vi) Pressure drop across ESP, DELPE (if unknown, then enter 1.5 inches of H_2O)

(viii) Radius of corona wires, RW (usually, 0.00127 m for 0.1-inch diameter wires)

(ix) Sneakage around plate section as a fraction, SN (may be zero due to plate extending into hopper, typical value around 0.05)

(x) Particle re-entrainment as a fraction, RR (usually, zero for wet ESPs and IWSs, typically 0.12 for dry ESPs)

(xi) Choice of either Field Charging or Both Field + Diffusion Charging.

Based on the above user inputs, the application first computes system design initialization parameters:

QWIREPLATE = VGESP * HPLATE * L	[m^3/sec]
AREAPLATES = HPLATE * DSECT	[m^2]
SCA = NS * AREAPLATES / QWIREPLATE	[sec/m]
ESCA = SCA * 5.08	[ft^2/1000 acfm]
EPS = 8.85×10^{-12}	[Coulombs/volt-meter]
TEMPKE = TEMPKV	[Kelvin]
SPRESSE = SPRESSV - DELPE	[inches of H_2O]
ABSPRESSESP = ABSPRESSV + SPRESSE/ 13.6	[inches of Hg]

where QWIREPLATE is the flowrate between single wire and plate in m^3/sec, AREAPLATES is the area of plate for wire-plate configuration in a section in m^2, SCA is the specific collection area in m^2 per m^3/sec or sec/m, ESCA is the specific collection area in English units of ft^2/1000 acfm, EPS is the space permittivity in coulombs/volt-m, TEMPKE is the inlet gas temperature of the ESP which is the same as that at the outlet of the venturi scrubber in Kelvin, SPRESSE is the static pressure in inches of H_2O and ABSPRESSESP is the absolute pressure in inches of Hg.

Next the application calculates the field strength (i.e., E in volt/m) beyond which sparkover (dead short when current goes to $\propto$ infinity) occurs. By convention and based on field experience, the actual operational field strength (E in volt/m) which is approximately E/1.75 is computed as follows (Turner et al., 1992):

$$E = 6.3\text{x}10^5 * (((273/\text{TEMPKE}) * (\text{ABSPRESSESP}/29.92))^{1.65}) / 1.75 \quad [\text{V/m}]$$

The voltage across the plates is then estimated as follows:

$$\text{VMAX} = E * L \quad [\text{Volts}] \ \text{Operational voltage.}$$

For a given plate-wire configuration, there is a critical field strength below which a corona is not formed (i.e., no charging takes place). The application utilizes the equations given in Turner et al. (1992) to check the voltage required to initiate corona at edge of wire such that **VMAX > VCOR** (VMAX: maximum voltage, VCOR: voltage required for corona generation).

$$\text{BB} = (293 * \text{ABSPRESSESP}) / (\text{TEMPKE} * 29.92)$$

$$\text{EMAX} = 3.126\text{x}10^6 * 0.6 * (\text{BB} + 0.03 * ((\text{BB}/\text{RW})^{0.5})) \quad [\text{Volt/m}]$$

where EMAX is the field strength at the edge of the round wires required to start corona.

$$VCOR = EMAX * RW * Ln((4 * L)/ (\pi * RW)) \qquad [Volt]$$

$$ECOR = VCOR / L \qquad [Volt/m]$$

where VCOR is the voltage required to start corona and ECOR is the corresponding field strength across space. If (ECOR > E) then the operational ESP field strength is less than that required to initiate a corona and the user is prompted to re-enter the ESP inputs to correct the discrepancy. In the ESP inputs screen, the user can either opt for Field Charging or both Field and Diffusion Charging mechanisms. The application algorithm is set up as follows with respect to variable ETIME to accommodate either option:

ETIME = 0 ....Field Charging.

ETIME = VGESP * DSECT ....Both Field + Diffusion Charging.

where ETIME is the residence time of the particles (in sec) in the ESP section. For the latter option, it is assumed that residence time through the ESP section is totally available for diffusion charging. Note that field or impact charging involves ion attachment to particles in the electric field and is very effective for particle sizes greater than 0.5 μm. Conversely, diffusion charging involves ion attachment to particles due to ion diffusion depending on thermal energy but not on the strength of the electric field and is very effective for particle sizes less than 0.2 μm. Next the application calculates the various terms for saturation charge (Turner et al., 1992):

$$LL = 2 * (5.3\times10^{-26}) * \pi * (1.38\times10^{-23}) * TEMPKE$$

$$KK(I) = 1 + (1.6\times10^{-19})^2 * (2\times10^{15}) * ETIME*DPSI(I)/ (2* 8.85\times10^{-12} * LL^{0.5})$$

$$Q(I) = 3 * \pi * EPS * E * DPSI(I)^2 + (2 * \pi * EPS * 1.38\times10^{-23} * TEMPKE * DPSI(I)/ (1.6\times10^{-19}) * Log(KK(I))$$

$$W(I) = Q(I) * KC(I) * E / (3 * \pi * VISQ * DPSI(I)) \qquad [m/sec]$$

$$PCQ(I) = EXP(-(SCA / NS) * W(I) / (1 - SN)) \qquad [\text{Deutsch-Andersen eqn.}]$$

$$PTESP(I) = SN + (1 - SN) * PCQ(I) + RR * (1 - SN) * (1 - PCQ(I))$$

where W(I) is the migration velocity in m/sec, DPSI(I) is similar to the mean particle size DP(I) in a given size increment except that it is expressed in meters, PCQ(I) is the particle penetration predicted by the Deutsch-Andersen equation and PTESP(I) is the total particle penetration in the first section of the ESP (recall that the algorithm can handle multiple ESP sections from 1 to NS as defined by user input). As discussed earlier, the algorithm keeps up with all the individual DPSI(I) particle sizes, computes the fractional efficiency for every particle size increment, the overall device efficiency and an overall system efficiency.

3.7 IWS CHARACTERISTICS

Like the ESP, the ionizing wet scrubber (IWS) is utilized in the APC system configuration for collection of finer particle fractions and is the focal point in a facility's ability to comply with regulatory particulate emission standards. However, unlike the ESP calculation option discussed earlier, the application's treatment of this device design module is somewhat different. IWS calculations utilize the Specific Collection Area (SCA)-specific empirical equations developed at UTK (please refer to Morse, 1996 for details). Since theoretical equations are not available, three equations were developed by Morse (1996) to characterize the IWS behavior which consists of an ionizer section followed by a packed bed in series. The packed bed is a cross-flow type containing 2-inch type-R tellerette packing.

The IWS is a very efficient particulate control device which has been field-tested to show particulate collection efficiencies as high as 99.7% (Morse, 1996). The principal particle collection mechanisms are inertial impaction and electrostatic forces. Studies have been performed at UTK to deepen the understanding of electrostatic collection (i.e., image force) and further quantify the electrostatic mechanism (Adams, 1997). The form of the empirical equations are given as follows (Morse, 1996):

<u>IWS Model: SCA= 62.6 ft²/1000 acfm</u>

$$Pt_{(62.6)IWS} = \exp[-(-0.48\, d_p^2 + 3.07\, d_p + 2.02)] \times [0.006199\, d_a^3 - 0.060103\, d_a^2 + 0.011302\, d_a + 0.77037]^{(L/8)}$$

<u>IWS Model: SCA= 34.8 ft²/1000 acfm</u>

$$Pt_{(34.8)IWS} = \exp[-(-0.43\, d_p^2 + 2.60\, d_p + 1.20)] \times [0.006199\, d_a^3 - 0.060103\, d_a^2 + 0.011302\, d_a + 0.77037]^{(L/8)}$$

<u>IWS Model: SCA= 28.5 ft²/1000 acfm</u>

$$Pt_{(28.5)IWS} = \exp[-(-0.66\, d_p^2 + 2.93\, d_p + 0.74)] \times [0.006199\, d_a^3 - 0.060103\, d_a^2 + 0.011302\, d_a + 0.77037]^{(L/8)}$$

where Pt is the fractional penetration, d_p is the physical diameter of the particle in μm, d_a is the aerodynamic diameter of the particle in μm, and L is the depth of the wetted packing in the direction of gas flow in feet. Figure 3-3 shows a graphical representation of the above model. Besides selecting the type of equation best suited for a user's needs, the only other input for IWS design is L in feet. If the user is not satisfied with the particle

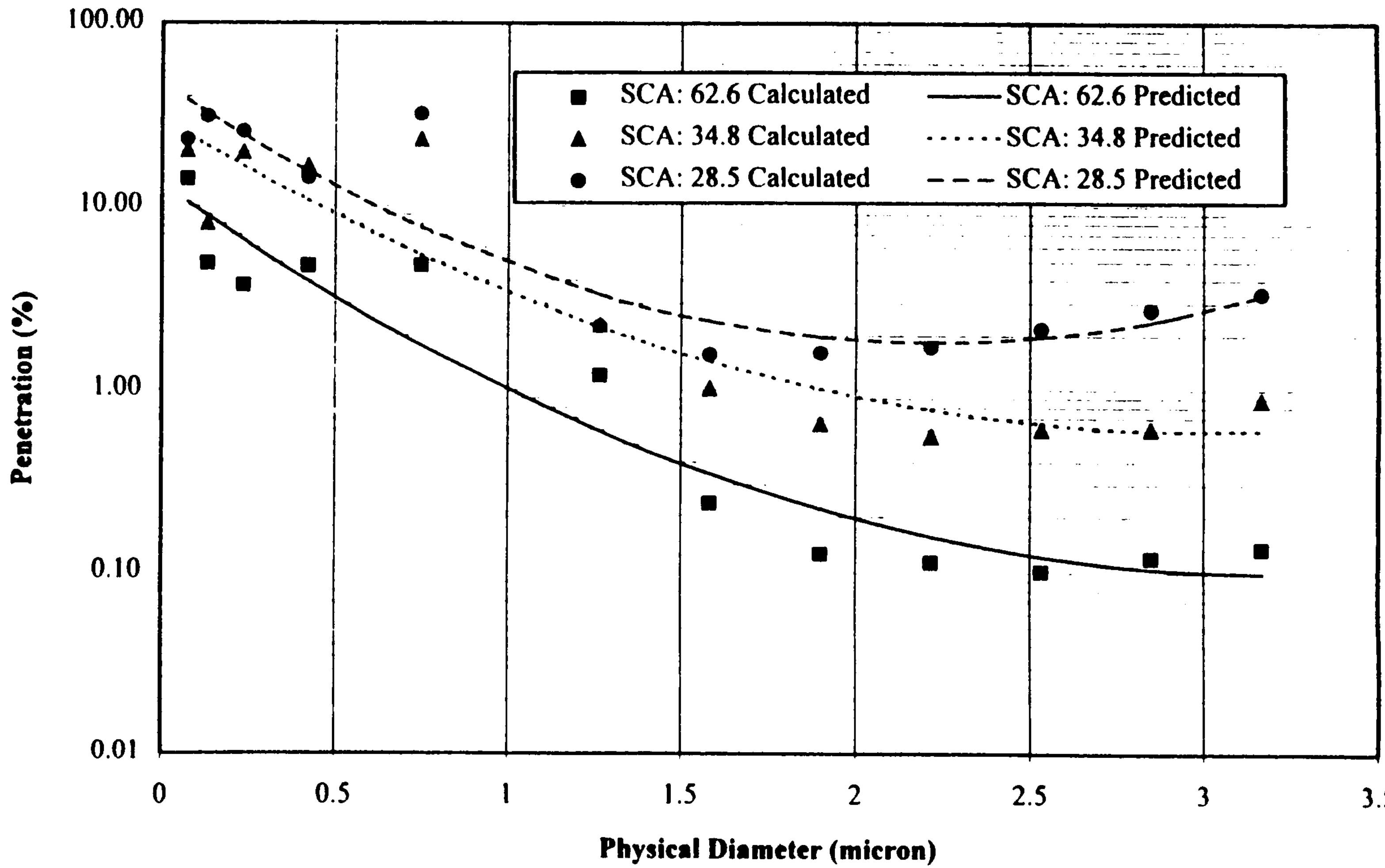

Figure 3-3. Predicted and Calculated Penetrations for IWS.
Source: Morse, Thomas C. "Characterization of Particle Collection Efficiency in an Ionizing Wet Scrubber." M.S. Thesis, University of Tennessee, Knoxville, TN (1996).

collection efficiency afforded by one IWS, the application provides for additional IWSs in series up to a total of four(4) IWSs. In addition to the device-specific control efficiency, the application algorithm calculates the overall system efficiency for the entire APC system selected by the user. The user should be aware that there are a number of parameters which are characteristic of the IWS which were held constant in the above studies such as wire-to-plate spacing, liquid-to-gas ratio, packing type, and plate configuration. Therefore, the equations are specific to the Ceilcote IWS system which was utilized in the UTK studies.

The above three equations are subject to the same limitations as the packed bed empirical equation discussed earlier, being valid only in the range $0.12 \leq d_a \leq 5$ μm (note that d_a is identical to the earlier nomenclature DAERO(I)). As in the earlier case, the user has to be cognizant of this limitation, especially for IWS efficiency predictions outside the valid particle size bounds. The application imposes an artificial efficiency prediction curb on particle sizes greater than $d_a = 5$ μm by reporting a constant fractional efficiency outside the valid d_a range which in effect translates to a more conservative IWS efficiency estimate, based on the general understanding of ESPs and inertial impactors.

4. VB APPLICATION INPUT/OUTPUT

This section provides a test design case and presents the respective screens a user would encounter during an actual application execution. The application startup and initial menus have already been discussed earlier in Chapter 2 and will not be reiterated here. After the user clicks Inlet Gas Inputs from the File menu structure, the user input screen for Inlet Gas Characteristics is loaded. The data input screen for inlet gas characteristics is shown in Figure 4-1. The screen has been designed to aid the user in following a sequence of data inputs shown as Steps 1 through 5. At every Step, the user selects appropriate inputs for gas temperature, molecular weight, pressure, flowrate, and moisture to complete the inlet gas characteristics.

Note that when the application is running, some of the text input boxes will already have data in them which the user should interpret as being default prompts provided by the designers of the application. Additionally, the application also contains a sprinkling of Read Me command buttons and labeled text on some of the user input screens. These command buttons and text usually provide additional information and/or technical guidance to aid the user navigate through the application. The user should feel the liberty to modify the default prompts as and when necessary by specific design needs.

For specifying gas temperature, the user has the option of either entering in Fahrenheit or Celsius units. Gas molecular weight can be specified either directly or through gas component volume fractions. Note that the sum of the gas component volume fractions should be 100%. Next, the user specifies the static pressure and

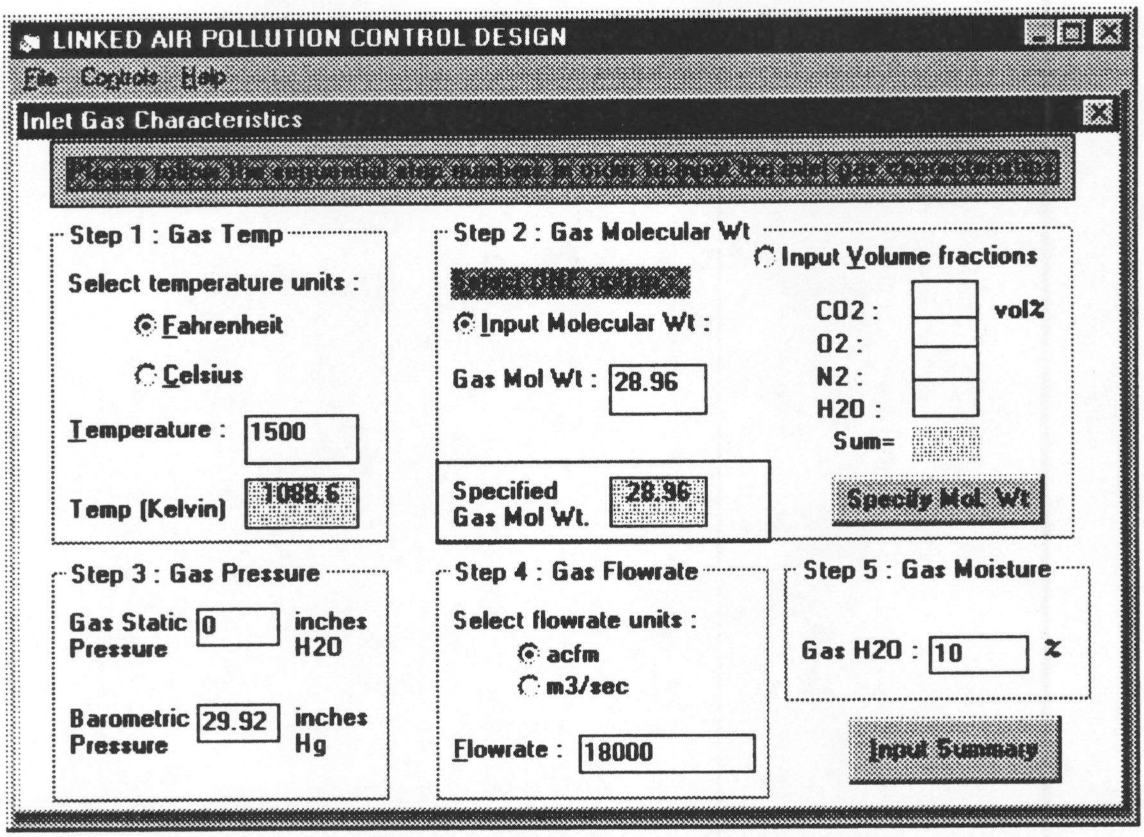

Figure 4-1. Data Input Screen for Inlet Gas Characteristics.

barometric pressure. Inlet gas flowrate units can either be specified in acfm (actual cubic feet per minute) or m^3/sec. In the last Step for specifying the gas moisture content, when the user selects the option for calculating gas molecular weight via. inputting gas component volume fractions, the application automatically enters the gas moisture content in the text input box. Lastly, the user clicks Input Summary for loading the Summary of Inlet Gas Characteristics screen shown in Figure 4-2. Note that if the user inadvertently clicks the Input Summary command button before completing the data inputs, a pop-up message box will warn the user that the data inputs have not been completed! This is a standard feature throughout the application whenever a user has inadvertently clicked a command button before completing a given procedure.

All the output summary screens for the application are empty when they are loaded. The user needs to click the Display command button to view the respective output summaries. Additionally, in the first output summary screen, the user needs to specify the drive location and filename for the comma-delimited text output file. Then if the user chooses to Print To File, the application writes the output data to a user-defined file which can be either printed directly to an external printer or further formatted with other software (e.g., Microsoft Excel) and then printed for a better looking output. Caution should be exercised here as not to define a filename that is already located in the drive of choice, as the resident file would be overwritten. Finally, the user clicks the Continue command button which unloads the summary output screen and reverts the user back to the MDI form or basic application palette (refer Figure 2-2).

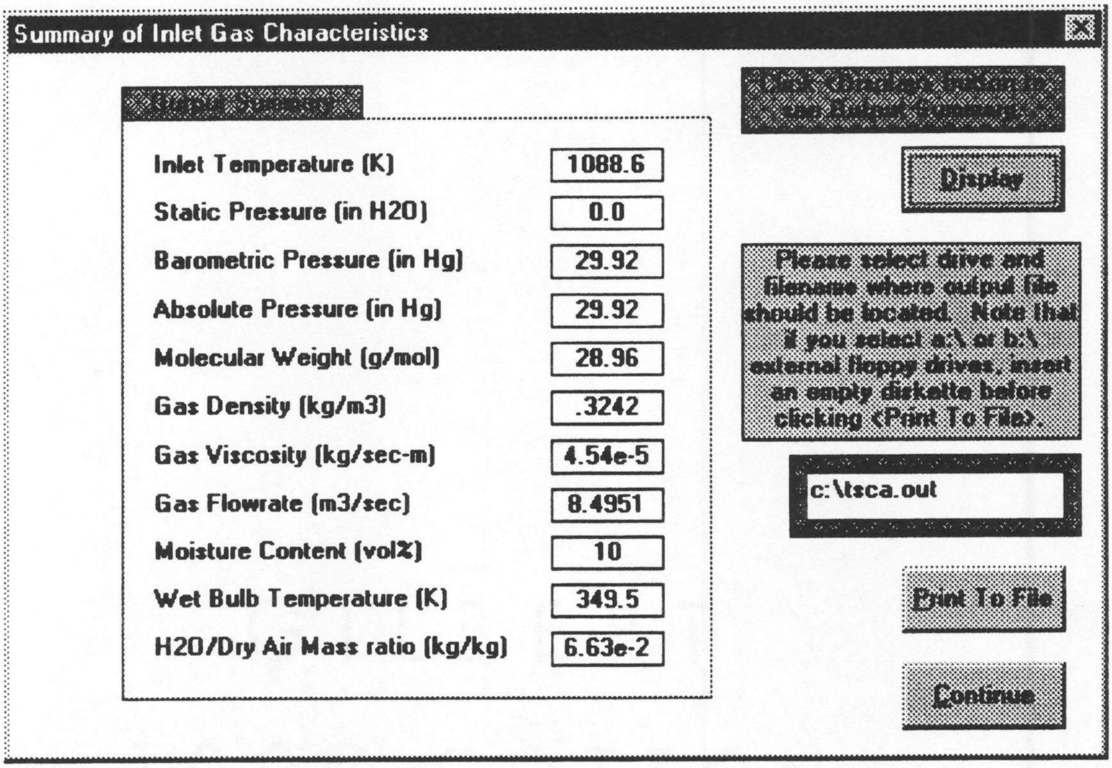

Figure 4-2. Data Output Screen for Inlet Gas Characteristics.

Now the user is ready to specify Inlet Particle Inputs from the File menu structure. Note that the menu selection Inlet Gas Inputs has been checked and grayed out (i.e., not available for loading for the current application session) since it has now been completed. The application has a logic built into it that prohibits a user from revisiting a prior menu selection once the user has specified the data inputs, reviewed the summary output screen and opted to continue to the next step in the application. Since the application is derived from a traditional textual procedural-based linked program (recall that the original code was written in BASIC and then modified in QBASIC), this logic was necessary to preserve calculation integrity without impairing the user-friendliness inherent in a GUI application. Also, the logic is useful for a novice user to develop a better understanding as to the logical steps in a design exercise without making improper choices.

The data input screen for inlet particle characteristics has been shown in Figure 4-3. The Read Me command button informs the user that the inlet particle size distribution may be specified either as a log-normal distribution or as actual mass fractions for a prescribed number of particle size increments. When the user clicks either of the above options, the application provides a pop-up message box which contains further technical guidance regarding data inputs. For the log-normal distribution, after the user finishes inputting the text input boxes, clicking the Compute Distribution command button shows the number of particle size increments for the chosen distribution. If the number of size increments (Z) is non-integer, the user should re-enter the distribution data. Similarly, in the other option, after the user finishes inputting the text input boxes, clicking the No.

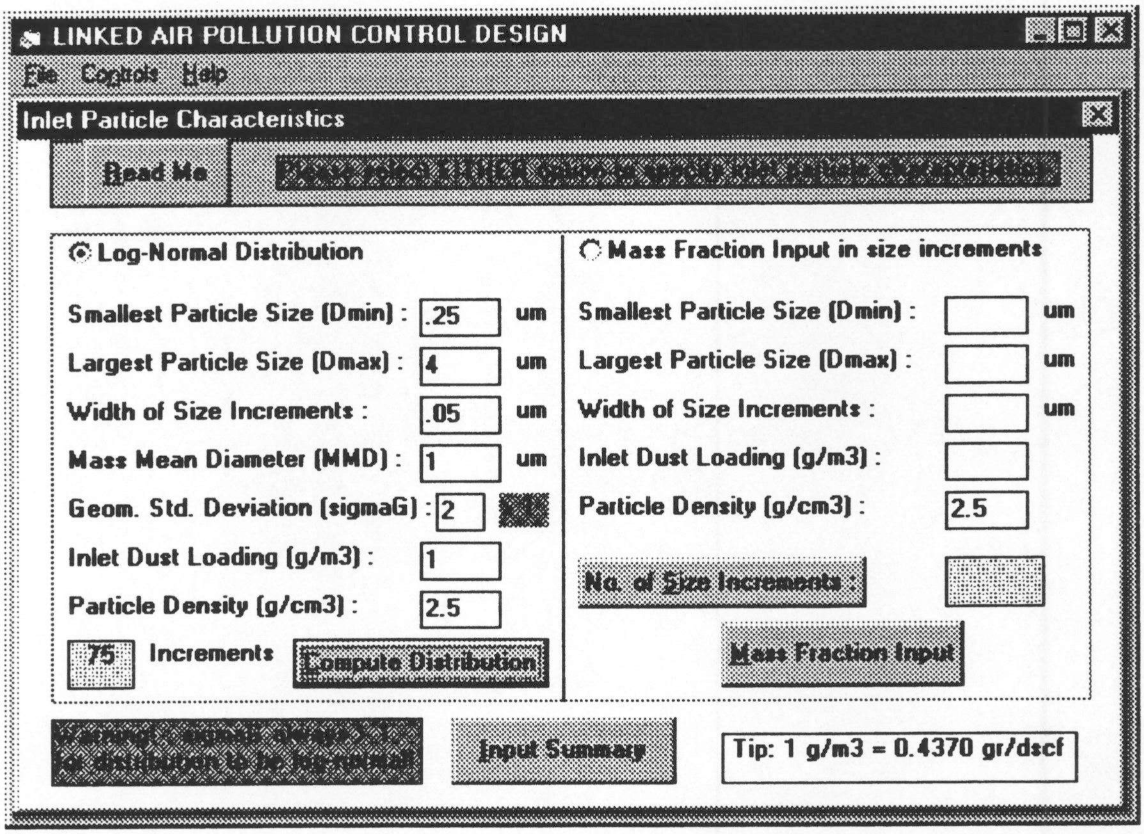

Figure 4-3. Data Input Screen for Inlet Particle Characteristics.

of Size Increments command button shows the value of Z. Again, if Z is non-integer, the user should re-enter the distribution data. Next, clicking the Mass Fraction Input command button invokes a pop-up input box Z times, once for every particle size increment. Finally, the user clicks the Input Summary for loading the Summary of Inlet Particle Characteristics screen shown in Figure 4-4. Again, as in the earlier summary output screen, the user clicks Display (to view the output summary), Print To File (optional) and Continue in succession. After the user is back on the MDI form, the application will gray out both menu choices under File: Inlet Gas Inputs and Inlet Particle Inputs. At this time, the only design choice available to the user will be under the Controls menu structure - Quench. The implication being that the user is ready to specify the quench tower characteristics.

The quench tower has a combined data input/output screen as shown in Figure 4-5. The text input boxes are located near the top of the form. When the form is loaded, two labels near the top of the form indicate the inlet gas saturation temperature (i.e., the wet bulb temperature calculated earlier) and the amount of water required for saturation in kg/sec. The technical guidance on the form indicates that the user should specify actual water injection rate greater than the amount of water required for saturation. For the quench tower, the user needs to provide only two inputs - actual water injection rate in kg/sec and the pressure drop across the quench tower in inches of H_2O. Next the user needs to click the Calculate command button to generate the output for the form. After the output has been displayed on the form, the user then has the option to Print To File and Continue on to the next device.

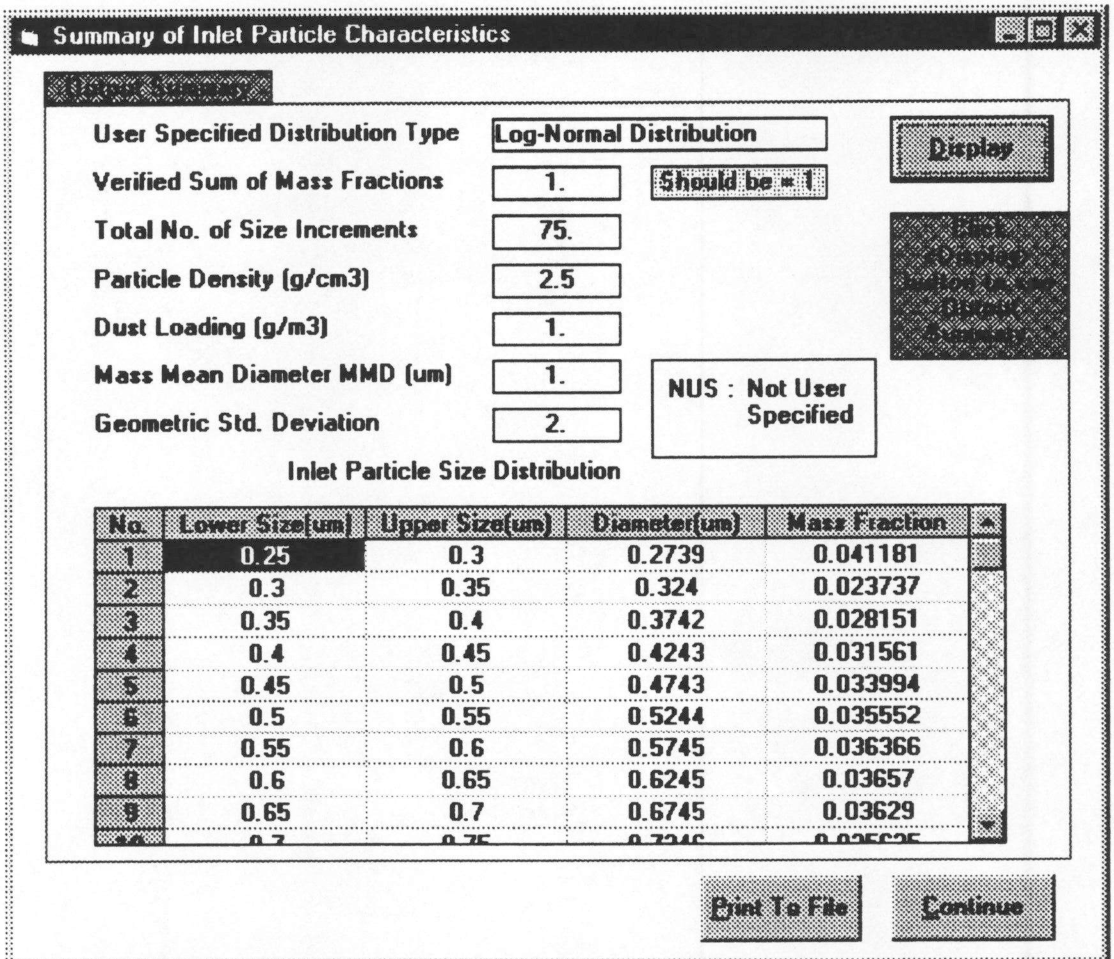

Figure 4-4. Data Output Screen for Inlet Particle Characteristics.

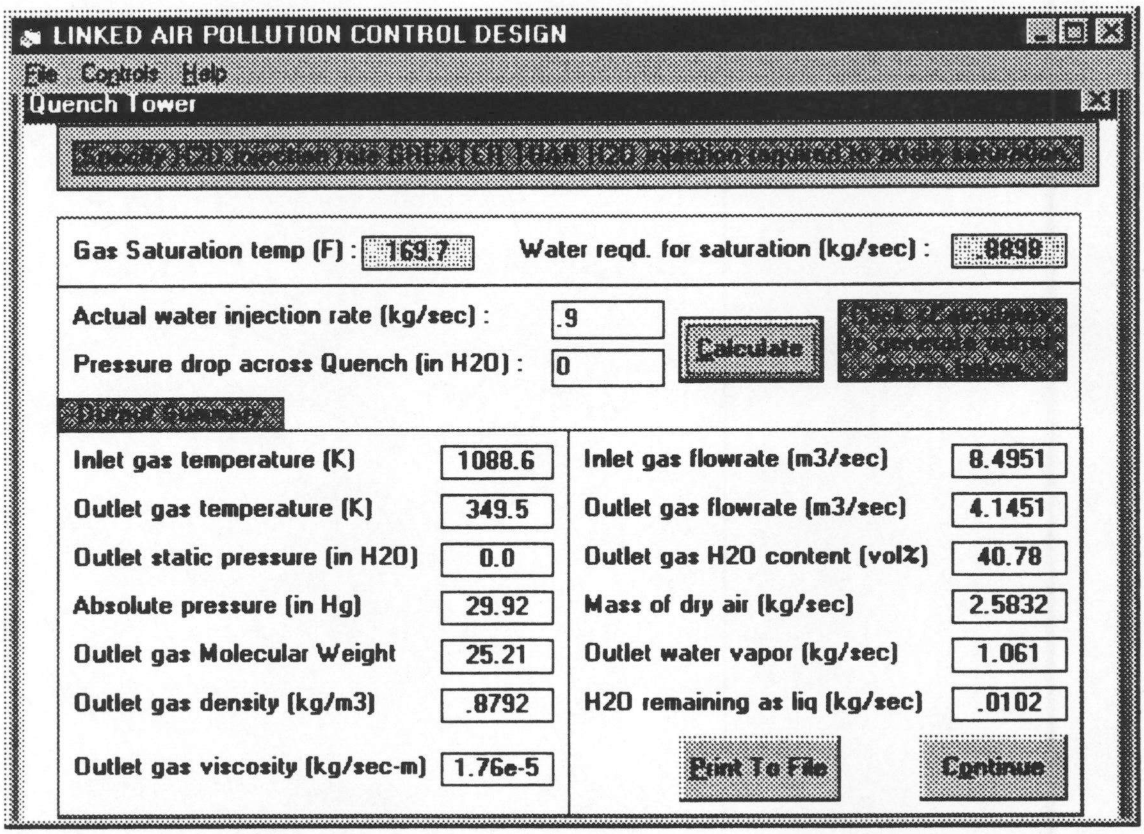

Figure 4-5. Data Input/Output Screen for Quench Tower Characteristics.

The next design choice available to the user under the Controls menu structure is the Venturi. The data input screen for the venturi scrubber has been shown in Figure 4-6. The technical guidance label on the top of the form informs the user that for venturi calculations, either pressure drop or throat velocity has to be specified. Additionally, it also provides the typical range for venturi liquid addition: 10-20 gal/1000 acfm of gas. The quench outlet gas flowrate is reflected on the top of the form in order to aid the user in the selection of the venturi liquid addition rate in gal/min. The form also has a command button labeled Venturi By-Pass, which provides default inputs as the required inputs in order to render the venturi calculations redundant and therefore, by-pass the venturi scrubber (i.e., enter both liquid addition rate and venturi pressure drop as zero, the throat velocity can be entered as any non-zero number and it will not have any consequence on the venturi scrubber calculations). After entering the appropriate venturi scrubber inputs, the user clicks the Output Summary command button to perform the venturi calculations. After the calculations are done, a message box informs the successful completion of the calculations. The Output Summary click event procedure also loads the output summary form shown in Figure 4-7. The output summary form has similar elements encountered in the earlier output summary forms, e.g., Display, Re-Visit (to repeat calculations starting with Quench), Print To File and Continue. The only notable difference in the output summary form for the venturi is the label near the bottom of the form which reflects the overall venturi efficiency. From the venturi scrubber onwards, the application starts calculating both individual device efficiency and overall system efficiency which are reported in the respective device output summary forms.

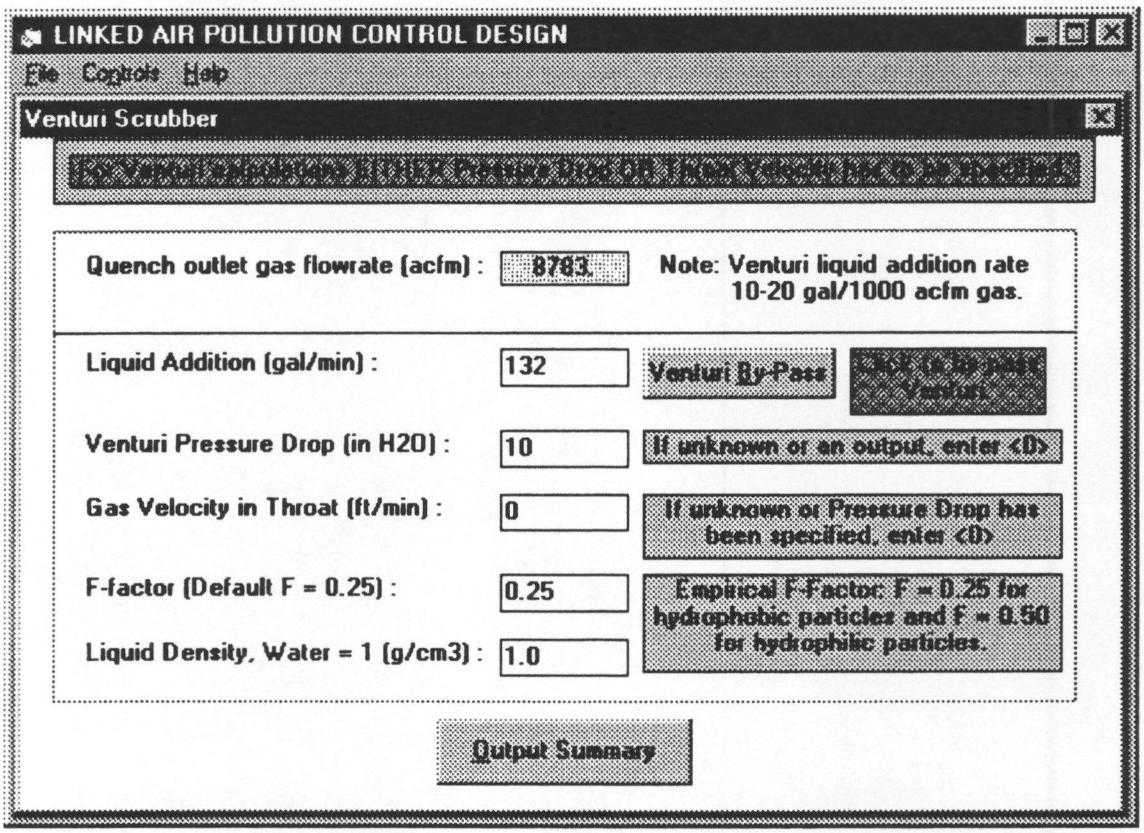

Figure 4-6. Data Input Screen for Venturi Scrubber Characteristics.

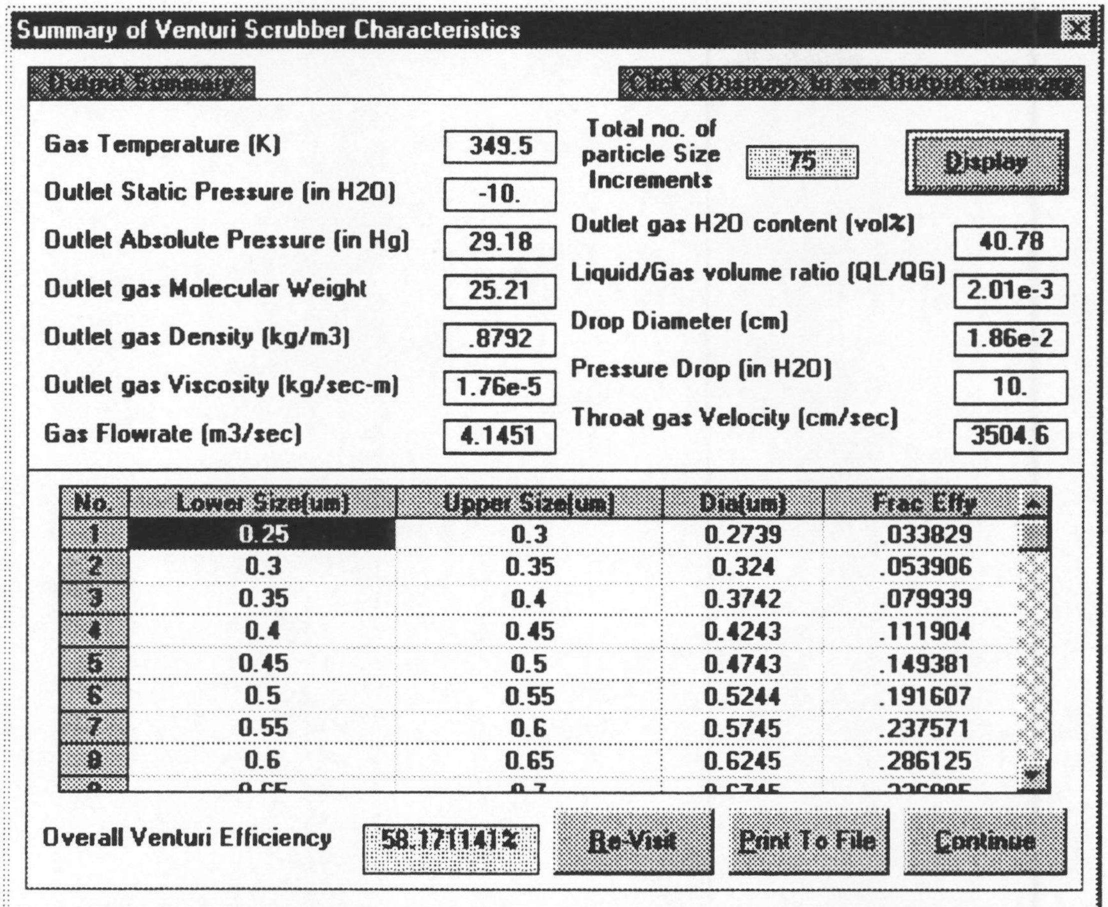

Figure 4-7. Data Output Screen for Venturi Scrubber Characteristics.

After the venturi, the next design choice available to the user under the Controls menu structure is the Packed Bed. The layout of the packed bed form is very similar to the quench in that it is a combined data input/output screen as shown in Figure 4-8. The Read Me command button on top of the form informs the user about the empirical nature of the packed bed equations and the limitations of the empirical equations, i.e., the valid particle size range of $0.12 \leq$ DAERO(I) ≤ 5 μm for which the packed bed equations are valid. The user should also be cognizant that for DAERO(I) ≥ 5 μm, the application sets a constant fractional efficiency outside the equation range which results in a more conservative packed bed efficiency estimate. The only user input for the packed bed is the depth of the wetted packing in feet following which the user clicks the Calculate command button which displays the output for the packed bed form. In order to by-pass the packed bed, the user can specify the depth of the wetted packing as zero. In addition to the size increment based fractional efficiencies, the application also reports the overall packed bed efficiency and the overall system efficiency. At this point in the application, the user has the option of specifying either ESP calculations or IWS calculations before proceeding to the next device. If the user specifies ESP calculations, then only the ESP design choice will be available under the Controls menu structure. The IWS-1 through IWS-4 design choices will be grayed out, i.e., not available. Similarly, if the user specifies IWS calculations, then the ESP design choice will be grayed out or vice versa.

Assuming the user specifies the ESP design choice under the Controls menu structure, then the data input form shown in Figure 4-9 will be loaded. The user inputs text boxes have technical guidance labels next to them in order to facilitate ease of data

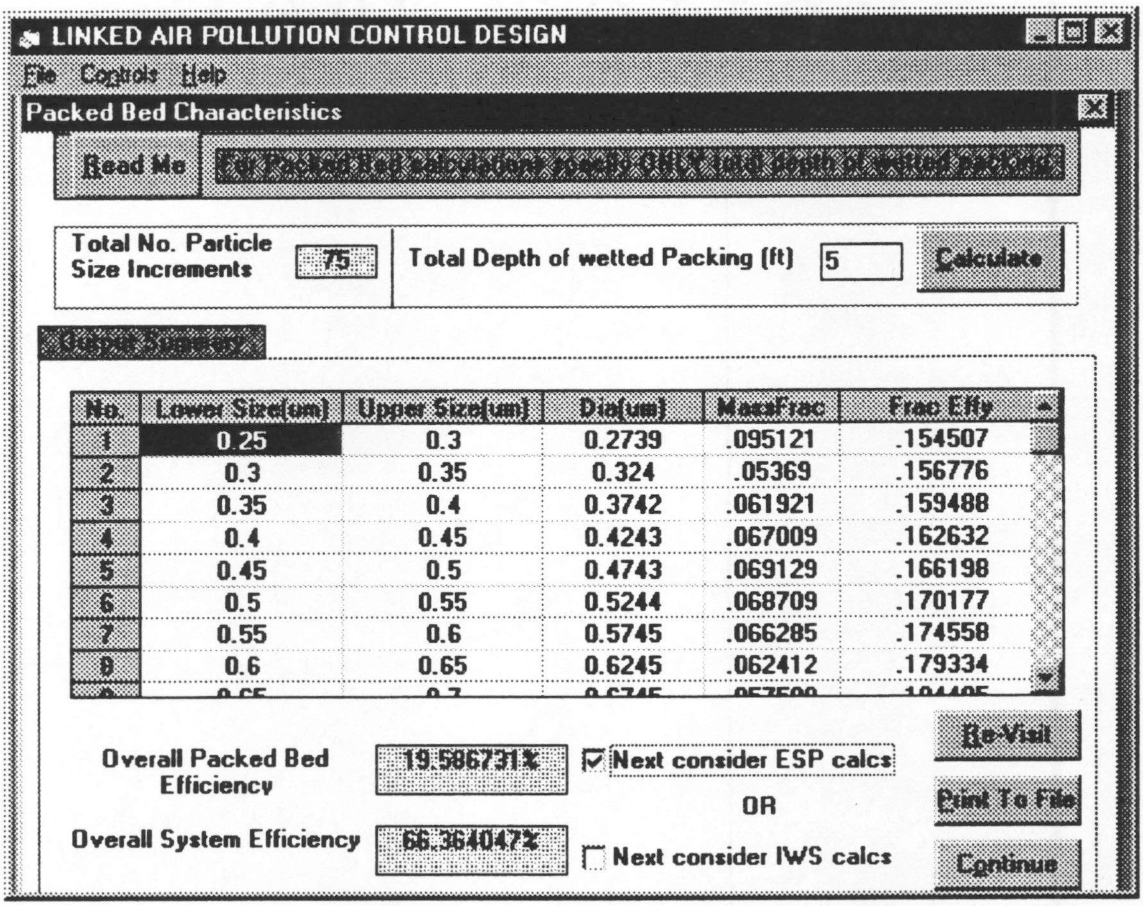

Figure 4-8. Data Input/Output Screen for Packed Bed Characteristics.

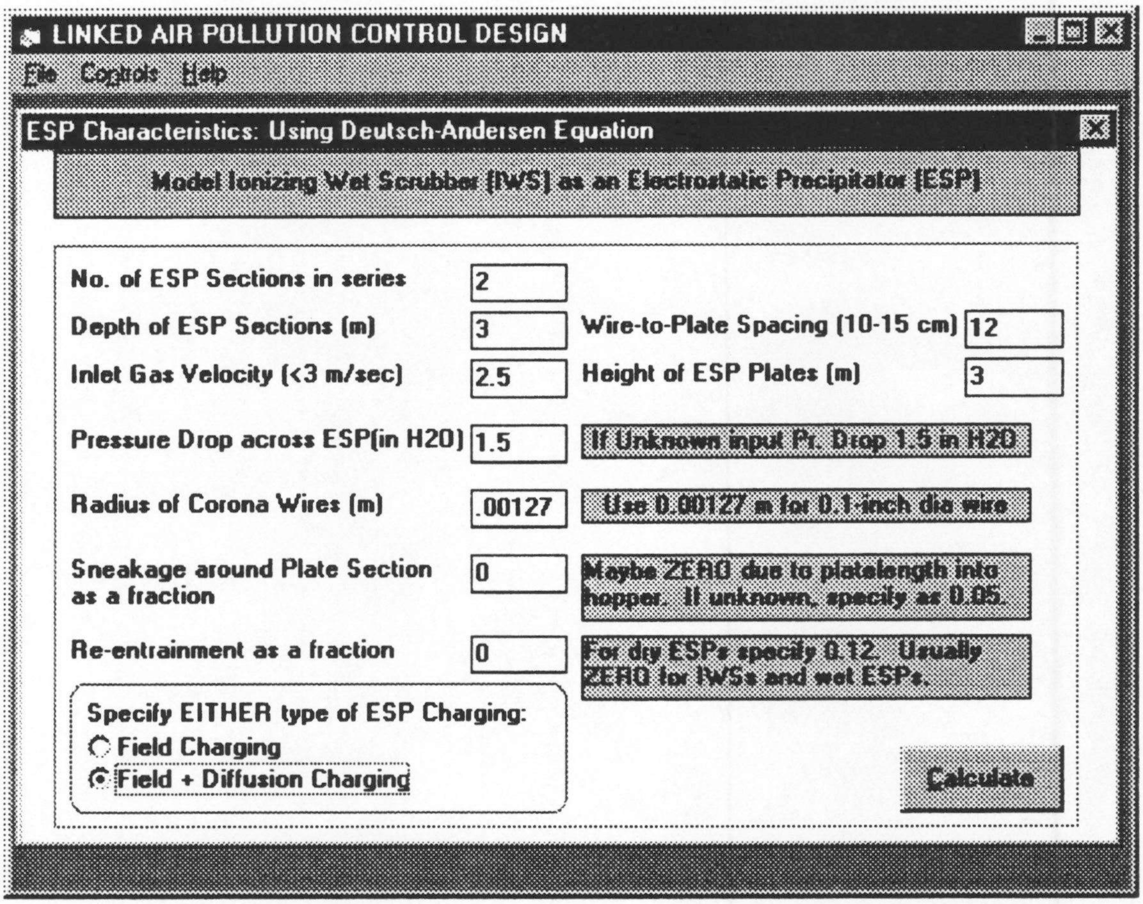

Figure 4-9. Data Input Screen for Electrostatic Precipitator (ESP) Characteristics.

input specification, e.g., the ESP pressure drop text input box has the following label "If Unknown input Pressure Drop 1.5 in H_2O. Lastly, the user specifies the type of particle charging mechanism, i.e., either Field Charging or Field + Diffusion Charging before clicking the Calculate command button which performs the design calculations. After the calculations have been successfully completed, the output summary form is loaded which is shown in Figure 4-10. The output summary form for the ESP calculations ends the application for the current session, i.e., the user needs to Exit Program (under File menu structure) after clicking the Continue command button on the output summary form prior to starting a new design session. When a user starts a new design session, it is recommended that the output filename be changed to reflect a new session, otherwise, the application will overwrite the existing output file if the same output filename is retained.

If the user specifies the IWS-1 design choice under the Controls menu structure, then the data input form shown in Figure 4-11 will be loaded. Under this design choice, the user has the option of adding up to four (4) IWSs in series. The user can also terminate a design session after any single IWS design. Since, all the IWS forms are identical, only IWS-1 will be discussed in this report. The IWS design form has a combined data input/output screen as shown in Figure 4-11. The Read Me command button on top of the form informs the user about the empirical nature of the IWS equations and the limitations of the empirical equations, i.e., the valid particle size range of $0.12 \leq$ DAERO(I) ≤ 5 μm for which the IWS equations are valid. The user should also be cognizant that for DAERO(I) ≥ 5 μm, the application sets a constant fractional efficiency outside the equation range which results in a more conservative IWS efficiency estimate.

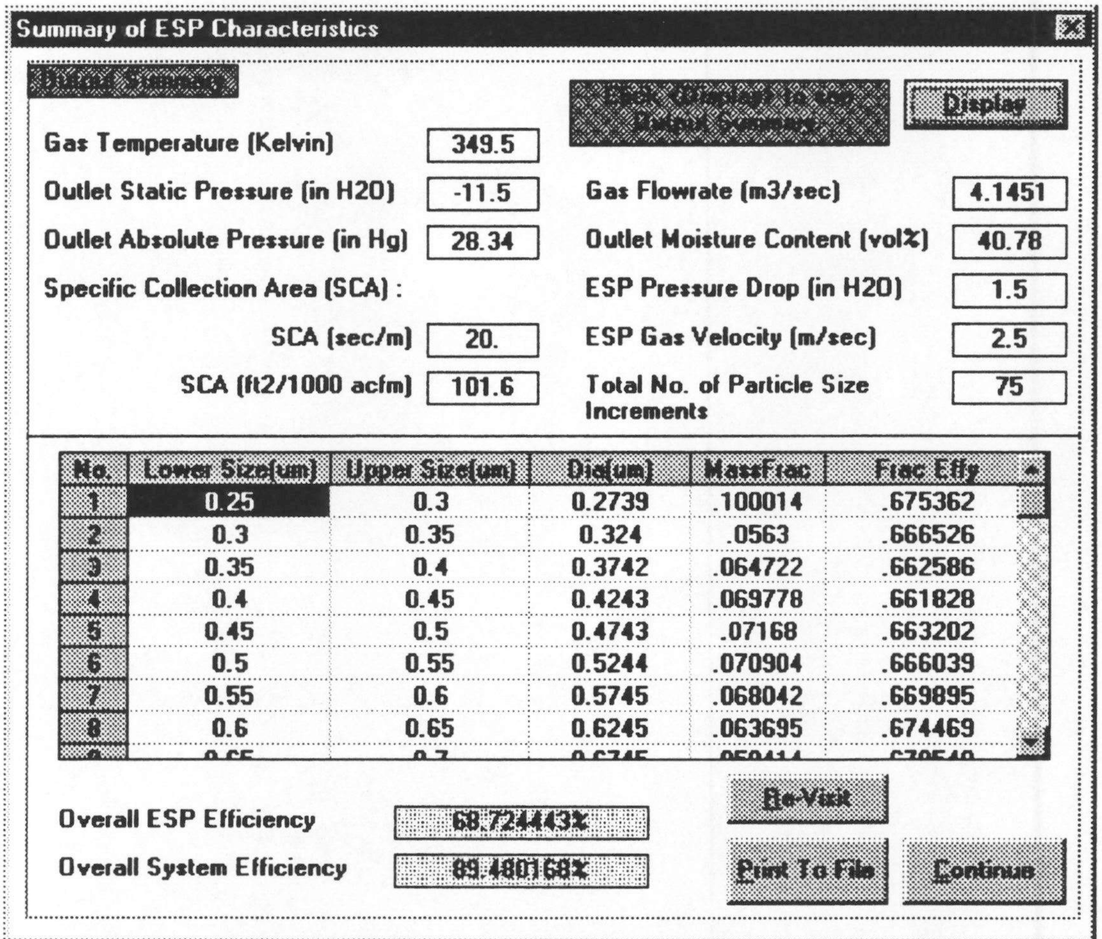

Figure 4-10. Data Output Screen for Electrostatic Precipitator (ESP) Characteristics.

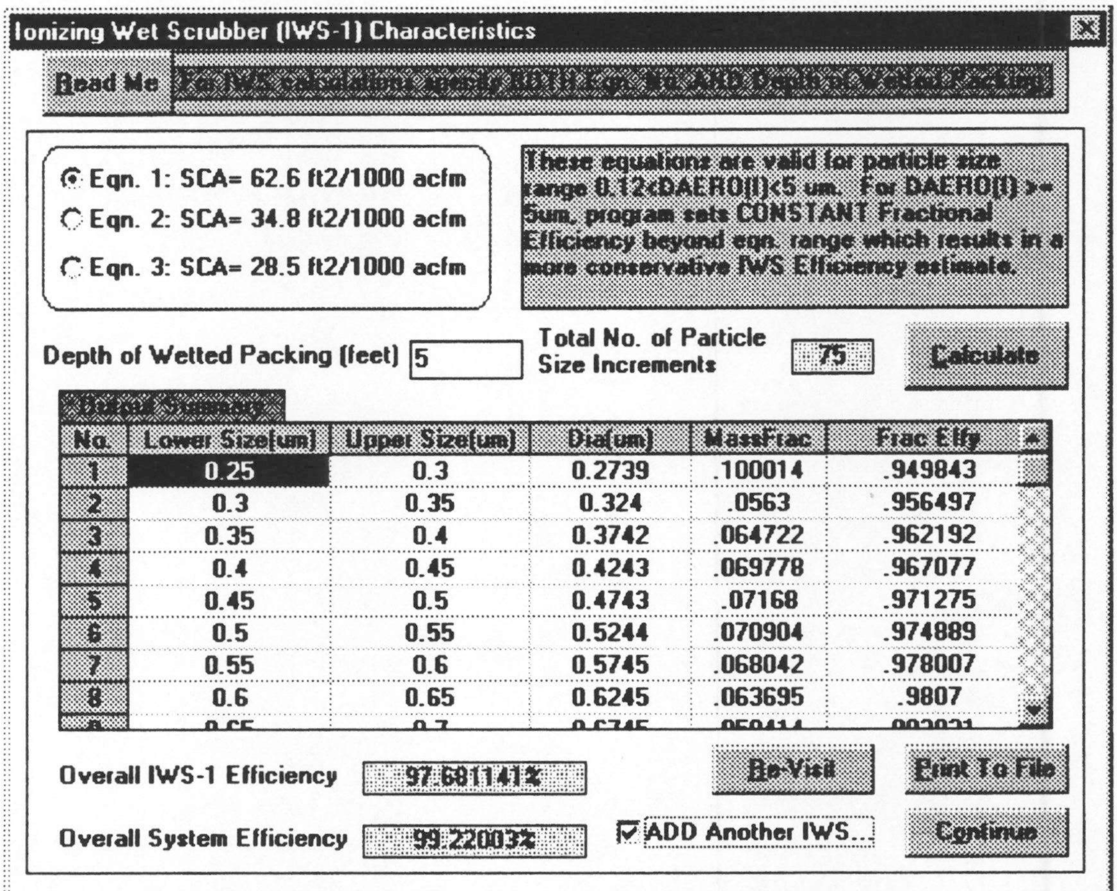

Figure 4-11. Data Input/Output Screen for Ionizing Wet Scrubber (IWS) Characteristics.

After specifying the type of IWS equation (i.e., select the SCA-specific empirical equation), the only user input for the IWS is the depth of the wetted packing in feet following which the user clicks the Calculate command button which displays the output for the IWS form. In addition to the size increment based fractional efficiencies, the application also reports the overall IWS efficiency and the overall system efficiency. As discussed earlier, upon completion of design of IWS-1, the user has the option of adding up to four (4) IWSs in series. After completing the application for the current session, the user needs to Exit Program (under File menu structure) after clicking the Continue command button on the IWS form prior to starting a new design session.

In this context, flexibility was incorporated into the program structure with the Re-Visit command button available on all control output screens starting with the venturi. It allows the user to repeat a set of calculations from Quench without reentering inlet gas and particle characteristics. In addition, the program was made more robust with logic that interrupts program execution if an error is detected and urges the user to re-enter input data at the screen where the program was halted. In addition, if the user inadvertently forgot to specify an output filename in the inlet gas characteristics output screen and wished to print the output from a subsequent output screen, the program will prompt the user for a new filename and execute the Print To File command.

5. CONCLUSIONS

The Visual Basic (VB) Air Pollution Control (APC) linked program simulating the TSCA facility system configuration can be used for numerous applications from actual design of ionizing wet scrubber (IWS) systems, design of individual control device components, and as an educational tool for teaching APC device design. This document contains a description of the relevant equations and methodology utilized in the development of this robust user-friendly VB application with an aesthetically appealing GUI (Graphical User Interface). The integrity of the algorithms were repeatedly checked against hand calculations for a test case simulating actual process conditions for the TSCA air pollution control system. Based on user-defined inlet gas and particle characteristics, the VB application utilizes a mix of classical efficiency predictor equations along with the various empirical equations developed at the University of Tennessee, Knoxville to predict both device-specific and overall system particle collection efficiencies for the TSCA facility APC system configuration. In addition, the application can be molded to define other system configurations (with by-passing certain control devices) containing any of the device elements included in the package to predict the particle collection efficiency for any number of inlet gas-particle scenarios.

BIBLIOGRAPHY

BIBLIOGRAPHY

Adams, Vince "Separation and Quantification of the Particle Removal Mechanisms in the Ionizing Wet Scrubber." Ph.D. Dissertation, University of Tennessee, Knoxville, TN (1997).

Calvert, S., Goldschmidt, J., Leith, David, Mehta, D. "Wet Scrubber System Study." *Scrubber Handbook*, vol.1, U.S. Department of Commerce, NTIS, PB-213016 (1972).

Cooper, David C., Alley, F.C. "Particulate Scrubbers: Theory and Design Considerations." *Air Pollution Control - A Design Approach*, 2nd ed., Waveland Press Inc. (1994).

Cooper, Hal B.H., Rossano, A.T. "Gas Flow Measurements." *Source Testing for Air Pollution Control*, 1st ed., McGraw-Hill Inc. (1971).

Crawford, Martin "Principles of Fluid Flow." *Air Pollution Control Theory*, 1st ed., MgGraw-Hill Inc. (1986).

Cunningham, M. Editor "*Visual Basic By Example*." 1st ed., Que Corporation (1994).

Keenan, J.H., Keyes, F.G., Hill, P.G., Moore, J.G. "Thermodynamic Properties of Water Including Vapor, Liquid, and Solid Phases." *Steam Tables*, 1st ed., John Wiley & Sons Inc. (1969).

Lide, David R. Editor "Viscosity of Gases." *CRC Handbook of Chemistry and Physics*, 73rd ed., CRC Press (1993).

Morse, Thomas C. "Characterization of Particle Collection Efficiency in an Ionizing Wet Scrubber." M.S. Thesis, University of Tennessee, Knoxville, TN (1996).

Nukiyama, S., Tanasawa, Y. "An Experiment on the Atomization of Liquid by Means of an Air Stream." *Transactions of the Society of Mechanical Engineers*, 4(14), Japan (1938).

Perry, Robert H., Green, Don W., Maloney, James O. Editors "Physical and Chemical Data: Thermodynamic Properties." *Perry's Chemical Engineers' Handbook*, 6th ed., McGraw-Hill Book Company (1984).

Potter, Bill, Taylor, M., Scott, B. "*Visual Basic: SuperBible*." 2nd ed., Waite Group Press (1993).

Turner, James H., Lawless, Phil A., Toshiaki, O., Coy, D.W., Greiner, G.P., McKenna, J.D., Vatavuk, W.A. "Electrostatic Precipitators." *Air Pollution Engineering Manual*, 1st ed., Air & Waste Management Association, Van Nostrand Reinhold (1992).

Wark, K., Warner, C.F. "Particulate: Terminal or Settling Velocity." *Air Pollution - Its Origin and Control*, 2nd ed., Harper Collins Publishers (1981).

APPENDICES

APPENDIX A

CODE FOR VISUAL BASIC APPLICATION

Code for Visual Basic Application

Front-End Screens

Source: Constant.txt

```
'
' Visual Basic global constant file. This file can be loaded into a code module.
' Some constants are commented out because they have duplicates (e.g., NONE appears
several places).
' If you are updating a Visual Basic application written with an older version, you should
replace your global constants with the constants in this file.
'
' General
'************************************************************
' Global Variable Declarations - KD
Global TEMPF, TEMPC, TEMPK, ACFM, FLOW, ROG, H2O, VIS, HM
Global MW, MWD, SPRESS, BPRESS, ABSPRESS, PWB, TWB, TWBF
Global I, Z, N, J, A, ZZ, B, XX
Global DMIN, DMAX, WI
Global DP(), DA(), SUMMASS, ROP(), GPERACM(), MMD()
Global SIGMAG(), MI()
Global MCSAT, MH2O, MH2OQ, WNETQ, MAIR, WIT
Global TEMPKQ, TEMPFQ, MWQ, SPRESSQ, ABSPRESSQ, ROGQ, H2OQ
Global DELPQ, VISQ, FLOWQ, ACFMQ
Global FLOWV, TEMPKV, H2OV, VISV, SPRESSV, ABSPRESSV
Global F, RL, QL, PR, VG, DELPV, DD
Global SIM1(), KC(), FK(), SI(), PT(), EFF(), PEFF(), TEFF()
Global MPV1(), MPV2(), MPV()
Global DPB, SIGVEFF, SIGMPV2
Global DAERO(), PBEFF(), TEFFPB(), PEFFPB(), MPPB()
Global SIGPBEFF, SYSPB
Global NS, DSECT, L, RW, SN, RR
Global TEMPKE, SPRESSE, ABSPRESSESP, SCA, ESCA, DELPE
Global VGESP, TEFFESP, SYSESP
Global MPESP1(), ESPEFF()
Global SYMPPB, SYMPIWS1, SYMPIWS2, SYMPIWS3
Global SYSIWS1, SYSIWS2, SYSIWS3, SYSIWS4
Global TEFFIWS1(), TEFFIWS2(), TEFFIWS3(), TEFFIWS4()
Global MPPIWS(), MPIWS1(), MPIWS2(), MPIWS3(), MPIWS4()
Global IWSEFF1(), IWSEFF2(), IWSEFF3(), IWSEFF4()
Global SIGIWS1EFF, SIGIWS2EFF, SIGIWS3EFF, SIGIWS4EFF
Global DIWS1, DIWS2, DIWS3, DIWS4
Global Drive() As String

*************************************************************************
```

```
Source: Form1.txt
Option Explicit

Sub cmdContinue_Click ()
   Unload frmFront
   frmMain.Show MODELESS
End Sub
```
****

```
Source: MDIForm1.txt
Option Explicit

Sub MDIForm_Load ()
   mnuParticle.Enabled = False
   mnuQuench.Enabled = False
   mnuVenturi.Enabled = False
   mnuPackbed.Enabled = False
   mnuESP.Enabled = False
   mnuIWS1.Enabled = False
   mnuIWS2.Enabled = False
   mnuIWS3.Enabled = False
   mnuIWS4.Enabled = False
   '****************************
   'mnuPrint.Enabled = False
   '****************************

End Sub

Sub mnuAboutProg_Click ()
   frmAboutprog.Show MODELESS
End Sub

Sub mnuContents_Click ()
   frmContents.Show MODELESS
End Sub

Sub mnuESP_Click ()
   frmESP.Show MODELESS
End Sub

Sub mnuExit_Click ()
   Dim Keypress
   Keypress = MsgBox("This will end your Program session.", 64 + 1, "Exit Program")
   Select Case Keypress
```

```
    Case IDOK: End
   End Select
End Sub

Sub mnuGas_Click ()
   frmGas1.Show MODELESS
End Sub

Sub mnuIWS1_Click ()
   frmIWS1.Show MODELESS
End Sub

Sub mnuIWS2_Click ()
   frmIWS2.Show MODELESS
End Sub

Sub mnuIWS3_Click ()
   frmIWS3.Show MODELESS
End Sub

Sub mnuIWS4_Click ()
   frmIWS4.Show MODELESS
End Sub

Sub mnuPackBed_Click ()
   frmPackBed.Show MODELESS
End Sub

Sub mnuParticle_Click ()
   frmPart1.Show MODELESS
End Sub

Sub mnuPrint_Click ()
   Dim Keypress
   Keypress = MsgBox("Program will create output file in specified [drive:\filename.out] and append outputs to output file during the current program session when user responds to Print To File command button(s) on different output screens.  The contents of the output file may be printed at a later date for a set of hardcopy program results. File format is comma-delimited text file which may be imported to a spreadsheet program for a better formatted look.", 48, "Print To File")
   Select Case Keypress
   Case IDOK:
   End Select
```

```
End Sub

Sub mnuQuench_Click ()
   frmQuench.Show MODELESS
End Sub

Sub mnuTechSupport_Click ()
   MsgBox "Contact: Dr. Wayne T. Davis at 73A Perkins Hall, Dept. of Civil & Environmental Engineering, The University of Tennessee, Knoxville TN 37996. Phone: (423)974-7728, Fax: (423)974-2669, E-Mail: wtdavis@utk.edu.", 64, "Technical Support"
End Sub

Sub mnuVenturi_Click ()
   frmVenturi.Show MODELESS
End Sub
```

****

Source: Form3.txt

```
Option Explicit

Sub cmdOK_Click ()
   Unload FrmAboutprog
End Sub

Sub Command1_Click ()
   Unload FrmAboutprog
End Sub
```

****

Source: Formconte.txt

```
Sub cmdOK_Click ()
   Unload frmContents
End Sub
```

****

Inlet Gas Characteristics

Source: Form4.txt

```
Option Explicit

Sub cmdCalmw_Click ()
   On Error GoTo cmdCalmw_Exit
   'Calculation of MW from user vol fraction inputs
   Dim CARBON, OXYGEN, NITROGEN, WATER, SUM
   'Mol wt calcs
   CARBON = Val(txtCarbon.Text)
   OXYGEN = Val(txtOxygen.Text)
```

```
   NITROGEN = Val(txtNitrogen.Text)
   WATER = Val(txtWatervap.Text)
   SUM = (CARBON + OXYGEN + NITROGEN + WATER)
   If (SUM <> 100) Then
      MsgBox "If Gas Molecular Weight is being specified through Input Volume
Fractions: Sum of gas volume fractions not equal to 100%.  Please re-enter percentages.",
16, "Warning!"
   Else
   lblVoltot.Caption = Format$(SUM, "###.")
   MW = (CARBON * 44 + OXYGEN * 32 + NITROGEN * 28 + WATER * 18) / 100
   lblMwcalc.Caption = Format$(MW, "##.##")
   txtMolwt.Text = ""
   txtGasrh.Text = txtWatervap.Text
   End If
   Exit Sub
   '*****************************************
cmdCalmw_Exit:
   MsgBox "An error occurred, please re-specify your inputs!", 16, "Warning!"
   txtCarbon.Text = ""
   txtOxygen.Text = ""
   txtNitrogen.Text = ""
   txtWatervap.Text = ""
   lblVoltot.Caption = ""
   lblMwcalc.Caption = ""
   txtGasrh.Text = ""
   txtMolwt.Text = ""
   Exit Sub

End Sub

Sub cmdGas_Click ()
   On Error GoTo cmdGas_Exit
   'Inform user that Gas inputs not provided
   If (lbltemp.Caption = "") Then
     MsgBox "Please input inlet gas characteristics completely before proceeding to Input
Summary !", 16, "Warning!"
   Else
   If (lblMwcalc.Caption = "") Then
     MsgBox "Please input inlet gas characteristics completely before proceeding to Input
Summary !", 16, "Warning!"
   Else
   If (txtGasflow.Text = "") Then
     MsgBox "Please input inlet gas characteristics completely before proceeding to Input
Summary !", 16, "Warning!"
```

```
   Else
   If (txtGasrh.Text = "") Then
     MsgBox "Please input inlet gas characteristics completely before proceeding to Input
Summary !", 16, "Warning!"
   Else
   '**********************************
   'Inlet gas viscosity VIS [kg/sec-m]
   VIS = (Exp(((Log(TEMPK) - 18.0075) / 1.431))) * .1
   'Dry Molecular wt MWD
   MWD = (MW - 18 * (H2O / 100)) / (1 - (H2O / 100))
   Dim HV, PW, CK

   'HV [lbmol Water/lbmol Dry Air]
   HV = H2O / (100 - H2O)
   'HM [lb Water/lb Dry Air]
   HM = HV * 18 / MWD
   'PW Vapor pr of water in gas
   PW = ABSPRESS * H2O / 100
   'Wetbulb temp TWB[K] by iterative solution of Carrier eqn
   'Wetbulb temp TWBF[F]
   'Sat vapor pr PWB [Pa] converted to [in Hg]
   For TWB = 273 To 473 Step .5
     PWB = Exp((-5800.22 / TWB) + 1.39149 - .048602 * TWB + (.417647 * 10 ^ (-4))
* (TWB ^ 2) - (.1445 * 10 ^ (-7)) * (TWB ^ 3) + 6.5459 * Log(TWB))
     PWB = PWB * 2.95 * 10 ^ (-4)
     TWBF = (9 / 5) * (TWB - 273) + 32
     CK = PWB - ((ABSPRESS - PWB) * (TEMPF - TWBF) / (2830 - 1.3 * TWBF))
     If (CK > PW) Then GoTo 10
   Next
10
   'Brings up Summary of Inlet gas characteristics
   frmGas2.Show MODELESS
   End If
   End If
   End If
   End If
   Exit Sub
   '****************************************
cmdGas_Exit:
   MsgBox "An error occurred, please re-specify your inputs!", 16, "Warning!"
   txtGastemp.Text = ""
   lbltemp.Caption = ""

   txtCarbon.Text = ""
```

```
    txtOxygen.Text = ""
    txtNitrogen.Text = ""
    txtWatervap.Text = ""
    lblVoltot.Caption = ""
    lblMwcalc.Caption = ""
    txtGasrh.Text = ""
    txtMolwt.Text = ""

    txtGasspr.Text = ""
    txtGasbpr.Text = ""

    txtGasflow.Text = ""
    Exit Sub

End Sub

Sub cmdGasflow_Click ()
    'Inlet gas flowrate calcs FLOW [m3/sec]
    If (optAcfm.Value = True) Then
      ACFM = Val(txtGasflow.Text)
      FLOW = (ACFM * .028316846) / 60
     Else
      If (optMetricflow.Value = True) Then
       FLOW = Val(txtGasflow.Text)
      End If
    End If
    If (txtGastemp.Text = "") Then lbltemp.Caption = "0"
    'Inlet gas Density calcs ROG [kg/m3]
    ROG = (MW / 22.4) * (273 / TEMPK) * (ABSPRESS / 29.92)

End Sub

Sub cmdGaspr_Click ()
    'Inlet gas pressure calcs - static pr SPRESS [in H2O]
    'barometric pr BPRESS and abs. pr ABSPRESS [in Hg]
    SPRESS = Val(txtGasspr.Text)
    BPRESS = Val(txtGasbpr.Text)
    ABSPRESS = 29.92 + SPRESS / 13.6
    optAcfm.Value = True
End Sub

Sub cmdGasrh_Click ()
    'Inform user that Gas inputs not provided
    If (lbltemp.Caption = "") Then
```

```
    MsgBox "Please input inlet gas characteristics completely before proceeding to Input
Summary !", 16, "Warning!"
   Else
   If (lblMwcalc.Caption = "") Then
    MsgBox "Please input inlet gas characteristics completely before proceeding to Input
Summary !", 16, "Warning!"
   Else
   If (txtGasflow.Text = "") Then
    MsgBox "Please input inlet gas characteristics completely before proceeding to Input
Summary !", 16, "Warning!"
   Else
   If (txtGasrh.Text = "") Then
    MsgBox "Please input inlet gas characteristics completely before proceeding to Input
Summary !", 16, "Warning!"
   Else
   'Gas moisture content H2O
   H2O = Val(txtGasrh.Text)
    MsgBox "Inlet gas characteristics calculations done successfully.  Please proceed to
Input Summary !", 48, "Attention"
   End If
   End If
   End If
   End If

End Sub

Sub cmdGastemp_Click ()
   'Inlet gas temperature calcs
   If (optFahrenheit(0).Value = True) Then
    TEMPF = Val(txtGastemp.Text)
    TEMPK = (((TEMPF - 32) / 1.8) + 273)
    lbltemp.Caption = Format$(TEMPK, "####.#")
    Else
    If (optCelsius(0).Value = True) Then
     TEMPC = Val(txtGastemp.Text)
     TEMPK = TEMPC + 273.2
     lbltemp.Caption = Format$(TEMPK, "####.#")
    End If
   End If
   If (txtGastemp.Text = "") Then lbltemp.Caption = "0"

End Sub

Sub cmdMolwt_Click ()
```

```
   'Inlet gas molecular weight calcs
   If (optMolwt.Value = True) Then
    MW = Val(txtMolwt.Text)
    lblMwcalc.Caption = Format$(MW, "##.##")
    'User needs to specify gas RH
    txtGasrh.Text = ""
   Else
    If (optCalmw.Value = True) Then
     'MW = Val(lblMw.Caption)
     'lblMwcalc.Caption = Format$(MW, "##.##")
     'Gas RH specified from input Water vol fraction
     txtGasrh.Text = txtWatervap.Text
   End If
   End If

   txtGasspr.TabIndex = 15
End Sub

Sub Form_Load ()
   'Inlet gas temperature calcs
   'If (optFahrenheit(0).Value = True) Then
     TEMPF = Val(txtGastemp.Text)
     TEMPK = (((TEMPF - 32) / 1.8) + 273)
     lbltemp.Caption = Format$(TEMPK, "####.#")
    'Else
     'If (optCelsius(0).Value = True) Then
      'TEMPC = Val(txtGastemp.Text)
      'TEMPK = TEMPC + 273.2
      'lbltemp.Caption = Format$(TEMPK, "####.#")
     'End If
   'End If
   'If (txtGastemp.Text = "") Then lbltemp.Caption = "0"
   '*****************************************************
   'Inlet gas pressure calcs - static pr SPRESS [in H2O]
   'barometric pr BPRESS and abs. pr ABSPRESS [in Hg]
   SPRESS = Val(txtGasspr.Text)
   BPRESS = Val(txtGasbpr.Text)
   ABSPRESS = 29.92 + SPRESS / 13.6
   optAcfm.Value = True
   '*****************************************************
   'MW = 28.96
   MW = Val(txtMolwt.Text)
   '*****************************************************
   'Inlet gas flowrate calcs FLOW [m3/sec]
```

```
   If (optAcfm.Value = True) Then
    ACFM = Val(txtGasflow.Text)
    FLOW = (ACFM * .028316846) / 60
   Else
    If (optMetricflow.Value = True) Then
     FLOW = Val(txtGasflow.Text)
    End If
   End If
   If (txtGastemp.Text = "") Then lbltemp.Caption = "0"
   'Inlet gas Density calcs ROG [kg/m3]
   ROG = (MW / 22.4) * (273 / TEMPK) * (ABSPRESS / 29.92)
   '*****************************************************
   H2O = Val(txtGasrh.Text)
   '*****************************************************

End Sub

Sub optCalmw_Click ()
   MsgBox "The sum of gas volume fractions should be 100%.", 48, "Attention!"
   txtMolwt.Text = ""
   lblMwcalc.Caption = ""
   txtGasrh.Text = ""

End Sub

Sub optCelsius_Click (Index As Integer)
   txtGastemp.Text = ""
   lbltemp.Caption = ""
End Sub

Sub optFahrenheit_Click (Index As Integer)
   txtGastemp.Text = ""
   lbltemp.Caption = ""
End Sub

Sub Option1_Click ()
   MsgBox "Sum of gas volume fractions should add to 100% !", 48, "Attention!"
End Sub

Sub optMolwt_Click ()
   'MsgBox "Default Molecular Weight for Air is 28.96", 64, "Information!"
   txtMolwt.Text = 28.96
   lblMwcalc.Caption = "28.96"
   txtCarbon.Text = ""
```

```
    txtOxygen.Text = ""
    txtNitrogen.Text = ""
    txtWatervap.Text = ""
    lblVoltot.Caption = ""

End Sub

Sub txtGasbpr_Change ()
    'Inlet gas pressure calcs - static pr SPRESS [in H2O]
    'barometric pr BPRESS and abs. pr ABSPRESS [in Hg]
    SPRESS = Val(txtGasspr.Text)
    BPRESS = Val(txtGasbpr.Text)
    ABSPRESS = 29.92 + SPRESS / 13.6
    optAcfm.Value = True

End Sub

Sub txtGasflow_Change ()
    'Inlet gas flowrate calcs FLOW [m3/sec]
    If (optAcfm.Value = True) Then
      ACFM = Val(txtGasflow.Text)
      FLOW = (ACFM * .028316846) / 60
     Else
      If (optMetricflow.Value = True) Then
       FLOW = Val(txtGasflow.Text)
      End If
    End If
    If (txtGastemp.Text = "") Then lbltemp.Caption = "0"
    'Inlet gas Density calcs ROG [kg/m3]
    ROG = (MW / 22.4) * (273 / TEMPK) * (ABSPRESS / 29.92)

End Sub

Sub txtGasrh_Change ()
    H2O = Val(txtGasrh.Text)
End Sub

Sub txtGasspr_Change ()
    'Inlet gas pressure calcs - static pr SPRESS [in H2O]
    'barometric pr BPRESS and abs. pr ABSPRESS [in Hg]
    SPRESS = Val(txtGasspr.Text)
    BPRESS = Val(txtGasbpr.Text)
    ABSPRESS = 29.92 + SPRESS / 13.6
    optAcfm.Value = True
```

```
End Sub

Sub txtGastemp_Change ()
   'Inlet gas temperature calcs
   If (optFahrenheit(0).Value = True) Then
     TEMPF = Val(txtGastemp.Text)
     TEMPK = (((TEMPF - 32) / 1.8) + 273)
     lbltemp.Caption = Format$(TEMPK, "####.#")
    Else
     If (optCelsius(0).Value = True) Then
      TEMPC = Val(txtGastemp.Text)
      TEMPK = TEMPC + 273.2
      lbltemp.Caption = Format$(TEMPK, "####.#")
     End If
   End If
   If (txtGastemp.Text = "") Then lbltemp.Caption = "0"

End Sub

Sub txtMolwt_Change ()
   'Inlet gas molecular weight calcs
   If (optMolwt.Value = True) Then
    MW = Val(txtMolwt.Text)
    lblMwcalc.Caption = Format$(MW, "##.##")
    'User needs to specify gas RH
    txtGasrh.Text = ""
   End If

   txtGasspr.TabIndex = 15


End Sub
***************************************************************************
Source: Frmgas2.txt
Option Explicit

Sub cmdContinue_Click ()
   Dim Keypress
   Keypress = MsgBox("Inlet gas characteristics have been specified.  Please proceed to select Inlet Particle Inputs from the Menu.", 48 + 1, "End Inlet Gas Inputs")
   Select Case Keypress
   Case IDOK:
   Unload frmGas1
```

```
   Unload frmGas2
   frmMain.mnuGas.Checked = True
   frmMain.mnuGas.Enabled = False
   frmMain.mnuParticle.Enabled = True
   End Select
End Sub

Sub cmdDisplay_Click ()
   lblTemp.Caption = Format$(TEMPK, "####.#")
   If (frmGas1.txtGasspr.Text = "" Or frmGas1.txtGasspr.Text = "0") Then
      lblSpr.Caption = Format$(SPRESS, "0.0")
   Else
      lblSpr.Caption = Format$(SPRESS, "###.##")
   End If
   lblBpr.Caption = Format$(BPRESS, "##.##")
   lblAbspr.Caption = Format$(ABSPRESS, "##.##")
   lblMolwt.Caption = Format$(MW, "##.##")
   lblDensity.Caption = Format$(ROG, "##.####")
   lblViscosity.Caption = Format$(VIS, "#.##e+##")
   'lblFlowrate.Caption = Format$(FLOW, "##.####")
   If (frmGas1.txtGasflow.Text = "" Or frmGas1.txtGasflow.Text = "0") Then
      lblFlowrate.Caption = Format$(FLOW, "0.0")
   Else
      lblFlowrate.Caption = Format$(FLOW, "##.####")
   End If

   'lblWater.Caption = Format$(H2O, "###")
   If (frmGas1.txtGasrh.Text = "" Or frmGas1.txtGasrh.Text = "0") Then
      lblWater.Caption = Format$(H2O, "0.0")
   Else
      lblWater.Caption = Format$(H2O, "###")
   End If


   lblWetbulb.Caption = Format$(TWB, "####.#")
   lblMassratio.Caption = Format$(HM, "#.##e+##")
End Sub

Sub cmdPrint_Click ()
   On Error GoTo cmdPrint_Exit
   Dim Keypress, RUNTIME
   Keypress = MsgBox("Print Inlet gas characteristics to user-specified output file.", 48 +
1, "Print To File")
   Select Case Keypress
```

```
    Case IDOK:
    '***********************************
    RUNTIME = Now
    'Open "a:\tsca.out" For Output As #1
    Drive = txtDrive.Text
    Open Drive For Output As #1
    Write #1,
    Write #1, "Visual Basic Program Output: Inlet Gas Characteristics"
    Write #1, "[File format is comma-delimited text file which may be imported"
    Write #1, "into a spreadsheet program for a better formatted look]"
    Write #1, "*******************************************"
    Write #1, "Current Session Filename :", Drive
    Write #1, "Date and Time of Run :", RUNTIME
    Write #1,
    Write #1, "Inlet Temperature (K):", Format$(TEMPK, "####.#")
    Write #1, "Static Pressure (in H2O):", Format$(SPRESS, "###.##")
    Write #1, "Barometric Pressure (in Hg):", Format$(BPRESS, "##.##")
    Write #1, "Absolute Pressure (in Hg):", Format$(ABSPRESS, "##.##")
    Write #1, "Molecular Weight:", Format$(MW, "##.##")
    Write #1, "Gas Density (kg/m3):", Format$(ROG, "##.####")
    Write #1, "Gas Viscosity (kg/sec-m):", Format$(VIS, "#.##e+##")
    Write #1, "Gas Flowrate (m3/sec):", Format$(FLOW, "##.####")
    Write #1, "Moisture Content (vol%):", Format$(H2O, "###")
    Write #1, "Wet Bulb Temperature (K):", Format$(TWB, "####.#")
    Write #1, "Water/Dry Air Mass Ratio:", Format$(HM, "#.##e+##")
    Write #1,
    Write #1, "*******************************************"
    Write #1,
    Close #1
    '************************************
    End Select
    Exit Sub
    '************************************
cmdPrint_Exit:
    MsgBox "An error occurred! Program will print to default output file [c:\tsca.out]", 16,
"Warning!"
    txtDrive.Text = ""
    txtDrive.Text = "c:\tsca.out"
    Exit Sub


End Sub
**************************************************************************
```

Inlet Particle Characteristics

Source: Frmpart1.txt

```
Option Explicit
Option Base 1

Sub cmdCompute_Click ()
  On Error GoTo cmdCompute_Exit
  If (optLog.Value = True) Then
    'Inform user that Particle inputs not provided
    If (txtDmin.Text = "") Then
    MsgBox "Please input inlet particle characteristics completely before proceeding to
Compute Distribution !", 16, "Warning!"
    Else
    If (txtDmax.Text = "") Then
    MsgBox "Please input inlet particle characteristics completely before proceeding to
Compute Distribution !", 16, "Warning!"
    Else
    If (txtIncwid.Text = "") Then
    MsgBox "Please input inlet particle characteristics completely before proceeding to
Compute Distribution !", 16, "Warning!"
    Else
    If (txtMMD.Text = "") Then
    MsgBox "Please input inlet particle characteristics completely before proceeding to
Compute Distribution !", 16, "Warning!"
    Else
    If (txtSigmaG.Text = "") Then
    MsgBox "Please input inlet particle characteristics completely before proceeding to
Compute Distribution !", 16, "Warning!"
    Else
    If (txtDust.Text = "") Then
    MsgBox "Please input inlet particle characteristics completely before proceeding to
Compute Distribution !", 16, "Warning!"
    Else
    If (txtRop.Text = "") Then
    MsgBox "Please input inlet particle characteristics completely before proceeding to
Compute Distribution !", 16, "Warning!"
    Else
    '***************************************************
    If (Val(txtSigmaG.Text) <= 1) Then
    MsgBox "SigmaG cannot be less than or equal to ONE for log-normal distribution.
Please re-enter value for sigmaG.", 16, "Warning!"
    Else
    '***************************************************
    If (Val(txtDmin.Text) >= Val(txtDmax.Text)) Then
```

```
   MsgBox "Please ensure that Dmin < Dmax!", 16, "Warning! "
   Else
   '****************************************************
'Global DMIN, DMAX, WI
   DMIN = Val(txtDmin.Text)
   DMAX = Val(txtDmax.Text)

   WI = Val(txtIncwid.Text)
   'Z: No. of particle size increments
   Z = (DMAX - DMIN) / WI
   lblSize.Caption = Z

ReDim MI(Z), DPSI(Z), CCF(Z), DP(Z), DA(Z + 1)
   'Calculate particle size increments
   DA(1) = DMIN
   For I = 1 To Z + 1
      DA(I) = DMIN + (I - 1) * WI
   Next

   'DP(1) = DMIN + (.5 * WI)
   'For I = 2 To Z
      'DP(I) = DP(1) + (I - 1) * WI
   'Next
   'Geometric mean value diameter
   For I = 1 To Z
      DP(I) = Sqr(DA(I) * DA(I + 1))
   Next

   'Program considers only single distribution per run
   'N: # of particle size distributions
   N = 1
ReDim MMD(N), SIGMAG(N), GPERACM(N), ROP(N), ASX(Z + 1)
ReDim ERFXDP(Z + 1), MLESSDP(Z + 1), X(Z + 1)
   For J = 1 To N
      MMD(J) = Val(txtMMD.Text)
      GPERACM(J) = Val(txtDust.Text)
      SIGMAG(J) = Val(txtSigmaG.Text)
      ROP(J) = Val(txtRop.Text)

      For I = 1 To (Z + 1)
         X(I) = (Log(DA(I) / (MMD(J)))) / (1.414 * Log(SIGMAG(J)))
         ASX(I) = X(I)
         If ASX(I) < 0 Then ASX(I) = -1 * ASX(I)
         A = 1: ZZ = 0:
```

```
            For B = 0 To 25
               If B = 0 GoTo E1
               A = A * B
E1:
               XX = ((2) / (3.1417 ^ .5)) * ((-1) ^ B) * ((ASX(I)) ^ (2 * B + 1)) / (A * (2 *
B + 1))
               ZZ = ZZ + XX
            Next
            ERFXDP(I) = ZZ
            If X(I) < 0 Then ERFXDP(I) = -1 * ERFXDP(I)
            MLESSDP(I) = .5 + .5 * ERFXDP(I)
         Next
      Next

      For I = 2 To (Z - 1)
         MI(I) = MLESSDP(I + 1) - MLESSDP(I)
      Next
      MI(1) = MLESSDP(2)
      MI(Z) = 1 - MLESSDP(Z)
CHECKSUM:
      SUMMASS = 0
      For I = 1 To Z
         SUMMASS = SUMMASS + MI(I)
      Next
     MsgBox "Inlet particle characteristics calculations done successfully.  Please proceed
to Input Summary !", 48, "Attention"
   '***************************************************
   End If
   End If
   End If
   End If
   End If
   End If
   End If
   End If
   End If
   End If
   '***************************************************
   If (optMassfrac.Value = True) Then
   'No event procedure
   End If
   Exit Sub
   '***************************************************
cmdCompute_Exit:
```

```
   MsgBox "An error occurred, please re-specify your inputs!", 16, "Warning!"
   txtDmin.Text = ""
   txtDmax.Text = ""
   txtIncwid.Text = ""
   txtMMD.Text = ""
   txtSigmaG.Text = ""
   txtDust.Text = ""
   txtRop.Text = ""
   lblSize.Caption = ""
   Exit Sub

End Sub

Sub cmdMassfrac_Click ()
   On Error GoTo cmdMassfrac_Exit
   If (optLog.Value = True) Then
   'No event procedure
   End If

   If (optMassfrac.Value = True) Then
      'Inform user that Particle inputs not provided
      If (txtDmin1.Text = "") Then
      MsgBox "Please input inlet particle characteristics completely before proceeding
with Mass Fraction Input !", 16, "Warning!"
      Else
      If (txtDmax1.Text = "") Then
      MsgBox "Please input inlet particle characteristics completely before proceeding
with Mass Fraction Input !", 16, "Warning!"
      Else
      If (txtIncwid1.Text = "") Then
      MsgBox "Please input inlet particle characteristics completely before proceeding
with Mass Fraction Input !", 16, "Warning!"
      Else
      If (txtDust1.Text = "") Then
      MsgBox "Please input inlet particle characteristics completely before proceeding
with Mass Fraction Input !", 16, "Warning!"
      Else
      If (txtRop1.Text = "") Then
      MsgBox "Please input inlet particle characteristics completely before proceeding
with Mass Fraction Input !", 16, "Warning!"
      Else
      If (lblSizeinc.Caption = "") Then
      MsgBox "Please input inlet particle characteristics completely before proceeding
with Mass Fraction Input !", 16, "Warning!"
```

```
        Else
        '****************************************************
    'Global DMIN, DMAX, WI

        DMIN = Val(txtDmin1.Text)
        DMAX = Val(txtDmax1.Text)
        WI = Val(txtIncwid1.Text)
        'Z: No. of particle size increments
        Z = (DMAX - DMIN) / WI

    ReDim MI(Z), DPSI(Z), CCF(Z), DP(Z), DA(Z + 1)
        'Calculate particle size increments
        DP(1) = DMIN + (.5 * WI)
        For I = 2 To Z
            DP(I) = DP(1) + (I - 1) * WI
        Next
        DA(1) = DMIN
        For I = 1 To Z + 1
            DA(I) = DMIN + (I - 1) * WI
        Next

    'Program considers only single distribution per run
    'N: # of particle size distributions
    N = 1
    ReDim GPERACM(N), ROP(N), MMD(N), SIGMAG(N)
    Dim UserInput
    'MMD and SIGMAG are inlcuded for array compatibility only
    For J = 1 To N
        GPERACM(J) = Val(txtDust1.Text)
        ROP(J) = Val(txtRop1.Text)
        MMD(J) = 0
        SIGMAG(J) = 0
MASSINPUT:
        For I = 1 To Z
        'Input MI(I): Mass fraction in increment, I

        UserInput = InputBox("Enter mass fraction for particle size increment and then hit
<ENTER>:", "Mass Fraction Input")
        MI(I) = UserInput

        Next
    Next
CHECKSUM1:
        ReDim SUMMFR(Z) As Single, SUMMFR1(Z) As Single    'Sum of mass fractions
```

```
    For I = 1 To 1
       SUMMFR(I) = MI(I)
    Next

    SUMMFR1(1) = SUMMFR(1)
    For I = 2 To Z
       SUMMFR1(I) = SUMMFR1(I - 1) + MI(I)
    Next
 If (SUMMFR1(Z) <> 1#) Then
    MsgBox "Sum of mass fractions NOT EQUAL TO ONE.  Please re-enter particle
mass fractions!", 16, "Warning!"
    GoTo MASSINPUT
 Else
    MsgBox "Inlet particle characteristics calculations done successfully.  Please proceed
to Input Summary !", 48, "Attention"

 End If
 End If
 End If
 End If
 End If
 End If
 End If
 End If

 Exit Sub
 '*****************************************************
cmdMassfrac_Exit:
 MsgBox "An error occurred, please re-specify your inputs!", 16, "Warning!"
 txtDmin1.Text = ""
 txtDmax1.Text = ""
 txtIncwid1.Text = ""
 txtDust1.Text = ""
 lblSizeinc.Caption = ""
 Exit Sub

End Sub

Sub cmdPartsum_Click ()
 If (optLog.Value = True) Then
    'Inform user that Particle inputs not provided
    If (txtDmin.Text = "") Then
```

```
        MsgBox "Please input inlet particle characteristics completely before proceeding to
the Input Summary !", 16, "Warning!"
        Else
        If (txtDmax.Text = "") Then
        MsgBox "Please input inlet particle characteristics completely before proceeding to
the Input Summary !", 16, "Warning!"
        Else
        If (txtIncwid.Text = "") Then
        MsgBox "Please input inlet particle characteristics completely before proceeding to
the Input Summary !", 16, "Warning!"
        Else
        If (txtMMD.Text = "") Then
        MsgBox "Please input inlet particle characteristics completely before proceeding to
the Input Summary !", 16, "Warning!"
        Else
        If (txtSigmaG.Text = "") Then
        MsgBox "Please input inlet particle characteristics completely before proceeding to
the Input Summary !", 16, "Warning!"
        Else
        If (txtDust.Text = "") Then
        MsgBox "Please input inlet particle characteristics completely before proceeding to
the Input Summary !", 16, "Warning!"
        Else
        If (txtRop.Text = "") Then
        MsgBox "Please input inlet particle characteristics completely before proceeding to
the Input Summary !", 16, "Warning!"
        Else
        If (lblSize.Caption = "") Then
        MsgBox "Please input inlet particle characteristics completely before proceeding to
the Input Summary !", 16, "Warning!"
        Else
        'Brings up Summary of Inlet particle characteristics
        frmPart2.Show MODELESS
        End If
        End If
        End If
        End If
        End If
        End If
        End If
        End If
    End If

    If (optMassfrac.Value = True) Then
```

```
'Inform user that Particle inputs not provided
If (txtDmin1.Text = "") Then
MsgBox "Please input inlet particle characteristics completely before proceeding to the Input Summary !", 16, "Warning!"
Else
If (txtDmax1.Text = "") Then
MsgBox "Please input inlet particle characteristics completely before proceeding to the Input Summary !", 16, "Warning!"
Else
If (txtIncwid1.Text = "") Then
MsgBox "Please input inlet particle characteristics completely before proceeding to the Input Summary !", 16, "Warning!"
Else
If (txtDust1.Text = "") Then
MsgBox "Please input inlet particle characteristics completely before proceeding to the Input Summary !", 16, "Warning!"
Else
If (txtRop1.Text = "") Then
MsgBox "Please input inlet particle characteristics completely before proceeding to the Input Summary !", 16, "Warning!"
Else
If (lblSizeinc.Caption = "") Then
MsgBox "Please input inlet particle characteristics completely before proceeding to the Input Summary !", 16, "Warning!"
Else
'If (lblDone.Caption = "") Then
'MsgBox "Please input inlet particle characteristics completely before proceeding to the Input Summary !", 16, "Warning!"
'Else
'Brings up Summary of Inlet particle characteristics
frmPart2.Show MODELESS
End If
End If
End If
End If
End If
End If
End If

End Sub

Sub cmdReadme_Click ()
```

```
MsgBox "The inlet particle size distribution may be specified EITHER as a Log-normal
distribution OR as actual mass fractions for a prescribed number of particle size
increments.", 64, "Inlet Particle Inputs"
End Sub

Sub cmdSizeint_Click ()
   On Error GoTo cmdSizeint_Exit
   If (optLog.Value = True) Then
   'No event procedure
   End If

   If (optMassfrac.Value = True) Then
     'Inform user that Particle inputs not provided
     If (txtDmin1.Text = "") Then
     MsgBox "Please input inlet particle characteristics completely before proceeding to
Mass Fraction Input !", 16, "Warning!"
     Else
     If (txtDmax1.Text = "") Then
     MsgBox "Please input inlet particle characteristics completely before proceeding to
Mass Fraction Input !", 16, "Warning!"
     Else
     If (txtIncwid1.Text = "") Then
     MsgBox "Please input inlet particle characteristics completely before proceeding to
Mass Fraction Input !", 16, "Warning!"
     Else
     If (txtDust1.Text = "") Then
     MsgBox "Please input inlet particle characteristics completely before proceeding to
Mass Fraction Input !", 16, "Warning!"
     Else
     If (txtRop1.Text = "") Then
     MsgBox "Please input inlet particle characteristics completely before proceeding to
Mass Fraction Input !", 16, "Warning!"
     Else
     '*****************************************************
     If (Val(txtDmin1.Text) >= Val(txtDmax1.Text)) Then
     MsgBox "Please ensure that Dmin < Dmax!", 16, "Warning! "
     Else
     '*****************************************************

    'Global DMIN, DMAX, WI

      DMIN = Val(txtDmin1.Text)
      DMAX = Val(txtDmax1.Text)
      WI = Val(txtIncwid1.Text)
```

```
    'Z: No. of particle size increments
    Z = (DMAX - DMIN) / WI
    lblSizeinc.Caption = Z

  End If
  End If
  End If
  End If
  End If
  End If
  End If
  Exit Sub
  '************************************
cmdSizeint_Exit:
  MsgBox "An error occurred, please re-specify your inputs!", 16, "Warning!"
  txtDmin1.Text = ""
  txtDmax1.Text = ""
  txtIncwid1.Text = ""
  'txtMMD.Text = ""
  'txtSigmaG.Text = ""
  txtDust1.Text = ""
  txtRop1.Text = ""
  lblSizeinc.Caption = ""
  Exit Sub

End Sub

Sub optLog_Click ()
  MsgBox "Program places any fraction LESS THAN Smallest size in the first size
increment and any fraction GREATER THAN Largest size in the last size increment. It is
recommended to specify minimum particle dia Dmin = MMD/(2*sigmaG) and maximum
particle dia Dmax = MMD*(2*sigmaG). Also ensure that the number of size increments is
an integer value, otherwise the particle size distribution will be in error.", 48, "Attention !"
  txtDmin1.Text = ""
  txtDmax1.Text = ""
  txtIncwid1.Text = ""
  txtDust1.Text = ""
  lblSizeinc.Caption = ""
  'lblDone.Caption = ""
End Sub

Sub optMassfrac_Click ()
  MsgBox "Please specify the fractional mass in each size increment. Be sure that the
sum of mass fractions equals ONE ! List any mass fraction fraction LESS THAN Smallest
```

size in the first size increment and any fraction GREATER THAN Largest size in the last size increment. Also ensure that the number of size increments is an integer value, otherwise the particle size distribution will be in error.", 48, "Attention !"

```
   txtDmin.Text = ""
   txtDmax.Text = ""
   txtIncwid.Text = ""
   txtMMD.Text = ""
   txtDust.Text = ""
   lblSize.Caption = ""
End Sub

Sub txtDmax1_Change ()
   lblSizeinc.Caption = ""
End Sub

Sub txtDmin1_Change ()
   lblSizeinc.Caption = ""

End Sub

Sub txtDust1_Change ()
   lblSizeinc.Caption = ""
End Sub

Sub txtIncwid1_Change ()
   lblSizeinc.Caption = ""
End Sub

Sub txtRop1_Change ()
   lblSizeinc.Caption = ""

End Sub
```

****

Source: Frmpart2.txt

```
Option Explicit
'Option Base 1

Sub Centercells ()
   'Sets the justification of the grid's cells to center
   Dim Cols As Integer
   For Cols = 0 To 4
      grdParticle.Col = Cols
      grdParticle.FixedAlignment(Cols) = 2
   Next Cols
```

```
   For Cols = 1 To 4
      grdParticle.ColAlignment(Cols) = 2
   Next Cols
End Sub

Sub cmdContinue_Click ()
   Dim Keypress
   Keypress = MsgBox("Inlet particle characteristics have been specified.  Please proceed
to select Quench (under Controls) from the Menu.", 48 + 1, "End Inlet Particle Inputs")
   Select Case Keypress
   Case IDOK:
   Unload frmPart1
   Unload frmPart2
   frmMain.mnuParticle.Checked = True
   frmMain.mnuParticle.Enabled = False
   frmMain.mnuQuench.Enabled = True
   End Select
End Sub

Sub cmdDisplay_Click ()
   If (frmPart1.optLog.Value = True) Then lblUserspec.Caption =
frmPart1.optLog.Caption
   If (frmPart1.optMassfrac.Value = True) Then lblUserspec.Caption = "Mass Fraction
Input by Size"

   lblSummass.Caption = Format$(SUMMASS, "##.##")
   lblIncrements.Caption = Format$(Z, "####.#")
   lblRop.Caption = Format$(ROP(1), "##.##")
   lblDust.Caption = Format$(GPERACM(1), "##.####")

   If (frmPart1.optLog.Value = True) Then
      lblMMD.Caption = Format$(MMD(1), "###.####")
      lblSigmaG.Caption = Format$(SIGMAG(1), "##.##")
   End If
   If (frmPart1.optMassfrac.Value = True) Then
      lblMMD.Caption = Format$("NUS")
      lblSigmaG.Caption = Format$("NUS")
   End If

   'lblMassratio.Caption = Format$(HM, "#.##e+##")
   '************************************************
   Dim Rows As Integer
```

```
    grdParticle.Col = 0
    For Rows = 1 To Z
        grdParticle.Row = Rows
        grdParticle.Text = Format$(Rows)
    Next Rows

    grdParticle.Col = 1
    For I = 1 To Z
        grdParticle.Row = I
        grdParticle.Text = Format$(DA(I), "0.####")
    Next I

    grdParticle.Col = 2
    For I = 1 To Z
        grdParticle.Row = I
        grdParticle.Text = Format$(DA(I + 1), "0.####")
    Next I

    grdParticle.Col = 3
    For I = 1 To Z
        grdParticle.Row = I
        grdParticle.Text = Format$(DP(I), "0.####")
    Next I

    grdParticle.Col = 4
    For I = 1 To Z
        grdParticle.Row = I
        grdParticle.Text = Format$(MI(I), "0.######")
    Next I


End Sub

Sub cmdPrint_Click ()
    On Error GoTo cmdPrint_Exit
    Dim Keypress, RUNTIME
    Keypress = MsgBox("Print Inlet Particle characteristics to user-specified output file.",
48 + 1, "Print To File")
    Select Case Keypress
    Case IDOK:
    '***********************************
    RUNTIME = Now
    'Open "a:\tsca.out" For Append As #1
    Open Drive For Append As #1
```

```
Write #1,
Write #1, "Visual Basic Program Output: Inlet Particle Characteristics"
Write #1, "********************************************"
Write #1, "Current Session Filename :", Drive
Write #1, "Date and Time of Run :", RUNTIME
Write #1,
If (frmPart1.optLog.Value = True) Then Write #1, "Log-Normal Distribution"
If (frmPart1.optMassfrac.Value = True) Then Write #1, "Mass Fraction Input in Size
Increments"
Write #1,
Write #1, "Verified Sum of Mass Fractions:", Format$(SUMMASS, "##.##")
Write #1, "Total No. of Particle Size Increments:", Z
Write #1, "Smallest Particle Size (um):", Format$(DMIN, "###.####")
Write #1, "Largest Particle Size (um):", Format$(DMAX, "###.####")
Write #1, "Width of Size Increments (um):", Format$(WI, "#.####")
Write #1, "Particle Density (g/cm3):", Format$(ROP(1), "##.##")
Write #1, "Dust Loading (g/m3):", Format$(GPERACM(1), "##.####")
If (frmPart1.optLog.Value = True) Then
   Write #1, "Mass Mean Diameter MMD (um):", Format$(MMD(1), "###.####")
   Write #1, "Geometric Standard Deviation:", Format$(SIGMAG(1), "##.##")
End If
If (frmPart1.optMassfrac.Value = True) Then
   Write #1, "Mass Mean Diameter MMD (um):", "Not User Specified"
   Write #1, "Geometric Standard Deviation:", "Not User Specified"
End If
Write #1,
Write #1, "No.", "Lower Size (um)", "-", "Upper Size (um)", "Diameter (um)", "Mass
Fraction"
For I = 1 To Z
   Write #1, I, Format$(DA(I), "0.####"), "-", Format$(DA(I + 1), "0.####"),
Format$(DP(I), "0.####"), Format$(MI(I), "0.######")
Next
Write #1,
Write #1, "********************************************"
Write #1,
Close #1
'***********************************
End Select
Exit Sub
'***********************************
cmdPrint_Exit:
MsgBox "An error occurred! You have not specified output filename. Program will
prompt for new output filename. Next click the command button <Print To File> to print
output to new filename.", 16, "Warning!"
```

```
   Drive = InputBox$("Enter new drive:\filename e.g., c:\tsca.out and then hit <Enter>.
Next click <Print To File> button to print output to new filename.", "Print To File",
"c:\tsca.out")
   Exit Sub

End Sub

Sub Form_Load ()
   'Defines justification of the cells
   'Call InitScrolls    'Initialize scroll bar
   Call Centercells    'Center all cell alignments
   Call Sizecells      'Specify width of cells
   Call Titles         'Initialize col and row titles
End Sub

Sub InitScrolls ()

End Sub

Sub Sizecells ()
   Dim Cols As Integer
   For Cols = 0 To 0
      grdParticle.ColWidth(Cols) = 500   'Twips
   Next Cols
   For Cols = 1 To 4
      grdParticle.ColWidth(Cols) = 1500   'Twips
   Next Cols

End Sub

Sub Titles ()
   'Fill in col titles
   grdParticle.Row = 0
   '****************************************
   grdParticle.Col = 0
   grdParticle.Text = "No."
   grdParticle.Col = 1
   grdParticle.Text = "Lower Size(um)"
   grdParticle.Col = 2
   grdParticle.Text = "Upper Size(um)"
   grdParticle.Col = 3
   grdParticle.Text = "Diameter(um)"
   grdParticle.Col = 4
   grdParticle.Text = "Mass Fraction"
```

```
'*******************************************

End Sub
***************************************************************************
```

Quench Tower Characteristics

Source: Frmquenc.txt

```
Option Explicit

Sub cmdCalc_Click ()
   On Error GoTo cmdCalc_Exit
   'Inform user that Quench inputs not provided
   If (txtWater.Text = "") Then
   MsgBox "Please input Quench tower parameters before proceeding with Calculate !",
16, "Warning !"
   Else
   If (txtDelP.Text = "") Then
   MsgBox "Please input Quench tower parameters before proceeding with Calculate !",
16, "Warning !"
   Else
   '*******************************************************
   Dim WI
   'Global TEMPKQ, TEMPFQ, MWQ, SPRESSQ, ABSPRESSQ, ROGQ
   'Global DELPQ, H2OQ, MAIR, MH2O, MCSAT, MH2OQ, WNETQ

   WI = Val(txtWater.Text)
   DELPQ = Val(txtDelP.Text)
   MH2OQ = MH2O + WIT
   'Water leaving quench chamber to sump or as carryover
   WNETQ = WI - WIT
   'Determine quench outlet temp
   TEMPKQ = TWB
   TEMPFQ = ((9 / 5) * (TEMPKQ - 273) + 32)
   MWQ = MWD * (1 - (MCSAT / 100)) + 18 * (MCSAT / 100)
   'Calculate static pressure
   SPRESSQ = SPRESS - DELPQ
   ABSPRESSQ = ABSPRESS + SPRESSQ / 13.6
   ROGQ = (MWQ / 22.4) * (273 / TEMPKQ) * (ABSPRESSQ / 29.92)
   H2OQ = MCSAT

   'Gas viscosity calcs
   Dim VISAIR, VISH2O, YAIR, YH2O, VISQ1, VISQ2
   'Global VISQ, FLOWQ, ACFMQ

   'VISQ = (Exp(((Log(TEMPKQ) - 18.0075) / 1.4311))) * .1
```

```
'Moist gas viscosity adjusted for moisture content & temp
'Calculate pure component viscosities for both gas and
'water vapor. Viscosity of mixture based on Perry [3-279]
'for low-pressure
VISAIR = .0000000406 * TEMPKQ + .00000663
VISH2O = .0000000382 * TEMPKQ - .00000176
YAIR = (100 - H2OQ) / 100
YH2O = H2OQ / 100
VISQ1 = (YAIR * VISAIR * Sqr(MWD)) + (YH2O * VISH2O * Sqr(18))
VISQ2 = (YAIR * Sqr(MWD)) + (YH2O * Sqr(18))
VISQ = VISQ1 / VISQ2

'Flow is (mass of dry air + mass water in + mass water evap)/ROGQ
'FLOWQ has units m3/sec
FLOWQ = (MAIR + MH2O + WIT) / ROGQ
ACFMQ = FLOWQ * 60 / .028316846

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

lblIntemp.Caption = Format$(TEMPK, "####.#")
lblOutemp.Caption = Format$(TEMPKQ, "####.#")
If (SPRESS = 0 And txtDelP.Text = "0") Then
   lblOutsp.Caption = Format$(SPRESSQ, "0.0")
Else
   lblOutsp.Caption = Format$(SPRESSQ, "###.##")
End If
lblAbssp.Caption = Format$(ABSPRESSQ, "##.##")
lblMolwt.Caption = Format$(MWQ, "##.##")
lblDensity.Caption = Format$(ROGQ, "##.####")
lblViscos.Caption = Format$(VISQ, "#.##e+##")

lblInflow.Caption = Format$(FLOW, "##.####")
lblOutflow.Caption = Format$(FLOWQ, "##.####")
lblOutwater.Caption = Format$(H2OQ, "###.##")
lblDryair.Caption = Format$(MAIR, "##.####")
lblMassw.Caption = Format$(MH2OQ, "##.####")
lblWatremn.Caption = Format$(WNETQ, "##.####")

End If
End If
Exit Sub
'**********************************************
cmdCalc_Exit:
MsgBox "An error occurred, please re-specify your inputs!", 16, "Warning!"
txtWater.Text = ""
```

```
    txtDelP.Text = ""
    Exit Sub

End Sub

Sub cmdContinue_Click ()
    'Inform user that Quench inputs not provided
    If (txtWater.Text = "") Then
    MsgBox "Please input Quench tower parameters before proceeding with Calculate !",
16, "Warning !"
    Else
    If (txtDelP.Text = "") Then
    MsgBox "Please input Quench tower parameters before proceeding with Calculate !",
16, "Warning !"
    Else
    If (lblIntemp.Caption = "") Then
    MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
    Else
    If (lblOutemp.Caption = "") Then
    MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
    Else
    If (lblOutsp.Caption = "") Then
    MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
    Else
    If (lblAbssp.Caption = "") Then
    MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
    Else
    If (lblMolwt.Caption = "") Then
    MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
    Else
    If (lblDensity.Caption = "") Then
    MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
    Else
    If (lblViscos.Caption = "") Then
    MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
    Else
    If (lblInflow.Caption = "") Then
```

```
MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
Else
If (lblOutflow.Caption = "") Then
MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
Else
If (lblOutwater.Caption = "") Then
MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
Else
If (lblDryair.Caption = "") Then
MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
Else
If (lblMassw.Caption = "") Then
MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
Else
If (lblWatremn.Caption = "") Then
MsgBox "Please input Quench tower parameters before proceeding with Continue !",
16, "Warning !"
Else

Dim Keypress
Keypress = MsgBox("Quench Tower characteristics have been specified.  Please
proceed to select Venturi (under Controls) from the Menu.", 48 + 1, "End Quench
Tower")
Select Case Keypress
Case IDOK:
Unload frmQuench
frmMain.mnuQuench.Checked = True
frmMain.mnuQuench.Enabled = False
frmMain.mnuVenturi.Enabled = True
End Select

End If
End If
End If
End If
End If
End If
End If
End If
```

```
    End If
    End If
    End If
    End If
    End If
    End If
    End If

End Sub

Sub cmdPrint_Click ()
    On Error GoTo cmdPrint_Exit
    Dim Keypress, RUNTIME
    Keypress = MsgBox("Print Quench Tower characteristics to user-specified output
file.", 48 + 1, "Print To File")
    Select Case Keypress
    Case IDOK:
    '**********************************
    RUNTIME = Now
    'Open "a:\tsca.out" For Append As #1
    Open Drive For Append As #1
    Write #1,
    Write #1, "Visual Basic Program Output: Quench Tower Characteristics"
    Write #1, "************************************************************"
    Write #1, "Current Session Filename :", Drive
    Write #1, "Date and Time of Run :", RUNTIME
    Write #1,
    Write #1, "Actual Water Injection Rate (kg/sec):", Format$(WI, "#.####")
    Write #1, "Pressure Drop Across Quench (in H2O):", Format$(DELPQ, "###.#")
    Write #1,
    Write #1, "Inlet Gas Temperature (K):", Format$(TEMPK, "####.#")
    Write #1, "Outlet Gas Temperature (K):", Format$(TEMPKQ, "####.#")
    Write #1, "Outlet Static Pressure (in H2O):", Format$(SPRESSQ, "###.##")
    Write #1, "Absolute Pressure (in Hg):", Format$(ABSPRESSQ, "##.##")
    Write #1, "Outlet Gas Molecular Weight:", Format$(MWQ, "##.##")
    Write #1, "Outlet Gas Density (kg/m3):", Format$(ROGQ, "##.####")
    Write #1, "Outlet Gas Viscosity (kg/sec-m):", Format$(VISQ, "#.##e+##")
    Write #1,
    Write #1, "Inlet Gas Flowrate (m3/sec):", Format$(FLOW, "##.####")
    Write #1, "Outlet Gas Flowrate (m3/sec):", Format$(FLOWQ, "##.####")
    Write #1, "Outlet Gas Water Content (vol%):", Format$(H2OQ, "###.##")
    Write #1, "Mass of Dry Air (kg/sec):", Format$(MAIR, "##.####")
    Write #1, "Outlet Water Vapor (kg/sec):", Format$(MH2OQ, "##.####")
    Write #1, "Water Remaining as Liquid (kg/sec):", Format$(WNETQ, "##.####")
```

```
   Write #1,
   Write #1, "****************************************************************"
   Write #1,
   Close #1
   '************************************
   End Select
   Exit Sub
   '************************************
cmdPrint_Exit:
   MsgBox "An error occurred! You have not specified output filename. Program will prompt for new output filename. Next click the command button <Print To File> to print output to new filename.", 16, "Warning!"
   Drive = InputBox$("Enter new drive:\filename e.g., c:\tsca.out and then hit <Enter>. Next click <Print To File> button to print output to new filename.", "Print To File", "c:\tsca.out")
   Exit Sub

End Sub

Sub Form_Load ()
   'Based on gas characteristics calculate amount of water
   'reqd to reach saturation.

   Dim HVSAT, HMSAT, MGAS, MFRACH2O
   'Global MCSAT,MH2O, MAIR, WIT

   'MCSAT: Moisture % at saturation
   MCSAT = (PWB / ABSPRESS) * 100
   'HVSAT: Mol fraction of water/dry air at saturation
   HVSAT = MCSAT / (100 - MCSAT)
   'HMSAT: Mass frac of water/dry air at saturation
   HMSAT = HVSAT * 18 / MWD
   'Mass of gas before water injection =
   'flowrate*density, kg/sec
   MGAS = FLOW * ROG
   'Mass fraction of water/gas before water injection
   MFRACH2O = HM / (HM + 1)
   'Mass of inlet water is MGAS*MFRACH2O in kg/sec
   MH2O = MGAS * MFRACH2O
   'Mass of dry air, MAIR in kg/sec
   MAIR = MGAS * (1 - MFRACH2O)
   'Theoretical water reqd to reach saturation is WIT
   WIT = (HMSAT * MAIR) - MH2O
   '*********************************************
```

```
   lblSattmp.Caption = Format$(TWBF, "####.#")
   lblWatreq.Caption = Format$(WIT, "##.####")

End Sub
```

****

Venturi Scrubber Characteristics

Source: Frmventu.txt

```
Option Explicit

Sub cmdBypass_Click ()
   Dim Keypress
   Keypress = MsgBox("You wish to by-pass Venturi.", 48 + 1, "Venturi By-Pass")
   Select Case Keypress
   Case IDOK:
   txtLiquid.Text = "0"
   txtDelP.Text = "0"
   txtGasVg.Text = "1000"
   'MsgBox "In order to by-pass Venturi - enter BOTH Liquid Addition (gpm) = 0 AND
Venturi Pressure Drop (in H2O) = 0.", 64, "By-Pass Venturi"
   End Select

End Sub

Sub cmdCalc_Click ()
   If (lblQwater.Caption = "") Then
      'Inform user that Venturi inputs not provided
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
      Else
      If (txtLiquid.Text = "") Then
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
      Else
      If (txtDelP.Text = "") Then
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
      Else
      If (txtGasVg.Text = "") Then
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
      Else
      If (txtFactor.Text = "") Then
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
```

```
    Else
    If (txtLiqden.Text = "") Then
    MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
    Else
  '*************************************************
  Dim F, RL, SFCT, LVIS
  'Global FLOWV, TEMPKV, H2OV, VISV, SPRESSV, ABSPRESSV
  'Global QL, PR, VG, DELPV, DD

  QL = Val(txtLiquid.Text)
  PR = Val(txtDelP.Text)
  VG = Val(txtGasVg.Text)

  If (PR = 0 And VG = 0) Then
    MsgBox "Pressure Drop and Throat Velocity BOTH cannot be zero!  EITHER
parameter has to be specified.  Please re-enter your inputs.", 16, "Warning !"
  Else

  F = Val(txtFactor.Text)
  RL = Val(txtLiqden.Text)
  'Convert to cgs to use venturi eqns from Calvert
  'PR: Pressure drop [cm H2O]
  PR = PR * 2.54
  'QL: g/sec or cm3/sec
  QL = QL * (3783 / 60)
  'FLOWV [cm3/sec] is FLOWQ*10^6 FLOWQ[m3/sec]
  FLOWV = FLOWQ * 10 ^ 6
  'Temp in venturi is same as outlet of quench
  TEMPKV = TEMPKQ
  H2OV = H2OQ
  'VG: [cm/sec]
  VG = VG * (30.48 / 60)
  'Viscosity is same as viscosity in quench tower unless
  'there is particle growth
  'Convert SI to CGS for VISV
  VISV = VISQ * 10
  '*************************************************
  'Calculate pressure drop OR velocity
  If (PR = 0) GoTo V2100
  If (VG = 0) GoTo V2200

V2100:
  PR = (.00103) * (VG ^ 2) * QL / FLOWV
```

```
  GoTo V1000

V2200:
  VG = ((PR * FLOWV) / (.00103 * QL)) ^ .5
  GoTo V1000
  '**************************************************
V1000:
  'VG [cm/sec] VG1 [fpm output purposes, not considered]
  'VG1 = VG * 60 / 30.48
  'Pressure drop [in H2O]
  DELPV = PR * (1 / 2.54)
  'Static pressure [in H2O]
  SPRESSV = SPRESSQ - DELPV
  'Absolute pressure [in Hg]
  ABSPRESSV = ABSPRESSQ + SPRESSV / 13.6
  'Calculate the drop size [cm] w/ Nukiyama Tanasawa eqn
  ' Use full-blown equation ...
  'DD = (50 / VG) + 91.8 * ((QL / FLOWV) ^ 1.5)
  'DD1 = DD * 10000   '[um] not considered
  'SFCT: Water surface tension in dynes/cm
  'LVIS: Water viscosity in poise
  SFCT = -.1676 * TEMPKV + 121.81
  LVIS = 2000000000000# * TEMPKV ^ -5.8009
  DD = (58600 / VG) * (SFCT / RL) ^ .5 + 597 * (LVIS / (SFCT * RL) ^ .5) ^ .45 *
(1000 * (QL / FLOWV)) ^ 1.5
  DD = DD / 10000    'units of cm
  'Calculate effy parameters for DP(Z)
  Dim RG, TV, LAMBDA
  ReDim SIM1(Z), KC(Z), FK(Z), SI(Z), PT(Z)
  ReDim EFF(Z), PEFF(Z), TEFF(Z)
  '********* Cunningham correction factor (KC) ********
  'KC(I) = 1 + ((.009730001) * (TEMPKV ^ .5) / DP(I))
  'Incorporate full-blown KC version
  'KC calc needs VISV [kg/m-sec] (nee VISQ, kg/m-sec),
  'TEMPKV(K), ABSPRESSV(in Hg), ROGQ(kg/m3), Dia of particle
  'DP(I)um(to m)
  'Gas constant RG= 8314.3 [J/kgmol-K]
  'TV: Mean thermal velocity of gas molecules [m/sec]
  'MWQ: Approx. gas mo. wt = 28.96 (adjusted in Quench)
  'LAMBDA: Mean free path of gas molecules [m]
  For I = 1 To Z
    RG = 8314.3
    TV = ((8 * RG * TEMPKV / (3.141592654 * MWQ))) ^ .5
    LAMBDA = (VISV / 10) / (.499 * ROGQ * TV)
```

```
    SIM1(I) = 1.257 + .4 * Exp(-.55 * (DP(I) * .000001) / LAMBDA)
    KC(I) = 1 + 2 * LAMBDA * SIM1(I) / (DP(I) * .000001)

    'Inertial impaction parameter - unitless
    SI(I) = VG * ROP(1) * KC(I) * ((DP(I) * (10 ^ (-4))) ^ 2) / (9 * VISV * DD)

    'Calvert effy parameter
    FK(I) = (-.7 - SI(I) * F + 1.4 * Log((SI(I) * F + .7) / .7) + (.49 / (.7 + SI(I) * F))) /
SI(I)
    PT(I) = Exp((2 * QL * VG * RL * DD * FK(I)) / (55 * FLOWV * VISV))
    EFF(I) = (1 - PT(I))
  Next
  ReDim MPV1(Z), MPV2(Z), MPV(Z)
  'Global SIGVEFF, SIGMPV2
  TEFF(0) = 0
  MPV2(0) = 0
  'Overall Effy is summation of MI(I)*EFF(I)/1.0

  For I = 1 To Z
    PEFF(I) = MI(I) * EFF(I)
    TEFF(I) = TEFF(I - 1) + PEFF(I)
    MPV1(I) = MI(I) * (1 - EFF(I))
    MPV2(I) = MPV2(I - 1) + MPV1(I)
  Next
  SIGVEFF = TEFF(Z) * 1

  'Sum of mass fractions penetrating venturi: MPV2(Z) n.e. 1 anymore
  'Adjust mass fractions to reflect sum(MPV(I))=1
  'Mass penetrating venturi MPV(I)
  SIGMPV2 = MPV2(Z) * 1
  For I = 1 To Z
    MPV(I) = MPV1(I) / SIGMPV2
  Next
  MsgBox "Venturi Calculations done successfully !  Please proceed to Output
Summary.", 48, "Attention"

  End If 'MsgBox for PR=0 and VG=0
  End If
  End If
  End If
  End If
  End If
  End If
```

```
End Sub

Sub cmdVenSum_Click ()
   On Error GoTo cmdVenSum_Exit
   If (lblQwater.Caption = "") Then
      'Inform user that Venturi inputs not provided
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
      Else
      If (txtLiquid.Text = "") Then
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
      Else
      If (txtDelP.Text = "") Then
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
      Else
      If (txtGasVg.Text = "") Then
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
      Else
      If (txtFactor.Text = "") Or (txtFactor.Text = "0") Then
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
      Else
      If (txtLiqden.Text = "") Then
      MsgBox "Please input Venturi parameters before proceeding with Calculate !", 16,
"Warning !"
      Else
   '************************************************
   Dim SFCT, LVIS
   'Global FLOWV, TEMPKV, H2OV, VISV, SPRESSV, ABSPRESSV
   'Global F, RL, QL, PR, VG, DELPV, DD

   QL = Val(txtLiquid.Text)
   PR = Val(txtDelP.Text)
   VG = Val(txtGasVg.Text)

   If (PR = 0 And VG = 0) Then
      MsgBox "Pressure Drop and Throat Velocity BOTH cannot be zero!  EITHER
parameter has to be specified.  Please re-enter your inputs.", 16, "Warning !"
   Else

   F = Val(txtFactor.Text)
```

```
    RL = Val(txtLiqden.Text)
    'Convert to cgs to use venturi eqns from Calvert
    'PR: Pressure drop [cm H2O]
    PR = PR * 2.54
    'QL: g/sec or cm3/sec
    QL = QL * (3783 / 60)
    'FLOWV [cm3/sec] is FLOWQ*10^6 FLOWQ[m3/sec]
    FLOWV = FLOWQ * 10 ^ 6
    'Temp in venturi is same as outlet of quench
    TEMPKV = TEMPKQ
    H2OV = H2OQ
    'VG: [cm/sec]
    VG = VG * (30.48 / 60)
    'Viscosity is same as viscosity in quench tower unless
    'there is particle growth
    'Convert SI to CGS for VISV
    VISV = VISQ * 10
    '************************************************
    'Calculate pressure drop OR velocity
    If (PR = 0) GoTo V1500
    If (VG = 0) GoTo V1600

V1500:
    PR = (.00103) * (VG ^ 2) * QL / FLOWV
    GoTo V2000

V1600:
    VG = ((PR * FLOWV) / (.00103 * QL)) ^ .5
    GoTo V2000
    '************************************************
V2000:
    'VG [cm/sec] VG1 [fpm output purposes, not considered]
    'VG1 = VG * 60 / 30.48
    'Pressure drop [in H2O]
    DELPV = PR * (1 / 2.54)
    'Static pressure [in H2O]
    SPRESSV = SPRESSQ - DELPV
    'Absolute pressure [in Hg]
    ABSPRESSV = ABSPRESSQ + SPRESSV / 13.6
    'Calculate the drop size [cm] w/ Nukiyama Tanasawa eqn
    ' Use full-blown equation ...
    'DD = (50 / VG) + 91.8 * ((QL / FLOWV) ^ 1.5)
    'DD1 = DD * 10000   '[um] not considered
    'SFCT: Water surface tension in dynes/cm
```

```
'LVIS: Water viscosity in poise
SFCT = -.1676 * TEMPKV + 121.81
LVIS = 2000000000000# * TEMPKV ^ -5.8009
DD = (58600 / VG) * (SFCT / RL) ^ .5 + 597 * (LVIS / (SFCT * RL) ^ .5) ^ .45 *
(1000 * (QL / FLOWV)) ^ 1.5
DD = DD / 10000    'units of cm
'Calculate effy parameters for DP(Z)
Dim RG, TV, LAMBDA
ReDim SIM1(Z), KC(Z), FK(Z), SI(Z), PT(Z)
ReDim EFF(Z), PEFF(Z), TEFF(Z)
'********* Cunningham correction factor (KC) ********
'KC(I) = 1 + ((.009730001) * (TEMPKV ^ .5) / DP(I))
'Incorporate full-blown KC version
'KC calc needs VISV [kg/m-sec] (nee VISQ, kg/m-sec),
'TEMPKV(K), ABSPRESSV(in Hg), ROGQ(kg/m3), Dia of particle
'DP(I)um(to m)
'Gas constant RG= 8314.3 [J/kgmol-K]
'TV: Mean thermal velocity of gas molecules [m/sec]
'MWQ: Approx. gas mo. wt = 28.96 (adjusted in Quench)
'LAMBDA: Mean free path of gas molecules [m]
For I = 1 To Z
   RG = 8314.3
   TV = ((8 * RG * TEMPKV / (3.141592654 * MWQ))) ^ .5
   LAMBDA = (VISV / 10) / (.499 * ROGQ * TV)
   SIM1(I) = 1.257 + .4 * Exp(-.55 * (DP(I) * .000001) / LAMBDA)
   KC(I) = 1 + 2 * LAMBDA * SIM1(I) / (DP(I) * .000001)

   'Inertial impaction parameter - unitless
   SI(I) = VG * ROP(1) * KC(I) * ((DP(I) * (10 ^ (-4))) ^ 2) / (9 * VISV * DD)

   'Calvert effy parameter
   FK(I) = (-.7 - SI(I) * F + 1.4 * Log((SI(I) * F + .7) / .7) + (.49 / (.7 + SI(I) * F))) /
SI(I)
   PT(I) = Exp((2 * QL * VG * RL * DD * FK(I)) / (55 * FLOWV * VISV))
   EFF(I) = (1 - PT(I))
Next
ReDim MPV1(Z), MPV2(Z), MPV(Z)
'Global SIGVEFF, SIGMPV2
TEFF(0) = 0
MPV2(0) = 0
'Overall Effy is summation of MI(I)*EFF(I)/1.0

For I = 1 To Z
   PEFF(I) = MI(I) * EFF(I)
```

```
    TEFF(I) = TEFF(I - 1) + PEFF(I)
    MPV1(I) = MI(I) * (1 - EFF(I))
    MPV2(I) = MPV2(I - 1) + MPV1(I)
  Next
  SIGVEFF = TEFF(Z) * 1

  'Sum of mass fractions penetrating venturi: MPV2(Z) n.e. 1 anymore
  'Adjust mass fractions to reflect sum(MPV(I))=1
  'Mass penetrating venturi MPV(I)
  SIGMPV2 = MPV2(Z) * 1
  For I = 1 To Z
    MPV(I) = MPV1(I) / SIGMPV2
  Next
  MsgBox "Venturi Calculations done successfully !  Please proceed to Output
Summary.", 48, "Attention"
  frmVen2.Show MODELESS
  End If 'MsgBox for PR=0 and VG=0
  End If
  End If
  End If
  End If
  End If
  End If
  Exit Sub
  '************************************************
cmdVenSum_Exit:
  MsgBox "An error occurred, please re-specify your inputs!", 16, "Warning!"
  txtLiquid.Text = ""
  txtDelP.Text = ""
  txtGasVg.Text = ""
  txtFactor.Text = ""
  txtLiqden.Text = ""
  Exit Sub

End Sub

Sub Command1_Click ()
  MsgBox "To by-pass the venturi - enter both Liquid Addition (gpm) = 0 AND Pressure
Drop (in H2O) = 0.", 64, "Venturi By-Pass"

End Sub

Sub Form_Load ()
  lblQwater.Caption = Format$(ACFMQ, "#######.#")
```

```
End Sub
**************************************************************************
Source: Frmven2.txt
Option Explicit
'Option base 1

Sub Centercells ()
   'Sets the justification of the grid's cells to center
   Dim Cols As Integer
   For Cols = 0 To 4
      grdVenturi.Col = Cols
      grdVenturi.FixedAlignment(Cols) = 2
   Next Cols
   For Cols = 1 To 4
      grdVenturi.ColAlignment(Cols) = 2
   Next Cols

End Sub

Sub cmdContinue_Click ()
   Dim Keypress
   Keypress = MsgBox("Venturi characteristics have been specified.  Please proceed to
select Packed Bed (under Controls) from Menu", 48 + 1, "End Venturi")
   Select Case Keypress
   Case IDOK:
   Unload frmVenturi
   Unload frmVen2
   frmMain.mnuVenturi.Checked = True
   frmMain.mnuVenturi.Enabled = False
   frmMain.mnuPackBed.Enabled = True
   End Select

End Sub

Sub cmdDisplay_Click ()
   lblGastmp.Caption = Format$(TEMPKV, "####.#")
   If (SPRESSV = 0) Then
      lblSpr.Caption = Format$(SPRESSV, "0.0")
   Else
      lblSpr.Caption = Format$(SPRESSV, "###.##")
   End If
   lblAbspr.Caption = Format$(ABSPRESSV, "###.##")
   lblMolwt.Caption = Format$(MWQ, "##.##")
```

```
lblGasden.Caption = Format$(ROGQ, "##.####")
lblGasvis.Caption = Format$(VISQ, "#.##e+##")
lblFlowrate.Caption = Format$(FLOWQ, "##.####")
lblIncr.Caption = Format$(Z)
lblWater.Caption = Format$(H2OV, "###.##")
lblLGratio.Caption = Format$(QL / FLOWV, "#.##e+##")
lblDropdia.Caption = Format$(DD, "#.##e+##")
If (PR = 0) Then
   lblPrdrop.Caption = Format$(DELPV, "0.0")
Else
   lblPrdrop.Caption = Format$(DELPV, "###.#")
End If
lblGasvel.Caption = Format$(VG, "#####.#")

If (QL = 0 And PR = 0) Then
   lblSigveff.Caption = Format$(SIGVEFF, "0.0%")
Else
   lblSigveff.Caption = Format$(SIGVEFF, "###.######%")
End If
'***********************************************
Dim Rows As Integer

grdVenturi.Col = 0
For Rows = 1 To Z
   grdVenturi.Row = Rows
   grdVenturi.Text = Format$(Rows)
Next Rows

'grdVenturi.Col = 1
'For I = 1 To Z
   'grdVenturi.Row = I
   'grdVenturi.Text = Format$(KC(I), "##.####")
'Next I

grdVenturi.Col = 1
For I = 1 To Z
   grdVenturi.Row = I
   grdVenturi.Text = Format$(DA(I), "0.####")
Next I

grdVenturi.Col = 2
For I = 1 To Z
   grdVenturi.Row = I
   grdVenturi.Text = Format$(DA(I + 1), "0.####")
```

```
    Next I

    grdVenturi.Col = 3
    For I = 1 To Z
       grdVenturi.Row = I
       grdVenturi.Text = Format$(DP(I), "0.####")
    Next I

    grdVenturi.Col = 4
    For I = 1 To Z
       grdVenturi.Row = I
       grdVenturi.Text = Format$(EFF(I), "#.######")
    Next I

End Sub

Sub cmdPrint_Click ()
    On Error GoTo cmdPrint_Exit
    Dim Keypress, RUNTIME
    Keypress = MsgBox("Print Venturi Scrubber characteristics to user-specified output
file.", 48 + 1, "Print To File")
    Select Case Keypress
    Case IDOK:
    '**********************************
    RUNTIME = Now
    'Open "a:\tsca.out" For Append As #1
    Open Drive For Append As #1
    Write #1,
    Write #1, "Visual Basic Program Output: Venturi Scrubber Characteristics"
    Write                                                                    #1,
"*****************************************************************"
    Write #1, "Current Session Filename :", Drive
    Write #1, "Date and Time of Run :", RUNTIME
    Write #1,
    Write #1, "Liquid Addition Rate (gal/min):", Format$(QL * 60 / 3783, "#####.#")
    Write #1, "F-Factor:", Format$(F, "#.##")
    Write #1, "Liquid Density (g/cm3):", Format$(RL, "##.##")
    Write #1,
    Write #1, "Gas Temperature (K):", Format$(TEMPKV, "####.##")
    Write #1, "Outlet Static Pressure (in H2O):", Format$(SPRESSV, "###.##")
    Write #1, "Outlet Absolute Pressure (in Hg):", Format$(ABSPRESSV, "###.##")
    Write #1, "Outlet Gas Molecular Weight:", Format$(MWQ, "##.##")
    Write #1, "Outlet Gas Density (kg/m3):", Format$(ROGQ, "##.####")
    Write #1, "Outlet Gas Viscosity (kg/sec-m):", Format$(VISQ, "#.##e+##")
```

```
  Write #1, "Gas Flowrate (m3/sec):", Format$(FLOWQ, "##.####")
  Write #1,
  Write #1, "Outlet Gas Water Content (vol%):", Format$(H2OV, "###.##")
  Write #1, "Liquid/Gas Volume Ratio (QL/QG):", Format$(QL / FLOWV, "#.##e+##")
  Write #1, "Drop Diameter (cm):", Format$(DD, "#.##e+##")
  Write #1, "Pressure Drop (in H2O):", Format$(DELPV, "###.#")
  Write #1, "Throat Gas Velocity (cm/sec):", Format$(VG, "#####.#")
  Write #1,
  Write #1, "Total No. of Particle Size Increments:", Z
  Write #1,
  Write #1, "No.", "Lower Size (um)", "-", "Upper Size (um)", "Diameter (um)", "Frac
Effy"
  For I = 1 To Z
    Write #1, I, Format$(DA(I), "0.####"), "-", Format$(DA(I + 1), "0.####"),
Format$(DP(I), "0.####"), Format$(EFF(I), "#.######")
  Next
  Write #1,
  Write #1, "Overall Venturi Efficiency:", Format$(SIGVEFF, "###.######%")
  Write #1,
  Write #1,
"************************************************************"
  Write #1,
  Close #1
  '*********************************
  End Select
  Exit Sub
  '*********************************
cmdPrint_Exit:
  MsgBox "An error occurred! You have not specified output filename. Program will
prompt for new output filename. Next click the command button <Print To File> to print
output to new filename.", 16, "Warning!"
  Drive = InputBox$("Enter new drive:\filename e.g., c:\tsca.out and then hit <Enter>.
Next click <Print To File> button to print output to new filename.", "Print To File",
"c:\tsca.out")
  Exit Sub

End Sub

Sub cmdRevisit_Click ()
  Dim Keypress
  Keypress = MsgBox("Do you wish to revisit earlier controls starting with Quench,
without re-specifying inlet gas and particle characteristics ?", 32 + 1, "Re-Visit Earlier
Controls")
  Select Case Keypress
```

```
    Case IDOK:
    Unload frmVenturi
    Unload frmVen2
    frmMain.mnuVenturi.Enabled = False

    frmMain.mnuQuench.Enabled = True
    frmMain.mnuQuench.Checked = False
    End Select

End Sub

Sub Command1_Click ()
    Dim Keypress
    Keypress = MsgBox("Do you wish to revisit earlier controls starting with Quench ?",
32 + 1, "Re-Visit Earlier Controls")
    Select Case Keypress
    Case IDOK:
    Unload frmVenturi
    Unload frmVen2
    frmMain.mnuVenturi.Enabled = False
    End Select

End Sub

Sub Form_Load ()
    'Defines justification of the cells
    Call Centercells    'Center all cell alignments
    Call Sizecells      'Specify width of cells
    Call Titles         'Initialize col and row titles

End Sub

Sub Sizecells ()
    Dim Cols As Integer
    For Cols = 0 To 0
        grdVenturi.ColWidth(Cols) = 500 'Twips
    Next Cols
    'For Cols = 1 To 1
        'grdVenturi.ColWidth(Cols) = 1000 'Twips
    'Next Cols
    For Cols = 1 To 2
        grdVenturi.ColWidth(Cols) = 2000 'Twips
    Next Cols
    For Cols = 3 To 3
```

```
      grdVenturi.ColWidth(Cols) = 1200 'Twips
   Next Cols
   For Cols = 4 To 4
      grdVenturi.ColWidth(Cols) = 1500 'Twips
   Next Cols

End Sub

Sub Titles ()
   'Fill in col titles
   grdVenturi.Row = 0
   '****************************************
   grdVenturi.Col = 0
   grdVenturi.Text = "No."
   'grdVenturi.Col = 1
   'grdVenturi.Text = "Kc"
   grdVenturi.Col = 1
   grdVenturi.Text = "Lower Size(um)"
   grdVenturi.Col = 2
   grdVenturi.Text = "Upper Size(um)"
   grdVenturi.Col = 3
   grdVenturi.Text = "Dia(um)"
   grdVenturi.Col = 4
   grdVenturi.Text = "Frac Effy"
   '****************************************


End Sub
************************************************************************
```

Packed Bed Characteristics

```
Source: Frmpackb.txt
Option Explicit

Sub Centercells ()
   'Sets justification of the grid's cells to center
   Dim Cols As Integer
   For Cols = 0 To 5
      grdPackbed.Col = Cols
      grdPackbed.FixedAlignment(Cols) = 2
   Next Cols
   For Cols = 1 To 5
      grdPackbed.ColAlignment(Cols) = 2
   Next Cols
```

```
End Sub

Sub chkESP_Click ()
   frmMain.mnuESP.Enabled = True
   '******************************
   frmMain.mnuIWS1.Enabled = False
End Sub

Sub chkIWS_Click ()
   frmMain.mnuIWS1.Enabled = True
   '*********************************
   frmMain.mnuESP.Enabled = False
End Sub

Sub cmdCalc_Click ()
   If (lblIncr.Caption = "") Then
   MsgBox "Please input Packed Bed parameters before proceeding with Calculate !", 16,
"Warning !"
   Else
   If (txtPBdepth.Text = "") Then
   MsgBox "Please input Packed Bed parameters before proceeding with Calculate !", 16,
"Warning !"
   Else
   '********************************************
   Dim SIGMPPB2
   ReDim DAERO(Z), PBEFF(Z), TEFFPB(Z), PEFFPB(Z)
   ReDim MPPB1(Z), MPPB2(Z), MPPB(Z)
   'Global DAERO(), PBEFF(), TEFFPB(), PEFFPB(), MPPB()
   'Global DPB, SIGPBEFF, SYSPB
   DPB = Val(txtPBdepth.Text)
   For I = 1 To Z
      DAERO(I) = DP(I) * (ROP(1) ^ .5)
      If DAERO(I) <= 5 Then
      PBEFF(I) = 1 - (((.006199 * DAERO(I) ^ 3) - (.060103 * DAERO(I) ^ 2) +
(.011302 * DAERO(I)) + .77037) ^ (DPB / 8))
      Else
      PBEFF(I) = PBEFF(I - 1)
      End If
   Next
   'Overall effy is summation of MPV(I)*PBEFF(I)/1.0
   'Original particle mass MI(I)
   'Mass penetrating venturi MPV(I)
   'Mass penetrating Packed Bed MPPB(I)
   TEFFPB(0) = 0
```

```
MPPB2(0) = 0

For I = 1 To Z
   PEFFPB(I) = MPV(I) * PBEFF(I)
   TEFFPB(I) = TEFFPB(I - 1) + PEFFPB(I)
   MPPB1(I) = MPV(I) * (1 - PBEFF(I))
   MPPB2(I) = MPPB2(I - 1) + MPPB1(I)
Next

'Sum of mass fractions penetrating PB: MPPB2(Z) n.e. 1 anymore
'Adjust mass fractions to reflect sum(MPPB(I))=1
'Mass penetrating Packed Bed MPPB(I)
SIGMPPB2 = MPPB2(Z) * 1
For I = 1 To Z
   MPPB(I) = MPPB1(I) / SIGMPPB2
Next
SIGPBEFF = TEFFPB(Z) * 1
SYSPB = 0
SYSPB = SIGVEFF + (SIGMPV2 * TEFFPB(Z))

'*********************************************
'Outputs on same screen
Dim Rows As Integer

grdPackbed.Col = 0
For Rows = 1 To Z
   grdPackbed.Row = Rows
   grdPackbed.Text = Format$(Rows)
Next Rows

grdPackbed.Col = 1
For I = 1 To Z
   grdPackbed.Row = I
   grdPackbed.Text = Format$(DA(I), "0.####")
Next I

grdPackbed.Col = 2
For I = 1 To Z
   grdPackbed.Row = I
   grdPackbed.Text = Format$(DA(I + 1), "0.####")
Next I

grdPackbed.Col = 3
For I = 1 To Z
```

```
      grdPackbed.Row = I
      grdPackbed.Text = Format$(DP(I), "0.####")
   Next I

   grdPackbed.Col = 4
   For I = 1 To Z
      grdPackbed.Row = I
      grdPackbed.Text = Format$(MPV(I), "#.######")
   Next I

   grdPackbed.Col = 5
   For I = 1 To Z
      grdPackbed.Row = I
      grdPackbed.Text = Format$(PBEFF(I), "#.######")
   Next I

   If (txtPBdepth.Text = "0") Then
      lblSigpbeff.Caption = Format$(SIGPBEFF, "0.0%")
   Else
      lblSigpbeff.Caption = Format$(SIGPBEFF, "###.######%")
   End If
   lblSyspb.Caption = Format$(SYSPB, "###.######%")
   '*******************************************
   End If
   End If

End Sub

Sub cmdContinue_Click ()
   Dim Keypress
   Keypress = MsgBox("Packed Bed characteristics have been specified. Please proceed
to select ESP or IWS (under Controls) from Menu", 48 + 1, "End Packed Bed")
   Select Case Keypress
   Case IDOK:
   Unload frmPackbed
   frmMain.mnuPackBed.Checked = True
   frmMain.mnuPackBed.Enabled = False
   End Select

End Sub

Sub cmdPrint_Click ()
   On Error GoTo cmdPrint_Exit
   Dim Keypress, RUNTIME
```

```
   Keypress = MsgBox("Print Packed Bed characteristics to user-specified output file.",
48 + 1, "Print To File")
   Select Case Keypress
   Case IDOK:
   '***********************************
   RUNTIME = Now
   'Open "a:\tsca.out" For Append As #1
   Open Drive For Append As #1
   Write #1,
   Write #1, "Visual Basic Program Output: Packed Bed Characteristics"
   Write #1, "***********************************************************"
   Write #1, "Current Session Filename :", Drive
   Write #1, "Date and Time of Run :", RUNTIME
   Write #1,
   Write #1, "Total No. of Particle Size Increments:", Z
   Write #1, "Total Depth of Wetted Packing (feet):", Format$(DPB, "###.#")
   Write #1,
   Write #1, "No.", "Lower Size (um)", "-", "Upper Size (um)", "Diameter (um)",
"MassFrac In", "Frac Effy"
   For I = 1 To Z
      Write #1, I, Format$(DA(I), "0.####"), "-", Format$(DA(I + 1), "0.####"),
Format$(DP(I), "0.####"), Format$(MPV(I), "#.######"), Format$(PBEFF(I),
"#.######")
   Next
   Write #1,
   Write #1, "Overall Packed Bed Efficiency:", Format$(SIGPBEFF, "###.######%")
   Write #1, "Overall System Efficiency:", Format$(SYSPB, "###.######%")
   Write #1,
   Write #1, "***********************************************************"
   Write #1,
   Close #1
   '***********************************
   End Select
   Exit Sub
   '***********************************
cmdPrint_Exit:
   MsgBox "An error occurred! You have not specified output filename. Program will
prompt for new output filename. Next click the command button <Print To File> to print
output to new filename.", 16, "Warning!"
   Drive = InputBox$("Enter new drive:\filename e.g., c:\tsca.out and then hit <Enter>.
Next click <Print To File> button to print output to new filename.", "Print To File",
"c:\tsca.out")
   Exit Sub
```

```
End Sub

Sub cmdReadme_Click ()
  MsgBox "Packed Bed calculations will follow the empirical equation developed at UTK under this contract.  The equation predicts the efficiency of a cross-flow Packed Bed containing 2-inch type-R Tellerette packing.  Equation valid for particle aerodynamic diameter DAERO(I):0.12-5 um.  For DAERO(I) >= 5um, program sets CONSTANT fractional efficiency outside equation range.  In order to by-pass the Packed Bed input Depth of Wetted Packing = 0.", 64, "Packed Bed Inputs"
End Sub

Sub cmdRevisit_Click ()
  Dim Keypress
  Keypress = MsgBox("Do you wish to revisit earlier controls starting with Quench, without re-specifying inlet gas and particle characteristics ?", 32 + 1, "Re-Visit Earlier Controls")
  Select Case Keypress
  Case IDOK:
  Unload frmPackbed
  frmMain.mnuPackBed.Enabled = False

  frmMain.mnuQuench.Enabled = True
  frmMain.mnuQuench.Checked = False
  frmMain.mnuVenturi.Checked = False
  End Select

End Sub

Sub Form_Load ()
  lblIncr.Caption = Format$(Z)
  'Defines justification of the cells
  Call Centercells   'Center all cell alignments
  Call Sizecells     'Specify width of cells
  Call Titles        'Initialize col and row titles

End Sub

Sub Sizecells ()
  Dim Cols As Integer
  For Cols = 0 To 0
    grdPackbed.ColWidth(Cols) = 500 'Twips
  Next Cols
  For Cols = 1 To 2
    grdPackbed.ColWidth(Cols) = 1500 'Twips
```

```
    Next Cols
    For Cols = 3 To 3
      grdPackbed.ColWidth(Cols) = 1200 'Twips
    Next Cols
    For Cols = 4 To 4
      grdPackbed.ColWidth(Cols) = 1000 'Twips
    Next Cols
    For Cols = 5 To 5
      grdPackbed.ColWidth(Cols) = 1500 'Twips
    Next Cols

End Sub

Sub Titles ()
    'Fill in col titles
    grdPackbed.Row = 0
    '***********************************
    grdPackbed.Col = 0
    grdPackbed.Text = "No."
    grdPackbed.Col = 1
    grdPackbed.Text = "Lower Size(um)"
    grdPackbed.Col = 2
    grdPackbed.Text = "Upper Size(um)"
    grdPackbed.Col = 3
    grdPackbed.Text = "Dia(um)"
    grdPackbed.Col = 4
    grdPackbed.Text = "MassFrac In"
    grdPackbed.Col = 5
    grdPackbed.Text = "Frac Effy"
    '***********************************


End Sub
```

****

ESP Characteristics

```
Source: Frmesp.txt
Option Explicit

Sub cmdCalc_Click ()
    On Error GoTo cmdCalc_Exit
    'Inform user that ESP inputs not provided
    If (txtSection.Text = "") Then
    MsgBox "Please input ESP parameters before proceeding with Calculate !", 16,
"Warning !"
```

```
Else
If (txtDepth.Text = "") Then
MsgBox "Please input ESP parameters before proceeding with Calculate !", 16,
"Warning !"
Else
If (txtGasVel.Text = "") Then
MsgBox "Please input ESP parameters before proceeding with Calculate !", 16,
"Warning !"
Else
If (txtPrDrop.Text = "") Then
MsgBox "Please input ESP parameters before proceeding with Calculate !", 16,
"Warning !"
Else
If (txtRadius.Text = "") Then
MsgBox "Please input ESP parameters before proceeding with Calculate !", 16,
"Warning !"
Else
If (txtSN.Text = "") Then
MsgBox "Please input ESP parameters before proceeding with Calculate !", 16,
"Warning !"
Else
If (txtRR.Text = "") Then
MsgBox "Please input ESP parameters before proceeding with Calculate !", 16,
"Warning !"
Else
If (txtSpace.Text = "") Then
MsgBox "Please input ESP parameters before proceeding with Calculate !", 16,
"Warning !"
Else
If (txtHeight.Text = "") Then
MsgBox "Please input ESP parameters before proceeding with Calculate !", 16,
"Warning !"
Else
If (optField.Value = False And optDiff.Value = False) Then
MsgBox "Please input ESP parameters before proceeding with Calculate !", 16,
"Warning !"
Else
'*********************************************
Dim HPLATE, QWIREPLATE
Dim AREAPLATES, EPS
Dim E, VMAX, BB, EMAX, VCOR, ECOR
Dim ETIME, S, LL, M
Dim SYMPPB
'Global TEMPKE, SPRESSE, ABSPRESSESP, SCA, ESCA, DELPE
```

```
'Global VGESP, MPESP1(), ESPEFF(), TEFFESP, SYSESP
'Global NS, DSECT, L, RW, SN, RR

NS = Val(txtSection.Text)
L = Val(txtSpace.Text)
L = L / 100 'Convert to SI (cm to m)
HPLATE = Val(txtHeight.Text)   'meters
DSECT = Val(txtDepth.Text)     'meters
VGESP = Val(txtGasVel.Text)    'm/sec
DELPE = Val(txtPrDrop.Text)    'in H2O

'Flowrate between single wire and plate
QWIREPLATE = VGESP * HPLATE * L
'Area of plate or section assoc. w/ wire and plate
AREAPLATES = HPLATE * DSECT
'Overall SCA in s/m
SCA = NS * AREAPLATES / QWIREPLATE
'Overall SCA in English system - ft2/1000 acfm
ESCA = SCA * 5.08
EPS = .00000000000885
'Temp is same as that at outlet of venturi (?)
'Cooling effect in Packed bed (?)
TEMPKE = TEMPKV
SPRESSE = SPRESSV - DELPE
ABSPRESSESP = ABSPRESSV + SPRESSE / 13.6
'Calc of field strength and voltage
'Calculate operational field strength
'Oper. Field strength = SPARKOVER E/1.75
E = (630000#) * (((273 / TEMPKE) * (ABSPRESSESP / 29.92)) ^ 1.65) / 1.75
'Calculate operational voltage
VMAX = E * L
'Check voltage reqd to initiate corona to make sure that
'VMAX>VCOR
BB = (293 * ABSPRESSESP) / (TEMPKE * 29.92)
RW = Val(txtRadius.Text)   'meters
'Calculate E reqd to start corona (E at edge of wire)
EMAX = (3126000) * .6 * (BB + .03 * ((BB / RW) ^ .5))
VCOR = EMAX * RW * Log((4 * L) / (3.1417 * RW))
ECOR = VCOR / L
If (ECOR >= E) Then
   MsgBox "Operational Field Strength is LESS than that required to initiate a corona.
Please re-enter your ESP inputs.", 16, "Warning !"
Else
SN = Val(txtSN.Text)
```

```
   RR = Val(txtRR.Text)
   'Convert particle size distribution from um to meters
   ReDim DPSI(Z)
   For I = 1 To Z
      DPSI(I) = DP(I) * .000001
   Next
   'With Packed Bed By-Pass: Mass In MPV(I) from VENTURI
   'From Packed Bed        : Mass In MPPB(I) from Packed Bed
   'Change MPV(I) to MPESP(I) which is amount entering

   ReDim MPESP1(Z), MPESP(Z)

   For I = 1 To Z
      MPESP1(I) = MPPB(I) * 1
      MPESP(I) = MPESP1(I) * 1
   Next

   If (optField.Value = True) Then
      ETIME = 0
      GoTo ESPEFFIC
   Else
   If (optDiff.Value = True) Then
      ETIME = VGESP * DSECT
      GoTo ESPEFFIC
   End If
   End If

ESPEFFIC:
   'Algorithm ignores effect of Packed Bed on particle
   'removal (??). Results show effect......
   'Migration velocity - m/sec
   'Viscosity is same as in Venturi (VISQ in SI units)
   ReDim KK(Z), Q(Z), W(Z), PCQ(Z), PTESP(Z), PS(NS)
   For S = 1 To NS
      For I = 1 To Z
      'Calculate terms for saturation charge
      'Saturation Charge = Field + Diffusion Charging
      LL = 2 * (5.3 * 10 ^ -26) * 3.1417 * (1.38 * 10 ^ -23) * TEMPKE
      KK(I) = 1 + ((1.6 * 10 ^ -19) ^ 2) * (2 * 10 ^ 15) * ETIME * DPSI(I) / (2 * (8.85 *
10 ^ -12) * (LL ^ .5))
      Q(I) = 3 * 3.1417 * EPS * E * (DPSI(I) ^ 2) + (2 * 3.1417 * EPS * (1.38 * 10 ^ -
23) * TEMPKE * DPSI(I) / (1.6 * 10 ^ -19)) * Log(KK(I))
      W(I) = Q(I) * KC(I) * E / (3 * 3.1417 * VISQ * DPSI(I))
      PCQ(I) = Exp(-(SCA / NS) * W(I) / (1 - SN))
```

```
  PTESP(I) = SN + (1 - SN) * PCQ(I) + RR * (1 - SN) * (1 - PCQ(I))
  Next
  'Amount passed thru ESP
  For I = 1 To Z
    MPESP(I) = PTESP(I) * MPESP(I)
  Next
  'Overall PN (penetration) of Section at size I
  PS(S) = 0
  For I = 1 To Z
    PS(S) = PS(S) + MPESP(I)
  Next
Next
'Overall efficiency
'MPESP(I) = Frac mass passing thru at size I
'Total mass fraction = 1
'Therefore, System PN = Sum MPESP(I)
M = 0
For I = 1 To Z
  M = MPESP(I) + M
Next
'System fractional efficiency = 1-M
'TEFFESP: ESP System efficiency (fraction)
TEFFESP = (1 - M) * 1
'ESP Effy for each increment (for all sections combined)
ReDim ESPEFF(Z)
'Overall System effy: Need to a/c for all upstream controls
For I = 1 To Z
  ESPEFF(I) = 1 - (MPESP(I) / MPESP1(I))
Next

SYMPPB = SIGMPV2 * (1 - TEFFPB(Z))
SYSESP = SYSPB + (SYMPPB * TEFFESP)

MsgBox "ESP Calculations done successfully !", 48, "Attention"
frmESPSum.Show MODELESS
End If 'assoc. w/ Warning for ECOR>= E
'**********************************************
End If
End If
End If
End If
End If
End If
End If
```

```
   End If
   End If
   End If
   Exit Sub
   '******************************************
cmdCalc_Exit:
   MsgBox "An error occurred, please re-specify your inputs!", 16, "Warning!"
   txtSection.Text = ""
   txtDepth.Text = ""
   txtGasVel.Text = ""
   txtPrDrop.Text = ""
   txtRadius.Text = ""
   txtSN.Text = ""
   txtRR.Text = ""
   txtSpace.Text = ""
   txtHeight.Text = ""
   optField.Value = False
   optDiff.Value = False
   Exit Sub


End Sub
```

****

Source: Frmespsu.txt

```
Option Explicit

Sub Centercells ()
   'Sets the justification of the grid's cells to center
   Dim Cols As Integer
   For Cols = 0 To 5
      grdESP.Col = Cols
      grdESP.FixedAlignment(Cols) = 2
   Next Cols
   For Cols = 1 To 5
      grdESP.ColAlignment(Cols) = 2
   Next Cols

End Sub

Sub cmdContinue_Click ()
   Dim Keypress
   Keypress = MsgBox("ESP characteristics have been specified.  Your current session is
OVER.  Thank You.", 48 + 1, "Program END...")
   Select Case Keypress
```

```
   Case IDOK:
   Unload frmESP
   Unload frmESPSum
   frmMain.mnuESP.Checked = True
   frmMain.mnuESP.Enabled = False
   End Select

End Sub

Sub cmdDisplay_Click ()
   lblGastemp.Caption = Format$(TEMPKE, "####.#")
   lblSpr.Caption = Format$(SPRESSE, "###.##")
   lblAbspr.Caption = Format$(ABSPRESSESP, "###.##")
   lblSCA1.Caption = Format$(SCA, "###.#")
   lblSCA2.Caption = Format$(ESCA, "####.#")
   lblFlow.Caption = Format$(FLOWQ, "##.####")
   lblWater.Caption = Format$(H2OV, "###.##")
   lblPrDrop.Caption = Format$(DELPE, "###.#")
   lblGasVel.Caption = Format$(VGESP, "###.#")
   '**********************************************
   Dim Rows As Integer

   grdESP.Col = 0
   For Rows = 1 To Z
      grdESP.Row = Rows
      grdESP.Text = Format$(Rows)
   Next Rows

   grdESP.Col = 1
   For I = 1 To Z
      grdESP.Row = I
      grdESP.Text = Format$(DA(I), "0.####")
   Next I

   grdESP.Col = 2
   For I = 1 To Z
      grdESP.Row = I
      grdESP.Text = Format$(DA(I + 1), "0.####")
   Next I

   grdESP.Col = 3
   For I = 1 To Z
      grdESP.Row = I
      grdESP.Text = Format$(DP(I), "0.####")
```

```
   Next I

   grdESP.Col = 4
   For I = 1 To Z
      grdESP.Row = I
      grdESP.Text = Format$(MPESP1(I), "#.######")
   Next I

   grdESP.Col = 5
   For I = 1 To Z
      grdESP.Row = I
      grdESP.Text = Format$(ESPEFF(I), "#.######")
   Next I
   '*********************************************
   lblTeffESP.Caption = Format$(TEFFESP, "###.######%")
   lblSysESP.Caption = Format$(SYSESP, "###.######%")

End Sub

Sub cmdPrint_Click ()
   On Error GoTo cmdPrint_Exit
   Dim Keypress, RUNTIME
   Keypress = MsgBox("Print ESP characteristics to user-specified output file.", 48 + 1,
"Print To File")
   Select Case Keypress
   Case IDOK:
   '*********************************
   RUNTIME = Now
   'Open "a:\tsca.out" For Append As #1
   Open Drive For Append As #1
   Write #1,
   Write #1, "Visual Basic Program Output: ESP Characteristics"
   Write #1, "**************************************************"
   Write #1, "Current Session Filename :", Drive
   Write #1, "Date and Time of Run :", RUNTIME
   Write #1,
   Write #1, "No. of ESP Sections in Series:", Format$(NS, "###.")
   Write #1, "Depth of ESP Sections (m):", Format$(DSECT, "###.#")
   Write #1, "Wire-to-Plate Spacing (cm):", Format$(L, "##.##")
   Write #1, "Radius of Corona Wires (m):", Format$(RW, "#.######")
   Write #1, "Sneakage Around Plate Section (fraction):", Format$(SN, "#.###")
   Write #1, "Re-entrainment (fraction):", Format$(RR, "#.###")
   Write #1,
   If (frmESP.optField.Value = True) Then
```

```
    Write #1, "Specified Type of ESP Charging:", "Field Charging"
  Else
  If (frmESP.optDiff.Value = True) Then
    Write #1, "Specified Type of ESP Charging:", "Field + Diffusion Charging"
    End If
  End If
  Write #1,
  Write #1, "Gas Temperature (K):", Format$(TEMPKE, "####.#")
  Write #1, "Outlet Static Pressure (in H2O):", Format$(SPRESSE, "###.##")
  Write #1, "Outlet Absolute Pressure (in Hg):", Format$(ABSPRESSESP, "###.##")
  Write #1, "Specific Collection Area (sec/m):", Format$(SCA, "###.#")
  Write #1, "Specific Collection Area (ft2/1000 acfm):", Format$(ESCA, "####.#")
  Write #1, "Gas Flowrate (m3/sec):", Format$(FLOWQ, "##.####")
  Write #1, "Outlet Moisture Content (vol%):", Format$(H2OV, "###.##")
  Write #1, "ESP Pressure Drop (in H2O):", Format$(DELPE, "###.#")
  Write #1, "ESP Gas Velocity (m/sec):", Format$(VGESP, "###.#")
  Write #1,
  Write #1, "Total No. of Particle Size Increments:", Z
  Write #1,
  Write #1, "No.", "Lower Size (um)", "-", "Upper Size (um)", "Diameter (um)",
"MassFrac In", "Frac Effy"
  For I = 1 To Z
    Write #1, I, Format$(DA(I), "0.####"), "-", Format$(DA(I + 1), "0.####"),
Format$(DP(I), "0.####"), Format$(MPESP1(I), "#.######"), Format$(ESPEFF(I),
"#.######")
  Next
  Write #1,
  Write #1, "Overall ESP Efficiency:", Format$(TEFFESP, "###.######%")
  Write #1, "Overall System Efficiency:", Format$(SYSESP, "###.######%")
  Write #1,
  Write #1, "******************************************************"
  Write #1,
  Close #1
  '************************************
  End Select
  Exit Sub
  '************************************
cmdPrint_Exit:
  MsgBox "An error occurred! You have not specified output filename. Program will
prompt for new output filename. Next click the command button <Print To File> to print
output to new filename.", 16, "Warning!"
  Drive = InputBox$("Enter new drive:\filename e.g., c:\tsca.out and then hit <Enter>.
Next click <Print To File> button to print output to new filename.", "Print To File",
"c:\tsca.out")
```

```
   Exit Sub

End Sub

Sub cmdRevisit_Click ()
   Dim Keypress
   Keypress = MsgBox("Do you wish to revisit earlier controls starting with Quench, without re-specifying inlet gas and particle characteristics ?", 32 + 1, "Re-Visit Earlier Controls")
   Select Case Keypress
   Case IDOK:
   Unload frmESP
   Unload frmESPSum
   frmMain.mnuESP.Enabled = False

   frmMain.mnuQuench.Enabled = True
   frmMain.mnuQuench.Checked = False
   frmMain.mnuVenturi.Checked = False
   frmMain.mnuPackBed.Checked = False
   End Select

End Sub

Sub Form_Load ()
   lblIncr.Caption = Format$(Z)
   'Defines justification of the cells
   Call Centercells    'Center all cell alignments
   Call Sizecells      'Specify width of cells
   Call Titles         'Initialize col and row titles

End Sub

Sub Sizecells ()
   Dim Cols As Integer
   For Cols = 0 To 0
      grdESP.ColWidth(Cols) = 500 'Twips
   Next Cols
   For Cols = 1 To 2
      grdESP.ColWidth(Cols) = 1500 'Twips
   Next Cols
   For Cols = 3 To 3
      grdESP.ColWidth(Cols) = 1200 'Twips
   Next Cols
   For Cols = 4 To 4
```

```
      grdESP.ColWidth(Cols) = 1000 'Twips
   Next Cols
   For Cols = 5 To 5
      grdESP.ColWidth(Cols) = 1500 'Twips
   Next Cols

End Sub

Sub Titles ()
   'Fill in col titles
   grdESP.Row = 0
   '*****************************************
   grdESP.Col = 0
   grdESP.Text = "No."
   grdESP.Col = 1
   grdESP.Text = "Lower Size(um)"
   grdESP.Col = 2
   grdESP.Text = "Upper Size(um)"
   grdESP.Col = 3
   grdESP.Text = "Dia(um)"
   grdESP.Col = 4
   grdESP.Text = "MassFrac In"
   grdESP.Col = 5
   grdESP.Text = "Frac Effy"
   '*****************************************

End Sub
************************************************************************
```

Ionizing Wet Scrubber-1 Characteristics

Source: FrmIWS1.txt

```
Option Explicit

Sub Centercells ()
   'Sets justification of the grid's cells to center
   Dim Cols As Integer
   For Cols = 0 To 5
      grdIWS.Col = Cols
      grdIWS.FixedAlignment(Cols) = 2
   Next Cols
   For Cols = 1 To 5
      grdIWS.ColAlignment(Cols) = 2
   Next Cols

End Sub
```

```
Sub chkAnother_Click ()
  frmMain.mnuIWS2.Enabled = True
End Sub

Sub cmdCalc_Click ()
  If (lblIncr.Caption = "") Then
  MsgBox "Please input IWS parameters before proceeding with Calculate !", 16,
"Warning !"
  Else
  If (txtDepth.Text = "") Then
  MsgBox "Please input IWS parameters before proceeding with Calculate !", 16,
"Warning !"
  Else
  If (optType1.Value = False And optType2.Value = False And optType3.Value = False)
Then
  MsgBox "Please input IWS parameters before proceeding with Calculate !", 16,
"Warning !"
  Else
  '*********************************************
  'From Packed Bed: Mass In MPPB(I) from PB
  Dim SIGMPIWS
  ReDim DAERO(Z)
  ReDim MPPIWS(Z), IWSEFF1(Z), TEFFIWS1(Z)
  ReDim PEFFIWS(Z), IWSPTB(Z), IWSPTA(Z), IWSPT(Z)
  'Global DIWS1, SYMPPB, SYSIWS1, TEFFIWS1(), MPIWS1()
  'Global MPPIWS(), IWSEFF1(), SIGIWS1EFF

  For I = 1 To Z
    MPPIWS(I) = MPPB(I) * 1
  Next
  DIWS1 = Val(txtDepth.Text)

  'User choice of IWS equation Type:
  If (optType1.Value = True) Then GoTo 5030
  If (optType2.Value = True) Then GoTo 5031
  If (optType3.Value = True) Then GoTo 5032

5030 :
  '********* SCA=62.6 Equation *********
  For I = 1 To Z
    DAERO(I) = DP(I) * (ROP(1) ^ .5)
    If DAERO(I) <= 5 Then
```

```
    IWSPTB(I) = ((.006199 * DAERO(I) ^ 3) - (.060103 * DAERO(I) ^ 2) + (.011302
* DAERO(I)) + .77037) ^ (DIWS1 / 8)
    IWSPTA(I) = -.48 * (DP(I) ^ 2) + (3.07 * DP(I)) + 2.02
    IWSPT(I) = (Exp(-IWSPTA(I))) * IWSPTB(I)
    IWSEFF1(I) = 1 - IWSPT(I)
   Else
    IWSEFF1(I) = IWSEFF1(I - 1)
    End If
  Next
  GoTo 5040

5031 :
  '********* SCA=34.8 Equation *********
  For I = 1 To Z
    DAERO(I) = DP(I) * (ROP(1) ^ .5)
    If DAERO(I) <= 5 Then
    IWSPTB(I) = ((.006199 * DAERO(I) ^ 3) - (.060103 * DAERO(I) ^ 2) + (.011302
* DAERO(I)) + .77037) ^ (DIWS1 / 8)
    IWSPTA(I) = -.43 * (DP(I) ^ 2) + (2.6 * DP(I)) + 1.2
    IWSPT(I) = (Exp(-IWSPTA(I))) * IWSPTB(I)
    IWSEFF1(I) = 1 - IWSPT(I)
   Else
    IWSEFF1(I) = IWSEFF1(I - 1)
    End If
  Next
  GoTo 5040

5032 :
  '********* SCA=28.5 Equation *********
  For I = 1 To Z
    DAERO(I) = DP(I) * (ROP(1) ^ .5)
    If DAERO(I) <= 5 Then
    IWSPTB(I) = ((.006199 * DAERO(I) ^ 3) - (.060103 * DAERO(I) ^ 2) + (.011302
* DAERO(I)) + .77037) ^ (DIWS1 / 8)
    IWSPTA(I) = -.66 * (DP(I) ^ 2) + (2.93 * DP(I)) + .74
    IWSPT(I) = (Exp(-IWSPTA(I))) * IWSPTB(I)
    IWSEFF1(I) = 1 - IWSPT(I)
   Else
    IWSEFF1(I) = IWSEFF1(I - 1)
    End If
  Next
  GoTo 5040

5040 :
```

```
'Overall Effy is summation of MPPIWS(I)*IWSEFF1(I)/1
ReDim MPIWSA(Z), MPIWSB(Z), MPIWS1(Z)

'Mass Penetrating IWS-1: MPIWS1(I)
TEFFIWS1(0) = 0
MPIWSB(0) = 0

For I = 1 To Z
   PEFFIWS(I) = MPPIWS(I) * IWSEFF1(I)
   TEFFIWS1(I) = TEFFIWS1(I - 1) + PEFFIWS(I)
   MPIWSA(I) = MPPIWS(I) * (1 - IWSEFF1(I))
   MPIWSB(I) = MPIWSB(I - 1) + MPIWSA(I)
Next

'Sum of mass fractions penetrating IWS-1: MPIWSB(Z) n.e. 1 anymore
'Adjust mass fractions to reflect sum(MPIWS1(I)) =1
'Mass penetrating IWS-1: MPIWS1(I)
SIGMPIWS = MPIWSB(Z) * 1
For I = 1 To Z
   MPIWS1(I) = MPIWSA(I) / SIGMPIWS
Next

SIGIWS1EFF = TEFFIWS1(Z) * 1
'Overall system effy: Need to a/c for all upstream controls
SYMPPB = SIGMPV2 * (1 - TEFFPB(Z))
SYSIWS1 = SYSPB + (SYMPPB * TEFFIWS1(Z))
'*********************************************
'Outputs on same screen
Dim Rows As Integer

grdIWS.Col = 0
For Rows = 1 To Z
   grdIWS.Row = Rows
   grdIWS.Text = Format$(Rows)
Next Rows

grdIWS.Col = 1
For I = 1 To Z
   grdIWS.Row = I
   grdIWS.Text = Format$(DA(I), "0.####")
Next I

grdIWS.Col = 2
For I = 1 To Z
```

```
      grdIWS.Row = I
      grdIWS.Text = Format$(DA(I + 1), "0.####")
   Next I

   grdIWS.Col = 3
   For I = 1 To Z
      grdIWS.Row = I
      grdIWS.Text = Format$(DP(I), "0.####")
   Next I

   grdIWS.Col = 4
   For I = 1 To Z
      grdIWS.Row = I
      grdIWS.Text = Format$(MPPIWS(I), "#.######")
   Next I

   grdIWS.Col = 5
   For I = 1 To Z
      grdIWS.Row = I
      grdIWS.Text = Format$(IWSEFF1(I), "#.######")
   Next I

   lblSIGIWSEFF.Caption = Format$(SIGIWS1EFF, "###.######%")
   lblSYSIWS1.Caption = Format$(SYSIWS1, "###.######%")
   '*********************************************
   End If
   End If
   End If

End Sub

Sub cmdContinue_Click ()
   Dim Keypress
   Keypress = MsgBox("IWS characteristics have been specified.  Please proceed to select
EITHER Additional IWS (under Controls) from Menu OR Exit Program.", 48 + 1, "End
IWS-1")
   Select Case Keypress
   Case IDOK:
   Unload frmIWS1
   frmMain.mnuIWS1.Checked = True
   frmMain.mnuIWS1.Enabled = False
   End Select

End Sub
```

```
Sub cmdPrint_Click ()
   On Error GoTo cmdPrint_Exit
   Dim Keypress, RUNTIME
   Keypress = MsgBox("Print IWS-1 characteristics to user-specified output file.", 48 + 1,
"Print To File")
   Select Case Keypress
   Case IDOK:
   '***********************************
   RUNTIME = Now
   'Open "a:\tsca.out" For Append As #1
   Open Drive For Append As #1
   Write #1,
   Write #1, "Visual Basic Program Output: IWS-1 Characteristics"
   Write #1, "****************************************************"
   Write #1, "Current Session Filename :", Drive
   Write #1, "Date and Time of Run :", RUNTIME
   Write #1,
   Write #1, "Total No. of Particle Size Increments:", Z
   If (frmIWS1.optType1) Then
   Write #1, "Type of SCA Equation Selected:", "SCA=62.6 FT2/1000 ACFM"
   Else
   If (frmIWS1.optType2) Then
   Write #1, "Type of SCA Equation Selected:", "SCA=34.8 FT2/1000 ACFM"
   Else
   If (frmIWS1.optType3) Then
   Write #1, "Type of SCA Equation Selected:", "SCA=28.5 FT2/1000 ACFM"
   Else
   End If
   End If
   End If
   Write #1,
   Write #1, "Total Depth of Wetted Packing (feet):", Format$(DIWS1, "###.#")
   Write #1,
   Write #1, "No.", "Lower Size (um)", "-", "Upper Size (um)", "Diameter (um)",
"MassFrac In", "Frac Effy"
   For I = 1 To Z
      Write #1, I, Format$(DA(I), "0.####"), "-", Format$(DA(I + 1), "0.####"),
Format$(DP(I), "0.####"), Format$(MPPIWS(I), "#.######"), Format$(IWSEFF1(I),
"#.######")
   Next
   Write #1,
   Write #1, "Overall IWS-1 Efficiency:", Format$(SIGIWS1EFF, "###.######%")
   Write #1, "Overall System Efficiency:", Format$(SYSIWS1, "###.######%")
```

```
  Write #1,
  Write #1, "******************************************************"
  Write #1,
  Close #1
  '************************************
  End Select
  Exit Sub
  '***********************************
cmdPrint_Exit:
  MsgBox "An error occurred! You have not specified output filename. Program will
prompt for new output filename. Next click the command button <Print To File> to print
output to new filename.", 16, "Warning!"
  Drive = InputBox$("Enter new drive:\filename e.g., c:\tsca.out and then hit <Enter>.
Next click <Print To File> button to print output to new filename.", "Print To File",
"c:\tsca.out")
  Exit Sub

End Sub

Sub cmdReadme_Click ()
  MsgBox "IWS calculations utilize the Specific Collection Area (SCA)-specific empirical
equations developed at UTK under this contract.  Three equations were developed to
characterize IWS behavior.  The IWS consists of an IONIZER and a PACKED BED in
series.", 64, "IWS Inputs"
End Sub

Sub cmdRevisit_Click ()
  Dim Keypress
  Keypress = MsgBox("Do you wish to revisit earlier controls starting with Quench,
without re-specifying inlet gas and particle characteristics ?", 32 + 1, "Re-Visit Earlier
Controls")
  Select Case Keypress
  Case IDOK:
  Unload frmIWS1
  frmMain.mnuIWS1.Enabled = False

  frmMain.mnuQuench.Enabled = True
  frmMain.mnuQuench.Checked = False
  frmMain.mnuVenturi.Checked = False
  frmMain.mnuPackBed.Checked = False
  frmMain.mnuESP.Checked = False
  End Select

End Sub
```

```
Sub Form_Load ()
  lblIncr.Caption = Format$(Z)
  'Defines justification of the cells
  Call Centercells   'Center all cell alignments
  Call Sizecells     'Specify width of cells
  Call Titles        'Initialize col and row titles

End Sub

Sub Sizecells ()
  Dim Cols As Integer
  For Cols = 0 To 0
    grdIWS.ColWidth(Cols) = 500 'Twips
  Next Cols
  For Cols = 1 To 2
    grdIWS.ColWidth(Cols) = 1500 'Twips
  Next Cols
  For Cols = 3 To 3
    grdIWS.ColWidth(Cols) = 1200 'Twips
  Next Cols
  For Cols = 4 To 4
    grdIWS.ColWidth(Cols) = 1000 'Twips
  Next Cols
  For Cols = 5 To 5
    grdIWS.ColWidth(Cols) = 1500 'Twips
  Next Cols

End Sub

Sub Titles ()
  'Fill in col titles
  grdIWS.Row = 0
  '*********************************
  grdIWS.Col = 0
  grdIWS.Text = "No."
  grdIWS.Col = 1
  grdIWS.Text = "Lower Size(um)"
  grdIWS.Col = 2
  grdIWS.Text = "Upper Size(um)"
  grdIWS.Col = 3
  grdIWS.Text = "Dia(um)"
  grdIWS.Col = 4
  grdIWS.Text = "MassFrac In"
```

```
    grdIWS.Col = 5
    grdIWS.Text = "Frac Effy"
    '***********************************

End Sub
'**********************************************************************
```

APPENDIX B

CALCULATION CHECK FOR VISUAL BASIC APPLICATION ALGORITHMS

Calculation Check for Visual Basic Application Algorithms

Inlet Gas Characteristics

This section provides a calculation check for inlet gas characteristics based on specified user inputs. The variable nomenclature used will be similar to that used earlier in the text in Section 3.1. In order to verify the calculations within the application, elements from the test design case calculations presented in Chapter 4.0 have been utilized. The user provided the following inputs:

1. Inlet gas temperature, TEMPF = 1500 °F
2. Inlet Gas Molecular Weight, MW = 28.96 g/gmol
3. Inlet Gas Static Pressure, SPRESS = 0 in H2O
4. Barometric Pressure, BPRESS = 29.92 in Hg
5. Inlet Gas Flowrate, ACFM = 18,000 acfm
6. Inlet Gas Moisture Content, H2O = 10% vol

Convert degrees Fahrenheit to Kelvin:
Gas Temperature, TEMPK = ((TEMPF - 32)/1.8) + 273° = 1088.6 K
In this case, the user has already specified the inlet gas molecular weight, MW. The alternative calculation via volume fractions has been checked against hand calculations.
Absolute Pressure, ABSPRESS = 29.92 + SPRESS/13.6 = 29.92 in Hg
Gas Density, ROG = (MW/22.4)*(273/TEMPK)*(ABSPRESS/29.92) kg/m^3
ROG = (28.96/22.4)*(273/1088.6)*(29.92/29.92) = 0.3242 kg/m^3
Gas Viscosity, VIS = (exp(((Ln(TEMPK) - 18.0075)/1.431)))*0.1 kg/sec-m
VIS = (exp(((Ln(1088.6) - 18.0075)/1.431)))*0.1 = 4.54e-5 kg/sec-m
Gas Flowrate, FLOW = (ACFM * 0.0283)/60 = 8.4951 m^3/sec
Vapor Pressure of water vapor in gas, PW = ABSPRESS*H2O/100 = 2.992 in Hg
Dry Molecular Weight, MWD = (MW-18*(H2O/100))/(1-(H20/100)) = 30.18 g/gmol
Molar ratio of dry water/dry gas, HV = H2O/(100-H2O) = 0.1111
Mass ratio of water to dry air, HM = HV*18/MWD = 0.1111*18/30.18 = 6.63e-2
The calculation of the wet bulb temperature involves the iterative solution of the Carrier equation, which cannot be readily duplicated by hand calculations. However, based on the application-derived value of TWB= 349.5K (wet bulb temperature), the value of CK (vapor pressure of water vapor in gas) will be compared against PW=2.992 in Hg.
PWB = exp ((-5800.22/TWB) + 1.39149 - 0.048602 * TWB + (0.417647 * 10^{-4}) * (TWB^2) - (0.1445 * 10^{-7}) * (TWB^3) + 6.5459 * Ln(TWB)) [Pa]
PWB = 41,361 Pascals
PWB = PWB * 2.95 * 10^{-4} [Conversion of PWB to inches of Hg]
PWB = 12.20 in Hg
TWBF = 1.8 * (TWB -273) + 32 = 169.7°F
CK = PWB - ((ABSPRESS - PWB) * (TEMPF-TWBF)/(2830 - 1.3 * TWBF))
CK= 12.2 -((29.92-12.2)*(1500-169.7)/(2830-1.3*169.7)) = 3.17 in Hg
Ignoring rounding errors, CK = PWB ≈ 3 in Hg

Inlet Particle Characteristics

This section provides a calculation check for inlet particle characteristics based on specified user inputs. The variable nomenclature used will be similar to that used earlier in the text in Section 3.2. In order to verify the calculations within the application, elements from the test design case calculations presented in Chapter 4.0 have been utilized. The user provided the following inputs:

1. Specified Particle Size Distribution: Log-Normal Distribution
2. Smallest Particle Size, DMIN = 0.25 μm
3. Largest Particle Size, DMAX = 4 μm
4. Width of Size Increments, WI = 0.05 μm
5. Mass Mean Diameter, MMD = 1 μm
6. Geometric Standard Deviation, SIGMAG = 2
7. Inlet Dust Loading, GPERACM = 1 g/m^3
8. Particle Density, ROP = 2.5 g/cm^3

Number of Particle Size Increments, Z = (DMAX - DMIN)/WI = 75
Due to the large number of particle size increments, the entire log-normal distribution cannot be readily duplicated by hand calculations. For the purposes of calculation verification, only the first two size increments will be considered. Calculate the bounds of the size increments, DA(I) - DA(I+1):
DA(1) = DMIN = 0.25 μm
DA(I) = DMIN + (I - 1)*WI
DA(2) = 0.25 + (2 - 1)*0.05 = 0.30 μm
DA(3) = 0.25 + (3 - 1)*0.05 = 0.35 μm
The first two size intervals will be 0.25 - 0.30 and 0.30 - 0.35 μm.
Calculate the geometric mean diameter DP(I) for the respective size intervals.
DP(I) = Sqr(DA(I) * DA(I+1))
∴ DP(1) = Sqr(0.25 * 0.30) = 0.2738μm, and DP(2) = Sqr(0.30 * 0.35) = 0.3240 μm.
Similarly, the application calculates the size increment bounds and geometric mean diameters for 75 size intervals.
Calculate the mass fraction MI(I) for the respective size intervals. The application logic is reiterated below for quick reference:

```
For I = 1 To (Z + 1)
X(I) = (Log(DA(I) / (MMD(J)))) / (1.414 * Log(SIGMAG(J)))
ASX(I) = X(I)
If ASX(I) < 0 Then ASX(I) = -1 * ASX(I)
A = 1: ZZ = 0:
For B = 0 To 25
If B = 0 GoTo E1
A = A * B
E1:
```

```
XX = ((2) / (3.1417 ^ .5)) * ((-1) ^ B) * ((ASX(I)) ^ (2 * B + 1)) / (A * (2 * B + 1))
ZZ = ZZ + XX
Next
ERFXDP(I) = ZZ
If X(I) < 0 Then ERFXDP(I) = -1 * ERFXDP(I)
MLESSDP(I) = .5 + .5 * ERFXDP(I)
Next
Next

For I = 2 To (Z - 1)
MI(I) = MLESSDP(I + 1) - MLESSDP(I)
Next
MI(1) = MLESSDP(2)
MI(Z) = 1 - MLESSDP(Z)
```

The numerical calculation proceeds as follows:

X(1) = ln(0.25/1)/(1.414*ln(2)) = -1.4144
X(2) = ln(0.30/1)/(1.414*ln(2)) = -1.2284
X(3) = ln(0.35/1)/(1.414*ln(2)) = -1.0711
$\therefore$ ASX(1) = 1.4144, ASX(2) = 1.2284, ASX(3) = 1.0711
For every ASX(I) value, ERFXDP(I) was calculated:
ERFXDP(1) = -0.9545, ERFXDP(2) = -0.9176 and ERFXDP(3) = -0.8701
MLESSDP(1) = 0.0227, MLESSDP(2) = 0.0412 and MLESSDP(3) = 0.0649
Thus, MI(1) = MLESSDP(2) = 0.0412 and
MI(2) = MLESSDP(3) - MLESSDP(2) = (0.0649 - 0.0412) = 0.0237

Ignoring rounding errors, these two mass fractions account for the first two size intervals for the log-normal particle size distribution. Note that for the mass fraction input in size increments, no such calculations are necessary. In that case, only Z, DA(I) are calculated as shown above. The user inputs the actual mass fraction data corresponding to the particle size distribution.

Quench Tower Characteristics

This section provides a calculation check for Quench Tower characteristics based on specified user inputs. The variable nomenclature used will be similar to that used earlier in the text in Section 3.3. In order to verify the calculations within the application, elements from the test design case calculations presented in Chapter 4.0 have been utilized. The user provided the following inputs:

1. Actual Water Injection Rate, WI = 0.9 kg/sec
2. Pressure Drop Across Quench, DELPQ = 0 in H2O

Before the user enters the above inputs, the application calculates the amount of water required for saturation, WIT as follows:

Moisture % at saturation, MCSAT = (PWB/ABSPRESS)*100

MCSAT = (12.2/29.92)*100 = 40.78% vol

Mole fraction of water/dry air, HVSAT = MCSAT/(100 - MCSAT)

HVSAT = 40.78/(100 - 40.78) = 0.6886

Mass fraction of water/dry air, HMSAT = HVSAT *18/ MWD

HMSAT = 0.6886 * 18/30.18 = 0.4107

Mass of gas flow, MGAS = FLOW * ROG = 8.4951 m^3/sec * 0.3242 kg/m^3

MGAS = 2.7541 kg/sec

Mass fraction of water/gas before injection, MFRACH2O = HM/(1+HM)

MFRACH2O = 6.63e-2 / (1 + 6.63e-2) = 0.0622

Mass of inlet water, MH2O = MGAS * MFRACH2O = 2.7541*0.0622 = 0.1713 kg/sec

Mass of dry air, MAIR = MGAS*(1 - MFRACH2O)

MAIR = 2.7541*(1-0.0622) = 2.5828 kg/sec

∴ WIT = (HMSAT * MAIR) - MH2O = (0.4107*2.5828)-0.1713 = 0.8894 kg/sec

In response to the above value, the user entered 0.9 kg/sec as the actual water injection rate. The remainder of the quench calculations are as follows:

Mass of water entering quench, MH2OQ = MH2O + WIT

MH2OQ = (0.1713 + 0.8894) = 1.0607 kg/sec.

Mass of water leaving quench as carryover, WNETQ = WI - WIT

WNETQ = (0.9 - 0.8894) = 0.0106 kg/sec

Quench outlet gas temperature, TEMPKQ = TWB = 349.5 K

In degrees Fahrenheit, gas temperature, TEMPFQ = 169.7 °F

Wet molecular weight, MWQ = MWD*(1-(MCSAT/100))+18*(MCSAT/100)

MWQ = 30.18 * (1- (40.78/100)) + 18*(40.78/100) = 25.21 g/gmol

Outlet Static Pressure, SPRESSQ = (SPRESS - DELPQ) = 0 in H2O

Absolute Pressure, ABSPRESSQ = ABSPRESS + SPRESSQ/13.6 = 29.92 in Hg

Gas Density, ROGQ = (MWQ/22.4)*(273/TEMPKQ)*(ABSPRESSQ/29.92)

ROGQ = (25.21/22.4)*(273/349.5)*(29.92/29.92) = 0.8791 kg/m^3

Outlet moisture content, H2OQ = MCSAT = 40.78%

Outlet gas flowrate, FLOWQ = (MAIR + MH2O + WIT)/ROGQ

FLOWQ = (2.5828 + 0.1713 + 0.8894)/0.8791 = 4.1446 m^3/sec

The outlet gas viscosity calculation is as follows:

At TEMPKQ = 349.5K, the viscosity of air and water vapor are given by the relationships

VISAIR = 4.06×10^{-8} * TEMPKQ + 6.63×10^{-6} kg/m-sec

VISAIR = 2.08×10^{-5} kg/m-sec

VISH2O = 3.82×10^{-8} * TEMPKQ - 1.70×10^{-6} kg/m-sec

VISH2O = 1.16×10^{-5} kg/m-sec

Mole fraction of dry air, YAIR = (100 - H2OQ)/100 = (100-40.78)/100 = 0.5922

Mole fraction of water, YH2O = H2OQ/100 = 0.4078

VISQ1 = (YAIR * VISAIR * Sqr(MWD)) + (YH2O * VISH2O * Sqr(18))

VISQ1 = (0.5922 * 2.08×10^{-5} * Sqr(30.18)) + (0.4078 * 1.16×10^{-5} * Sqr(18))

VISQ1 = 8.68×10^{-5}

VISQ2 = (YAIR * Sqr(MWD)) + (YH2O * Sqr(18))
VISQ2 = (0.5922 * Sqr(30.18)) + (0.4078 * Sqr(18))
VISQ2 = 4.9835
VISQ = VISQ1/VISQ2 = 8.68x10-5 / 4.9835 = 1.74×10^{-5} kg/sec-m

Venturi Scrubber Characteristics

This section provides a calculation check for Venturi Scrubber characteristics based on specified user inputs. The variable nomenclature used will be similar to that used earlier in the text in Section 3.4. In order to verify the calculations within the application, elements from the test design case calculations presented in Chapter 4.0 have been utilized. The user provided the following inputs:

1. Liquid Addition Rate, QL = 132 gal/min
2. Venturi Pressure Drop, PR = 10 in H2O
3. Gas Velocity in Throat,VG = 0 ft/min [since, venturi pressure drop has been specified]
4. F-Factor, F = 0.25 [default]
5. Liquid Density, RL = 1 g/cm^3 [water]

The application converts the pressure drop PR (in H2O) to cm H2O:
PR = 10 * 2.54 = 25.4 cm H2O
The liquid addition rate QL is converted to g/sec or cm^3/sec [equivalent for water]:
QL = 132 * 3783/60 = 8322.6 cm^3/sec
The flowrate FLOWQ (which displays on top of the screen) is converted to cm^3/sec:
FLOWV = FLOWQ * 10^6 = 4.1446×10^6 cm^3/sec
Temperature and moisture content in the venturi is the same as in the quench outlet:
TEMPKV = TEMPKQ = 349.5 K
H2OV = H2OQ = 40.78% vol
Convert unchanged gas viscosity from quench outlet to cgs units:
VISV = VISQ * 10 = 1.74×10^{-5} x 10 = 1.74×10^{-4} g/sec-cm
Since pressure drop PR was specified, the application calculates the throat velocity, VG:
VG = $((PR*FLOWV)/(0.00103*QL))^{0.5}$ cm/sec
VG = $((25.4 * 4.1446\times10^6)/(0.00103 * 8322.6))^{0.5}$ = 3504 cm/sec
Application reconverts pressure drop PR to fps units:
DEPLV = PR/2.54 = 25.4/2.54 = 10 in H2O
Venturi Static Pressure, SPRESSV = SPRESSQ - DELPV in H2O
SPRESSV = (0 - 10) = -10 in H2O
Venturi Absolute Pressure, ABSPRESSV = ABSPRESSQ + SPRESSV/13.6
ABSPRESSV = 29.92 + (-10)/13.6 = 29.18 in Hg

Calculate collector droplet diameter DD with the Nukiyama-Tanasawa equation. First the application calculates the water surface tension SFCT (dynes/cm) and viscosity LVIS (poise) terms:
SFCT = -0.1676 * TEMPKV + 121.81 dynes/cm

SFCT = -0.1676 * 349.5 + 121.81 = 63.23 dynes/cm
LVIS = 2e+12 * $TEMPKV^{-5.8009}$ poise
LVIS = 2e+12 * $349.5^{-5.8009}$ = 0.00352 poise [g/cm-sec]
The Nukiyama-Tanasawa equation is given by:
DD = $58{,}600/V_G * (\sigma/\rho_L)^{0.5} + 597 * (\mu_L/(\sigma/\rho_L)^{0.5})^{0.45} * (1000 * QL/FLOWV)^{1.5}$ [in μm]
Substitute VG = 3504 cm/sec, σ = SFCT = 63.23 dynes/cm, ρ_L = RL = 1 g/cc, μ_L = LVIS = 0.00352 poise, QL = 8322.6 cc/sec and FLOWV = 4.1446×10^6 cc/sec
∴ DD = 185.56 μm = 185.56 μm /10000 = 0.0186 cm

Next, the application calculates the Cunningham correction factor, KC(I) for every DP(I). For calculation verification purposes, only the first two size intervals, i.e., DP(1) and DP(2) will be considered. From the earlier inlet particle characteristics calculations,
DP(1) = 0.2738 μm and DP(2) = 0.3240 μm.
Now Universal Gas Constant, RG = 8314.3 J/kgmol-K
Mean Thermal Velocity of Gas Molecules, TV = $((8*RG*TEMPKV/(3.1415*MWQ)))^{0.5}$
TV = $((8*8314.3*349.5/(3.1415*25.21)))^{0.5}$ = 541.78 m/sec
Mean Free Path of Gas Molecules, LAMBDA = (VISV/10)/(0.499*ROGQ*TV) m
LAMBDA = $(1.74 \times 10^{-4}/10)/(0.499*0.8791*541.78)$ = 7.3×10^{-8} m
SIM1(I) = 1.2570 + 0.4*exp(-0.55*DP(I)*0.000001)/LAMBDA)
KC(I) = 1 + 2*LAMBDA*SIM1(I)/(DP(I)*0.000001)
Now SIM1(1) = 1.3078 and SIM1(2) = 1.2918
And KC(1) = 1.6703 and KC(2) = 1.5821

Finally, the application calculates the particle collection efficiency with the Calvert equations:
Calculate the inertial impaction efficiency parameter, SI(I)
SI(I) = $VG*ROP*KC(I)*((DP(I)*10^{-4})^2)/(9*VISV*DD)$
SI(1) = $3504*2.5*KC(1)*((DP(1)*10^{-4})^2)/(9*1.74 \times 10^{-4}*0.0186)$
SI(1) = 0.3766 and SI(2) = 0.4995
Calculate the dimensionless Calvert efficiency paramater, FK(I)
FK(I) = (-0.7 - SI(I)*F + 1.4*ln((SI(I)*F+0.7)/0.7) + (0.49/(0.7+SI(I)*F)))/SI(I)
FK(1) = -0.001246 and FK(2) = -0.002069
The particle penetration PT(I) and collection efficiency EFF(I) is given by:
PT(I) = exp((2*QL*VG*RL*DD*FK(I))/(55*FLOWV*VISV))
EFF(I) = 1 - PT(I)
PT(1) = $\exp((2*8322.6*3504*1*0.0186*FK(1))/(55*4.1446 \times 10^6*1.74 \times 10^{-4}))$
PT(1) = 0.96649
∴EFF(1) = 1 - PT(1) = 1 - 0.96649 = 0.0335*100 = 3.35%
PT(2) = $\exp((2*8322.6*3504*1*0.0186*FK(2))/(55*4.1446 \times 10^6*1.74 \times 10^{-4}))$
PT(2) = 0.94498
∴EFF(2) = 1 - PT(2) = 1 - 0.94498 = 0.055*100 = 5.50%
Within rounding errors, the above calculations support the application output. The above calculations are repeated to generate overall venturi efficiency which cannot be readily checked with hand calculations. However, the algorithm has been verified otherwise.

Packed Bed Characteristics

This section provides a calculation check for Packed Bed characteristics based on specified user inputs. The variable nomenclature used will be similar to that used earlier in the text in Section 3.5. In order to verify the calculations within the application, elements from the test design case calculations presented in Chapter 4.0 have been utilized. The user provided the following inputs:

1. Total Depth of Wetted Packing, DPB = 5 feet

In this case, the calculations are straightforward. The Packed Bed empirical prediction equation is of the form:

Aerodynamic Diameter, $DAERO(I) = DP(I)^{*}\ ROP^{0.5}$

Fractional Efficiency, $PBEFF(I) = 1 - (((0.006199 * DAERO(I)^3) - (0.060103 * DAERO(I)^2) + (0.011302 * DAERO(I)) + 0.77037)^{(DPB/8)}$

For calculation verification purposes, only the first two size intervals, i.e., DP(1) and DP(2) will be considered. From the earlier inlet particle characteristics calculations,

DP(1) = 0.2738 μm and DP(2) = 0.3240 μm.

$DAERO(1) = DP(I)^{*}\ ROP^{0.5}$

DAERO(1) = 0.4329 μm and DAERO(2) = 0.5123 μm

$PBEFF(I) = 1 - (((0.006199 * DAERO(I)^3) - (0.060103 * DAERO(I)^2) + (0.011302 * DAERO(I)) + 0.77037)^{(DPB/8)}$

PBEFF(1) = 0.1545*100 = 15.45% and PBEFF(2) = 0.1568*100 = 15.68%

Within rounding errors, the above calculations support the application output. The above calculations are repeated to generate overall Packed Bed efficiency and overall System efficiency which cannot be readily checked with hand calculations. However, the algorithm has been verified otherwise. Also note that the Packed Bed empirical equation is valid in the range $0.12 \leq DAERO(I) \leq 5$ μm. The application imposes an artificial efficiency prediction curb on particle sizes greater than DAERO(I) = 5 μm by reporting a constant fractional efficiency value equal to the efficiency at 5 μm for aerodynamic particle diameters greater than 5 μm which, in effect, translates to a more conservative overall Packed Bed efficiency estimate.

ESP Characteristics

This section provides a calculation check for ESP characteristics based on specified user inputs. The variable nomenclature used will be similar to that used earlier in the text in Section 3.6. In order to verify the calculations within the application, elements from the test design case calculations presented in Chapter 4.0 have been utilized. The user provided the following inputs:

1. No. of ESP Sections in Series, NS = 2

2. Depth of ESP Sections, DSECT = 3 m
3. Wire-to-Plate Spacing, L = 12 cm = 0.12 m
4. Inlet Gas Velocity, VGESP = 2.5 m/sec
5. Height of ESP Plates, HPLATE = 3 m
6. Pressure Drop Across ESP, DELPE = 1.5 in H2O
7. Radius of Corona Wires, RW = 0.00127 m
8. Sneakage, SN = 0
9. Re-entrainment, RR = 0
10. Type of Particle Charging : Field + Diffusion Charging

Calculate flowrate between single wire and plate.
QWIREPLATE = VGESP * HPLATE * L m^3/sec
QWIREPLATE = (2.5 * 3 * 0.12) = 0.90 m^3/sec
Calculate area of plate or section associated with wire and plate.
AREAPLATES = HPLATE * DSECT = 3*3 = 9 m^2
Calculate overall SCA (Specific Collection Area) in sec/m.
SCA = NS * AREAPLATES/ QWIREPLATE
SCA = 2 * 9/ 0.90 = 20 sec/m
Calculate overall SCA in English units - ft^2/1000 acfm.
ESCA = SCA * 5.08 = 20 * 5.08 = 101.6 ft^2/1000 acfm
Gas Temperature in ESP is same as that at the outlet of venturi,
TEMPKE = TEMPKV = 349.5 K
Static Pressure, SPRESSE = SPRESSV - DELPE = -10 - 1.5 = -11.5 in H2O
Absolute Pressure, ABSPRESSESP = ABSPRESSV+SPRESSE/13.6
ABSPRESSESP = 29.18 + (-11.5)/13.6 = 28.33 in Hg
Calculate actual operational field strength, E (volts/m)
E = $6.3\text{x}10^5$ * (((273/TEMPKE)* (ABSPRESSESP/29.92))$^{1.65}$)/1.75
E = $6.3\text{x}10^5$ * (((273/349.5)* (28.33/29.92))$^{1.65}$)/1.75 = 218,854 volts/m
Calculate operational voltage, VMAX (volts)
VMAX = E * L = 218,854 * 0.12 = 26,262 volts = 26.26 KV
Check voltage required to initiate corona, VCOR such that VMAX > VCOR.
Calculate field strength to initiate corona, EMAX.
BB = (293 * ABSPRESSESP)/ (TEMPKE * 29.92) = 0.7938
EMAX = $3.126\text{x}10^6$ * 0.6 * (BB + 0.03 * ((BB/RW)$^{0.5}$))
EMAX = $3.126\text{x}10^6$ * 0.6 * (0.7938 + 0.03 * ((0.7938/0.00127)$^{0.5}$)) = $2.89\text{x}10^6$ volts/m
Calculate voltage required to start corona, VCOR (volts)
VCOR = EMAX * RW * ln((4 * L)/(3.1417 * RW)) volts
VCOR = $2.89\text{x}10^6$ * 0.00127 * ln((4 * 0.12)/(3.1417 * 0.00127)) volts
VCOR = 17,581 volts = 17.58 KV
ECOR = VCOR /L = 17,581/0.12 = 146,506 volts/m.
Since, ECOR < E, therefore ok.
Convert DP(I) in μm to DPSI(I) in meters for ESP calculations.
DPSI(1) = 0.2738 * 10^{-6} = $2.73\text{x}10^{-7}$ m
DPSI(2) = 0.3240 * 10^{-6} = $3.24\text{x}10^{-7}$ m

The user selected Field + Diffusion type of particle charging.
ETIME = VGESP * DSECT = 2.5 * 3 = 7.5 m^2/sec
Calculate terms for saturation charge and ESP efficiency:
EPS = 8.85×10^{-12}
LL = 2*(5.3×10^{-26})*3.1417*(1.38×10^{-23})*TEMPKE = 1.61×10^{-45}
KK(I) = 1 + $(1.6 \times 10^{-19})^2$ * 2×10^{15} * ETIME * DPSI(I)/ (2*(8.85×10^{-12})*($LL^{0.5}$))
Q(I) =3 * 3.1417 * EPS * E * DPSI(I)^2 + (2 * 3.1417 * EPS * 1.38×10^{-23} * TEMPKE * DPSI(I)/ (1.6×10^{-19})) * ln(KK(I))
WI(I) = Q(I) * KC(I) * E / (3 * 3.1417 * VISQ * DPSI(I))
PCQ(I) = exp(-(SCA/NS) * WI(I) / (1 - SN))
Penetration, PTESP(I) = SN + (1 - SN) * PCQ(I) + RR * (1 - SN) * (1 - PCQ(I))
ESP Efficiency, EFFESP(I) = 1 - PTESP(I)

For calculation verification purposes, only the first two size intervals, i.e., DP(1) and DP(2) will be considered. From the earlier inlet particle characteristics calculations, DP(1) = 0.2738 μm and DP(2) = 0.3240 μm.

KK(1) = 1 + $(1.6 \times 10^{-19})^2$ * 2×10^{15} * 7.5 * 2.73×10^{-7}/ (2*(8.85×10^{-12})*($LL^{0.5}$))
KK(1) = 147,608.24 and ln(KK(1)) = 11.9023
KK(2) = 1 + $(1.6 \times 10^{-19})^2$ * 2×10^{15} * 7.5 * 3.24×10^{-7}/ (2*(8.85×10^{-12})*($LL^{0.5}$))
KK(2) = 175,183.22 and ln(KK(2)) = 12.0736

Q(1) =3 * 3.1417 * 8.85×10^{-12} * 218,854 * $(2.73 \times 10^{-7})^2$ + (2 * 3.1417 * 8.85×10^{-12} * 1.38×10^{-23} * TEMPKE * 2.73×10^{-7}/ (1.6×10^{-19})) * 11.9023)
Q(1) = 6.81×10^{-18}
Q(2) =3 * 3.1417 * 8.85×10^{-12} * 218,854 * $(3.24 \times 10^{-7})^2$ + (2 * 3.1417 * 8.85×10^{-12} * 1.38×10^{-23} * TEMPKE * 3.24×10^{-7}/ (1.6×10^{-19})) * 12.0736)
Q(2) = 8.47×10^{-18}

WI(1) = Q(1) * KC(1) * 218,854 / (3 * 3.1417 * 1.74×10^{-5} * DPSI(I))
WI(1) = 6.81×10^{-18} * 1.6703 * 218,854/ (3 * 3.1417 * 1.74×10^{-5} * 2.73×10^{-7})
WI(1) = 0.05560
WI(2) = Q(1) * KC(1) * 218,854 / (3 * 3.1417 * 1.74×10^{-5} * DPSI(I))
WI(2) = 8.47×10^{-18} * 1.5821 * 218,854/ (3 * 3.1417 * 1.74×10^{-5} * 3.24×10^{-7})
WI(2) = 0.05519

Simplifying the calculation for one ESP Section, i.e., NS = 1, SCA = 10 s/m
PCQ(1) = exp(-(10/1) * WI(1) / (1 - 0))
PCQ(1) = exp(-10 * WI(1)) = exp(-10 * 0.05560) = 0.5735
PCQ(2) = exp(-(10/1) * WI(2) / (1 - 0))
PCQ(2) = exp(-10 * WI(2)) = exp(-10 * 0.05519) = 0.5758

Penetration, PTESP(1) = 0 + (1 - 0) * PCQ(1) + 0 * (1 - SN) * (1 - PCQ(I))
PTESP(1) = PCQ(1) = 0.5735

ESP Efficiency, EFFESP(1) = 1 - PTESP(1) = 1 - 0.5735 = 0.4265
Similarly, PTESP(1) = PCQ(2) = 0.5758
ESP Efficiency, EFFESP(2) = 1 - PTESP(2) = 1 - 0.5758 = 0.4242

Within rounding errors, the above calculations support the application output for single ESP Section, i.e., NS=1. The above calculations are repeated to generate overall ESP efficiency and overall System efficiency which cannot be readily checked with hand calculations. However, the algorithm has been verified otherwise.

IWS-1 Characteristics

This section provides a calculation check for IWS-1 characteristics based on specified user inputs. The variable nomenclature used will be similar to that used earlier in the text in Section 3.7. In order to verify the calculations within the application, elements from the test design case calculations presented in Chapter 4.0 have been utilized. The user provided the following inputs:

1. Total Depth of Wetted Packing, DIWS1 = 5 feet
2. Selected SCA = 62.6 ft^2/1000 acfm

In this case, the calculations are straightforward. The IWS empirical prediction equation for the specified SCA is of the form:
Aerodynamic Diameter, $DAERO(I) = DP(I)* ROP^{0.5}$
$IWSPTB(I) = ((0.006199 * DAERO(I)^3) - (0.060103 * DAERO(I)^2) + (0.011302 * DAERO(I)) + 0.77037)^{(DIWS1/8)}$
$IWSPTA(I) = -0.48 * DP(I)^2 + (3.07 * DP(I)) + 2.02$
Fractional Penetration, IWSPT(I) = (exp(-IWSPTA(I))) * IWSPTB(I)
Fractional Efficiency, IWSEFF1(I) = 1 - IWSPT(I)

For calculation verification purposes, only the first two size intervals, i.e., DP(1) and DP(2) will be considered. From the earlier inlet particle characteristics calculations,
DP(1) = 0.2738 μm and DP(2) = 0.3240 μm.
$DAERO(1) = DP(I)* ROP^{0.5}$
DAERO(1) = 0.4329 μm and DAERO(2) = 0.5123 μm
$IWSPTB(1) = ((0.006199 * DAERO(I)^3) - (0.060103 * DAERO(I)^2) + (0.011302 * DAERO(I)) + 0.77037)^{(DIWS1/8)}$
IWSPTB(1) = 0.8452 and IWSPTB(2) = 0.8432
$IWSPTA(1) = -0.48 * DP(1)^2 + (3.07 * DP(1)) + 2.02$
IWSPTA(1) = 2.8246
$IWSPTA(2) = -0.48 * DP(2)^2 + (3.07 * DP(2)) + 2.02$
IWSPTA(2) = 2.9643
Fractional Penetration, IWSPT(1) = (exp(-IWSPTA(1))) * IWSPTB(1)
IWSPT(1) = (exp(-2.8246)) * 0.8452 = 0.050148
Fractional Efficiency, IWSEFF1(1) = 1 - IWSPT(1) = 1 - 0.050148 = 0.9498

IWSEFF1(1) = 0.9498 * 100 = 94.98%

Fractional Penetration, IWSPT(2) = (exp(-IWSPTA(2))) * IWSPTB(2)
IWSPT(2) = (exp(-2.9643)) * 0.8432 = 0.04351
Fractional Efficiency, IWSEFF1(2) = 1 - IWSPT(2) = 1 - 0.04351 = 0.9565
IWSEFF1(2) = 0.9565 * 100 = 95.65%

Within rounding errors, the above calculations support the application output. The above calculations are repeated to generate overall IWS-1 efficiency and overall System efficiency which cannot be readily checked with hand calculations. However, the algorithm has been verified otherwise. Also note that the IWS empirical equation is valid in the range $0.12 \leq$ DAERO(I) ≤ 5 μm. The application imposes an artificial efficiency prediction curb on particle sizes greater than DAERO(I) = 5 μm by reporting a constant fractional efficiency value equal to the efficiency at 5 μm for aerodynamic particle diameters greater than 5 μm which, in effect, translates to a more conservative overall IWS efficiency estimate.

*** 0 ***

VITA

Kaushik Deb was born in Calcutta, India on October 20, 1960. He attended La Martiniere for Boys in Calcutta and graduated from high school in 1979. After high school he entered Jadavpur University, Calcutta and received a Bachelor's degree in civil engineering in 1983. After graduation he joined the Master's program in environmental engineering at the Indian Institute of Technology, Kharagpur and graduated in 1984. He worked for two years as a Project Air Pollution Control Engineer with Andrew Yule Ltd. (India) from 1985-87. In fall 1987, he entered the Master's program in atmospheric science at North Carolina State University, Raleigh, NC. In 1989, he left N.C. State to join Dames & Moore Inc. at Chicago to work for their Air Services Group. He worked with Dames & Moore from 1989-95. In fall 1995, he left Dames & Moore to join the graduate program in environmental engineering at the University of Tennessee at Knoxville. He completed the requirements for his Master's degree in August, 1997.

He is currently a member of the Air and Waste Management Association (AWMA) and an Associate Member of Sigma Xi, the national scientific research society.