1992

Particle Size and Velocity Measurements in Two Phase Flows Using Laser Doppler Velocimetry

Joseph Parker
Western Michigan University, joecparker@sbcglobal.net

PARTICLE SIZE AND VELOCITY MEASUREMENTS IN TWO PHASE FLOWS USING LASER DOPPLER VELOCIMETRY

HONORS COLLEGE FINAL PAPER
AND
SENIOR DESIGN FINAL PAPER

JOSEPH PARKER

ADVISOR:
DR. PARVIZ MERATI

ADDITIONAL SUPPORT BY:
JAMES JOSAITIS

WINTER 1992

DEPARTMENT OF MECHANICAL AND
AERONAUTICAL ENGINEERING

WESTERN MICHIGAN UNIVERSITY

ABSTRACT

Laser Doppler Velocimetry is used in fluid mechanics to measure the velocity of particles in fluid flows. A small particle exhibits motion closely corresponding to the fluid behavior because of its ability to follow high frequency flow fluctuations. Large particles do not follow the flow field precisely but represent a second phase. This project used LDV to measure the size and velocity of particles in two phase flows by computer analysis of the Doppler signal.

TABLE OF CONTENTS

INTRODUCTION

Standard experimental techniques in fluid flow studies involve the use of Laser Doppler Velocimetry to measure velocities of small particles seeded in the flow. Small particles are selected because they follow the highest frequency flow fluctuations indicative of the true fluid flow field. Larger particles have a frequency response much less than small particles and the fluid itself. Therefore, a flow seeded with large particles is actually a two phase flow in which the liquid and particle phases have dissimilar flow characteristics. Determining the flow properties of each phase independently can provide insight into how the phases interact, which leads to a description of the overall flow behavior. The difficulty in analyzing two phase flows lies in the experimental and computational techniques used to separate the two phases.

This project's goal was to separate particle phase signal (large particles) from fluid phase signal (small particles). Also, the particle signals were used to determine size and velocity.

LASER DOPPLER VELOCIMETRY THEORY

Introduction

Laser Doppler Velocimetry is a non-intrusive experimentation technique for the investigation of fluid flow in gasses and liquids. At a point location in a clear fluid, velocity and turbulence intensity can be measured. These measurements are based on the signal produced by a particle passing through a volume of interference fringes.

A planar wave laser beam is split into two equal and parallel beams and are crossed after passing through a focusing lens. Figure 1 shows the major components of an LDV system: laser, beamsplitter, frequency shifter, focusing lens, photodetector, and processor. Signals are collected by the photodetector, analyzed by the TSI IFA 550 signal processor, and the results are recorded and displayed on the CompuAdd 333 PC.

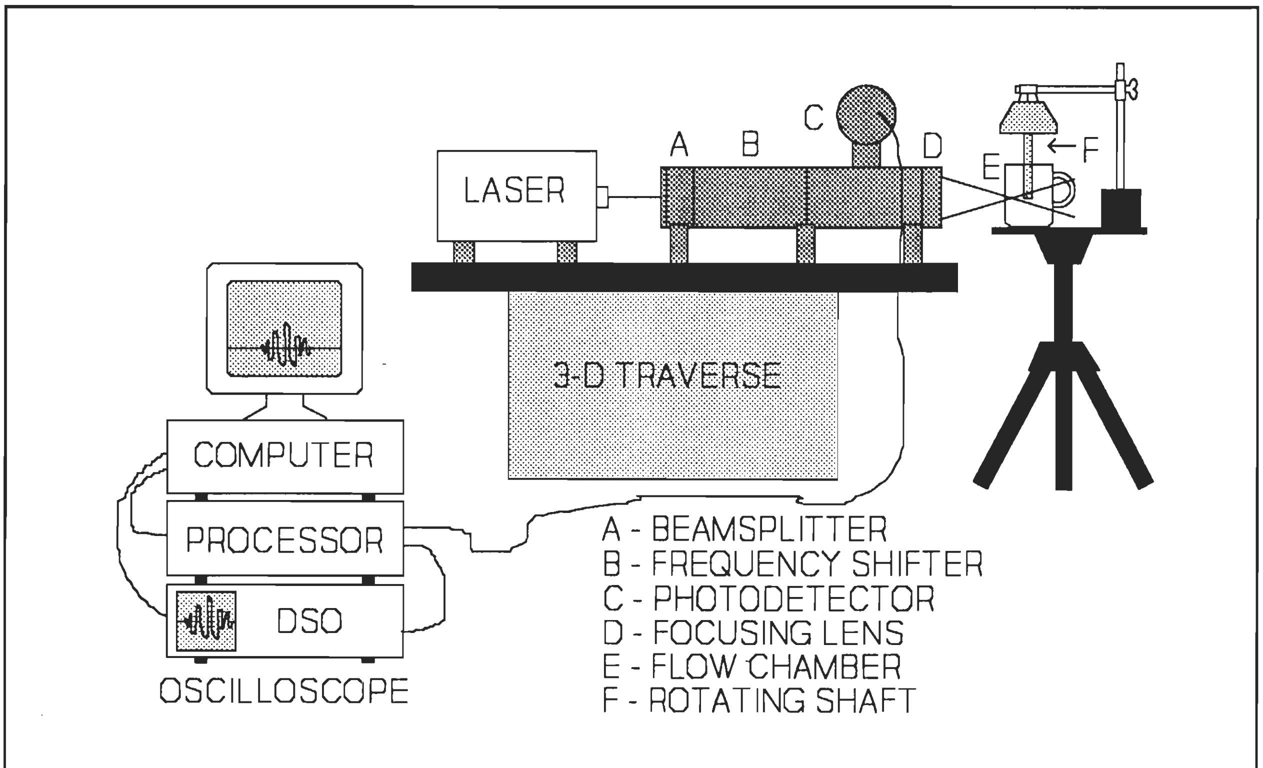


Figure 1: LDV and experimental setup

Filtered signal is also displayed on the Tektronix 2232 Digital Storage Oscilloscope (DSO). Although unused in this project, the frequency shifter determines the flow direction. The beam crossing is located inside a clear glass of distilled water.

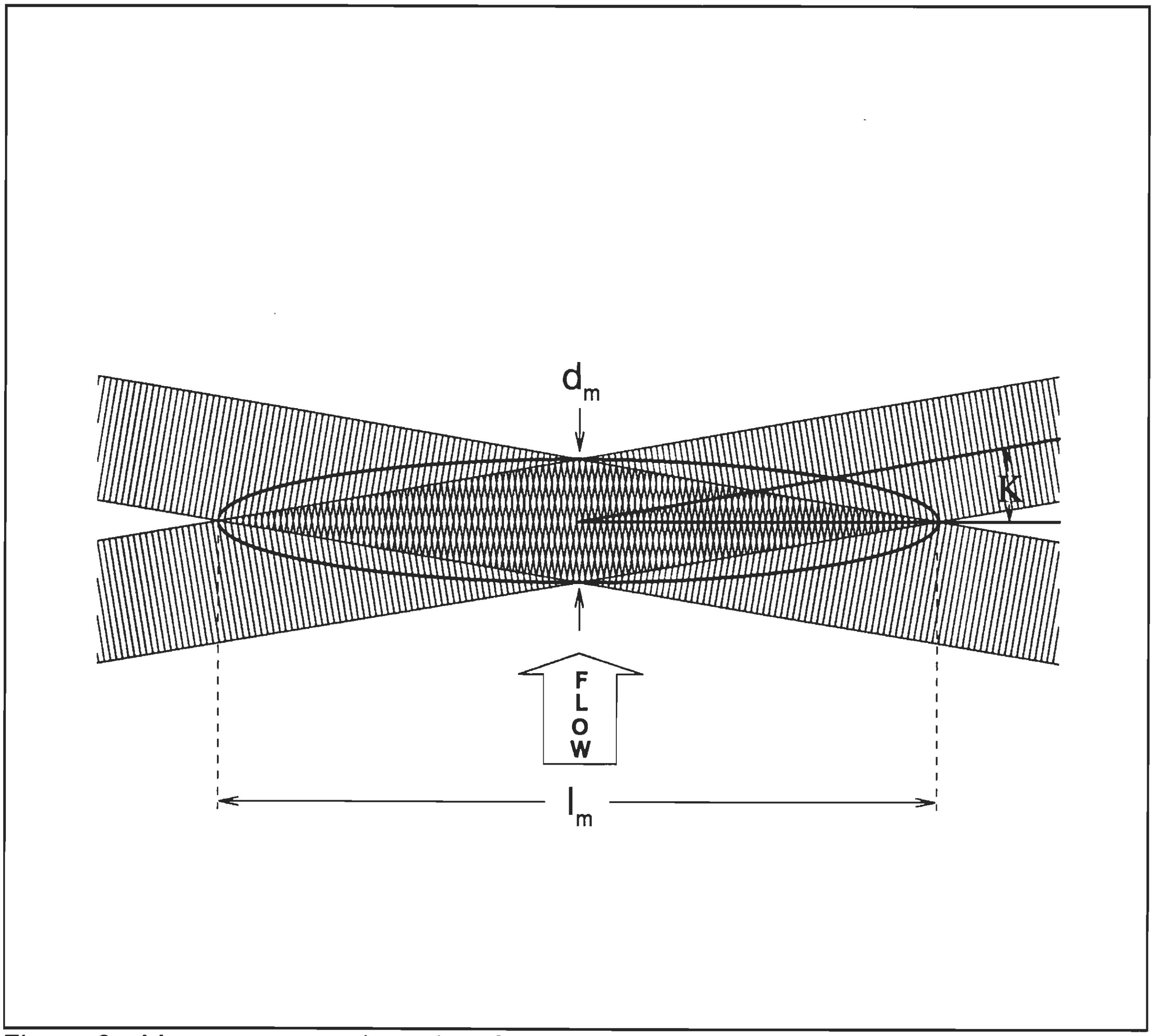


Figure 2: Measurement volume interference fringes

<u>Interference Fringes</u>

A football-shaped measurement volume of constructive and destructive, high and low intensity light fringes is created at the point of beam intersection. Figure 2 illustrates this phenomenon. By convention, the laser beams and the measurement volume are

defined to exist out to e^{-2} (13%) of the centerline intensity. Because the beams physically exist out to zero intensity, the measurement volume occupies an ellipsoidal space greater than the diamond shape of the geometrical beam crossing. Following Mie interference fringe scattering theory, Farmer (1972) developed equation (1) for the intensity distribution at any coordinate in the measurement volume.

$$I_o=2I\exp\left[\left(-2/b_o^2\right)\left[X^2+Y^2+\left(Z^2\alpha^2/4\right)\right]\right]* \\ *\left[\cosh\left(2YZ\alpha/b_o^2\right)+\cos\left(K\alpha Y-g\right)\right] \qquad (1)$$

where:

- I = Center-line intensity of one beam at the geometric center of the probe volume
- b_o = $d_m/2$
- α = 2K
- K = $2\pi/\lambda$
- d_m = Measurement volume diameter
- K = Half angle between the two beams
- λ = Beam wavelength

Cartesian coordinates with respect to the center of the measurement volume are represented by X, Y, and Z. g is a function that accounts for relative phase differences in the beams due to differences in path length in arriving at the probe volume. For optimum fringe contrast, g is set to zero.

The exponential component follows a Gaussian distribution of the beam crossing intensity as if there was no interference. The cosine component establishes the intensity oscillation due to the fringes. Figure 3 is a plot of intensity ratio, I/Imax, across the measurement volume center. I_{max} is the intensity at the center of the measurement volume. Notice that the intensity ratio is e^{-2} at dm/2.

For this project, a focusing lens with a focal length of 122 mm was used. This provided a measurement volume with the following dimensions: length (l_m) = 0.60 mm,

width (d_m) = 0.12 mm, and the fringe spacing (d_f) = 0.0013 mm. The fact that the measurement volume is essentially a point location is one distinct advantage of LDV over other measurement systems.

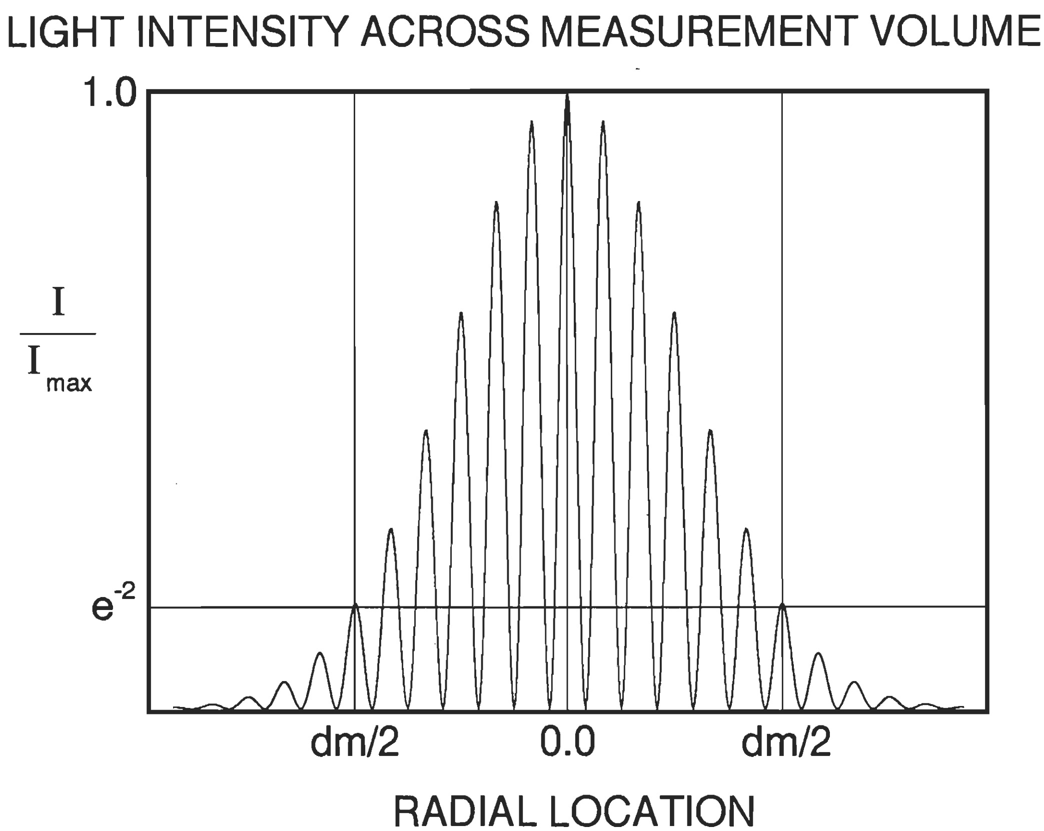


Figure 3: Light intensity across probe volume center

Particle Velocity

As a particle in the fluid passes through the measurement volume fringes, light is scattered in all directions off its surface. The scattered light is collected by a photodetector aligned with the transmitting optics (see Figure 1) and converted into voltage. The resulting burst has the characteristic shape of the measurement volume

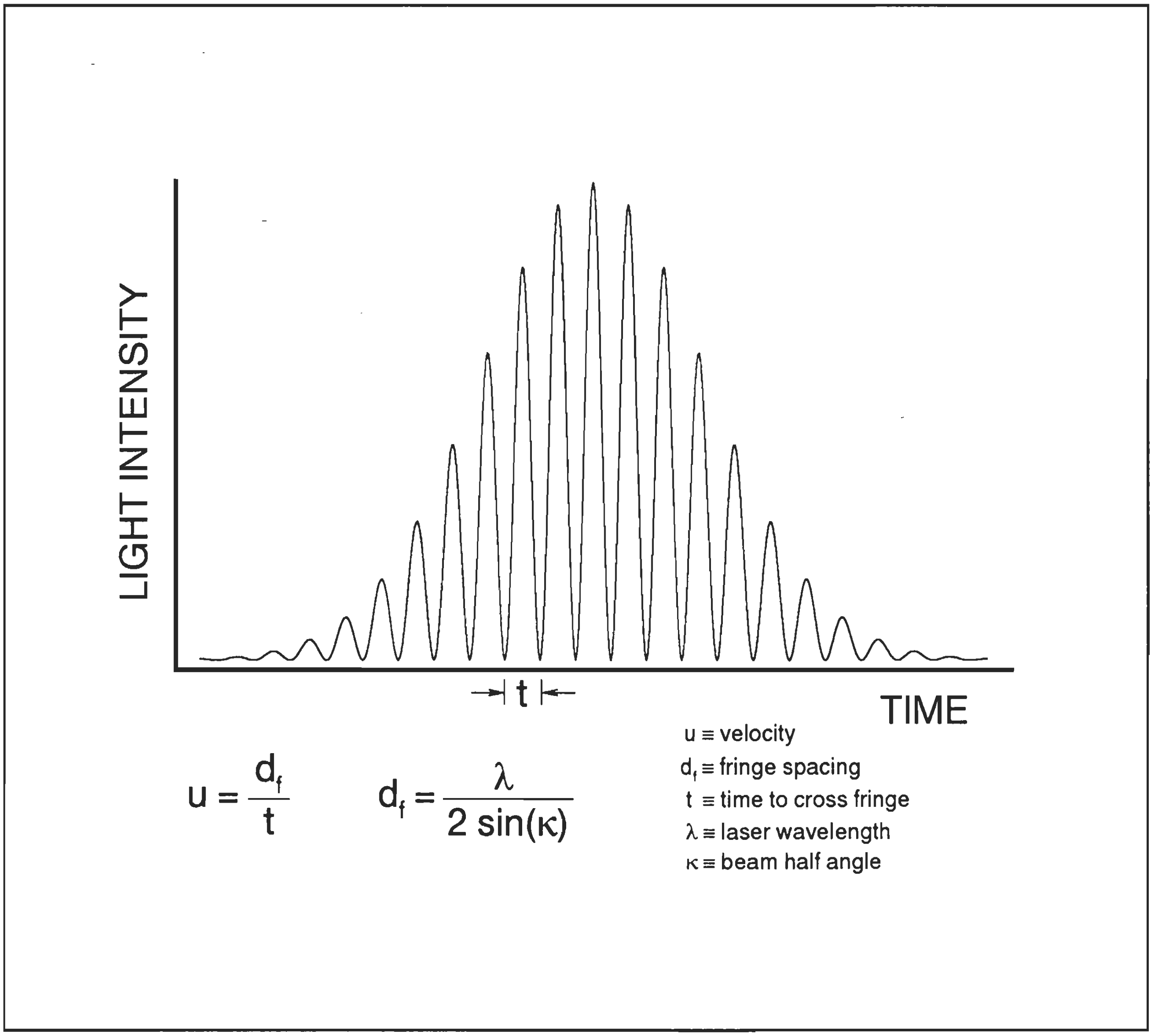


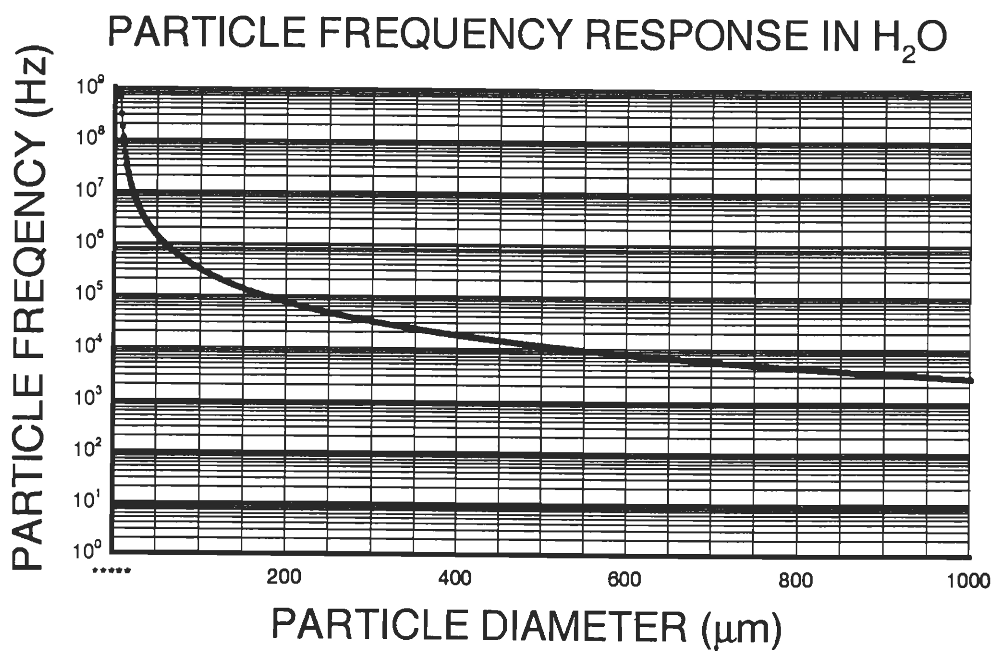
Figure 4: Light intensity scattered off a moving particle

intensity but with respect to time (Figure 4). Particle velocity, u, is calculated by dividing the fringe spacing, d_f, by the time it takes the particle to cross one fringe. Doppler frequency is the inverse of this time. Because fringe spacing is a function of the laser and optics only (see equation in Figure 4), doppler frequency is the remaining unknown. The IFA 550 signal processor from TSI reads the photodetector signal, filters it, and calculates the particle velocity for each burst. A one component LDV system measures the component of the velocity perpendicular to the bisector of the two beams as shown in Figure 2.

PARTICLE THEORY

Particle Selection

If LDV is used to measure the velocity of gasses and liquids, the flow must be seeded with particles which scatter light. The selection of particle size and type is an important consideration because of the particle's frequency response to flow fluctuations. In general, the fluid is capable of high frequency fluctuations. Seed particles must be able to closely follow the flow to provide representative velocities. Figure 5 and the



$$f = \frac{1}{2\pi\tau} \qquad \tau = \frac{\rho_p d^2}{18\mu}$$

$f \equiv$ particle frequency
$\rho_p \equiv$ particle density
$d \equiv$ particle diameter
$\mu \equiv$ viscosity of fluid

Figure 5: Particle frequency response

included equations are used to select adequate particle sizes. For polystyrene particles in water, diameters less than 25 μm sufficiently follow the flow. As particle diameter becomes greater, the particles become less responsive to flow fluctuations and can be considered a second phase. Typically, 9 micron silver coated particles are used in water.

For measurements in two phase flows consisting of fluid and particulate, it is important to differentiate between signals from different particle sizes. In order to measure the velocity of the fluid phase, only the small particles should be examined. In order to measure the velocity of the particulate phase, only the large particles should be examined.

Particle Signal

The left column of figures in Figure 6 shows a small particle passing through the center of the measurement volume and its representative signal. The right column shows a large particle passing through the same location and its representative signal. Because the large particle covers many fringes at any location in the measurement volume, its signal amplitude is greater than if a few fringes are covered at any location. From the intensity equation [equation (1)], light intensity is calculated at any point in the measurement volume. For the large particle, if the intensity is calculated for a range of locations (integration over the particle diameter), this returns an intensity greater than for any individual point. It is because of this principle that there is a relationship between particle size and amplitude. It may not be feasible to integrate the intensity equation to find a characteristic intensity (amplitude) for a given particle size because of the particle's reflective properties or its shape. A calibration curve with particle diameter versus amplitude should be used.

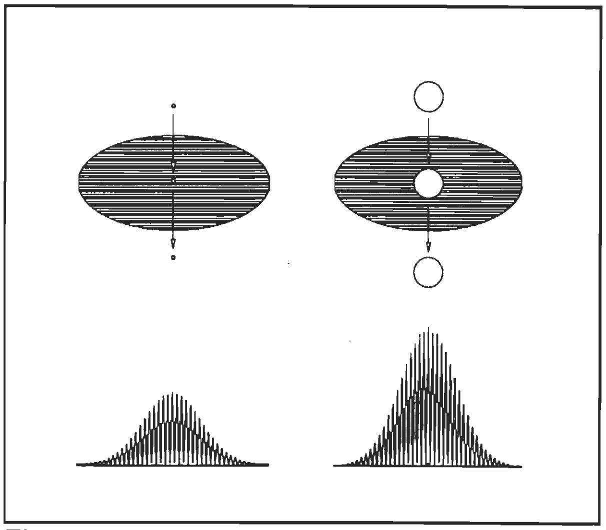


Figure 6: Large and small particles passing through probe volume center

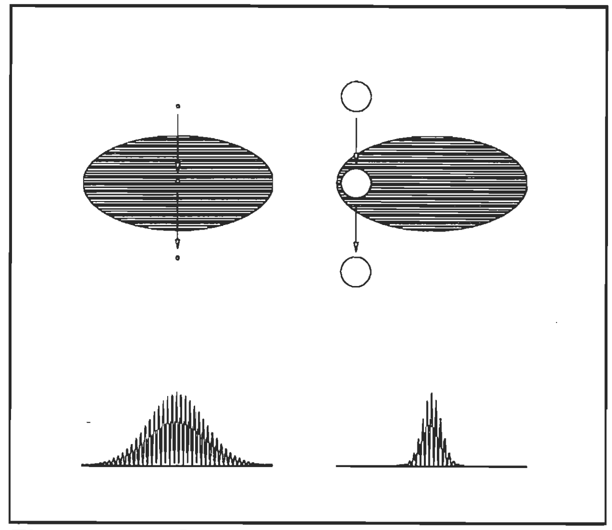


Figure 7: Small and large particles producing equal amplitudes

What happens when a particle does not pass through the center of the probe volume? For example, in Figure 7, a small particle passing through the measurement volume center might have the same amplitude as a large particle passing through an edge. In order to establish a relationship between particle size and amplitude, the amplitude must be normalized to the measurement volume center. Notice that for the large particle to produce the same amplitude burst as the small particle, the path through the measurement volume is shorter; less fringes are crossed.

Amplitude Correction

Each signal is composed of three factors: maximum amplitude, number of fringes, and doppler frequency, all of which can be measured. With the maximum amplitude and number of fringes, Yang (1991) used a Gaussian function to calculate the amplitude of a particle as if it had passed through the center of the measurement volume:

$$A(D_p)=A_{max}(D_p,I)*\exp\left[2\left(1-\frac{N^2}{N_{max}^2}\right)\right] \qquad (2)$$

where:

$A(D_p)$ = Calculated amplitude representing particle size passing through volume center

A_{max} = Maximum measured amplitude from particle signal

N_{max} = Maximum number of fringes

N = Measured number of fringes from particle signal

By using this equation, the large particle burst in Figure 7 is corrected to the same configuration as Figure 6. With this tool, all bursts are compared to the same standard location so that one amplitude corresponds to one particle size.

Calibration is the key to this method. By analyzing a known particle size, a characteristic amplitude results. This is repeated for various particle sizes until a complete calibration curve of amplitude versus particle size is created. Because of a particle's reflective capacity and its shape, calibration is required for different types of particles as well as particle sizes. For example, silver coated particles would produce a different amplitude than polystyrene particles. Also, spherical particles and rectangular particles would produce different amplitudes.

BURST ANALYSIS

Introduction

After using the Tektronix Digital Storage Oscilloscope to save a 4096 data point waveform, the first task of burst analysis is to establish criteria used to discriminate between clean burst signals and noise. The waveform generated by a burst without the pedestal is a periodic, Gaussian, amplitude modulated wave. These facts suggest that the criteria to be used in burst identification should be derived from one or both of these items. The loosely used term "noise" consists of high frequency and white noise which is always present during data acquisition. Identifying the characteristics of noise can also be useful in burst/noise discrimination.

Autocorrelation

Autocorrelation is a measure how well data correlates with itself. Equation (3) (Bendat, Piersol, 1971) evaluates the autocorrelation normalized with respect to zero lag by direct computations:

$$A(r)=\frac{R_x(rh)}{R_x(0)}=\frac{\frac{1}{N-r}\sum_{n=1}^{N-r} x_n x_{n+r}}{\frac{1}{N}\sum_{n=1}^{N}(x_n)^2} \qquad r=0,1,2,\ldots,m \tag{3}$$

where:

- x_n = data value
- N = number of data values
- r = lag number
- m = maximum lag number (m < N)
- rh = displacement

Noise is random and has little, if any, correlation. The value of autocorrelation,

A(r), for noise initially starts at 1, and immediately drops to zero as r increases. The value of autocorrelation for a constant value is 1 because at any lag, the data value is constant.

Initial attempts at implementing the autocorrelation for burst/noise discrimination were unsuccessful because the parameter being analyzed was inherently random. Amplitude variation of the entire signal was autocorrelated in hopes that regions of noise and signal would be revealed. The autocorrelation function appeared to follow the oscillating amplitude variation but with a phase shift. This result follows the theoretical autocorrelation of a sine wave but proved nothing for burst/noise discrimination.

One characteristic of a burst signal that is relatively constant is the doppler frequency. Assuming that a particle passing through the measurement volume does not experience an appreciable change in velocity over such a small region, the doppler frequency generated by that particle would remain constant. This also infers that the time between fringe crossings is constant. If the autocorrelation of the time between zero voltage crossings for a waveform returns a value close to 1, the signal is a clean burst. This approach provided excellent results for discrimination. For a data set of 16 zero crossings (as for the TSI signal processor), the value of autocorrelation varied less than $\pm$ 0.05 from 1. The autocorrelation for 16 zero crossings of noise provided results characteristic of random data, immediately dropping from 1 to zero.

Although the criterion for determining the difference between burst signal and noise functioned correctly, it was determined (after considerable attempts) that complete computer automation to establish burst locations is too complex for the scope of this project. Special cases, such as overlapping bursts, proved to be difficult to eliminate. Therefore, the initial determination of burst start and finish locations would be a decision

by the user and not an automatic function of the software. The software does, however, evaluate many possible combinations of start and finish locations around the manually selected points.

Gaussian Curve Fit

Theoretically, the shape of a burst without the pedestal is such that the variation of local maximum amplitude (within each zero crossing) versus time follows a Gaussian function. Since raw data is rarely "clean" enough to analyze directly, a least squares curve fit of the maximum local amplitudes based on a Gaussian function is calculated for the range of data selected by the user. The characteristic equation is of the form $y = A_{max}\exp[(-B(t-t_m)^2]$ while the linear form is:

$$y' = \ln(y) = C - B(t-t_m)^2 \qquad C = \ln(A_{max}) \tag{4}$$

where:

$$B = \frac{\frac{1}{n}\sum(t_i-t_m)^2\sum y_i' - \sum(t_i-t_m)^2 y_i'}{\sum(t_i-t_m)^4 - \frac{1}{n}\left[\sum(t_i-t_m)^2\right]^2} \tag{5}$$

$$C = \frac{1}{n}\sum\left[y_i' + B(t_i-t_m)^2\right] \tag{6}$$

All summations are for i from 1 to n, the maximum number of data points. t_m is the time corresponding to the maximum amplitude of the doppler signal and t_i is the time for a local maximum amplitude. y_i is the local maximum amplitude.

A measure of the curve fit's quality is given by the correlation coefficient:

$$r=\left[1-\frac{\sigma_{y,t}^{2}}{\sigma_{y}^{2}}\right]^{\frac{1}{2}} \tag{7}$$

where σ_y and $\sigma_{y,t}$ are defined as:

$$\sigma_{y}=\left[\frac{\sum(y_{i}-y_{m})^{2}}{n-1}\right]^{\frac{1}{2}} \tag{8}$$

$$\sigma_{y,t}=\left[\frac{\sum(y_{i}-y_{ic})^{2}}{n-2}\right]^{\frac{1}{2}} \tag{9}$$

y_{ic} are the computed values of y_i from the curve fit for the same value of t. To provide the best possible curve fit, the software recalculates the curve fit for a range of zero crossings around the user selected data range until the best correlation coefficient occurs.

<u>Calculations</u>

Because the curve fit represents the best mathematical model of the actual burst, doppler frequency, number of fringes, and maximum amplitude are derived from its coefficients. Once these parameters are known, it is possible to convert the maximum amplitude given by the curve fit into an amplitude referenced to the center of the measurement volume through equation (2).

After the entire waveform is averaged and the mean value removed, the zero crossings are calculated by linear interpolation. Assuming the waveform is symmetric about zero amplitude, the absolute value is used to double the number of data points. Local maximum amplitudes of the absolute value of the waveform are obtained with a maximum value capture routine. The local maximum amplitudes for the selected burst are then curve fitted and the correlation coefficient is determined. Based on the curve fit, $A_{max}(D_p, l)$ is calculated.

Autocorrelation of the time between zero crossings is calculated for the selected burst. If the autocorrelation is at least 0.90, the doppler frequency is calculated by: 1) finding the average time between zero crossings for the burst, 2) multiplying the average by 2 for a complete cycle, and 3) inverting the value. The time it takes a particle to cross the probe volume is determined by: 1) finding t_m, 2) finding the time $t(e^{-2})$ corresponding to $e^{-2}A_{max}$, and 3) calculating $2[t_m - t(e^{-2})]$. The number of fringes is the doppler frequency multiplied by the particle passing time. Particle velocity is the fringe spacing, based on optics, multiplied by the doppler frequency.

PROGRAMMING

Data acquisition and signal processing of waveforms is accomplished with 3 separate programs written in QuickBASIC. Appendices A, B and C contain the source codes. The first program, PWAVES, captures waveforms from the oscilloscope and converts them into Tecplot-format raw data files. The second program, MANAZAC, allows the user to manually select bursts, curve fitting and autocorrelation are performed, and several parameters are calculated. Finally, MANALYZE converts the amplitude and number of fringes obtained with MANAZAC into a corrected amplitude and executes a statistical analysis.

Waveform Capture

Figure 8 is a schematic of a typical LDV system on the left and a general flow chart of waveform capture on the right. The basic LDV system utilizes the TSI program "FIND" to measure real time velocity based on the processor's calculations, while the particle sizing system relies on oscilloscope waveform capture for post processing of bursts. The flow chart shows the structure of the PWAVES waveform capture program. The initial routine prompts the user to define a family name for the type of particle the data will represent. A name abbreviating the size and type of particle reduces future confusion when selecting waveform files for processing. Also, the user is prompted for a filename starting increment. This number is included with the family name in order to identify individual waveform files. Once the filing scheme is prepared, various Tektronix SPD subroutines are called such that the waveform on the oscilloscope is transferred to the computer. The waveform absolute value is then plotted on the monitor.

At this point, the user has the option to either save or reject the waveform based

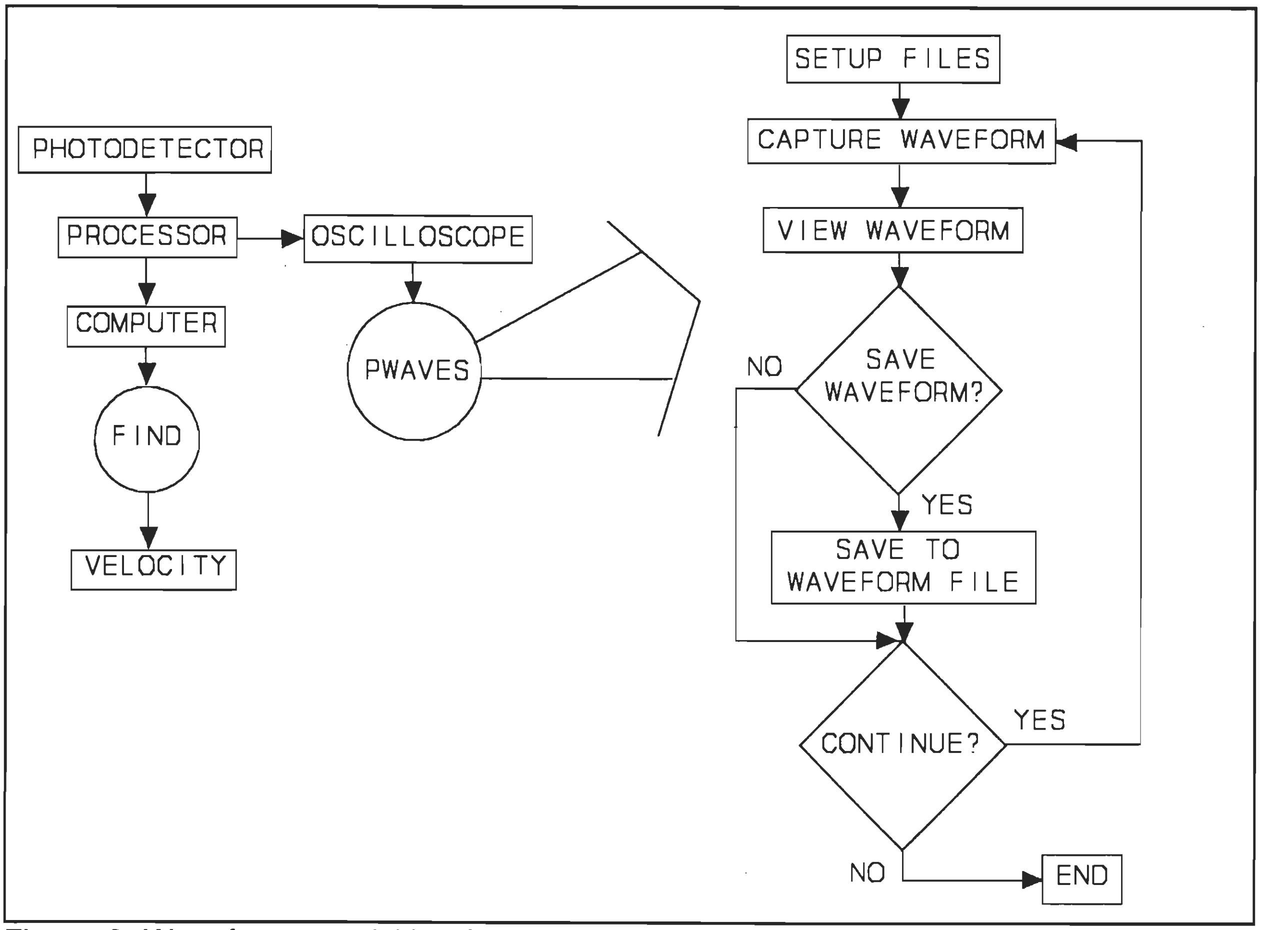


Figure 8: Waveform acquisition flowchart

At this point, the user has the option to either save or reject the waveform based on a well defined burst and the amount of noise. With experience, the user can make judgements on the likelihood of obtaining a curve fit with a high correlation coefficient. If the decision is to reject the waveform, the system is reset and readied to acquire another waveform. If the decision is to save the waveform, the data is saved to the computer hard drive using the family name and increment. The next filename is incremented by one. This type of file management allows the user to concentrate on capturing waveforms instead of typing filenames and writing over files by mistake. After deciding the fate of the current waveform, the user may continue capturing data or exit the program. This process should be repeated until enough waveforms have been saved such that a proper statistical analysis can be performed.

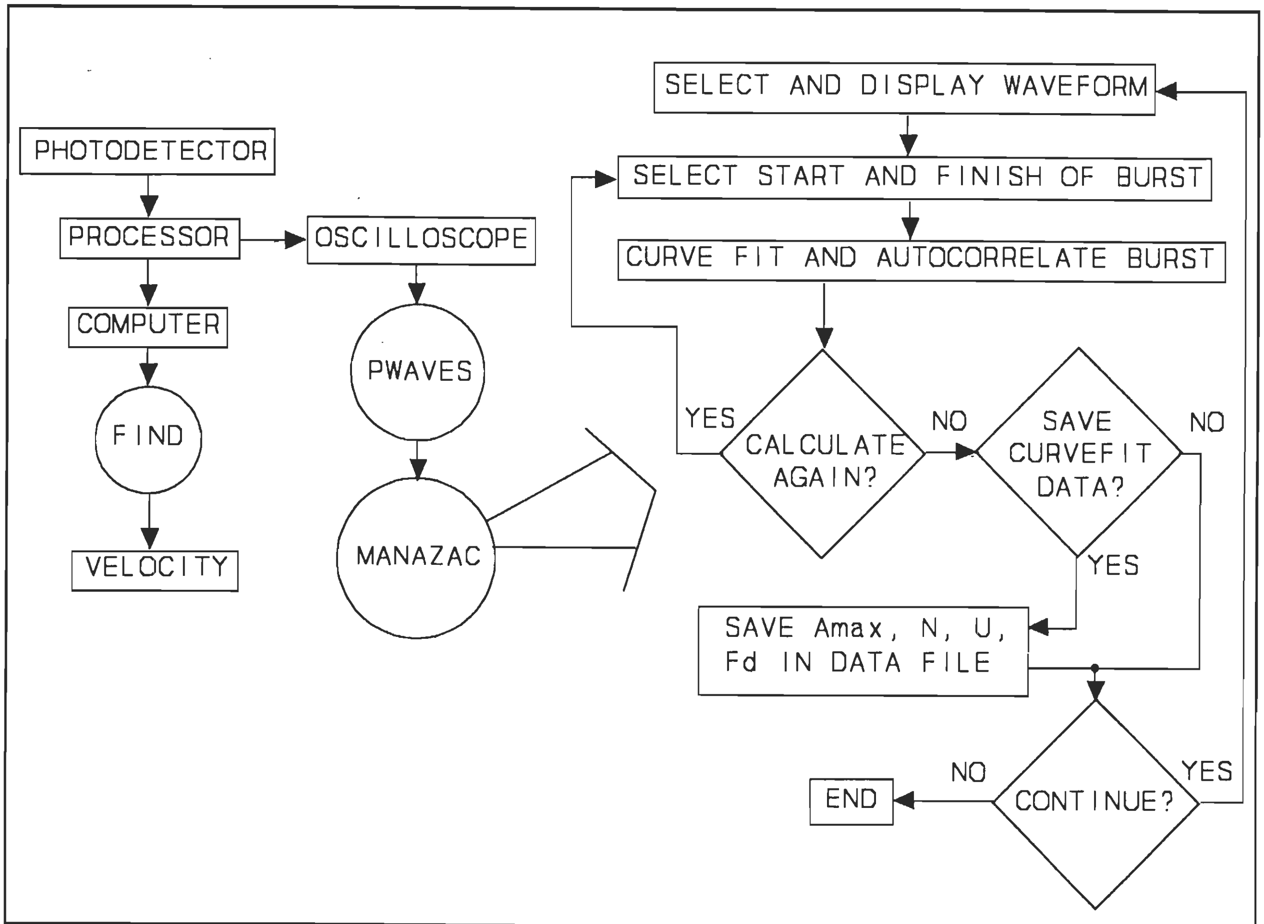


Figure 9: Signal processing flowchart

Signal Processing

Figure 9 represents the next phase of signal processing in which the waveform files created using PWAVES are curve fitted to yield maximum amplitude, number of fringes, velocity and doppler frequency. The first routine in MANAZAC, (**MAN**ual **A**mplitude **Z**ero crossing **A**utocorrelation **C**urve fit), sets up a directory of the waveform files created in PWAVES and enables the user to select a particular file for study. Next, MANAZAC displays the waveform with a list of commands executable by pressing the first command letter: manually select the end points of a burst (S and F), perform curve fit and other calculations (C), save curve fit data (N), select another waveform (ESC), and exit

(Q). Two hidden commands provide a screen redraw (R) and the curve fit and waveform saved in a Tecplot file (T).

The first step in calculating a curve fit is to select a data range. With the "Start" and "Finish" commands, a marker is moved on the screen to the estimated start and finish location of a possible burst. Start and finish locations are located at zero crossings. When the command "Calculate" is executed, the curve fit and autocorrelation are calculated for +/- 5 zero crossings around the start and finish markers. For the 121 possible combinations, the curve fit corresponding to the best correlation coefficient is retained. Calculations of maximum amplitude, number of fringes, and doppler frequency are based on this best curve fit. The curve fit is then plotted over the waveform and the calculated parameters are displayed on the screen.

After observing the results, the user decides whether or not the values for autocorrelation and correlation coefficient are acceptable. Values for both higher than 0.9 are generally acceptable; however, a visual check of the curve fitted waveform can also be a criterion for rejecting a result. It may be beneficial to run the calculations again with the best locations. This allows a slightly new range of data. Also, if the start and finish locations for the best curve fit are unreasonable, new locations should be selected for calculations.

Like PWAVES, the results can be saved or rejected. If the option is to save the curve fit results, the user is prompted to create a new data file or to append a previous file. It is necessary to name a new file only for the first time through this routine because once a data file is created for the family of particles being analyzed, the data file is appended for the entire family. Once the fate of the curve fit data is decided, the user may select another waveform for analysis or exit the program.

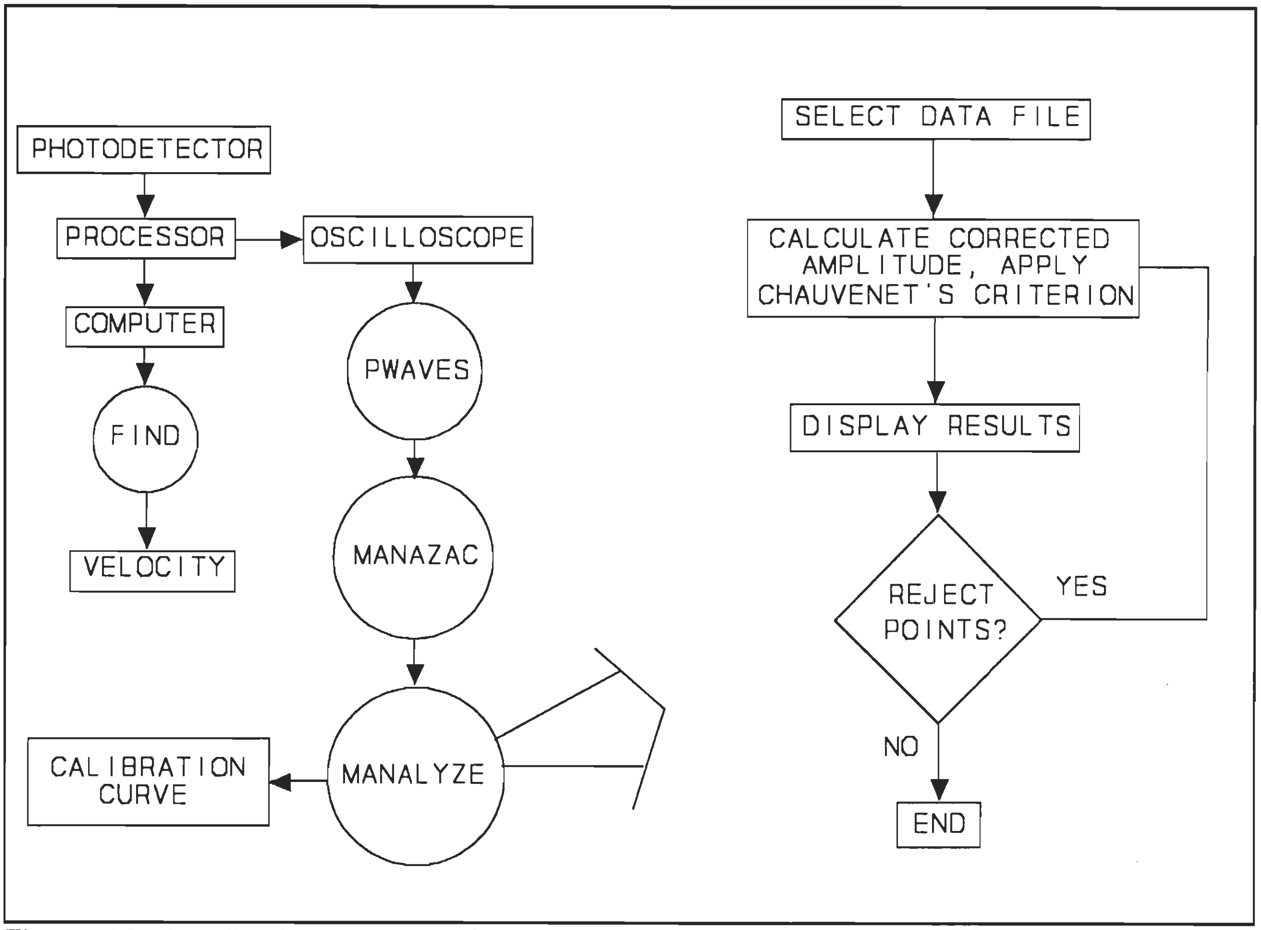


Figure 10: Amplitude correction flowchart

Amplitude Correction

The final step in data processing is a program called MANALYZE (Figure 10), which converts the measured maximum amplitude to a value as if the particle were to pass through the center of the measurement volume. The user selects a data file created by MANAZAC which contains the maximum amplitude and number of fringes for each individual burst for a family of waveforms. Using the amplitude conversion of equation (2), MANALYZE calculates the corrected maximum amplitude for each burst dependant on the number of fringes for each burst. For a family of corrected maximum amplitudes, the average value and standard deviation is then displayed along with the actual data for

each burst. The velocity and Chauvenet's criterion is also listed for each burst. Chauvenet's criterion states that "a data point can be rejected if the probability of obtaining the particular deviation from the mean is less than 1/2n" (Holman, 1966). The particular deviation is defined by $(x_m - x_i)/\sigma$ where x_m is the mean and x_i are the data. Points may be discarded by the user dependant on if they meet Chauvenet's criterion or by inspection. The data set is recalculated without the selected points.

Calibration

This entire 3 program process must be repeated for each particle size when performing an initial calibration. Once enough particle sizes have a corrected amplitude associated with them, a calibration curve for that type of particle can be constructed relating measured amplitude with particle size. Different particle types require their own calibration curve, therefore a library of calibration curves for different particle types should be created.

RESULTS

Two sizes of polystyrene particles and one size of silver coated particles were analyzed using the technique described in this report. Figures 11 and 12 represent MANAZAC examples of bursts used in the determination of the corrected maximum amplitude for 10 and 100 μm particles of polystyrene, respectively. Notice that the 10 μm particle waveform contains one well defined, smooth, and symmetrical burst. The correlation coefficient of 0.982 states that 98.2% of the data correlates well with the curve fit. An autocorrelation of 1.015 implies that the burst is clean and the particle velocity is constant. The 120 mm focal length lens produces a maximum number of 96 fringes. This particle crosses 84 fringes, close to the probe volume center. Also notice that the ratio of signal amplitude to noise amplitude (on either side of the burst), or signal-to-noise ratio, is very large. This fact supports the conclusion that the burst is clean.

The 100 μm particle waveform of Figure 12 is more difficult to resolve because of the signal before and after the large burst. These regions may be noise or small bursts. Although the correlation coefficient for the large burst is 0.969, which defines a clean Gaussian burst, the general shape appears quite different than the 10 μm particle burst. This is due to the different number of fringes and maximum amplitude. The asymmetric burst on the far right may be due to multiple particles.

Figure 13 is a screen dump of the results of MANALYZE for a 10 μm polystyrene particle family. For each burst (i): AMax is the measured maximum amplitude, N is the number of fringes, u is the velocity, $A(D_p)$ is the corrected maximum amplitude, and

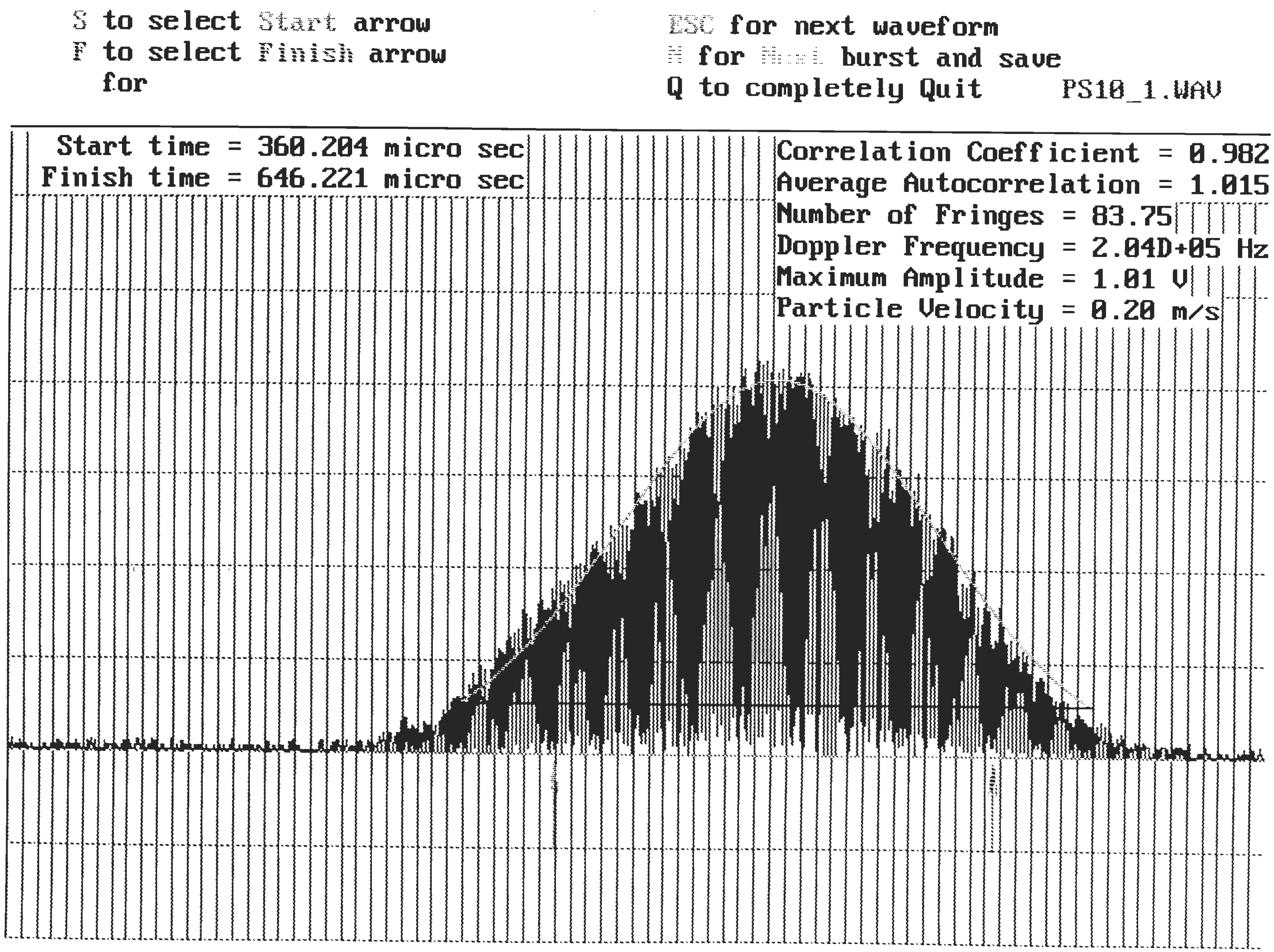


Figure 11: 10 µm polystyrene particle waveform

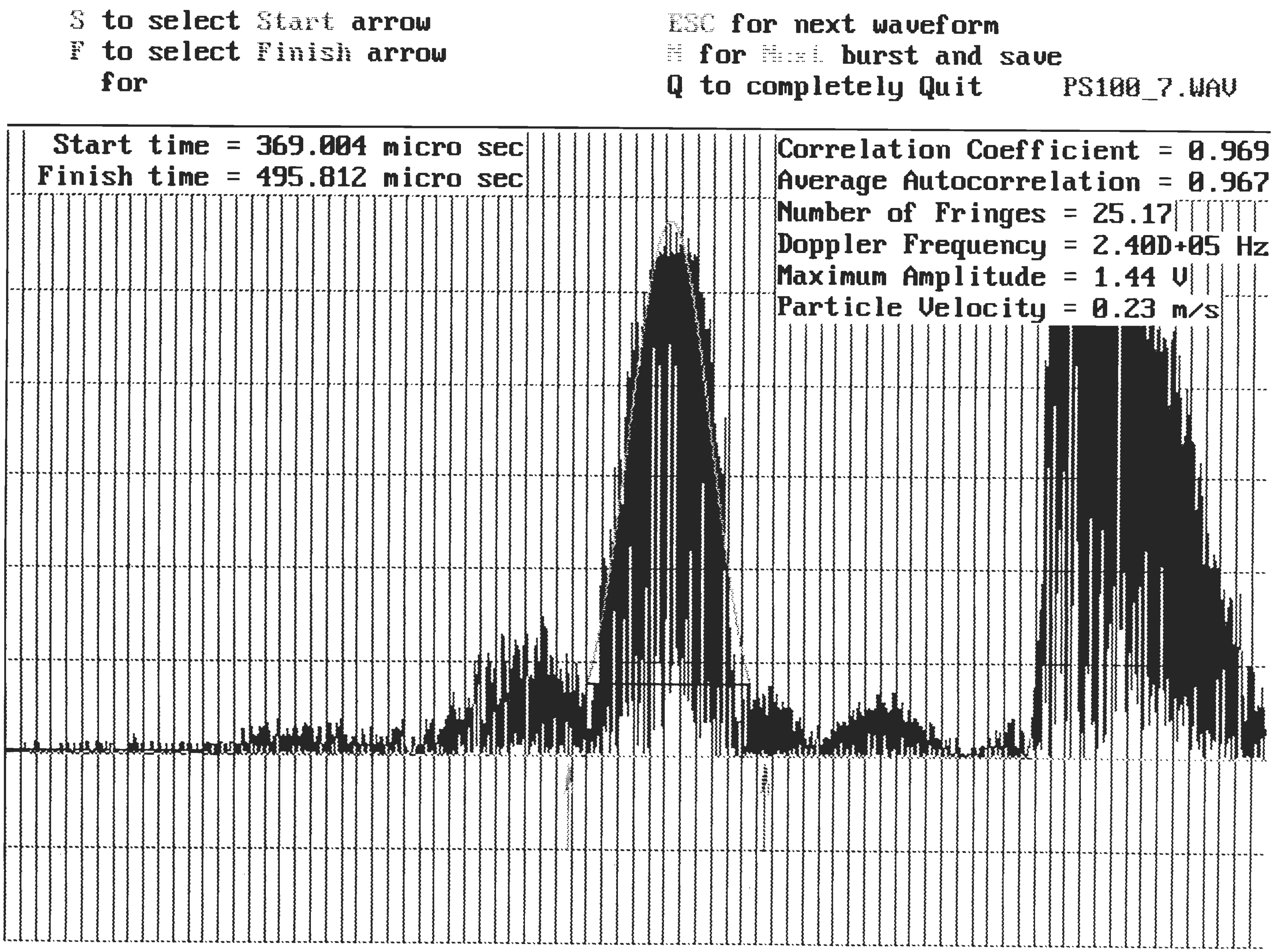


Figure 12: 100 µm polystyrene particle waveform

$$\frac{d_i}{\sigma} = \frac{|A(D_p) - A_a(D_p)|}{\sigma} \tag{10}$$

where $A_a(D_p)$ is the average value of $A(D_p)$. Averages and standard deviations are calculated for all parameters. Based on Chauvenet's criterion for 14 particles, if the ratio of di/σ is greater than 2.13, data points can be rejected. None of the 10 μm particles fall in this range. Therefore, the average corrected amplitude of 2.30 Volt is the characteristic amplitude for 10 μm polystyrene particles.

PS10_1.DAT

i	AMax	N	A(Dp)	di/σ	u
1	1.018	83.7	1.63	1.42	0.20
2	0.897	65.5	2.60	0.63	0.19
3	1.304	87.3	1.83	1.00	0.21
4	0.591	60.0	2.00	0.65	0.23
5	0.890	62.1	2.84	1.12	0.19
6	0.878	68.1	2.36	0.12	0.23
7	0.847	75.2	1.83	1.01	0.22
8	1.083	65.4	3.16	1.80	0.18
9	1.222	76.3	2.54	0.51	0.19
10	0.784	60.4	2.62	0.66	0.21
11	0.819	76.3	1.70	1.28	0.21
12	1.096	71.5	2.66	0.74	0.20
13	0.694	66.4	1.96	0.72	0.23
14	1.158	74.6	2.54	0.50	0.20
AVG	0.949	70.9	2.30		0.21
STD	0.205	8.4	0.47		0.02

Nmax = 95.8

Delete some data points?

Figure 13: MANALYZE results for 10 μm polystyrene particle

Figures 14 and 15 show the MANALYZE results for a family of 100 μm polystyrene particles before and after rejecting data points. Note the improvement in the standard deviation after removing the outlying points. Also, the average corrected amplitude is increased.

Figure 16 is the first calibration curve established with the particle sizing system for polystyrene particles. Although only two points are plotted, the large voltage difference between them proves that the system works. It confirms the theory that particles of larger size return a higher amplitude when referenced to the center of the measurement volume. Each point was plotted from the results obtained in Figures 13 and 15. Note that a 9 μm silver coated particle returns a higher voltage than a 10 μm

polystyrene particle. This point appears to be inconsistent with the theory simply because, in general, a smaller particle should return a lower voltage. However, since the 9 µm particle is silver coated, its reflective properties are much greater than the relatively opaque polystyrene. Therefore, one would expect a small particle with excellent reflectivity to return a higher voltage than a particle with lower reflective properties. Standard LDV practice uses 9 µm silver coated particles to seed flows because of their great reflectivity.

PS100_1.DAT

i	AMax	N	A(Dp)	di/σ	u
1	1.237	31.5	7.36	0.61	0.22
2	0.229	23.6	1.50	1.58	0.23
3	1.276	42.4	6.37	0.24	0.26
4	0.247	27.0	1.55	1.56	0.21
5	1.143	77.2	2.30	1.28	0.24
6	1.438	25.2	9.25	1.32	0.23
7	1.309	23.1	8.61	1.08	0.26
8	1.131	18.4	7.76	0.76	0.16
9	1.215	25.7	7.77	0.76	0.17
10	0.172	25.7	1.10	1.73	0.19
11	1.047	28.1	6.52	0.30	0.16
12	1.048	26.9	6.61	0.33	0.17
13	0.798	9.2	5.79	0.02	0.14
14	1.139	35.4	6.41	0.26	0.15
15	1.231	27.6	7.71	0.74	0.14
16	0.928	38.0	5.01	0.27	0.16
AVG	0.974	30.3	5.73		0.19
STD	0.405	14.6	2.67		0.04

Nmax = 95.8

Delete some data points?

Figure 14: MANALYZE results for 100 µm polystyrene particle before eliminating points

PS100_1.DAT

i	AMax	N	A(Dp)	di/σ	u
1	1.237	31.5	7.36	0.31	0.22
3	1.276	42.4	6.37	0.82	0.26
7	1.309	23.1	8.61	1.74	0.26
8	1.131	18.4	7.76	0.77	0.16
9	1.215	25.7	7.77	0.77	0.17
11	1.047	28.1	6.52	0.66	0.16
12	1.048	26.9	6.61	0.55	0.17
13	0.798	9.2	5.79	1.48	0.14
14	1.139	35.4	6.41	0.78	0.15
15	1.231	27.6	7.71	0.70	0.14
AVG	1.143	26.8	7.09		0.18
STD	0.151	9.0	0.88		0.05

Nmax = 95.8

Delete some data points?

Figure 15: MANALYZE results for 100 µm polystyrene particle after eliminating points

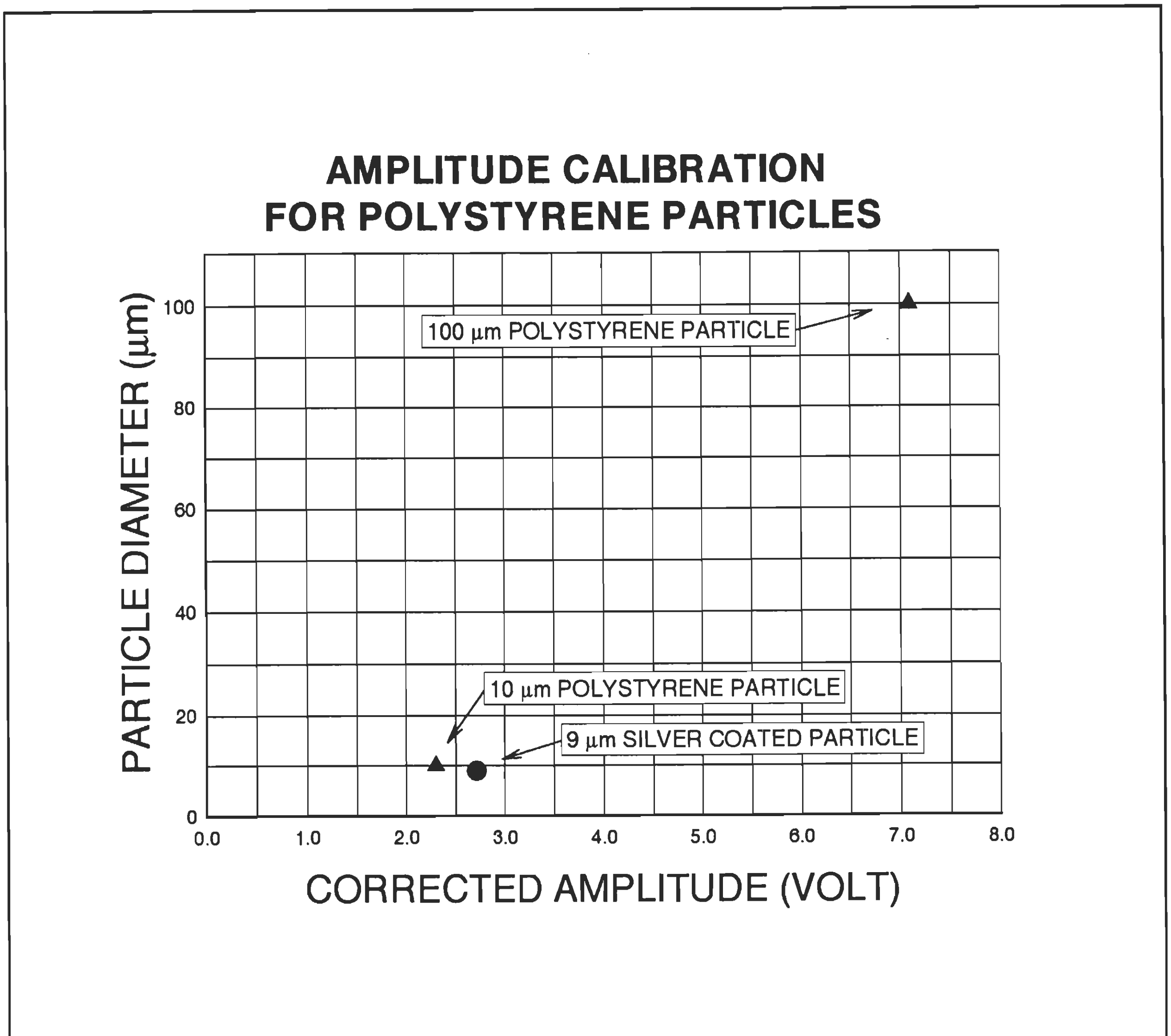


Figure 16: Relationship between size and amplitude for polystyrene particles

UNCERTAINTY ANALYSIS

In order to assess the range of deviation in the particle size measurement, a standard uncertainty analysis was applied to equation (10).

$$A(D_p)=A_{max}(D_p,l)*e^{2\left(1-\frac{N^2}{N_{max}^2}\right)} \tag{10}$$

The uncertainty in the amplitude measurement, $\omega_{A(Dp)}$ is shown in equation (11). This equation shows that the partial derivatives of $A(D_p)$ with respect to each of the variables $A_{max}(D_p,l)$, N and N_{max} provide the uncertainty in the amplitdue measurement.

$$\omega_{A(D_p)}=\left[\left(\frac{\delta A(D_p)}{\delta A_{max}(D_p,l)}*\omega_{A_{max}(D_p,l)}\right)^2+\left(\frac{\delta A(D_p)}{\delta N}*\omega_N\right)^2+\left(\frac{\delta A(D_p)}{\delta N_{max}}*\omega_{N_{max}}\right)^2\right]^{\frac{1}{2}} \tag{11}$$

The partial derivatives of $A(D_p)$ are given as follows:

$$\frac{\delta A(D_p)}{\delta A_{max}(D_p,l)}=e^{2\left(1-\frac{N^2}{N_{max}^2}\right)} \tag{12}$$

$$\frac{\delta A(D_p)}{\delta N}=-\frac{4N}{N_{max}^2}*A_{max}(D_p,l)*e^{2\left(1-\frac{N^2}{N_{max}^2}\right)} \tag{13}$$

$$\frac{\delta A(D_p)}{\delta N_{max}}=\frac{4N}{N_{max}^3}*A_{max}(D_p,l)*e^{2\left(1-\frac{N^2}{N_{max}^2}\right)} \tag{14}$$

The uncertainty in N_{max} is assumed to be zero (ω_{Nmax} = 0) because it is a quantity that remains constant throughout the experiment. N_{max} is derived from the measurement volume dimensions which are a function of the optical setup. During the course of this

experiment, the optical configuration was not changed. The uncertainty in the amplitude equation is now represented by:

$$\omega_{A(D_p)} = \left[\left(\frac{\delta A(D_p)}{\delta A_{max}(D_p,l)} * \omega_{A_{max}(D_p,l)}\right)^2 + \left(\frac{\delta A(D_p)}{\delta N} * \omega_N\right)^2\right]^{\frac{1}{2}} \tag{15}$$

Substituting, $\omega_{A(Dp)}$ becomes...

$$\omega_{A(D_p)} = \left[\left(e^{2\left(1-\frac{N^2}{N_{max}^2}\right)} * \omega_{A_{max}(D_p,l)}\right)^2 + \left(\frac{4N}{N_{max}^2} * A_{max}(D_p,l) * e^{2\left(1-\frac{N^2}{N_{max}^2}\right)} * \omega_N\right)^2\right]^{\frac{1}{2}} \tag{16}$$

The remaining variables in equation 7 are defined as such:

N^2/N^2_{max} = The average ratio of measured fringes to the maximum possible number of fringes. During the experiment, this ratio never exceeded 0.9 and most bursts that were curve fitted had a ratio higher than 0.5.

$\omega_{Amax(Dp,l)}$ = The uncertainty in the actual measured amplitude. This is a direct function of the instrument uncertainty in the oscilloscope. For the Tektronix 2232 model scope, the uncertainty was given as 1/25th of a division on the voltage scale. For all tests, the divisions were set at 0.5 volts. This defines the uncertainty in the measured amplitude as $\pm$ 0.02 volts.

$A_{max}(Dp,l)$ = An estimate of the highest typical amplitude measured during the experiment. The best average value for the data used in the calibration is $\approx$ 1.5 volts.

ω_N = The uncertainty in the measured number of fringe crossings. The resolution of all the acceptable waveforms permitted single fringe crossings to be observed. Therefore the uncertainty in this measurement in assumed to be $\pm$ 1 fringe.

N_{max} = The maximum number of fringe crossings. Constant value dependant on the geometry of the control volume, in this case 96 fringes.

For the 10 μm particles, the ratio of N to N_{max} was quite large. With an estimate of 0.8, equation 16 becomes:

$$\omega_{A(D_p)} = \left[\left(e^{0.72} * 0.02\,volts\right)^2 + \left(\frac{4*0.8}{96} * e^{0.72} * 1.5\,volts * 1 \right)^2 \right]^{\frac{1}{2}}$$

$$\omega_{A(Dp)} = 0.11 \text{ volts}$$

The slope of the calibration curve provides the conversion factor necessary to determine the actual uncertainty in the particle size measurement:

$$slope = \frac{(100-10)\,\mu m}{(7-2)\,volts} = 18\,\frac{\mu m}{volts}$$

$$\omega_{measured\ size} = slope * \omega_{A(D_p)} = \pm 1.99\,\mu m$$

The 100 μm particles exhibited a N/N_{max} ratio closer to 0.5, which changed the value of $\omega_{measured\ size}$ to equal ± 2.99 μm.

To summarize, the uncertainty analysis for the particle sizing system was ± 1.99 μm for 10 μm particles and ± 2.99 μm for 100 μm particles. These results are acceptable considering that the deviations in the particles themselves are ± 2 μm. Technically, the system becomes less accurate for larger sized particles, but the resolution of a large particle compared to the system uncertainty for such a particle is high enough that there is no difficulty discerning the difference between a large and small particle.

CONCLUSIONS

Based on the characteristic corrected amplitudes for 10 and 100 μm polystyrene particles, the amplitude correction theory and analysis methodology is valid. Also, the amplitudes for different types of particles produced different amplitudes.

One of the proposed objectives was to develop a completely automated system. Compared to Yang's (1991) manual-intensive system, this project removed many manual calculations but does require some intelligent user input. From this perspective, the developed system excels above any manual system because it draws from the combined computer/common sense judgment pool.

Calibration curves can be established for various particle types. Polystyrene, PTFE (2 types), shale, and silver coated particles were all tested individually. The polystyrene and silver coated particles produced the finest Gaussian bursts probably because of their precisely manufactured spherical shape. Calibration would account for randomly eccentric shapes.

Although this project concluded its feasibility in a crude testing chamber, actual particle sizing experiments would require calibration curve development in the actual flow chamber. This practice would account for reflections, refractions, and other noise inherent in the flow situation. To prove the theories and methodology, the testing chamber used was adequate.

The complete particle sizing system was composed of previously available hardware and an additional $590.75 computer/oscilloscope interface. Although an additional $35 was spent on a futile testing chamber, this project has saved the Department of Mechanical and Aeronautical Engineering the $50,000 approximate cost of commercially available systems.

RECOMMENDATIONS

Although this project concluded the feasibility of particle sizing using Laser Doppler Velocimetry, the following are a few recommendations for future use of this technique:

1) Further testing in more predictable flow conditions is necessary to establish the calibration routine. We experimented with 2 different particle sizes. This is not enough for an accurate calibration curve. Also, the flow chamber used was a clear glass coffee cup with a rotating humidifier post stirring the water. A chamber with a known velocity profile would help to prove the performance of velocity measurements along with particle sizing.
2) For particle sizing experiments, calibrate with several sets of homogeneous single size particles in the experimental chamber. The size ranges and particle types used for this project were inconsistent but the main focus was on proving the technique. Conduct actual experiments with similar particle types, mixtures of different particle types (not sizes) will provide illogical results.
3) Processing time is at least 9 minutes for each acceptable waveform because of the programming, even though all programs are independently executable. Reducing processing time will save total experiment time, not to mention operator frustration.
4) Ideally, real time signal processing could be performed with modifications to the LDV signal processor. Because the processor already employs various burst discrimination methods, it may not be too difficult to calculate the maximum burst amplitude and number of fringes. This final step would be a modification by TSI, the LDV manufacturer.

ACKNOWLEDGEMENTS

Thank you to Dr. Parviz Merati for his support and advice for this project and life/school in general. Thank you to James Josaitis for his able hands and mind as a partner on this project. Thank you to my wife Caryn for her undying support and love even though, as of late, school is my first home. Welcome April 24!

BIBLIOGRAPHY

Adrian, Ronald J., Laser Velocimetry. T. & A.M. report # 442, University of Illinois, 1980.

Adrian, Ronald J., Fingerson, Leroy M., Kaufman, Stanley L., Laser Velocimetry Theory, Applications and Techniques, University of Illinois, 1987.

Atakan, S., Chigier, N.A., Ungut, A., Yule, A.J., Particle Size and Velocity Measurement by Laser Anemometry, AIAA paper #77-214, 1977.

Bendat, J. S., Piersol, A. G., Random Data: Analysis and Measurement Procedures, Wiley-Interscience, 1971.

Brayton, D. B., Small Particle Signal Characteristics of a Dual-Scatter Laser Velocimeter, Applied Optics, vol. 13, 1974, pp. 2346-2351.

Durst, F., Studies of Particle Motion by Laser Doppler Techniques, Proceedings of the Dynamic Flow Conference, 1978, pp. 345-372.

Farmer, W. M., Measurement of Particle Size and Concentrations Using LDV Techniques, Proceedings of the Dynamic Flow Conference, 1978, pp. 373-396.

Farmer, W. M., Measurement of Particle Size, Number Density, and Velocity Using a Laser Inferometer, Applied Optics, vol. 11, 1972, pp. 2603-2612.

Farmer, W. M., Sample Space for Particle Size and Velocity Measuring Interferometers, Applied Optics, vol. 15, 1976, pp. 1984-1989.

Goldschmidt, Victor W., Measurements in Two Phase Flow, Proceedings of the Dynamic Flow Conference, 1978, pp. 289-319.

Holman, J. P., Experimental Methods for Engineers, McGraw-Hill Book Company, 1984.

Lee, S.L., Srinivasan J., Measurement of Local Size and Velocity Probability Density Distributions in Two Phase Suspension Flows by Laser Doppler Technique, International Journal of Multiphase Flow, vol. 4, 1978, pp. 141-155.

Wyle, C. R., Advanced Engineering Mathematics, McGraw-Hill, 1960.

Yang, Yi, Experiments and Theory on Gas and Cohesive Particles Flow Behavior and Agglomeration in the Fluidized Bed Systems, Illinois Institute of Technology, 1991.

APPENDIX

A: PWAVES

B: MANAZAC

C: MANALYZE

```
' $INCLUDE: 'tekspd.bi'
'*****************************************************************************
'                          PWAVES1.BAS
'
'      This program grabs a 4 kByte waveform from the oscilloscope and
'      saves it to a TecPlot-ready file.  Most oscilloscope commands
'      were taken from Tektronix's SPD program by permission from M. Ely.
'      INPUT parameters: family file name, increment number
'      OUTPUT parameters: TecPlot-ready waveform file

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

DIM Volt(4100), T(4100)
status% = 1
wave% = 1
normalize$ = "YES"
family$ = "BURST"
fileinc% = 1
source$ = "CH1%"

SCREEN 12

top:
i$ = ""

menu:
        CLS
        LOCATE 4, 16: COLOR 14: PRINT "  PARTICLE SIZING WAVEFORM
            CAPTURE PROGRAM"
        LOCATE 5, 16: COLOR 12: PRINT "——————————————————————————"
        LOCATE 7, 16: COLOR 15: PRINT "  Oscilloscope must read data on"; :
            COLOR 10: PRINT " Channel 1"

        COLOR 15: LOCATE 11, 8: PRINT "F";
        COLOR 7: PRINT "amily Filename  ->"

        COLOR 15: LOCATE 13, 8: PRINT "S";
        COLOR 7: PRINT "tarting increment  ->"

        COLOR 15: LOCATE 15, 8: PRINT "C";
        COLOR 7: PRINT "heck directory files  ->"
        LOCATE 15, 48: COLOR 15: PRINT "Files *.WAV"

        COLOR 12: LOCATE 23, 70: PRINT "E"; : COLOR 7: PRINT "xit"
        COLOR 10: LOCATE 23, 60: PRINT "R"; : COLOR 7: PRINT "un"
        COLOR 15: LOCATE 23, 58: PRINT "<": LOCATE 23, 64: PRINT ">   <":
            LOCATE 23, 75: PRINT ">"
```

```
        COLOR 15: LOCATE 11, 48: PRINT family$
        COLOR 15: LOCATE 13, 47: PRINT fileinc%

    IF i$ <> "" THEN RETURN


'menu control section

    DO
        i$ = INKEY$
        i$ = UCASE$(i$)

     IF i$ = "F" THEN
        COLOR 15
        dsum$ = family$
        LOCATE 11, 48: PRINT dsum$; : PRINT CHR$(2)
        DO
            DO
            d$ = INKEY$
            LOOP UNTIL d$ <> ""
            IF ASC(d$) = 8 AND LEN(dsum$) > 0 THEN
                dsum$ = LEFT$(dsum$, (LEN(dsum$) - 1))
            END IF
            IF ASC(d$) <> 13 AND ASC(d$) <> 8 THEN dsum$ = dsum$ + d$
            IF LEN(dsum$) > 6 THEN dsum$ = LEFT$(dsum$, 6)
            LOCATE 11, 48: PRINT dsum$; : PRINT CHR$(2); : PRINT
                    SPACE$(8)
        LOOP UNTIL d$ = CHR$(13)
     family$ = UCASE$(dsum$)
     GOSUB menu
     END IF

    'increment filename input cont
     IF i$ = "S" THEN
        COLOR 15: LOCATE 13, 47: PRINT fileinc%
        LOCATE 13, 48: INPUT ; "", fileinc%
        GOSUB menu
     END IF

    'check that crazy directory!
     IF i$ = "C" THEN
        CLS
        LOCATE 7, 5: COLOR 15
        SHELL "DIR *.WAV /W"
        PRINT : PRINT : PRINT
        COLOR 11: PRINT "        Press any key to continue"
```

```
            DO
            a$ = INKEY$
            LOOP WHILE a$ = ""
            CLS : COLOR 15
            GOSUB menu
        END IF
        IF i$ = "E" THEN CLOSE : END

      LOOP WHILE i$ <> "R"

TakeIt:     'acquisition loop

DO
      CLS
      'filename increment
        filenum1$ = STR$(fileinc%)
        filenum2$ = RIGHT$(filenum1$, (LEN(filenum1$) - 1))
        OutFile$ = family$ + filenum2$ + ".WAV"

      'Enable floating point exception handling.
            CALL benablfpe

      'creates error data storage file
            CALL berrfile("PWAVES1.ERR", status%)

      'Reserve memory for waveforms
            dummy& = SETMEM(-40000)

      'Acquire waveform
            CALL bget2230(wave%, WFLOAT%, CH1%, POLL%, T10S%, TRUE%,
                  status%)
            IF status% <> 0 THEN CALL brpterr(status%, "Waveform acquisition
                  failed."): END

      'Creats Ascii file from waveform
            CALL bwftoafile(wave%, OutFile$, status%)
            IF status% <> 0 THEN CALL brpterr(status%, "Waveform to file failed."):
                  END

      'free wave%
            CALL bfreewf(wave%, status%)
            IF status% <> 0 THEN CALL brpterr(status%, "Free waveform failed"): END

      'return wavform memory to system
            dummy& = SETMEM(40001)

      GOSUB ConvertIt
```

```
LOOP UNTIL Q$ = "Q"
GOTO top
END


ConvertIt: 'Converts a useless SPD file into a voltage and time TecPlot file
Nmax = 4096
OPEN OutFile$ FOR INPUT AS #1
FOR i = 1 TO 13
        INPUT #1, crap$
NEXT i
INPUT #1, SCA$
FOR i = 1 TO 15
        INPUT #1, crap$
NEXT i
length2% = LEN(SCA$) - 4
Time1$ = RIGHT$(SCA$, length2%)
xpos% = VAL(RIGHT$(Time1$, 4))
length4% = LEN(Time1$) - 4
Base1% = VAL(LEFT$(Time1$, length4%))
Time2 = (Base1% * 10 ^ (xpos%)) * 10 ^ (6)

DumSum = 0
FOR i = 1 TO Nmax
        INPUT #1, Volt(i), trash$
        IF i > 1 THEN T(i) = T(i - 1) + Time2
        IF i = 1 THEN T(i) = 0
        DumSum = DumSum + Volt(i)
NEXT i
CLOSE #1
Max = 0
FOR i = 1 TO Nmax
        Volt(i) = Volt(i) - DumSum / (Nmax)
        IF ABS(Volt(i)) > Max THEN LET Max = ABS(Volt(i))
NEXT i

GOSUB PlotIt

IF Reject$ = "S" THEN
        OPEN OutFile$ FOR OUTPUT AS #2
        PRINT #2, "VARIABLES = Time, Volt"
        PRINT #2, "ZONE T = ONE"
        FOR i = 1 TO Nmax
                PRINT #2, T(i); Volt(i)
        NEXT i
        CLOSE #2
        fileinc% = fileinc% + 1
```

```
END IF
IF Reject$ = "R" THEN
        SHELL "DEL " + OutFile$
END IF

LOCATE 27, 58: PRINT SPACE$(18)
LOCATE 27, 15: COLOR 13: PRINT "Q "; : COLOR 15: PRINT "to"; : COLOR 13: PRINT
" Quit";
COLOR 15: PRINT ", any other key to collect next waveform"
DO
Q$ = UCASE$(INKEY$)
LOOP WHILE Q$ = ""

RETURN


PlotIt:    ' Plots the grabbed waveform and displays it on the screen for decision
upper = 3
lower = -.5

CLS  '  Set up Window
LOCATE 1, 2: PRINT "Horizontal: 10 micro sec/div"
LOCATE 2, 2: PRINT "  Vertical: 0.5 V/div"
WINDOW (0, lower)-(T(Nmax), upper)
FOR i = 0 TO T(Nmax) STEP 10
        PSET (i, lower), 8
        FOR j = lower TO upper STEP .5
                LINE -(i, j), 8
        NEXT j
NEXT i
FOR i = lower TO upper STEP .5
        PSET (0, i), 8
        FOR j = 0 TO T(Nmax) STEP T(Nmax - 1)
                LINE -(j, i), 8
        NEXT j
NEXT i

'  Plot Absolute Burst Voltage
PSET (T(1), ABS(Volt(1))), 7
FOR i = 2 TO Nmax
        LINE -(T(i), ABS(Volt(i))), 6
        'PSET (i, Volt(i)), 14
NEXT i
LINE (0, lower)-(0, upper), 5
LINE (0, 0)-(T(Nmax), 0), 5

COLOR 15
```

```
LOCATE 2, 40: COLOR 14: PRINT "Potential "; OutFile$
LOCATE 3, 39: COLOR 7: PRINT "————————————"
LOCATE 27, 57: COLOR 12: PRINT "R"; : COLOR 7: PRINT "eject"
LOCATE 27, 70: COLOR 10: PRINT "S"; : COLOR 7: PRINT "ave": COLOR 15
LOCATE 27, 55: PRINT "<": LOCATE 27, 64: PRINT ">"
LOCATE 27, 68: PRINT "<": LOCATE 27, 75: PRINT ">"

DO
Reject$ = INKEY$
Reject$ = UCASE$(Reject$)
LOOP UNTIL Reject$ = "R" OR Reject$ = "S"

RETURN
```

```
'*****************************************************************************
'                         MANAZAC2.BAS
'
'      This program analyzes the waveforms saved by PWAVES1 by curve
'      fitting the local maximum amplitudes and autocorrelating the
'      zero crossings.
'
'      INPUT parameters: TecPlot-ready waveform file with 4096 points
'         of voltage and time, laser wavelength, half angle
'      OUTPUT parameters: correlation coefficient, autocorrelation coeff,
'         maximum amplitude (Amax), number of fringes (N), doppler
'         frequency (Fd), particle velocity (u)
'
'*****************************************************************************


DEFDBL A-Z
Nwaymax = 5000: Nmax = 1000
DIM Volt(Nwaymax), Tinc(Nwaymax)
DIM TZ(Nmax), AmpMax(Nmax), TampMax(Nmax)
DIM Acor(Nmax), Rx(Nmax)
DIM What$(500), WhereFile$(50, 6)
CONST PI = 3.141592654#
CONST ESC = 27, DOWN = 80, UP = 72, LEFT = 75, RIGHT = 77
CONST PGDN = 81, RETURF = 13, Funk1 = 59

NS = -1
Nmax = 4096
dirwf$ = "*.wav"
dirns$ = "*.dat"

SCREEN 12: COLOR 15: CLS
LOCATE 5: INPUT "    Enter the beam half angle Kappa (degrees)  -> ", Kap
LOCATE 7: INPUT "     Enter the laser wavelength (nano meter)  -> ", Lambda
IF Lambda < 100 THEN
   DO
   INPUT "      Enter the laser wavelength (nano meter)  -> ", Lambda
   LOOP UNTIL Lambda > 100
END IF
Lambda = Lambda * 10 ^ (-9)
Kap = Kap * PI / 180

GOSUB Waveform
GOSUB Menu

END      ' Note: this is the end of the program's main body.  All other
         '    executions are in subroutines
```

```
Menu:     'The menu is located at the top of the screen
  DO
   LOCATE 1, 1: PRINT SPACE$(80): LOCATE 2, 1: PRINT SPACE$(80)
   LOCATE 3, 1: PRINT SPACE$(80): LOCATE 4, 1: PRINT SPACE$(80)
   COLOR 15: LOCATE 4, 1: PRINT
"______________________________________________________________________________"
   COLOR 10: LOCATE 1, 5: PRINT "S"; : COLOR 15: PRINT " to select"; : COLOR 10:
                    PRINT " Start"; : COLOR 15: PRINT " arrow"
   COLOR 12: LOCATE 2, 5: PRINT "F"; : COLOR 15: PRINT " to select"; : COLOR 12:
                    PRINT " Finish"; : COLOR 15: PRINT " arrow"
   COLOR 14: LOCATE 3, 5: PRINT "C"; : COLOR 15: PRINT " for"; : COLOR 14:
                    PRINT " Calculations"
   COLOR 7: LOCATE 3, 43: PRINT "Q"; : COLOR 15: PRINT " to completely"; : COLOR
                    7: PRINT " Quit"
   COLOR 13: LOCATE 1, 43: PRINT "ESC"; : COLOR 15: PRINT " for next waveform"
   COLOR 11: LOCATE 2, 43: PRINT "N"; : COLOR 15: PRINT " for"; : COLOR 11:
                    PRINT " Next"; : COLOR 15: PRINT " burst and save"
   COLOR 14: LOCATE 2, 34: PRINT "MAIN": LOCATE 3, 34: PRINT "MENU": COLOR
                    15
   LOCATE 5, 3: PRINT USING " Start time = ###.### micro sec"; TampMax(Start1)
   LOCATE 6, 3: PRINT USING "Finish time = ###.### micro sec"; TampMax(Finish1)
   LOCATE 3, 68: COLOR 9: PRINT File1$: COLOR 15

   DO: i$ = INKEY$
   LOOP UNTIL i$ <> ""
   SELECT CASE UCASE$(i$)
      CASE "S"
         GOSUB Start
      CASE "F"
         GOSUB Finish
      CASE "C"
         GOSUB Acorr
      CASE "N"
         GOSUB NextSave
      CASE "R"
         GOSUB Plot1
         GOSUB Plot2
      CASE "T"
         GOSUB TecPlot
      CASE "Q"
         END
   END SELECT
   IF ASC(i$) = ESC THEN GOSUB Waveform

LOOP

RETURN
```

```
Acorr:  'This subroutine curvefits the local maximum amplitudes, autocorrelates
        'the time between zero crossings, other necessary parameters

RrrMax = 0  'These loops set up the starting and finishing increments over which
s1 = 6      'all calculations will be performed
DO
s1 = s1 - 1
LOOP UNTIL Start1 - s1 > 0
s2 = 6
DO
s2 = s2 - 1
LOOP UNTIL Start1 + s2 < (Tmax - 1)
F1 = 6
DO
F1 = F1 - 1
LOOP UNTIL Finish1 - F1 > 0
f2 = 6
DO
f2 = f2 - 1
LOOP UNTIL Finish1 + f2 < (Tmax - 1)

Kinc = 10
FOR kStart1 = (Start1 - s1) TO (Start1 + s2)
a = 100: SOUND a, a / 100
a = 500: SOUND a, a / 500
a = 100: SOUND a, a / 100
COLOR 12: LOCATE 2, 75: PRINT USING "##"; Kinc: COLOR 15
FOR jFinish1 = (Finish1 - F1) TO (Finish1 + f2)

N = jFinish1 - kStart1 + 1
IF N <= 2 THEN GOTO SkipItAll
SumSqr = 0
' Autocorrelation for r=0
FOR m = kStart1 TO jFinish1
   SumSqr = SumSqr + (TZ(m) - TZ(m - 1)) ^ 2
NEXT m
Ro = SumSqr / N

'  Determine autocorrelation for specified range
FOR r = 0 TO N - 1
   XSqr = 0
   FOR m = kStart1 TO (jFinish1 - r)
      XSqr = XSqr + (TZ(m) - TZ(m - 1)) * (TZ(m + r) - TZ(m + r - 1))
   NEXT m
   Rx(r) = XSqr / (N - r)
   Acor(r) = Rx(r) / Ro
NEXT r
```

```
AcorSum = 0
FOR r = 0 TO CINT(N / 2)
   AcorSum = AcorSum + Acor(r)
NEXT r
AcorAvg = AcorSum / (CINT(N / 2))

'  Curve Fit the Local Maximum Amplitudes for 121 start and finish combinations
one = 0: two = 0: three = 0: four = 0: ysum = 0: DelTZsum = 0
Tmid = TampMax(kStart1) + (TampMax(jFinish1) - TampMax(kStart1)) / 2
FOR i = kStart1 TO jFinish1
   one = one + (Tmid - TampMax(i)) ^ 2
   two = two + LOG(AmpMax(i))
   three = three + ((Tmid - TampMax(i)) ^ 2) * (LOG(AmpMax(i)))
   four = four + (Tmid - TampMax(i)) ^ 4
   ysum = ysum + AmpMax(i)
   DelTZsum = DelTZsum + (TZ(i + 1) - TZ(i - 1))
NEXT i
B = ((one * two / N) - three) / (four - ((one ^ 2) / N))
C = (two + (B * one)) / N

ym = ysum / N
yiym = 0: yiyc = 0
FOR i = kStart1 TO jFinish1
   y = EXP(C - B * (Tmid - TampMax(i)) ^ 2)
   yiym = yiym + (AmpMax(i) - ym) ^ 2
   yiyc = yiyc + (AmpMax(i) - y) ^ 2
NEXT i
sigy = (yiym / (N - 1))
sigx = (yiyc / (N - 2))
IF sigx < sigy THEN Rrr = SQR(1 - (sigx / sigy)) ELSE Rrr = 0
IF Rrr > RrrMax THEN
   RrrMax = Rrr: RrrSt = kStart1: RrrFt = jFinish1: RrrAcorAvg = AcorAvg
   RrrB = B: RrrC = C: RrrTmid = Tmid: RrrN = N: RrrDelTZsum = DelTZsum
END IF
SkipItAll:
NEXT jFinish1
Kinc = Kinc - 1
NEXT kStart1
LOCATE 2, 75: PRINT SPACE$(2)

Rrr = RrrMax: Start1 = RrrSt: Finish1 = RrrFt: AcorAvg = RrrAcorAvg
B = RrrB: C = RrrC: Tmid = RrrTmid: N = RrrN: DelTZsum = RrrDelTZsum

FDopp = N / (DelTZsum * 10 ^ (-6))
MaxAmp = EXP(C)
IF B > 0 THEN NTime = ((Tmid + SQR(2 / B)) - (Tmid - SQR(2 / B))) ELSE NTime = 0
NFring = (NTime * 10 ^ (-6)) * FDopp
```

```
Velocity = FDopp * Lambda / (2 * SIN(Kap))

GOSUB Plot2
RETURN


Plot1:  'This subroutine plots the burst, zero crossings, and local max amplitudes
upper = 0
DO
upper = upper + 2
LOOP UNTIL upper > Max
lower = -.5

CLS  '  Set up Window
WINDOW (0, lower)-(Tinc(Nmax), upper)
LINE (0, 0)-(Nmax, 0), 5
FOR i = 0 TO Tinc(Nmax) STEP 2
   FOR j = lower TO upper STEP .25
      PSET (i, j), 8
   NEXT j
NEXT i
FOR i = 0 TO Tinc(Nmax) STEP 10
   PSET (i, lower), 8
   FOR j = lower TO upper STEP .5
      LINE -(i, j), 8
   NEXT j
NEXT i
LINE (0, lower)-(0, upper), 5

'  Plot Absolute Burst Voltage
PSET (Tinc(1), ABS(Volt(1))), 7
FOR i = 2 TO Nmax
   LINE -(Tinc(i), ABS(Volt(i))), 6
   'PSET (i, Volt(i)), 14
NEXT i

'  Plot Zero Crossings
FOR i = 1 TO Tmax - 1
   PSET (TZ(i), 0), 11
   CIRCLE (TZ(i), 0), .1, 11
NEXT i

'  Plot Local Maximum Amplitudes
FOR i = 1 TO Tmax - 2
   PSET (TampMax(i), AmpMax(i)), 15
   CIRCLE (TampMax(i), AmpMax(i)), .1, 15
NEXT i
```

```
LINE (TZ(Start1), -.02)-(TZ(Start1), -.25), 10
LINE (TZ(Start1 - 1), -.1)-(TZ(Start1), -.02), 10
LINE (TZ(Start1), -.02)-(TZ(Start1 + 1), -.1), 10
LINE (TZ(Finish1), -.02)-(TZ(Finish1), -.25), 12
LINE (TZ(Finish1 - 1), -.1)-(TZ(Finish1), -.02), 12
LINE (TZ(Finish1), -.02)-(TZ(Finish1 + 1), -.1), 12

RETURN


Plot2:    ' This subroutine plots the local maximum amplitude curvefit
Start2 = 0
IF B > 0 THEN a = Tmid - SQR(2 / B) ELSE a = 1
DO
Start2 = Start2 + .05
y = EXP(C - B * (Tmid - Start2) ^ 2)
LOOP UNTIL y >= EXP(C - 2)
Finish2 = Start2
DO
Finish2 = Finish2 + .05
y = EXP(C - B * (Tmid - Finish2) ^ 2)
LOOP UNTIL y <= EXP(C - 2)

y = EXP(C - B * (Tmid - (Start2 - .05)) ^ 2)
PSET ((Start2 - .05), y), 10
FOR i = (Start2) TO (Finish2) STEP .05
   y = EXP(C - B * (Tmid - i) ^ 2)
   IF y > 1000 THEN y = 1000
   LINE -(i, y), 10
NEXT i

yexpmax = EXP(C - 2)
IF B > 0 THEN LINE ((Tmid - SQR(2 / B)), yexpmax)-((Tmid + SQR(2 / B)), yexpmax),
15

LINE (0, lower)-(0, upper), 5
LOCATE 5, 50: PRINT USING "Correlation Coefficient = #.###"; Rrr
LOCATE 6, 50: PRINT USING "Average Autocorrelation = #.###"; AcorAvg
LOCATE 7, 50: PRINT USING "Number of Fringes = ##.##"; NFring
LOCATE 8, 50: PRINT USING "Doppler Frequency =##.##^^^^ Hz"; FDopp
LOCATE 9, 50: PRINT USING "Maximum Amplitude =##.## V"; MaxAmp
LOCATE 10, 50: PRINT USING "Particle Velocity =##.## m/s"; Velocity

RETURN
```

```
FileRead:   'This subroutine does the fancy file management by printing the
            'directory to the screen and allowing the user to cursor through
            'the directory and select the appropriate file.
CLS
COLOR 15
Quest$ = "N"
ChangeDir:
LOCATE 1, 25: COLOR 15: PRINT "Press RETURN to select, ESC to change search
string"
IF NSA$ = "Y" THEN LOCATE 2, 25: COLOR 11: PRINT "Press F1 to create a new file"
LOCATE 1, 3: COLOR 10: PRINT "FILES:"
IF Quest$ = "Y" THEN
   LOCATE 1, 10: PRINT dir$ + CHR$(2)
   DO
      DO
      d$ = INKEY$
      LOOP UNTIL d$ <> ""
      IF ASC(d$) = 8 AND LEN(dir$) > 0 THEN
         dir$ = LEFT$(dir$, (LEN(dir$) - 1))
      END IF
      IF ASC(d$) <> 13 AND ASC(d$) <> 8 THEN dir$ = dir$ + d$
      LOCATE 1, 10: PRINT dir$ + CHR$(2) + SPACE$(1)
   LOOP UNTIL d$ = CHR$(13)
END IF

CLS
LOCATE 1, 25: COLOR 15: PRINT "Press RETURN to select, ESC to change search
string"
IF NSA$ = "Y" THEN LOCATE 2, 25: COLOR 11: PRINT "Press F1 to create a new file"
LOCATE 1, 3: COLOR 10: PRINT "FILES:"
LOCATE 1, 10: COLOR 10: PRINT dir$

SHELL "dir " + dir$ + " > wav.tmp"
OPEN "wav.tmp" FOR INPUT AS #1
i = 1
DO UNTIL EOF(1)
   INPUT #1, What$(i)
   i = i + 1
LOOP
CLOSE #1
SHELL "del wav.tmp"
imax = i
filemax = imax - 8

COLOR 7
rows = 20
roff = 2
```

```
columax = INT(filemax / rows) + 1
lcfmax = filemax - rows * (columax - 1)
IF columax > 5 THEN
   columax = 5
   LOCATE 2, 5: COLOR 12
   PRINT "Number of files exceeds screen limits.  Press <PGDN> for more."
   lcfmax = rows
END IF
COLOR 7
column = -16
jjinc = 0
FOR co = 1 TO columax
   column = column + 17
   FOR r = 1 TO rows
      jjinc = jjinc + 1
      LOCATE (r + roff), column
      IF jjinc <= filemax THEN
         WhereFile$(r, co) = LEFT$(What$(jjinc + 5), 12)
         PRINT WhereFile$(r, co)
      END IF
   NEXT r
NEXT co

r = 1: co = 1
co1 = 17 * co - 16
LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
DO
   DO
      Choice$ = INKEY$
   LOOP WHILE Choice$ = ""
   IF LEN(Choice$) = 1 THEN
      SELECT CASE Choice$
         CASE CHR$(ESC)
            Quest$ = "Y"
            GOTO ChangeDir
      END SELECT
   END IF
   IF LEN(Choice$) <> 1 THEN
      Choice$ = RIGHT$(Choice$, 1)
      SELECT CASE Choice$
         CASE CHR$(DOWN)
            LOCATE (r + roff), co1: COLOR 7: PRINT WhereFile$(r, co)
            r = r + 1
            IF r > rows THEN r = 1: co = co + 1
            IF co > columax THEN co = 1: r = 1
            IF co = columax AND r > lcfmax THEN r = 1: co = 1
            co1 = 17 * co - 16
```

```
      LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
   CASE CHR$(UP)
      LOCATE (r + roff), co1: COLOR 7: PRINT WhereFile$(r, co)
      r = r - 1
      IF r < 1 AND co = 1 THEN r = lcfmax: co = columax
      IF r < 1 AND co > 1 THEN r = rows: co = co - 1
      co1 = 17 * co - 16
      LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
   CASE CHR$(RIGHT)
      LOCATE (r + roff), co1: COLOR 7: PRINT WhereFile$(r, co)
      co = co + 1
      IF co > columax THEN co = 1
      IF co = columax AND r > lcfmax THEN r = lcfmax
      co1 = 17 * co - 16
      LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
   CASE CHR$(LEFT)
      LOCATE (r + roff), co1: COLOR 7: PRINT WhereFile$(r, co)
      co = co - 1
      IF co < 1 THEN co = columax
      IF co = columax AND r > lcfmax THEN r = lcfmax
      co1 = 17 * co - 16
      LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
   CASE CHR$(Funk1)
      LOCATE 1, 3: PRINT SPACE$(20)
      LOCATE 1, 3: COLOR 10: PRINT "FILE:"
      LOCATE 1, 10: PRINT CHR$(2): dsum$ = ""
      DO
         DO
         d$ = INKEY$
         LOOP UNTIL d$ <> ""
         IF ASC(d$) = 8 AND LEN(dsum$) > 0 THEN
            dsum$ = LEFT$(dsum$, (LEN(dsum$) - 1))
         END IF
         IF ASC(d$) <> 13 AND ASC(d$) <> 8 THEN dsum$ = dsum$ + d$
         LOCATE 1, 10: PRINT dsum$ + CHR$(2) + SPACE$(1)
      LOOP UNTIL d$ = CHR$(13)
      RETURN
   CASE CHR$(PGDN)
      IF columax = 5 AND lcfmax = rows THEN
         CLS
         columax = INT((filemax - 100) / rows) + 1
         lcfmax = (filemax - 100) - rows * (columax - 1)
         IF columax <= 5 THEN LOCATE 2, 5: PRINT SPACE$(63)
         IF columax > 5 THEN
            columax = 5
            lcfmax = rows
            LOCATE 2, 5: COLOR 12
```

```
                PRINT "Number of files exceeds screen limits.  Press <PGDN> for
                                        more."
            END IF
            COLOR 7
            column = -16: jjinc = 100
            FOR cj = 1 TO columax
                column = column + 17
                FOR ri = 1 TO rows
                    jjinc = jjinc + 1
                    LOCATE (ri + roff), column
                    IF jjinc <= filemax THEN
                        WhereFile$(ri, cj) = LEFT$(What$(jjinc + 5), 12)
                        PRINT WhereFile$(ri, cj)
                    END IF
                NEXT ri
            NEXT cj
            r = 1: co = 1
            LOCATE 1, 25: COLOR 15: PRINT "Press RETURN to select, ESC to
                            change search string"
            IF NSA$ = "Y" THEN LOCATE 2, 25: COLOR 11: PRINT "Press F1 to
                            create a new file"
            LOCATE 1, 3: COLOR 10: PRINT "FILES:"
            LOCATE 1, 10: COLOR 10: PRINT dir$
            co1 = 17 * co - 16
            LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
        END IF

    END SELECT
  END IF
  Filed$ = WhereFile$(r, co)
LOOP UNTIL Choice$ = CHR$(RETURF)
Filed$ = LEFT$(Filed$, (INSTR(Filed$, " ") - 1)) + "." + RIGHT$(Filed$, 3)
CLS
RETURN


Waveform:  'This subroutine calculates the zero crossings and local
        'maximum amplitudes for the input file.
dir$ = dirwf$
GOSUB FileRead
dirwf$ = dir$
File1$ = Filed$
OPEN File1$ FOR INPUT AS #1
FOR i = 1 TO 3
  INPUT #1, crap$
NEXT i
Max = 0
```

```
FOR i = 1 TO Nmax
   INPUT #1, Tinc(i), Volt(i)
   IF Volt(i) > Max THEN Max = Volt(i)
NEXT i
CLOSE #1

'  Find Zero Crossings and Max Amplitude
j = 1
FOR i = 1 TO Nmax
   Tmax = i
   Amax = 0
   IF Volt(j) > 0 THEN
      DO
         j = j + 1
         IF j > Nmax THEN GOTO EndAZ
         IF ABS(Volt(j)) > Amax THEN Amax = ABS(Volt(j)): TAmax = Tinc(j)
      LOOP UNTIL Volt(j) < 0
      GOTO Calc
   END IF
   IF Volt(j) < 0 THEN
      DO
         j = j + 1
         IF j > Nmax THEN GOTO EndAZ
         IF ABS(Volt(j)) > Amax THEN Amax = ABS(Volt(j)): TAmax = Tinc(j)
      LOOP UNTIL Volt(j) > 0
   END IF
Calc:
   IF Volt(j) <> 0 THEN
      TZ(i) = Tinc(j - 1) - (Volt(j - 1)) * (Tinc(j - 1) - Tinc(j)) / (Volt(j - 1) - Volt(j))
   END IF
   IF Volt(j) = 0 THEN
      TZ(i) = Tinc(j)
      j = j + 1
   END IF
   IF i > 1 THEN
      AmpMax(i - 1) = Amax
      TampMax(i - 1) = TAmax
   END IF
NEXT i
EndAZ:
Start1 = Tmax / 3
Finish1 = 2 * Tmax / 3
GOSUB Plot1
RETURN
```

```
NextSave:     ' This subroutine saves the calculated parameters
CLS          ' to an appendable file
NS = NS + 1
NSA$ = "Y"
dir$ = dirns$
GOSUB FileRead
dirns$ = dir$
NSA$ = "N"
IF Choice$ = CHR$(RETURF) THEN
   File3$ = Filed$
   OPEN File3$ FOR APPEND AS #1
   PRINT #1, NS, MaxAmp, NFring, Velocity, FDopp, Rrr, AcorAvg
   CLOSE #1
END IF
IF Choice$ = CHR$(Funk1) THEN
   File3$ = dsum$
   OPEN File3$ FOR OUTPUT AS #1
   PRINT #1, Lambda, Kap
   PRINT #1, NS, MaxAmp, NFring, Velocity, FDopp, Rrr, AcorAvg
   CLOSE #1
END IF

GOSUB Plot1
GOSUB Plot2
RETURN


TecPlot:  ' This subroutine sends the data and curvefit points to a TecPlot file
CLS : LOCATE 5, 5: PRINT "Please enter TecPlot-format output file name:"
PRINT : PRINT : PRINT : PRINT : FILES "*.prn"
LOCATE 7, 10: PRINT "Data filename: "; File1$
LOCATE 8, 10: INPUT " New filename: ", file2$

SCALE$ = "A"
PLAY "o1 X" + VARPTR$(SCALE$)

OPEN file2$ FOR OUTPUT AS #1
PRINT #1, "VARIABLES = X, Y"
PRINT #1, "ZONE T=DATA"
FOR i = 1 TO Nmax
   PRINT #1, Tinc(i), (Volt(i) + 5)
NEXT i
PRINT #1, "ZONE T=CURVED"
FOR i = (Start2 - .05) TO Finish2 STEP .05
   y = EXP(C - B * (Tmid - i) ^ 2)
   PRINT #1, i, (y + 5)
NEXT i
```

```
PRINT #1, "ZONE T=EXPSQR"
PRINT #1, (Tmid - SQR(2 / B)), (yexpmax + 5)
PRINT #1, (Tmid + SQR(2 / B)), (yexpmax + 5)
CLOSE #1
GOSUB Plot1
GOSUB Plot2

SCALE$ = "A"
PLAY "o0 X" + VARPTR$(SCALE$)
RETURN


Start:    'This subroutine interactively locates the burst start marker
LOCATE 1, 1: PRINT SPACE$(80): LOCATE 2, 1: PRINT SPACE$(80)
LOCATE 3, 1: PRINT SPACE$(80): LOCATE 4, 1: PRINT SPACE$(80)
COLOR 15: LOCATE 4, 1: PRINT
"________________________________________________________________________________"
LOCATE 2, 20: PRINT "SELECT STARTING POINT WITH < AND > KEYS"
LOCATE 3, 20: PRINT "      Press RETURN to save point"

LINE (TZ(Start1), -.02)-(TZ(Start1), -.25), 10
LINE (TZ(Start1 - 1), -.1)-(TZ(Start1), -.02), 10
LINE (TZ(Start1), -.02)-(TZ(Start1 + 1), -.1), 10

DO
a$ = INKEY$
LOOP UNTIL a$ = "," OR a$ = "."

DO
   LINE (TZ(Start1), -.02)-(TZ(Start1), -.25), 0
   LINE (TZ(Start1 - 1), -.1)-(TZ(Start1), -.02), 0
   LINE (TZ(Start1), -.02)-(TZ(Start1 + 1), -.1), 0

   IF a$ = "." THEN Start1 = Start1 + 1
   IF a$ = "," THEN Start1 = Start1 - 1
   IF Start1 < 1 THEN Start1 = 1

   LINE (TZ(Start1), -.02)-(TZ(Start1), -.25), 10
   LINE (TZ(Start1 - 1), -.1)-(TZ(Start1), -.02), 10
   LINE (TZ(Start1), -.02)-(TZ(Start1 + 1), -.1), 10

   LOCATE 5, 3: PRINT USING " Start time = ###.### micro sec"; TampMax(Start1)
   LOCATE 6, 3: PRINT USING "Finish time = ###.### micro sec"; TampMax(Finish1)

   DO
   a$ = INKEY$
   LOOP UNTIL a$ <> ""
```

```
    IF ASC(a$) = 13 THEN a$ = "Y"

LOOP UNTIL a$ = "Y"

RETURN


Finish:   'This subroutine interactively locates the burst finish markers
LOCATE 1, 1: PRINT SPACE$(80): LOCATE 2, 1: PRINT SPACE$(80)
LOCATE 3, 1: PRINT SPACE$(80): LOCATE 4, 1: PRINT SPACE$(80)
COLOR 15: LOCATE 4, 1: PRINT
"________________________________________________________________________________"
LOCATE 2, 20: PRINT " SELECT ENDING POINT WITH < AND > KEYS"
LOCATE 3, 20: PRINT "       Press RETURN to save point"

LINE (TZ(Finish1), -.02)-(TZ(Finish1), -.25), 12
LINE (TZ(Finish1 - 1), -.1)-(TZ(Finish1), -.02), 12
LINE (TZ(Finish1), -.02)-(TZ(Finish1 + 1), -.1), 12

DO
a$ = INKEY$
LOOP UNTIL a$ = "," OR a$ = "."

DO
    LINE (TZ(Finish1), -.02)-(TZ(Finish1), -.25), 0
    LINE (TZ(Finish1 - 1), -.1)-(TZ(Finish1), -.02), 0
    LINE (TZ(Finish1), -.02)-(TZ(Finish1 + 1), -.1), 0

    IF a$ = "." THEN Finish1 = Finish1 + 1
    IF a$ = "," THEN Finish1 = Finish1 - 1
    IF Finish1 < 1 THEN Finish1 = 1

    LINE (TZ(Finish1), -.02)-(TZ(Finish1), -.25), 12
    LINE (TZ(Finish1 - 1), -.1)-(TZ(Finish1), -.02), 12
    LINE (TZ(Finish1), -.02)-(TZ(Finish1 + 1), -.1), 12

    LOCATE 5, 3: PRINT USING " Start time = ###.### micro sec"; TampMax(Start1)
    LOCATE 6, 3: PRINT USING "Finish time = ###.### micro sec"; TampMax(Finish1)

    DO
    a$ = INKEY$
    LOOP UNTIL a$ <> ""
    IF ASC(a$) = 13 THEN a$ = "Y"

LOOP UNTIL a$ = "Y"

RETURN
```

```
'******************************************************************************
'                          MANALYZ1.BAS
'
'  This program analyzes the data file created in MANAZAC2 and produces
'  the corrected particle amplitude.
'  INPUT parameters: half angle, laser wavelength, Amax, N, u
'  OUTPUT parameters: corrected amplitude, average corrected amplitude
'     and standard deviation, average velocity
'
'******************************************************************************

DEFDBL A-Z
N = 50
DIM Ad0(N), MaxAmp(N), Nfring(N), Velocity(N), FDopp(N), Rrr(N), AcorAvg(N)
DIM What$(500), WhereFile$(50, 6), Skip(N)
CONST PI = 3.141592654#
CONST ESC = 27, DOWN = 80, UP = 72, LEFT = 75, RIGHT = 77
CONST PGDN = 81, RETURF = 13, Funk1 = 59

Lambda = 514 / 1.33 * 10 ^ (-9)
DE2 = (.65) * 10 ^ (-3)
dir$ = "*.DAT"

top:
SCREEN 12: COLOR 15: CLS

GOSUB FileRead

CLS
LOCATE 5, 5
COLOR 14: PRINT "     Enter the focal length (mm, please)        FL = "; : COLOR 15:
INPUT "", FL

OPEN Filed$ FOR INPUT AS #1
INPUT #1, Lambda, Kap
PRINT
Max = 0
i = 1

DO WHILE NOT EOF(1)
   INPUT #1, NS, MaxAmp(i), Nfring(i), Velocity(i), FDopp(i), Rrr(i), AcorAvg(i)
   i = i + 1
LOOP
CLOSE #1
Max = i - 1
```

```
dm = 4 * FL * 10 ^ (-3) * Lambda / (PI * DE2)
df = Lambda / (2 * SIN(Kap))
NMax = dm / df
s = 1


DO
'         Calculate the averages
Ad0Sum = 0: Asum = 0: usum = 0: Nsum = 0
FOR i = 1 TO Max
   FOR j = 1 TO s
      IF Skip(j) = i THEN GOTO SkipItAll1
   NEXT j
   Ad0(i) = MaxAmp(i) * EXP(2 * (1 - (Nfring(i) / NMax) ^ 2))
   Ad0Sum = Ad0Sum + Ad0(i)
   Asum = Asum + MaxAmp(i)
   usum = usum + Velocity(i)
   Nsum = Nsum + Nfring(i)
SkipItAll1:
NEXT i

BMax = Max - (s - 1)
AvgAd0 = Ad0Sum / BMax: AvgAmax = Asum / BMax
AvgVel = usum / BMax: AvgN = Nsum / BMax
Ad0Sqr = 0
'         Calculate the standard deviation
FOR i = 1 TO Max
   FOR j = 1 TO s
      IF Skip(j) = i THEN GOTO SkipItAll2
   NEXT j
   Ad0Sqr = Ad0Sqr + (Ad0(i) - AvgAd0) ^ 2
SkipItAll2:
NEXT i
Ad0STD = SQR(Ad0Sqr / (BMax - 1))

'         Print results to screen
CLS : PRINT : PRINT
PRINT "                    "; Filed$
PRINT
PRINT "   i    AMax     N     Ad0    di/σ     u    "
p$ = "  ##   #.###   ##.#   ##.##   #.##   ##.##  "
FOR i = 1 TO Max
   FOR j = 1 TO s
      IF Skip(j) = i THEN GOTO SkipItAll3
   NEXT j
   disig = ABS(AvgAd0 - Ad0(i)) / Ad0STD  ' This is Chauvenet's criterion
   PRINT USING p$; i; MaxAmp(i); Nfring(i); Ad0(i); disig; Velocity(i)
```

```
SkipItAll3:
NEXT i

pp$ = " AVGS   #.###   ##.#   ##.##          ##.##  "
PRINT : PRINT USING pp$; AvgAmax; AvgN; AvgAd0; AvgVel
PRINT USING "  Nmax = ##.#   Ad0STD = #.###"; NMax; Ad0STD
PRINT
'       Based on Chauvenet's criterion and intuition, prompt to reject points
PRINT "  Delete some data points? ";
DO: ques$ = INKEY$: LOOP WHILE ques$ = ""
PRINT : ques$ = UCASE$(ques$)
IF ques$ = "Y" THEN
   DO
      INPUT "    Enter number to skip (0 to analyze) -> ", Skip(s)
      s = s + 1
   LOOP UNTIL Skip(s - 1) = 0
   s = s - 1
END IF
LOOP WHILE ques$ = "Y"

PRINT
PRINT "   Try again?"
DO
b$ = UCASE$(INKEY$)
LOOP WHILE b$ = ""
IF b$ = "Y" THEN REDIM Skip(50): GOTO top:   ' Reset the Skip array before restarting

END


FileRead:    'This subroutine does the fancy file management by printing the
             'directory to the screen and allowing the user to cursor through
             'the directory and select the appropriate file.
COLOR 15
Quest$ = "N"
ChangeDir:
LOCATE 1, 25: COLOR 15: PRINT "Press RETURN to select, ESC to change search
string"
LOCATE 1, 3: COLOR 10: PRINT "FILES:"
IF Quest$ = "Y" THEN
   LOCATE 1, 10: PRINT dir$ + CHR$(2)
   DO
      DO
      d$ = INKEY$
      LOOP UNTIL d$ <> ""
      IF ASC(d$) = 8 AND LEN(dir$) > 0 THEN
         dir$ = LEFT$(dir$, (LEN(dir$) - 1))
```

```
        END IF
        IF ASC(d$) <> 13 AND ASC(d$) <> 8 THEN dir$ = dir$ + d$
        LOCATE 1, 10: PRINT dir$ + CHR$(2) + SPACE$(1)
    LOOP UNTIL d$ = CHR$(13)
END IF

CLS
LOCATE 1, 25: COLOR 15: PRINT "Press RETURN to select, ESC to change search
string"
LOCATE 1, 3: COLOR 10: PRINT "FILES:"
LOCATE 1, 10: COLOR 10: PRINT dir$

SHELL "dir " + dir$ + " > wav.tmp"
OPEN "wav.tmp" FOR INPUT AS #1
i = 1
DO UNTIL EOF(1)
    INPUT #1, What$(i)
    i = i + 1
LOOP
CLOSE #1
SHELL "del wav.tmp"
imax = i
filemax = imax - 8

COLOR 7
rows = 20
roff = 2
columax = INT(filemax / rows) + 1
lcfmax = filemax - rows * (columax - 1)
IF columax > 5 THEN
    columax = 5
    LOCATE 2, 5: COLOR 12
    PRINT "Number of files exceeds screen limits.  Press <PGDN> for more."
    lcfmax = rows
END IF
COLOR 7
column = -16
jjinc = 0
FOR co = 1 TO columax
    column = column + 17
    FOR r = 1 TO rows
        jjinc = jjinc + 1
        LOCATE (r + roff), column
        IF jjinc <= filemax THEN
            WhereFile$(r, co) = LEFT$(What$(jjinc + 5), 12)
            PRINT WhereFile$(r, co)
        END IF
```

```
    NEXT r
NEXT co

r = 1: co = 1
co1 = 17 * co - 16
LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
DO
    DO
        Choice$ = INKEY$
    LOOP WHILE Choice$ = ""
    IF LEN(Choice$) = 1 THEN
        SELECT CASE Choice$
            CASE CHR$(ESC)
                Quest$ = "Y"
                GOTO ChangeDir
        END SELECT
    END IF
    IF LEN(Choice$) <> 1 THEN
        Choice$ = RIGHT$(Choice$, 1)
        SELECT CASE Choice$
            CASE CHR$(DOWN)
                LOCATE (r + roff), co1: COLOR 7: PRINT WhereFile$(r, co)
                r = r + 1
                IF r > rows THEN r = 1: co = co + 1
                IF co > columax THEN co = 1: r = 1
                IF co = columax AND r > lcfmax THEN r = 1: co = 1
                co1 = 17 * co - 16
                LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
            CASE CHR$(UP)
                LOCATE (r + roff), co1: COLOR 7: PRINT WhereFile$(r, co)
                r = r - 1
                IF r < 1 AND co = 1 THEN r = lcfmax: co = columax
                IF r < 1 AND co > 1 THEN r = rows: co = co - 1
                co1 = 17 * co - 16
                LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
            CASE CHR$(RIGHT)
                LOCATE (r + roff), co1: COLOR 7: PRINT WhereFile$(r, co)
                co = co + 1
                IF co > columax THEN co = 1
                IF co = columax AND r > lcfmax THEN r = lcfmax
                co1 = 17 * co - 16
                LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
            CASE CHR$(LEFT)
                LOCATE (r + roff), co1: COLOR 7: PRINT WhereFile$(r, co)
                co = co - 1
                IF co < 1 THEN co = columax
                IF co = columax AND r > lcfmax THEN r = lcfmax
```

```
            co1 = 17 * co - 16
            LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
         CASE CHR$(PGDN)
            IF columax = 5 AND lcfmax = rows THEN
               CLS
               columax = INT((filemax - 100) / rows) + 1
               lcfmax = (filemax - 100) - rows * (columax - 1)
               IF columax <= 5 THEN LOCATE 2, 5: PRINT SPACE$(63)
               IF columax > 5 THEN
                  columax = 5
                  lcfmax = rows
                  LOCATE 2, 5: COLOR 12
                  PRINT "Number of files exceeds screen limits.  Press <PGDN> for
                                    more."
               END IF
               COLOR 7
               column = -16: jjinc = 100
               FOR cj = 1 TO columax
                  column = column + 17
                  FOR ri = 1 TO rows
                     jjinc = jjinc + 1
                     LOCATE (ri + roff), column
                     IF jjinc <= filemax THEN
                        WhereFile$(ri, cj) = LEFT$(What$(jjinc + 5), 12)
                        PRINT WhereFile$(ri, cj)
                     END IF
                  NEXT ri
               NEXT cj
               r = 1: co = 1
               LOCATE 1, 25: COLOR 15: PRINT "Press RETURN to select, ESC to
                              change search string"
               IF NSA$ = "Y" THEN LOCATE 2, 25: COLOR 11: PRINT "Press F1 to
                              create a new file"
               LOCATE 1, 3: COLOR 10: PRINT "FILES:"
               LOCATE 1, 10: COLOR 10: PRINT dir$
               co1 = 17 * co - 16
               LOCATE (r + roff), co1: COLOR 14: PRINT WhereFile$(r, co)
            END IF

      END SELECT
   END IF
   Filed$ = WhereFile$(r, co)
LOOP UNTIL Choice$ = CHR$(RETURF)
Filed$ = LEFT$(Filed$, (INSTR(Filed$, " ") - 1)) + "." + RIGHT$(Filed$, 3)
CLS
RETURN
```