IN-35
1622
p44

NASA Technical Memorandum 103720

Software Manual for Operating Particle Displacement Tracking Data Acquisition and Reduction System

Mark P. Wernet
Lewis Research Center
Cleveland, Ohio

January 1991

NASA

PARTICLE DISPLACEMENT TRACKING

Mark P. Wernet
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

INTRODUCTION

Particle Imaging Velocimetry (PIV) is a quantitative velocity measurement technique for measuring instantaneous planar cross sections of a flow field.[1-4] In traditional PIV, the light from a high power double pulsed laser source is formed into a thin (1mm thick) sheet and used to illuminate a planar cross section of a seeded flow. The only restriction on the seed particle size is that they have a small enough aerodynamic diameter to accurately follow the fluid motion. In the traditional PIV setup, the images of the particles illuminated in the double pulse exposure are recorded on photographic plates. The quantitative velocity data is obtained by using a low power laser beam to interrogate the photograph. There are different regimes of particle density which ultimately affect the details of the data reduction technique. Typically a Young's fringe method is used to estimate the velocity vector data. The low power laser beam is approximately 1mm in diameter, hence, the recorded photograph must be evaluated at many thousands of points to obtain the whole velocity vector field. The processing time per interrogation point varies from a few seconds to a few minutes, depending on the array processing and computer hardware employed. More recently, a modification to the traditional PIV technique was introduced which eliminated the directional ambiguity problem.[5]

The PIV technique offers the user very high precision (1%), and if performed properly, directionally resolved velocity vector estimates over an extended planar flow cross section. The disadvantages of the PIV technique are the high equipment costs and complexity of operation. Special array processors are required to perform the data reduction. The processing time per photographic plate is several hours at best. The recorded photographic plates must be chemically processed, which increases the time lag between when the data were recorded and when the velocity vector information is obtained. Usually, it is difficult to ascertain if the recording conditions are properly set until after the photographic plates have been developed, which is long after the experiment ended.

The Particle Displacement Tracking (PDT) technique is an all electronic PIV data acquisition and reduction procedure. There has been other work devoted to the development of all, or partial electronic PIV.[6-12] None of these techniques, however, are as efficient and simple as the PDT technique. The PDT technique uses a low power, continuous wave laser and a Charged Coupled Device (CCD) camera to electronically record the particle images[13]. A frame-grabber board in a PC computer supplies all of the data acquisition support and data reduction processing. No chemical processing steps are involved. No specialized array processors are required. The particle images can be viewed on an RGB monitor prior to acquiring the data to insure that all of the recording parameters are properly set. The data processing time is typically less than 100 seconds in total on an 80386 PC. The accuracy of the velocity estimates is on the order of 5%, and varies inversely with the magnitude of the velocity vector. In summary, the PDT technique is a simple, fast, easily implemented data reduction technique for PIV data. No chemical processing steps are involved, the system costs are moderately low, and the reduced data are available for analysis seconds after the data are acquired.

This manual describes the hardware and software requirements for implementing the PDT system. No specific details are given about the laser source, optics, particulate seeding, or CCD cameras. These parameters vary depending on the specifics of the flow field under study. The reader is directed to the open literature on PIV for more information on selecting system parameters and hardware. No claim is made that the software routines for reducing and analyzing the data are completely optimized. Although much effort has been devoted to obtaining the fastest data reduction speeds possible, the user may determine more efficient methods for implementing the concepts defined in the PDT technique.

1). HARDWARE REQUIREMENTS

All of the data acquisition and processing routines discussed herein run under MS-DOS V3.0 and higher. The data acquisition routines are real mode programs, meaning they will run on an 8088/8087, 80286/80287, or an 80386/80387 based computer. However, the data reduction and graphical analysis programs are written in Fortran 77 and compiled using a DOS-Extender, which generates protected mode code. The data reduction programs will only run on an 80386 based computer which must have at least an 80387 coprocessor installed. A Weitek 3167 coprocessor may also be installed in addition to the 80387 math coprocessor. The Microway NDP Fortran Version 2.1 compiler used to compile the Fortran code supports both the 80387 or the Weitek 3167. The Weitek code will execute 3-5 times faster than the 80387 code.

The real mode data acquisition programs require approximately 100KBytes of memory space to run. The protected mode programs run in Extended memory above 1MByte. In order to run the protected mode program the 80386 based computer must have at least 3MBytes of Extended memory above the 1Mbyte base memory, for a total system memory of 4MBytes. The graphical analysis program uses 1MByte of extended memory to generate high resolution (300dpi) plots in the computer memory and then transfers the image to a printer. The default printer supported by the graphical analysis program is the HP Laserjet+ and HP Laserjet Series II with at least 1.5Mbytes of memory installed in the printer. Both the data acquisition and graphical analysis routines use menu based screens. The data acquisition routines require the mouse for operation, while the graphical analysis routines can use either the mouse or the keypad arrow keys.

All of the data acquisition and data reduction software use text graphics for the user interface. The text graphics support EGA graphics resolution and above. The only other graphic requirements are for the graphical analysis routine. The graphical analysis routine allows the user to display the acquired velocity vector data in a 2-D velocity vector map, and other graphing options. The supported graphic resolutions are EGA 640×350, VGA 640×480, and Video Seven VRAM VGA 1024×768 resolutions.

The data acquisition hardware was installed in a Compaq 386/25MHz computer, equipped with both an Intel 80387 and a Weitek 3167 coprocessor. The data acquisition board supported by the data acquisition routines is an EPIX 4-MEG video board. The EPIX 4-MEG video board was chosen because of the large on board memory buffering capacity and the extremely flexible image sizing memory architecture. The default EPIX board settings were used (interrupt vector=3, memory segment=0xD, dma channel=2, register base=0x280). The only modification was the use of a 12.5MHz oscillator, available from EPIX for the 4-MEG video board. The 12.5MHz oscillator controls the sampling frequency of the RS-170 video signal. Using this oscillator frequency produces nearly square pixels (@640×480 pixel resolution), hence, no aspect ratio correction is required in the velocity vector data. An RGB monitor is used to monitor the frame-grabber operations. Either a single frequency or multiple frequency RGB monitor can be used. Throughout the remainder of this document the RGB monitor connected to the EPIX 4-MEG video board will be referred to as the system monitor. The data can be acquired from a Charge Coupled Device (CCD) camera, or any other RS-170 compatible camera. The image acquisition format was 640×240 pixel video fields. The EPIX board allows acquisition of either fields or frames, fields were used to obtain the highest sampling frequency, 1/60 second. At this video resolution, 27 image buffers can be stored on the EPIX board. The images acquired from an RS-170 video source are interlaced, thus, every other row of a 640×480 pixel image would be blank, so only 240 lines are saved. For more information on the video formats see reference 14.

In summary, the hardware requirements are:

Computer System:
80386 based PC run MS-DOS Version 3.0 or higher
equipped with an 80387
optionally equipped with a Weitek 3167
4MBytes total system memory (3MBytes of Extended)
EGA, VGA, or Video Seven VRAM VGA w/512kbyte VRAM
Monitor to support the above graphics card
(NEC MultiSync 4D works well with the Video Seven VRAM VGA)
Microsoft compatible mouse

Hardcopy:
HP Laserjet+, HP Laserjet Series II, or
HP Laserjet Series III with at least 1.5MBytes of memory

Frame-Grabber:
EPIX 4-MEG Video board equipped with the optional 12.5MHz oscillator. The following EPIX software is also
required: 4MDRIVER C image acquisition subroutine library
PXIPL C image processing subroutine library
recommended: 4MIP interactive image processing package

System monitor:
RGB or composite video monitor to display images acquired from the EPIX 4-MEG video board, and to display the RS-170 video images from the CCD array camera used to acquire the video data.

Video Camera:
RS-170 video camera for acquiring the video images. The pixel resolution of the camera should be at least 600h×400v. The camera sensitivity depends on the amount of scattered light collected. For low light level sources and small particle sizes (5μm), an intensified camera is recommended.

2) SOFTWARE COMPILERS AND LIBRARIES

Two software compilers were used to produce the executable programs for data acquisition and reduction. The real mode data acquisition program was created via Microsoft C Version 5.1 compiler along with the Microsoft QuickC Version 2.0 screen graphics include file and graphics library. The mouse support for the text screen menus was obtained via the Microsoft mouse library 'MOUSE.LIB'. As mentioned above, the software support for the EPIX frame-grabber board functions were obtained via EPIX's software subroutine libraries.

The protected mode programs were written in Fortran 77 and compiled using Microway's NDP-Fortran-386 Version 2.1 compiler. The text screen menus were generated using the NDP built in graphics library 'LGREX.LIB'. The LGREX library also contains support for a Microsoft compatible mouse, which is used in conjunction with the screen menus in the graphical analysis program. The high resolution screen graphics and hardcopy support were obtained using Media Cybernetic's HALO Professional Fortran libraries. The HALO Professional libraries are compatible with Microway's NDP Fortran compiler.

In summary, the compilers and libraries used were:

Real Mode C program:

Microsoft C Version 5.1 Compiler	
Microsoft QuickC Version 2.0	
include file:	'GRAPH.H'
graphics library:	'GRAPHICS.LIB'
Microsoft Mouse Library:	'MOUSE.LIB'
EPIX subroutine library:	'M4OBJM.LIB'
EPIX Image Processing Library:	'PXIPLM.LIB'

Protected Mode Fortran Programs:

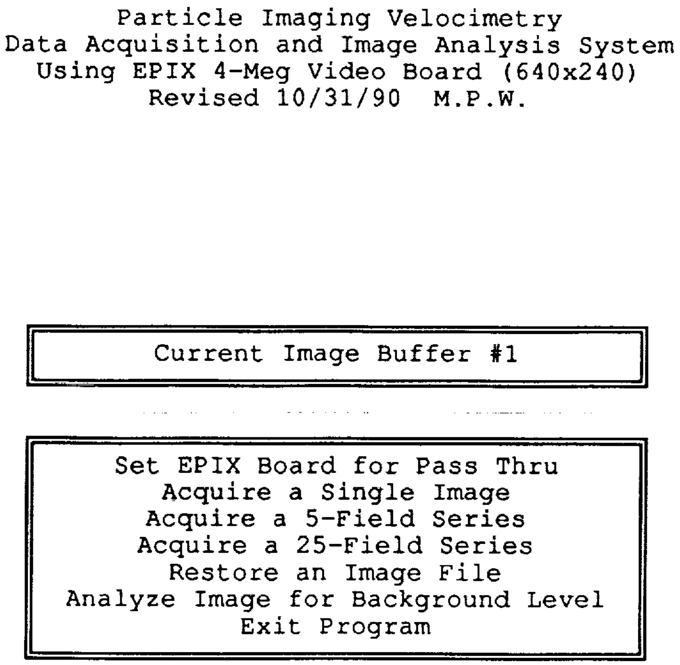
Microway NDP-Fortran-386 Version 2.1 Compiler with the built in screen graphics library:	'LGREX.LIB'
Media Cybernetics HALO Professional	
graphics interface module:	'AHDNDPF.OBJ'
graphics kernel:	'AHDNDPF.KRN'
screen drivers:	'AHDIMBE.DSP'
	'AHDIBMV.DSP'
	'AHDV7VP.DSP'
virtual graphics device:	'AHDVRI.DSP'
printer driver:	'AHDLJTP.PRT'
fonts:	'AHD106.FNT'
	'AHD107.FNT'
	'AHD201.FNT'

3). DATA ACQUISITION ROUTINES

The data acquisition routines are all written in Microsoft C Version 5.1 and can all be executed from a single menu based master program. The EPIX 4-MEG video board comes with a subroutine library of C routines. Two of the EPIX libraries were used: M4OBJM.LIB and PXIPLM.OBJ for the medium size memory model. The Microsoft Mouse (MOUSE.LIB) library was also used for mouse support. The screen graphics were done using Microsoft QuickC Version 2.0 graphics. The graphics include file was from QuickC Version 2.0, along with the graphics library. The QuickC graphics library is compatible with Microsoft C V5.1.

The C program 'PDTMNLS.C' performs all of the requisite data acquisition and permits analysis of the acquired images for boundaries and background level. The main menu is displayed at the start of the program. Both the keyboard and a Microsoft compatible mouse can be used to select menu items. The mouse is moved forward and back to select items, and the left button is depressed to execute a menu item. The keypad up and down arrow keys can also be used to select the menu items. To execute a menu item press the enter key. The home and end keys move the menu pointer to the top and bottom of the menu respectively. The pageup and pagedown keys are used to select the currently displayed image buffer on the system monitor. The buffer number currently being displayed is shown at the top of the menu "Current Image Buffer #" on the computer screen.

The PDTMNLS main menu appears below:



```
             Particle Imaging Velocimetry
        Data Acquisition and Image Analysis System
          Using EPIX 4-Meg Video Board (640x240)
               Revised 10/31/90  M.P.W.




                 Current Image Buffer #1


                Set EPIX Board for Pass Thru
                  Acquire a Single Image
                 Acquire a 5-Field Series
                Acquire a 25-Field Series
                   Restore an Image File
           Analyze Image for Background Level
                      Exit Program
```

The currently selected item has a highlight bar around it. By using the mouse or keypad arrow keys, the highlighted option can be changed. The function of each menu item is as follows:

Set EPIX Board For Pass Thru

Set the EPIX board so that the RS-170 video signal connected to the Video-In line is displayed on the system monitor in a pass through mode.

Acquire a Single Image

This option acquires the next full video frame (640×480) that comes in on the Video-In line and displays it on the system monitor. This is an interlaced full video frame. The user is queried for the full name of the file to store the image. This feature is used for evaluating the system setup, or acquiring a single image which may contain a reference scale.

Acquire a 5-Field Series

This option acquires a 5-field sequence of images. The five video fields are subsequently used to generate a single velocity vector map using the Fortran data reduction software routines. The user is queried for the number of inter-field intervals between the acquired fields. The acquired image size is 640 pixels horizontally × 240 lines. The field intervals are in multiples of 1/60 of a second. By selecting a large number of video fields to elapse between acquired images, the motion of low velocity particles can be recorded. Selecting 1 video field interval between acquired images corresponds to the maximum velocity which can be recorded, which acquires 5 adjacent video fields. The user is also prompted for the file name root to store the acquired image series. The name length must be no longer than 7 characters. A buffer number and suffix number are appended to the end of the entered file name root. For example, if the root 'TEST' is entered, the five images are stored in files TEST1.PIV, TEST2.PIV, TEST3.PIV, TEST4.PIV, and TEST5.PIV. The *.PIV suffix is used to denote raw particle image files.

Image acquisition begins after the user enters the file name root and presses the return key. At the end of acquisition, the image acquired into buffer #1 is displayed on the system monitor. The five video fields are stored in image buffers 1 through 5 on the EPIX board. The pageup/pagedown keys can be used to page through the acquired images. The top of the main menu indicates the currently displayed image buffer. The images remain in the first five buffers until another operation is performed. An additional operation performed by the software is to add together the images in buffer 1 through 5 and store the result in image buffer #6. By viewing image buffer #6, the user can evaluate whether an appropriate inter-field time interval has been selected for the flow field under study. Since the acquired sequence of video images will subsequently be processed individually, overlapping particle images in the summed image sequence displayed in buffer #6 are permitted. The particles must have moved by some detectable distance for the PDT technique to work. The minimum allowable particle displacement between two exposures is 1 pixel.

Acquire 25-Field Series

This option acquires five 5-field series, for a total of 25 sequential video fields. The 25-field series are used to produce 5 velocity vector map files via the Fortran data reduction routines. The five velocity vector files can be used to show the particle trajectories over an extended period (5 acquisition sets).

The user is queried for the number of inter-field intervals between the acquired fields. The acquired image size is 640 pixels horizontally × 240 lines. The field intervals are in multiples of 1/60 of a second. By selecting a large number of video fields to elapse between acquired images, the motion of low velocity particles can be recorded. Selecting 1 video field interval between acquired images corresponds to the maximum velocity which can be recorded, which acquires 25 adjacent video fields. The user is also prompted for the file name root to store the acquired image series. The name length must be no longer than 6 characters. A buffer number and suffix number are appended to the end of the entered file name root. For example, if the root 'TEST' is entered, the twenty-five images are stored in files TEST1.PIV, TEST2.PIV, TEST3.PIV, TEST4.PIV, ... TEST25.PIV. The '.PIV' suffix is used to denote raw image files.

Image acquisition begins after the user enters the file name root and presses the return key. At the end of acquisition, the image acquired into buffer #1 is displayed on the system monitor. The twenty-five video fields are stored in image buffers 1 through 25 on the EPIX board. The pageup/pagedown keys can be used to page through the acquired images. The top of the main menu indicates the currently displayed image buffer. The images remain in the first twenty-five buffers until another operation is performed.

Restore an Image File

This menu option restores an image file from disk to the EPIX board frame buffer #1, which can then be viewed on the system monitor. The image file can be either a video field (640×240) or a video frame (640×480). The software automatically detects the size of the image and changes the EPIX board to be in the appropriate interlace mode (on or off). The user is prompted for the full file name to be restored. This option is useful for examining a previously acquired and saved image.

Analyze Image for Boundaries

The purpose of this routine is to allow the user to determine the boundaries of the image. In subsequent processing steps, only the rectangular region containing the actual moving particles is desired. Bounding surfaces, or other unwanted sources of stray light in the periphery of the image can be eliminated by recording the coordinates of the active velocity area only. If the velocity field fills the entire field of view, or if there are no extraneous objects present in the image other than the particle images against a black background, then this step is not required.

When selected, this option queries the user for the name of an image file to be restored to the EPIX board frame buffer #1, which can then be viewed on the system monitor. The image file can be either a video field (640×240) or a video frame (640×480). The software automatically detects the size of the image and changes the EPIX board to be in the appropriate interlace mode (on or off). Once the image has been restored, a cursor with intersecting horizontal and vertical lines will appear on top of the image displayed on the system monitor. The cursor coordinates in pixel units are displayed on the computer screen. The mouse is used to move the cursor around on the system monitor. As the cursor moves on the system monitor, the coordinates on the computer screen are updated to show the current cursor coordinates. The coordinates displayed are the x (horizontal 0-639) and y (vertical 0-479). If the image being examined is a video field, the vertical coordinate increments by 2. For a full frame video image, the vertical coordinate increments by 1. By depressing the left button, the current cursor position is stored on the computer screen. The active cursor coordinates are now displayed underneath the stored coordinates on the computer screen. By depressing the left mouse button again the now current cursor coordinates are stored underneath the previously stored coordinates, and the active coordinates are displayed further down on the computer screen. The cursor coordinates are stored on the screen as many times as the left mouse button is depressed. By pressing the "PRINT SCREEN" key, a hardcopy of the screen will be transferred to the printer. After the images have been analyzed, the printer can be form fed to obtain the copy of the recorded cursor coordinates. The two pairs of coordinates which will be used in subsequent processing steps are the lower left corner and upper right corner coordinates. Hence, these two coordinate pairs should be recorded. The routine is terminated by depressing the right mouse button.

Figure 1 shows a sample PIV image with some glare on the right side of the figure. The particle image data do not completely fill the field of view. The boxed region shows the desired processing region. The x and y extents of the boxed region must be recorded as described above.

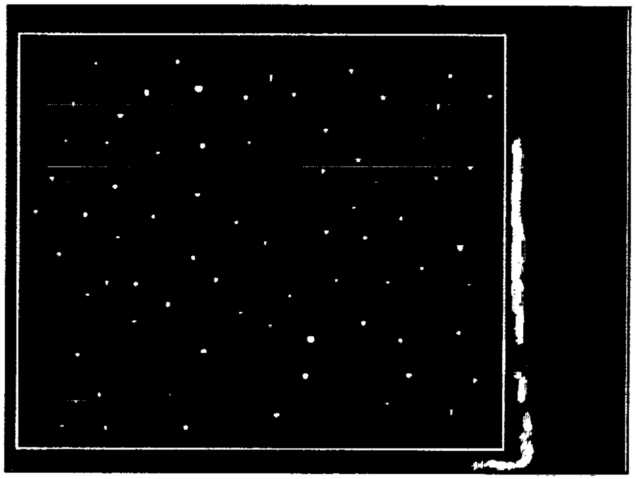


Figure 1: Particle image file showing some glare at the right of the figure. The boxed area indicates the region to be processed.

Analyze Image for Background Level

The purpose of the background analysis operation is to determine the appropriate image threshold level for the PARTICLE IMAGE BOUNDARY PROCESSING stage of data reduction. This option queries the user for the name of an image file to be restored to the EPIX board frame buffer #1, which can then be viewed on the system monitor. The image file can be either a video field (640×240) or a video frame (640×480). The software automatically detects the size of the image and changes the EPIX board to be in the appropriate interlace mode (on or off). Once the image has been restored, the computer screen will show the current threshold level in grey levels. The initial value is zero; all of the image information is displayed. By moving the mouse forward and backward, the threshold level can be increased or decreased. The actual threshold level currently in effect is displayed on the computer screen and the effect of the threshold operation on the image can be observed on the system monitor. This operation is terminated by depressing the left mouse button.

In order to accurately determine the particle centroids, the individual particle images must have high contrast. The background level **MUST** be made zero by adjusting the threshold level. Typically, when an image is acquired, camera noise or stray light tend to add a constant background level to the image. The background level may be as low as a few grey levels or as high as 250. For 8-bit images the maximum grey level is 256. Every effort should be made to reduce the amount of background light prior to acquiring the video images. Although high background levels can be removed, they reduce the dynamic range of the subsequent image. High dynamic range maximizes the accuracy of the particle centroid estimation.

The threshold level is adjusted until the particle images are displayed on a completely black background. The particle images must be clearly resolved individual entities. Adjacent particle images must not be touching. Figure 2a shows a close up view of some recorded particle images with a threshold level of 0. Notice that some of the particle images are connected by non-zero valued pixels. By increasing the threshold level, the non-zero pixel values in between the adjacent particle images can be eliminated yielding completely independent particle images. Figure 2b shows the same image as in 2a with a threshold level of 40 grey levels. The particle images are now clearly defined on a black background. The threshold level which produces the best image contrast is recorded. The background level analysis procedure typically need be done for only one image out of the acquired series, since the image to image background level variations are very small. Care must be taken to insure that not too many of the particle images are removed by the selected threshold level. There is a subjective tradeoff between eliminating faint particle images and clearly resolving the boundaries of brighter particle images.

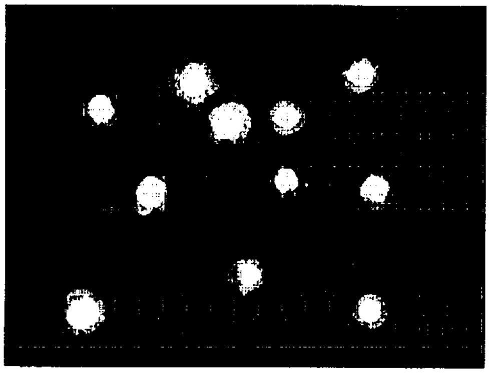


Figure 2a: Sample raw image file showing particle images with non-zero background level.

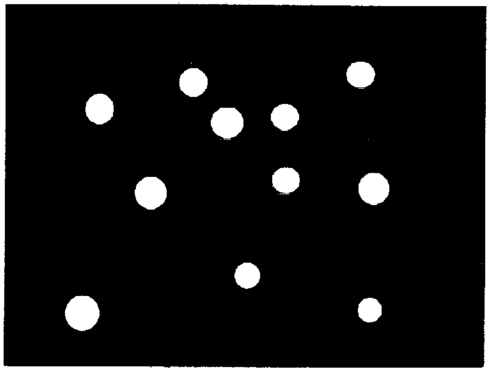


Figure 2b: Same image as shown in figure 2a, however, a threshold level has been used to clearly resolve the particle images and make the background level zero.

Exit Program

The function of this menu option is obvious. Program operation is terminated.

4). DATA REDUCTION ROUTINES

The data reduction routines were all written in Fortran 77 and compiled using Microway's NDP-Fortran-386 compiler for MS-DOS. All of the routines incorporate text based graphics via NDP's built-in graphics features. An EGA graphics board or better is required to use these text based graphics. The data reduction routines have previously been written to run in

640KBytes of memory for use with Microsoft's Fortran compiler. The conversion is simple, replace all of the INTEGER*4 image storage arrays in the NDP version with CHARACTER*1 arrays. The Microsoft Fortran compiler executes very efficiently with character data. Also, the grey level comparison statements must be changed to character comparisons, instead of integer comparisons. The text based graphics used in NDP Fortran must be modified for the Microsoft Fortran text graphics routines.

There are essentially two stages to the data reduction process. First, the acquired raw particle image *.PIV video fields obtained via the EPIX frame-grabber board must be processed to determine the particle image centroids. The particle image centroids are determined for each image in the 5 field sequence. A single time history particle image centroid file is then created which contains the single pixel positions of the particle images recorded on all 5 fields in the image sequence. The time history file format is 640×480 pixels by 8 bits, the same as the original image data. The pixels representing the particle positions have their amplitudes coded by 2^I, where I is the frame number in the sequence in which the particle was detected. The time history files then serve as the input data to the second stage of the data reduction process. A description of the boundary processing algorithm is given in Appendix I.

The second stage of the data reduction process is the Particle Displacement Tracking stage. The time history file is searched for the location of all pixels with amplitudes equal to 2^1, which corresponds to the initial particle locations in the 5 field sequence. The initial particle locations are then used as the center point of a circular search region. The distance between the 2^1 amplitude center pixel and 2^2 amplitude pixels in the search region are used to estimate the particle's direction of travel and displacement between exposures. The displacements determined from the 2^1 - 2^2 particle pairs in the search region are then used to project where the particle will have progressed to in the 3rd, 4th, and 5th exposures. For a valid velocity vector identification, the pixel amplitudes at the projected 3rd, 4th, and 5th positions must be equal to 2^3, 2^4, and 2^5, respectively. The details of the PDT algorithm are discussed in reference 13.

There are four data reduction routines included in the PDT package; BOUND5, BOUND25, BSPDTLS, and PDTLS. Briefly, the BOUND5 and BOUND25 routines perform the first stage boundary processing of the particle image data and generate time history files with the *.TS# suffix. The PDTLS routine performs the Particle Displacement Tracking process on the time history files to detect velocity vectors. The BSPDTLS routine performs both of the data reduction stages without the generation of the intermediate time history files. A more detailed description of each routine follows.

The protected mode Fortran programs require a DOS-Extender loader program for execution.

The form of the DOS command line call for running one of the NDP-Fortran programs is:

```
          C:\> RUN386 BOUND5

      where:
          RUN386:        Microway DOS-Extender program loader
```

BOUND5

The BOUND5 routine performs the boundary processing algorithm on a single 5-image sequence. The user is prompted for the XXXXXX#.PIV file name root, and the single time history file name to store the resulting image. The first 7 characters of the file name are entered, and the five raw image data file names are generated by the program. For example, given the file name root "TEST", the five file names generated by PDTMNLS.EXE are: TEST1.PIV, TEST2.PIV, ... TEST5.PIV. The routine queries the user for the type of image file, field or frame. The data acquisition program PDTMNLS.EXE used to acquire the images for the PDT system acquires image fields in 1/60 second intervals. Hence, the user should respond to the query with the default reply of a field. Since the data are fields and not frames, the data files only contain 240 lines of image data. The horizontal resolution is still 640 pixels. As the program reads in the field data files, the horizontal rows are doubled in order to produce a full 640×480 pixel image. The routine also requests the user to enter a processing threshold level. The threshold level was previously determined via the "Analyze Image for Background Level" routine from the data acquisition program. For high contrast images with little background noise a typical threshold level is 10 grey levels. The user is also queried for the boundaries of the image to process. The image boundaries were previously determined via the "Analyze Image for Boundaries" from the data acquisition program. If there are no obstructions or flare light in the field of view, then the default values can be used. Only the rectangular area defined by the lower left and upper right corners of the bounded area are processed. The results of this routine are then used as the input to the PDTLS routine to identify the velocity vectors from the amplitude coded particle time history information.

BOUND25

The BOUND25 routine performs boundary processing on a 25-field sequence of images. The images are processed in five groups of 5-fields, producing 5 time history files. The advantage of the 25-field series acquisition is that more data are collected and particles can be tracked over an extended interval. A particle which is tracked across all 25-fields will produce 5 velocity vectors oriented head to tail in succession. Acquiring multiple data sets sequentially can be used in the data analysis to elucidate the flowing fluid motion. More details of 'movies' will be given in the Graphical Data Analysis Section below.

The user is prompted for the XXXXX#.PIV file name root, and for the processing threshold grey level. For the raw image series file names, only the first 5 characters of the file name are entered, and the twenty-five raw image data file names are generated by the program. For the time history storage file names, the same 5 character root is used. The program generates five file names with the threshold level incorporated into the name prefix, and the suffix is '*.TS#', where # is the number of the group of 5 images out of the 25 image sequence. For example, if the file name root is "TEST", then the data files read in are TEST1.PIV to TEST25.PIV. If the processing threshold level was 25, then the storage file names for the time history files are TEST025.TS1, TEST025.TS2, ... TEST025.TS5. Three spaces are used for the threshold level to permit accurate recording of threshold levels larger than 100 grey levels. Hence, in the time history file names, the first 5 characters are selected by the user to identify the data set, the next 3 characters are used to encode the processing threshold level, and the file suffix is TS# identifies the file as a centroid processed time history file, where # is the index of the group of 5-fields out of the 25-field sequence. For example, file TEST025.TS1 contains the coded time history information from files TEST1.PIV, TEST2.PIV, TEST3.PIV, TEST4.PIV, and TEST5.PIV. File TEST025.TS2 contains the coded time history information form image files TEST6.PIV to TEST10.PIV, and so on. Again, as stated above for the BOUND5 routine, the threshold level was previously determined via the "Analyze Image for Background Level" routine from the data acquisition program. For high contrast images with little background noise a typical threshold level is 10 grey levels. The routine queries the user for the type of image file, field or frame. The data acquisition program PDTMNLS used to acquire the images for the PDT system acquires image fields in 1/60 second intervals. Hence, the user should respond to the query with the default reply of a field. Since the data are fields and not frames, the data files only contain 240 lines of image data. The horizontal resolution is still 640 pixels. As the program reads in the field data files, the horizontal rows are doubled in order to produce a full 640×480 pixel image. The user is also queried for the boundaries of the image to process. The image boundaries were previously determined via the "Analyze Image for Boundaries" from the data acquisition program. If there are no obstructions or flare light in the field of view, then the default values can be used. Only the rectangular area defined by the lower left and upper right corners of the bounded area are processed. The results of this routine are used as input to the PDTLS routine, which identifies the velocity vectors from the amplitude coded particle image time history information.

PDTLS

The PDTLS routine performs the second stage of the PDT processing. The time history files created in the first stage are now processed to identify velocity vectors. The program queries the user for the number of time history files to be processed. For each file to be processed, the user is prompted for name of each time history file, the name of the file in which to store the velocity vector data, and the search region size in pixels. The entire file names must be entered. The search region size is selected corresponding to the maximum expected particle displacement between exposures. A typical search region size is 10 pixels. The shorter the displacement, the more nearly linear the expected particle path will be. Longer paths will deviate from the assumed linear particle trajectory. Larger search regions also increase the probability of a false identification. For more details regarding the PDT search algorithm see reference 13. For more details about the probability of a false identification see reference 15.

The velocity vector data are written to the user selected file name in formatted file type. The first number in the file is the number of detected velocity vectors. The format for the number of vectors is FORMAT(I4). Following the integer number of vectors in the file are the velocity vector data. There are four numbers per velocity vector: x,y-coordinates (pixels), vector magnitude (pixels/time of 5 field acquisition), and vector direction (degrees). The x,y coordinates are the coordinates of the **MIDPOINT** of the velocity vector. The plotting routines contained in this package adhere to this standard, and correctly display the velocity vector. The reason the midpoint is used is because the velocity vector is the average velocity of a particle over all 4 displacements (between the five exposures), hence, the mean position of the particle is at the midpoint of the velocity vector. Actually four particle displacements are determined. The four particle displacements are from the motion of the particle between exposures 1&2, 2&3, 3&4, and 4&5. The sum total of the individual displacements is used as the velocity vector magnitude and written to the output file. The format for the vector data is FORMAT(4F12.4). Following the velocity vector data are two comment lines. The first line tells how many velocity vectors were detected, and the second line tells the amount of cpu processing time required in the PDT processing stage. The comment lines are merely for archival purposes.

BSPDTLS

The BSPDTLS routine performs the same operations as both the BOUND25 and PDTLS routines in a single program. The program operates on 25-field acquired series. The raw image files are read in, and the velocity vector data files are written out at the end of the program. The time history image generated from the centroid processing is passed to a PDT processing subroutine to perform the velocity vector identification. No intermediate time history images are written to files, which saves disk storage space. The option of batch mode or interactive mode processing is available. In batch mode, the program queries the user for the number of 25-field image sequences to process and then queries all the requisite information for the number of data series specified. In interactive mode, only one 25 field image series is processed at a time.

The advantage of the 25-field series acquisition is that more data are collected and particles can be tracked over an extended interval. A particle which is tracked across all 25-fields will produce 5 velocity vectors oriented head to tail in succession. Acquiring multiple data sets sequentially can be used in the data analysis to elucidate the flowing fluid motion. More details of 'movies' will be given in the Graphical Data Analysis Section below.

The user is prompted for the first five characters (XXXXX#.PIV) of the file name root, and for the processing threshold grey level. For the raw image file names, only the first 5 characters of the file name are entered, and the twenty-five raw image data file names are generated by the program. For the velocity vector output files, the program generates five file names with the threshold level incorporated into the name prefix, and the suffix is 'DS#', where # is the number of the group of 5 images out of the 25 image sequence. For example, if the file name root is "TEST", then the data files read in are TEST1.PIV to TEST25.PIV. If the processing threshold level was 25, then the velocity vector output file names are TEST025.DS1, TEST025.DS2, ... TEST025.DS5. Three spaces are used for the threshold level to permit accurate recording of threshold levels larger than 100 grey levels. Hence, in the output velocity vector file names, the first 5 characters are selected by the user to identify the data set, the next 3 characters are used to encode the processing threshold level, and the file suffix is '*.DS#', where # is the index of the group of 5-fields out of the 25-field sequence. Again, as stated above for the BOUND5 and BOUND25 routines, the threshold level was previously determined via the "Analyze Image for Background Level" routine from the data acquisition program. For high contrast images with little background noise a typical threshold level is 10 grey levels. The routine queries the user for the type of image file, field or frame. The data acquisition program PDTMNLS used to acquire the images for the PDT system acquires image fields in 1/60 second intervals. Hence, the user should respond to the query with the default reply of a field. Since the data are fields and not frames, the data files only contain 240 lines of image data. The horizontal resolution is still 640 pixels. As the program reads in the field data files, the horizontal rows are doubled in order to produce a full 640×480 pixel image. The user is also queried for the boundaries of the image to process. The image boundaries were previously determined via the "Analyze Image for Boundaries" from the data acquisition program. If there are no obstructions or flare light in the field of view, then the default values can be used. Only the rectangular area defined by the lower left and upper right corners of the bounded area are processed.

In addition to the boundary processing information, the program also requests the region size for the PDT processing as in the PDTLS program. The search region size is selected corresponding to the maximum expected particle displacement between exposures. A typical search region size is 10 pixels. The shorter the displacement, the more nearly linear the particle path will be. Longer paths will deviate from the assumed linear particle trajectory. Larger search regions also increase the probability of a false identification. For more details regarding the PDT search algorithm see reference 13. For more details about the probability of a false identification see reference 15.

The velocity vector data are written to the XXXXXZZZ.DS# file names in a formatted file type. The first number in the file is the number of detected velocity vectors. The format for the number of vectors is FORMAT(I4). Following the integer number of vectors in the file are the velocity vector data. There are four numbers per velocity vector: x,y-coordinates (pixels), vector magnitude (pixels/time of 5 filed acquisition), and vector direction (degrees). The x,y coordinates are the coordinates of the **MIDPOINT** of the velocity vector. The plotting routines contained in this package adhere to this standard, and correctly display the velocity vector. The reason the midpoint is used is because the velocity vector is the average velocity of the particle over all 4 displacements (between five exposures), hence, the mean position of the particle is at the midpoint of the velocity vector. Actually four particle displacements are determined. The four particle displacements are from the motion of the particle between exposures 1&2, 2&3, 3&4, and 4&5. The sum total of the individual displacements is used as the velocity vector magnitude and written to the output file. The format for the vector data is FORMAT(4F12.4). Following the velocity vector data are two comment lines. The first line tells how many velocity vectors were detected, and the second line tells the amount of cpu processing time required in the PDT processing stage. The comment lines are merely for archival purposes.

5). GRAPHICAL DATA ANALYSIS ROUTINES

The graphical data analysis and display routines are useful for analyzing both the time history data files and the reduced velocity vector maps. Two routines are included, the first routine POINTS, graphs the time history files generated from the boundary processing stage. The second program, PDTGRAF, is used for displaying the velocity vector maps, interpolated velocity vector grids, iso-velocity vector contours, and flow streamlines.

The graphical data analysis routines are written in Fortran 77 and compiled with Microway's NDP-Fortran-386 compiler. The screen based text graphics and Microsoft compatible mouse support are done using the Microway built-in GREX library. The on screen graphics (EGA, VGA and superVGA) and 300 dpi hardcopy support on HP Laserjet style printers are accomplished using Media Cybernetic's HALO Professional graphics primitives library.

POINTS

The POINTS routine is used for displaying the particle centroid time history files on the computer screen or producing a hardcopy. At least initially, it is very instructive to examine the time history files generated from the boundary processing programs so that the user understands the effect of changing the inter-field image acquisition interval. Typical particle displacements are on the order of 10 pixels between exposures, hence, by viewing the particle centroid files, the user can determine if the displacements are in the proper range. Recall that the time history files contain the single pixel particle centroids whose amplitudes indicate from which field in the 5-field sequence the particle image was recorded. Since there were 5 exposures, 5 symbols are used to code the centroids. A complete particle displacement record over all 5 images consists of the five different symbols in the appropriate order.

The POINTS program utilizes a command line argument for selecting the current display resolution. The format of the program command line call and range of command line argument values are:

```
C:\> RUN386 POINTS IARG
```

where:
RUN386: Microway DOS-Extender program loader
IARG = 0, or
no argument: Video Seven VRAM VGA 1024×768×16 colors
IARG = 1: IBM VGA 640×480×16 colors
IARG = 2: IBM EGA 640×350×16 colors

Regardless of the screen resolution, the printer resolution is always 300 dpi. Other video boards and screen resolutions can be supported by selecting the appropriate HALO Professional screen driver and video mode.

The program queries the user where the files are to be displayed; on the computer screen, or on the system printer. The computer screen is the default. Next, the program queries for the number of files to be displayed or printed. The user is then asked to enter the file name(s) to be plotted. After all the file names have been entered, the program will either sequentially display the images on the computer screen, or print 2 copies of each image on the printer. For on screen displays, the user presses the return key when finished viewing the graph to either display the next image or end the program.

For the on screen displays, both color and symbol shapes are used to distinguish the particle image centroids from different fields in the 5-field sequence. Color coding the symbols makes up for the low resolution of the computer screen. However, on the printer hardcopy, only black and white are available. The symbol size used for both the onscreen displays and for hardcopies are approximately twice as large as a normal pixel would be from a time history file 640×480 pixel image. The key to the symbol shapes and colors are displayed at the top of the graph, just underneath the file name. The five symbols used are:

Exposure #	Symbol Type
1	■
2	+
3	∘
4	•
5	×

Hence, a successful particle displacement record which would be identified as a valid velocity vector would appear as:

■ + ∘ • ×
————————▶

The velocity vector corresponding to the particle displacement pattern is shown here underneath the symbols just as an example. The particle direction is unambiguous due to the time history encoding. The ■ marks the first exposure and the × marks the fifth exposure, thus, the particle is moving from left to right. The full resolution of an HP-Laserjet style printer is used for generating the 300 dpi image. A Virtual Raster Interface (VRI) device is created within the computer's extended memory, which has the equivalent resolution of a 300 dpi image. The program displays a banner indicating that the image is being transferred to the printer. Two copies of the graph are generated. At the completion of the print operation, the program returns to the main menu. Typical print times on a 25MHz 80386 PC are 3-4 minutes. If an all black page is ejected from the printer, then an expanded memory driver may be present. Extended memory must be used with the VRI driver. A sample print out of a time history data file is shown in figure 3 below.

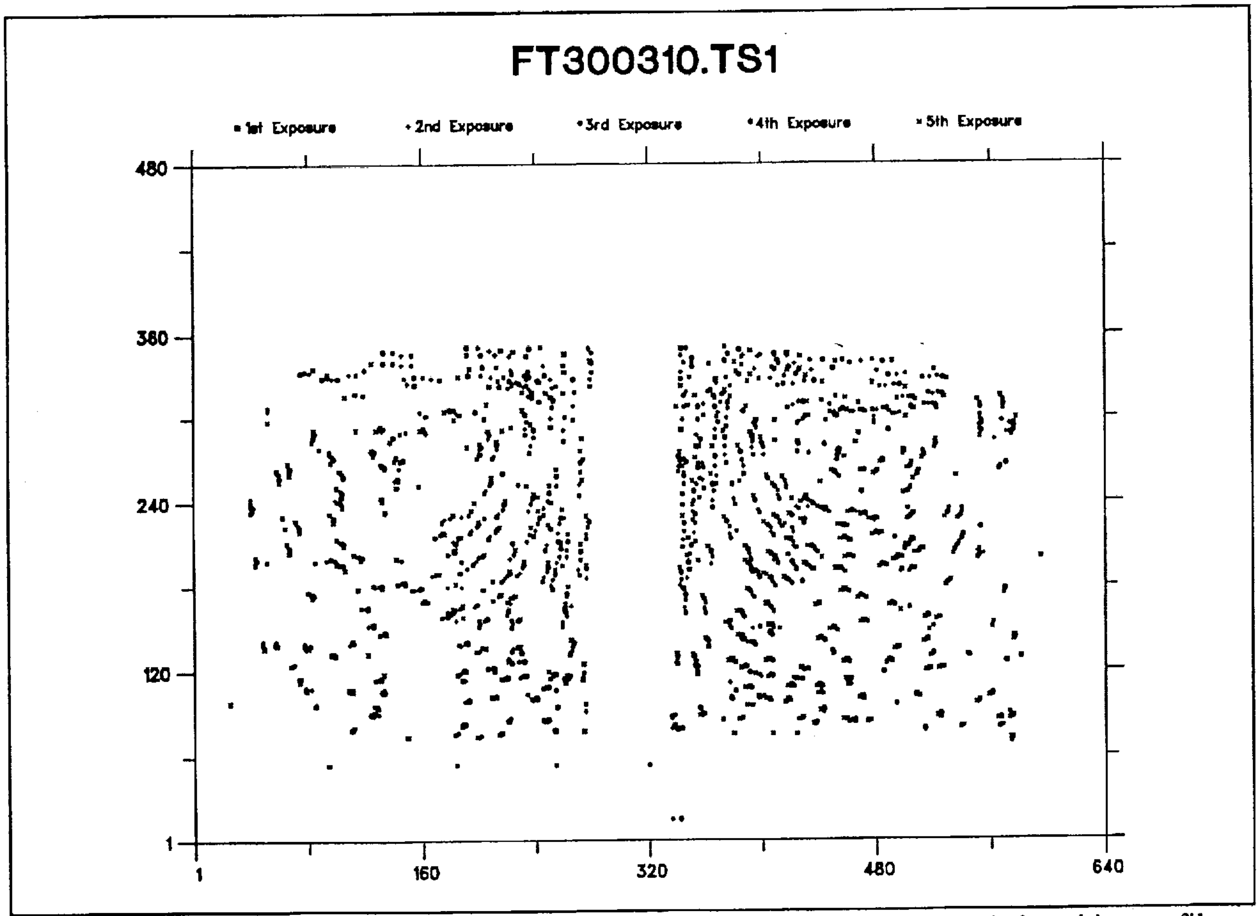


Figure 3: Graph produced from the POINTS program displaying a typical time history file.

PDTGRAF

The PDTGRAF program is a large collection of data analysis and plotting routines for interpreting and displaying the reduced velocity vector data. The input data to the PDTGRAF program are the velocity vector map files generated either from the PDTLS or BSPDTLS programs.

The PDTGRAF program utilizes a command line argument for selecting the current display resolution. The format of the program command line call and range of command line argument values are:

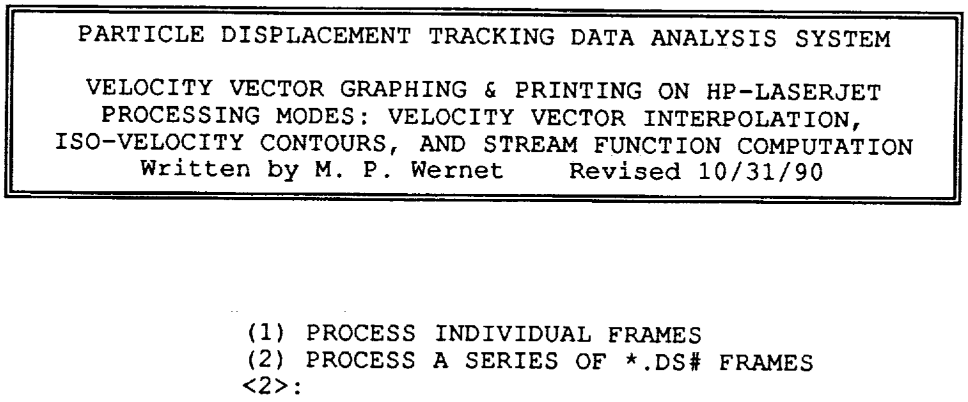
```
          C:\> RUN386 PDTGRAF IARG

     where:
          RUN386:        Microway DOS-Extender program loader
          IARG = 0, or
          no argument: Video Seven VRAM VGA 1024x768x16 colors
          IARG = 1:      IBM VGA 640x480x16 colors
          IARG = 2:      IBM EGA 640x350x16 colors
```

regardless of the screen resolution, the printer resolution is always 300 dpi. Other video boards and screen resolutions can be supported by selecting the appropriate HALO Professional screen driver and video mode.

The PDTGRAF program displays multiple data entry screens before the data processing menu is reached. There are three data entry screens before the main menu screen. The functions of each data entry screen are described below.

The initial screen, screen #1, displayed by the PDTGRAF program shows a banner message and options of processing individual frames or a series of frames:

(SCREEN #1)



```
        PARTICLE DISPLACEMENT TRACKING DATA ANALYSIS SYSTEM

         VELOCITY VECTOR GRAPHING & PRINTING ON HP-LASERJET
           PROCESSING MODES: VELOCITY VECTOR INTERPOLATION,
     ISO-VELOCITY CONTOURS, AND STREAM FUNCTION COMPUTATION
            Written by M. P. Wernet    Revised 10/31/90



                  (1) PROCESS INDIVIDUAL FRAMES
                  (2) PROCESS A SERIES OF *.DS# FRAMES
                  <2>:
```

In all of the program queries to the user in the PDTGRAF program, a default input value is always displayed in <>. The default option is selected by just pressing the return key. The default option for the above screen is option 2. In most instances the format of the data type to be read are also denoted in brackets []. For example, integer reads are designated [I2], real data type reads are denoted [F6.2].

The individual frame option is for plotting single velocity vector map files processed using the BOUND5 and PDTLS routines. Screen #2 asks the user how many individual files are to be read, and then queries the user for the file name(s) of the input data files. For each data file name entered, the user is also asked to enter a scale factor for the data file. The scale factor refers to magnitude of the velocity vectors in each file relative to the other data files entered. The scale factor is used for scaling data sets with different inter-field acquisition times and defines the magnitude of the velocity vectors in the subsequent graphing operations. For only a single input file, a scale factor of 1 is appropriate. However, if two data files are to be read in, VECTOR1.DAT and VECTOR2.DAT, which were recorded with inter-frame times of 5 and 10 field intervals (5 × 1/60, and 10 × 1/60 second intervals), then the ratio of the velocity vector magnitude scales is 2:1. The scale for the 5/60 second data set would be 1.0, while the scale for the 10/60 second data set would be 0.5. Hence, the ratio of the inter-frame acquisition times is used for scaling the different data sets relative to the maximum velocity data set (shortest inter-frame time). The maximum velocity data set is usually entered first, followed by the successively slower velocity data sets.

The PROCESS A SERIES OF *.DS FRAMES (option 2, screen #1) also queries the user for the number of series to be read, and then queries the user for each series file name root. Each series contains 5 data files. The user enters the file name root such as 'TEST010.DS', and the program will read in data files TEST010.DS1, TEST010.DS2, ... TEST010.DS5 for each series specified. The user is asked to enter a velocity scale for each series of velocity vector data files. Again, the scale factors are used to scale data sets with different inter-frame acquisition times relative to each other.

In the PIV work performed at LeRC, a file naming convention has been adopted which encodes some of the experimental setup parameters in the file name. For example, in the series file names:

B0536025.DS1
B0536025.DS2
B0536025.DS3
B0536025.DS4
B0536025.DS5

The letter 'B' is used to denote the experiment type. The '05' section is used to encode the inter-frame time in field intervals, therefore, '05' means 5 field intervals, or 5/60 of a second between acquired images in the sequence. The '36' is the run number of experiment type 'B'. The '025' number indicates the processing threshold level used in the boundary processing of the original PIV images. The suffixes 'DS#' are used to denote the five data sets generated from the original 25-field image sequence. Typically, several data sets with different inter-frame intervals are plotted on the same graph. Each data set has approximately a 10:1 dynamic range (remember the 10 pixel displacements). Hence, by adding three data series together with inter-frame acquisition times of 05, 10, and 20, yielding velocity magnitude scales of 1, 0.5, and 0.25, one can obtain a velocity diagram with a dynamic range of (10 × 1/0.25) 40:1.

The necessity of simplifying the data series entry procedure has driven the development of the following file entry coding. For example suppose the user wants to enter 3 data series files named B0536025.DS(1-5), ,B1036025.DS(1-5), and B2036025.DS(1-5). When queried by the program for the number of series to be read in, the user enters 3 just as they normally would. Instead of typing the entire file name roots for all three series and their respective velocity vector magnitude scales, however, when prompted for the first series file name root the user enters the following string:

```
$B-05-10-20-36025.DS
```

The program has been told that 3 series are to be read. The '$' character informs the program that a coded string has been entered which contains all 3 file name roots. The hyphens in the code string are used to delimit the inter-frame acquisition intervals in integral multiples of video fields (1/60 second). For consistency, two field positions (i.e. -05- not -5-) are used for all inter-frame time intervals. The inter-frame acquisition intervals must be entered in order from shortest to longest, which corresponds to the highest to lowest velocities. The program generates the file names and automatically reads in all three data series B0536025.DS(1-5), B1036025.DS(1-5), and B2036025.DS(1-5), for a total of 15 files. After entering the coded file name string, the user is prompted to enter the scale factor for the first series. The user should enter the scale factor desired for the maximum velocity data series, which is the first data series in the coded string. The scale factor can be greater than or less than unity. The value of the scale factor sets the relative length of all of the velocity vectors in the graphs. If the vectors are too short, then increase the scale factor, or, if the vectors are too long, then decrease the scale factor. The program uses the user entered scale factor and the inter-frame acquisition times to calculate the scale factors for all the other data series. In the example case above, the inter-frame times are 5, 10, and 20 field intervals. If the user entered a scale factor of 1.0 for data series B0536025.DS(1-5), then the scale factors for the other two data series would be computed to be 1.0 × 05/10 = 0.5 for B1036025.DS(1-5), and 1.0 × 05/20 = 0.25 for B2036025.DS(1-5). The automated data entry scheme simplifies the data entry process for the user.

After all of the data files have been read, the program queries the user for the velocity scale. The velocity scale is the conversion factor from the particle image pixel displacements to real world dimensions of cm/sec or m/sec. If no velocity scale is entered, (the default option), then a scale key is not included in the velocity vector graphs. The velocity scale is computed based on the highest velocity data set, or correspondingly, the data set with the minimum inter-frame exposure time. The velocity scale is computed from two parameters, the image scale L_{scale} [m/pixels], and the total exposure time ΔT.

The image scale is obtained by placing a reference scale (ruler) in the plane of the light sheet illumination. The data acquisition program PDTMNLS "Acquire a Single Image" operation can be used to acquire an image containing the reference scale. Then the PDTMNLS "Analyze Image for Boundaries" operation can be used to find the number of pixels between two points on the reference object. Alternatively, the EPIX interactive menu based image processing package 4MIP can be used to acquire an image and determine the number of pixels between two points (x_1,y_1) and (x_2,y_2) on the reference object. The distance between the two points is given by $D=[(x_2-x_1)^2 + (y_2-y_1)^2]^{1/2}$. The image scale is then simply the physical distance between the two reference points on the scale object divided by the distance in pixels, D, between the two points. A typical PIV setup has an image scale of approximately L_{scale} = 200μm/pixel, which corresponds to a 128×96mm field of view in a 640×480 pixel image.

The total exposure time is the sum total of the four inter-frame acquisition intervals. For example, if the inter-frame image acquisition interval was 5 video fields, then the total exposure time is $\Delta T = 4\times5\times1/60 = 0.333$ seconds. The total time interval corresponds the time elapsed from when the particle was first recorded on field #1 to the last record of the particle on field #5.

The velocity scale is computed from the product L_{scale} and $1/\Delta T$. Hence, for the example quantities given above, the velocity scale is:

$$V_{scale} = L_{scale} / \Delta T$$

$$= 200\mu\text{m/pixel} / 0.333 \text{ seconds}$$

$$V_{scale} = 6.0\text{E-4 m/pixel}\cdot\text{sec}$$

When a non-zero velocity scale is entered above, a velocity scale is displayed in the 2-D velocity vector plots. The displayed scale can be either the maximum velocity in the data set, which is used as a reference scale for the velocity vectors in the graph, or the velocity scale can be the mean velocity of all of the velocity vectors in the data set. After the user enters the velocity scale into the program, the program queries the user for which type of velocity to use in the scale computation, the maximum velocity or the mean velocity. The default is to use the maximum velocity. For the maximum velocity, the program searches the velocity vector data to find the maximum vector magnitude. The maximum magnitude reference velocity vector is drawn at under the graph title along with the numerical value of the vector, in cm/sec. Alternatively, the mean velocity option can be selected, which causes the program to compute the mean velocity vector magnitude from all of the data vectors. The standard deviation is also computed with the mean vector magnitude. A reference velocity vector is drawn under the graph title along with the mean vector magnitude and standard deviation in cm/sec.

The last parameter to be entered in screen #2 is the graph title. By default, the program generates a title from the last entered file name. The default title is accepted by just pressing the return key. If a different title is desired, type the new title and then press the return key. The maximum title length is 40 characters. For example, if single data file input is selected, the default title is exactly the last data file name. If series data file input is selected, then the default title is the last entered file name root with the added suffix '(1-5)', denoting that a series of data files have been read. Hence, if the series file name root was 'B0536025.DS', the default title is 'B0536025.DS(1-5)'. Again, if a different title is desired, type in the new file name at the prompt and press return. Finally, If a coded multiple series file root is entered (beginning with a '$'), then the default file name is the coded string minus the '$' character and with the added suffix '(1-5)', denoting that all of the coded series have been read.

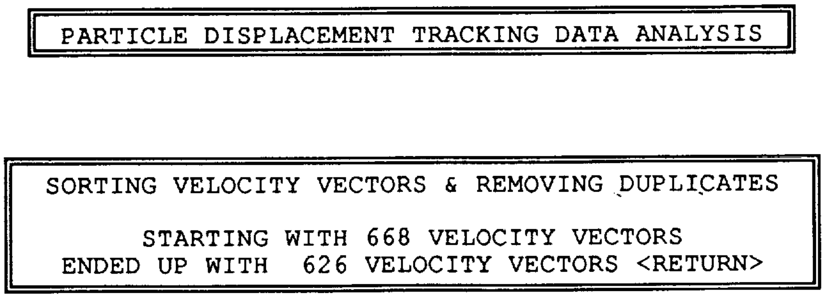
In summary, an example of a screen #2 display from PDTGRAF is shown below, where the user inputs are shown in **BOLD**:

(SCREEN #2)

```
                  PARTICLE DISPLACEMENT TRACKING DATA ANALYSIS

 ONE SERIES, OR MULTIPLE SERIES TO BE ADDED [I2] <1>: 1
 FOR AUTOMATED SERIES INPUT ENTER $X-##-##-##-XXYYY.ZZ
 ENTER SERIES# 1 FILE ROOT [A40]: B0536025.DS
 ENTER MAGNITUDE SCALE FOR FILE SERIES# 1 [F6.2] <1.00>: 1.0

                     READING SERIES FILE: B0536025.DS1
                     READING SERIES FILE: B0536025.DS2
                     READING SERIES FILE: B0536025.DS3
                     READING SERIES FILE: B0536025.DS4
                     READING SERIES FILE: B0536025.DS5

  VELOCITY SCALE IS ON A TOTAL EXPOSURE BASIS, ([m/pixel]/(TOTAL TIME))
         TOTAL TIME = N*ΔT, (WHERE N = TOTAL # OF EXPOSURES - 1)
 VELOCITY SCALE CORRESPONDS TO THE MAXIMUM VELOCITY DATA SET (MINIMUM ΔT)

  ENTER THE VELOCITY SCALE [(m/s)/pixel] < 0.00>: 6.0E-4
  USE WHICH VELOCITY FOR SCALE (MAX=0,MEAN=1) <0>: 0
  ENTER THE GRAPH TITLE <B0536025.DS(1-5)>:
```

The maximum number of velocity vectors which can be read is 4096. The internal storage arrays must all be redimensioned if more than 4096 velocity vectors are to be read. As the velocity vector data are read in, the x,y coordinates of the are adjusted to the vector tail coordinate. Hence, the velocity vector data internal to the PDTGRAF program are treated as tail coordinate data. Screen #3 is simply an information screen, no user input is required at this stage. After all of the information about the number and type of data files, velocity scale, scale type, and graph type have been entered, the program scans the data to remove any duplicate vectors. Duplicate vectors can occur when multiple data sets are added together. A duplicate vector occurs when two velocity vectors occur with their tail coordinates within 2 pixels of each other. The program searches through the data arrays detected velocity vectors which are too close or on top of each other. When a duplicate pair is found, the program keeps the larger magnitude vector and discards the lower magnitude velocity vector. The larger magnitude vector is retained because larger displacements, corresponding to higher velocities, are more accurate than smaller displacement particle records.[2] Screen #3 displays the total number of velocity vectors which were initially read form the files, and when the sort has completed, the program updates the screen #3 display to show the number of 'UNIQUE VELOCITY VECTORS'. The number of removed velocity vectors is not of great importance, but merely lets the user know how many data vectors are left for subsequent processing. The duplicate sorting operation is very fast, and typically completes within 1 second. A sample screen #3 is shown below.

(SCREEN #3)

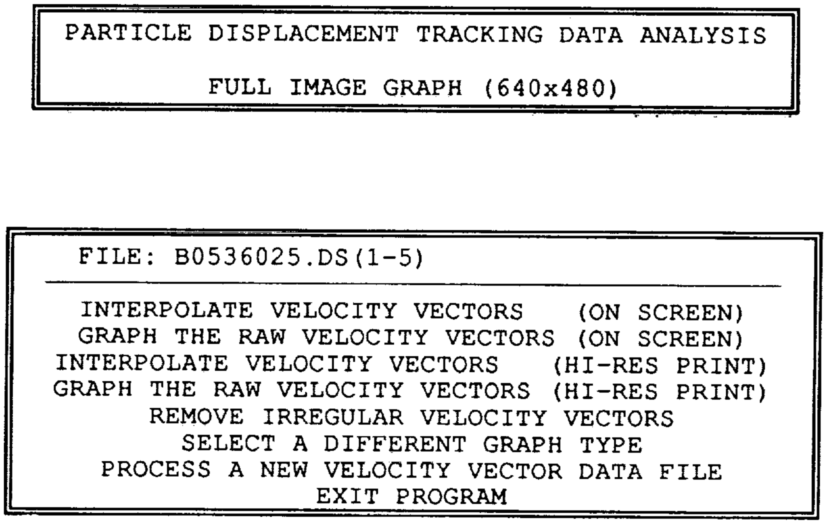


```
PARTICLE DISPLACEMENT TRACKING DATA ANALYSIS
```

```
SORTING VELOCITY VECTORS & REMOVING DUPLICATES

     STARTING WITH 668 VELOCITY VECTORS
 ENDED UP WITH  626 VELOCITY VECTORS <RETURN>
```

After the display is updated with the number of velocity vectors remaining following the sort, the user simply presses return to go on to screen #4, which is the main menu in the PDTGRAF program. The main menu screen is shown below.

(SCREEN #4)



```
PARTICLE DISPLACEMENT TRACKING DATA ANALYSIS

         FULL IMAGE GRAPH (640x480)
```

```
  FILE: B0536025.DS(1-5)
  -----------------------------------------------
  INTERPOLATE VELOCITY VECTORS    (ON SCREEN)
  GRAPH THE RAW VELOCITY VECTORS (ON SCREEN)
 INTERPOLATE VELOCITY VECTORS    (HI-RES PRINT)
 GRAPH THE RAW VELOCITY VECTORS (HI-RES PRINT)
       REMOVE IRREGULAR VELOCITY VECTORS
         SELECT A DIFFERENT GRAPH TYPE
    PROCESS A NEW VELOCITY VECTOR DATA FILE
                 EXIT PROGRAM
```

The PDTGRAF main menu screen shows all of the data processing and display options. The PDTGRAF program was written to display both the full frame 640×480 pixel images and a special 2:1 aspect graph type. The 2:1 aspect graph type is used to display data obtained from the Surface Tension Driven Convection Experiment (STDCE) setup at NASA LeRC. Any subsequent reference to a 2:1 aspect graph, or to STDCE, are particular to this experiment. The STDCE graph options were retained in the software to demonstrate different graph types. The main banner on screen #4 displays the current graph type. The default graph type is 'FULL IMAGE GRAPH (640x480)'. The graph type may be changed through a menu option which will be discussed below.

The 'FULL IMAGE GRAPH' plots the velocity vector data on a 640×480 pixel graph. All images obtained using the PDT technique can be displayed in this graph type. The graph title is displayed at the top of the data plot, along with a subtitle which designates the data type in the graph. The possible data types are 'RAW DATA', 'INTERPOLATED #GRID PTS = XX', 'VELOCITY CONTOURS', and 'FLOW STREAMLINES'. The subtitle is automatically set according to the type of graph selected. A velocity scale will also be included at the top of the graph if a non-zero velocity scale has been entered.

If the user should happen to select the STDCE graph type, then the message banner at the top of the main menu screen will read 'STDCE 2:1 ASPECT GRAPH'. The STDCE graph type assumes the data were obtained from a 2:1 (width to height) aspect ratio flow cell. In the STDCE setup at NASA LeRC, a cylindrical reservoir filled with silicon oil and seeded with Al_2O_3 particles is illuminated by a sheet of light. In the same plane as the light sheet, a CO_2 laser beam heats the fluid surface. The surface tension gradient imposed on the fluid surface drives a circularly symmetric counter rotating flow pattern. In another configuration, a tubular heater is placed in the center of the reservoir instead having the CO_2 laser beam impinge on the fluid surface. Hence, the STDCE graph type is designed to handle both of these configurations. The user must enter the coordinates bounding the 2:1 test cell within the 640×480 pixel field of view. If the heater was used, then the heater coordinates must also be entered. The STDCE graphs display the file generated title header, as in the FULL GRAPH type discussed above. In addition, the user can specify a main title for the STDCE graph, which appears at the top of the graph. The axes are normalized, thus, the radial coordinate, or x-axis runs from -1 to 1, and the vertical coordinate, or y-axis runs from 0 to 1. Subtitles, indicating the type of graph selected, are also displayed under the first two main titles. Below the subtitle are two lines of text. The first line indicates the number of data vectors read from the files and the number used in the current graph. When interpolation operations are selected, the number of vectors used in the graph defaults to the number of grid points selected. The second line of text displays the pixel coordinates of the test cell entered by the user. If the heater option has been selected, then a rectangular box is draw at the coordinates delimiting the heater position. A sample STDCE graph type will be shown in a later section.

The menu portion of screen #4 displays the current graph title, (derived from the data file names read). The menu pointer is indicated by the highlighted text. The menu pointer can be moved either via a Microsoft compatible mouse, or by using the numeric keypad arrow keys. The home and end keys can be used to place the menu pointer at the top or bottom of the menu, respectively. Some of the menu items can be changed by choosing the 'SELECT A DIFFERENT GRAPH TYPE' menu option. Each menu item will now be discussed in detail.

INTERPOLATE VELOCITY VECTORS (ON SCREEN)

The '(ON-SCREEN)' designation simply means that the graph will be displayed on the computer's graphics screen. The interpolation option is used to transform the randomly sampled flow field data obtained from the PDT technique to a uniform grid of velocity vectors. There must be at least 100 total velocity vectors present in the data files read for the interpolation to be performed properly. The distribution of the detected velocity vectors also affects the quality of the interpolation. A uniform distribution of velocity vectors is desired for optimum performance. For the FULL FRAME GRAPH type, the velocity vector data are interpolated over the region where velocity vectors are known to exist. When the velocity vector data are read, the minimum and maximum x,y coordinates are recorded. These limits are used to set the boundaries of the interpolated grid. For the STDC graph type, the grid boundaries are determined by the dimensions of the 2:1 aspect ratio test cell. In either case, the velocity vector data are interpolated over an N×N grid, where N is the number of grid points entered by the user. The number of grid points is the same in both dimensions, even though the data extents may not have a 1:1 aspect ratio. The maximum number of grid points is 64. The internal storage arrays are dimensioned to this size. After entering the number of grid points, the user is queried whether or not to save the interpolated values to a disk file. These two queries, which appear under the main menu are shown below:

```
ENTER NUMBER OF GRID POINTS [I2]: 32
SAVE INTERPOLATED VALUES TO A FILE (0=NO/1=YES) <0>:
```

The default option is not to save the interpolated values to a file. If the save to a file option is selected, the user is prompted for the file name, and after completing the interpolation the program writes the x, y, magnitude, and angle (degrees) values to the file. The x,y coordinates are the coordinates of the velocity vector tail, not the midpoint. After the file option has been addressed, the program computes the interpolated grid of velocity vectors. During the interpolation operation, a message banner is displayed on the computer screen indicating the percentage of the interpolation computation that has been completed. At the completion of the interpolation, the program plots the interpolated velocity vector grid on the computer screen. The resolution of the on-screen graph has been previously determined by the command line argument value passed to the PDTGRAF program.

The interpolation algorithm starts by generating the x_g,y_g grid point from the minimum and maximum data extents. The algorithm then sorts the velocity vector data according to the distance away from the x_g,y_g grid coordinate. The twenty nearest neighbors to the grid point are used in a weighted linear least squares analysis. A planar surface is fit to the velocity vector data, and used to interpolate the velocity vector magnitude and direction at the x_g,y_g coordinate. The weighting factors used in the analysis are the inverse distances from the grid point. When the data density in the region of the interpolation grid point is high, a good estimate of the local velocity is obtained. However, there may exist regions in the flow where the data density is low, yielding estimates which exhibit some random deviations from the expected flow. The interpolation routine is quite efficient. The data sorting routines were obtained from reference 16. Reference 17 gives a good description of different data interpolation algorithms, from which the algorithm used in PDTGRAF was derived.

After viewing the graph displayed on the computer screen, press the return key to come back to the PDTGRAF main menu screen.

GRAPH THE RAW VELOCITY VECTORS (ON SCREEN)

The second menu item, which is also the default option, is for plotting the raw velocity vector files on the computer screen. The velocity vector map obtained contains the randomly sampled velocity vectors detected using the PDT technique. There are three suboptions presented after selecting this menu option. The suboptions are:

```
            SINGLE COLOR (S), MULTICOLOR (C), OR MOVIE (M) <S>:
```

The default option is for a SINGLE COLOR graph, which displays all of the velocity vectors in a single color. The next two options 'MULTICOLOR' and 'MOVIE' are only available for velocity vector series data. The 'MULTICOLOR' option displays the velocity vector data color coded by the file series number. Recall that the velocity vector file series are read in with the suffix designation '.DS1', '.DS2', ... '.DS5'. The file number order in the series is recorded for each velocity vector. The series order number is used to set the colors of the displayed vectors. This graph type is useful for tracing particle trajectories over the 25-field image acquisition interval. The velocity vectors depicting a particle trajectory will be oriented head to tail in a row, color coded by the order of detection.

The 'MOVIE' option allows the user to elucidate the flow field motion over the 25-field acquisition interval. Again, the data series file numbers are used to time code the data files as they are read into the program. The time code records are maintained even for multiple data series reads. Each velocity vector has a number from 1 to 5 associated with it, depicting to which frame in the 5 frame series it belongs. Fluid motion is observed by sequentially displaying all velocity vectors associated each successive 5 field group in the series. Hence, 5 video frames are sequentially displayed on the computer screen. The up and down arrow keys on the keypad can be used to increase or decrease the frame rate, respectively. The graph is terminated by pressing the return key, after which the main menu is redisplayed.

INTERPOLATE VELOCITY VECTORS (HI-RES PRINT)

The third menu item is used for generating a hardcopy of the interpolated velocity vector graphs. The full resolution of an HP-Laserjet style printer is used for generating the 300 dpi image. A Virtual Raster Interface (VRI) device is created within the computer's extended memory, which has the equivalent resolution of a 300 dpi image. For the hardcopy option, the user is not asked if the interpolated values are to be written to a file. The only user entry is the number of grid points, just as for the on-screen version of the interpolation option. During the interpolation operation, a banner is displayed on the computer screen indicating the percentage of the interpolation computation that has been completed. Then, the program displays a banner indicating that the image is being transferred to the printer. Two copies of the graph are generated. At the completion of the print operation, the program returns to the main menu. Typical print times on a 25MHz 80386 PC are 3-4 minutes. If an all black page is ejected from the printer, then an expanded memory driver may be present. Extended memory must be used with the VRI driver. A sample print out of an interpolated velocity vector data set is shown in figure 4 below.

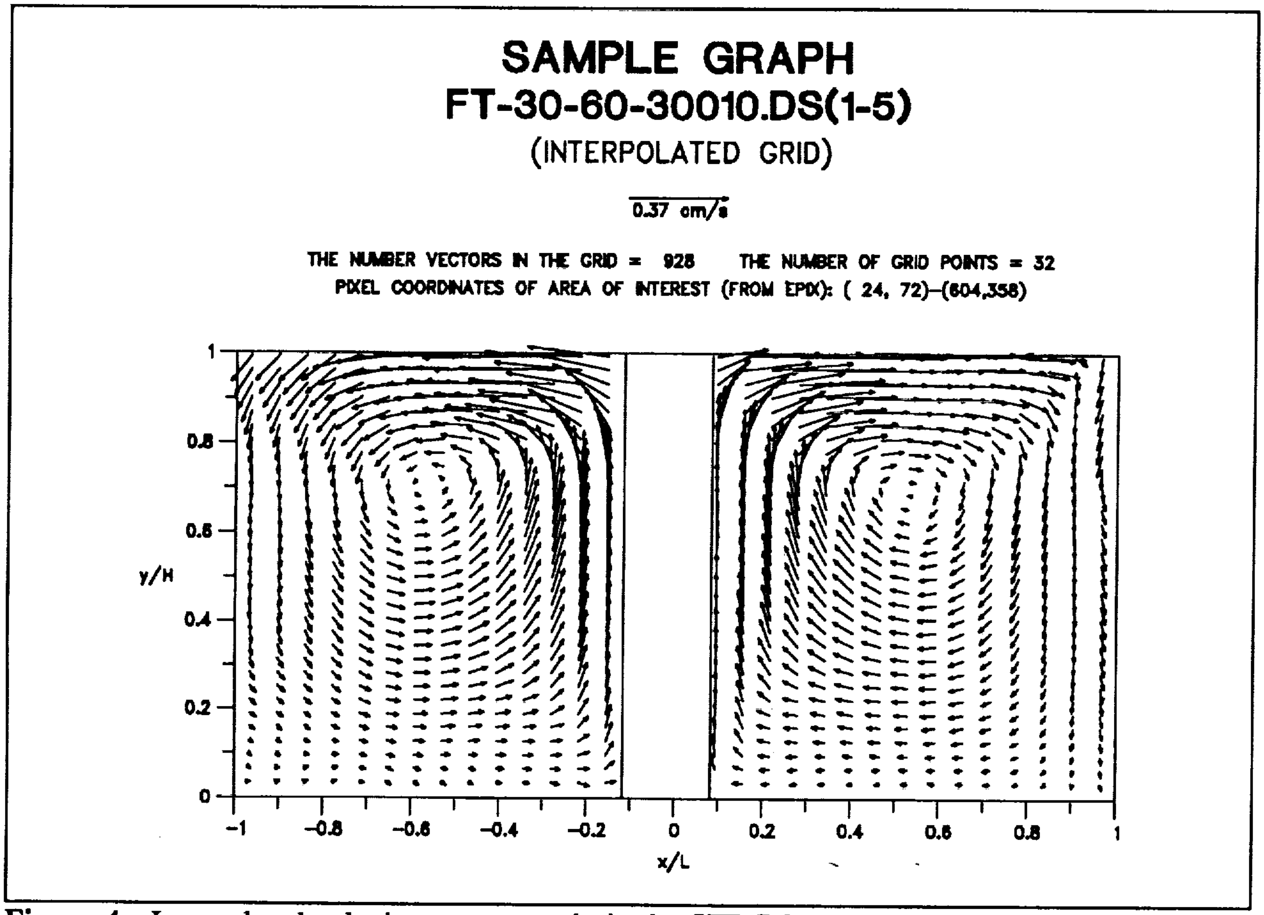


Figure 4: Interpolated velocity vector graph, in the STDC 2:1 aspect graph format with heater.

GRAPH THE RAW VELOCITY VECTORS (HI-RES PRINT)

The fourth menu option is used for generating a hardcopy of the raw velocity vector data graphs. The full resolution of an HP-Laserjet style printer is used for generating the 300 dpi image. A Virtual Raster Interface (VRI) device is created within the computer's extended memory, which has the equivalent resolution of a 300 dpi image. For the hardcopy option, the user is not asked about single color, multicolor, or movie options. A simple 2-color (black on white) image is generated. The program displays a banner indicating that the image is being transferred to the printer. Two copies of the graph are generated. At the completion of the print operation, the program returns to the main menu. Typical print times range from 3 to 4 minutes on a 25MHz 80386 PC. If an all black page is ejected from the printer, then an expanded memory driver may be present. Extended memory must be used with the VRI driver. A sample printout of a series of velocity vector raw data is shown in figure 5 below.

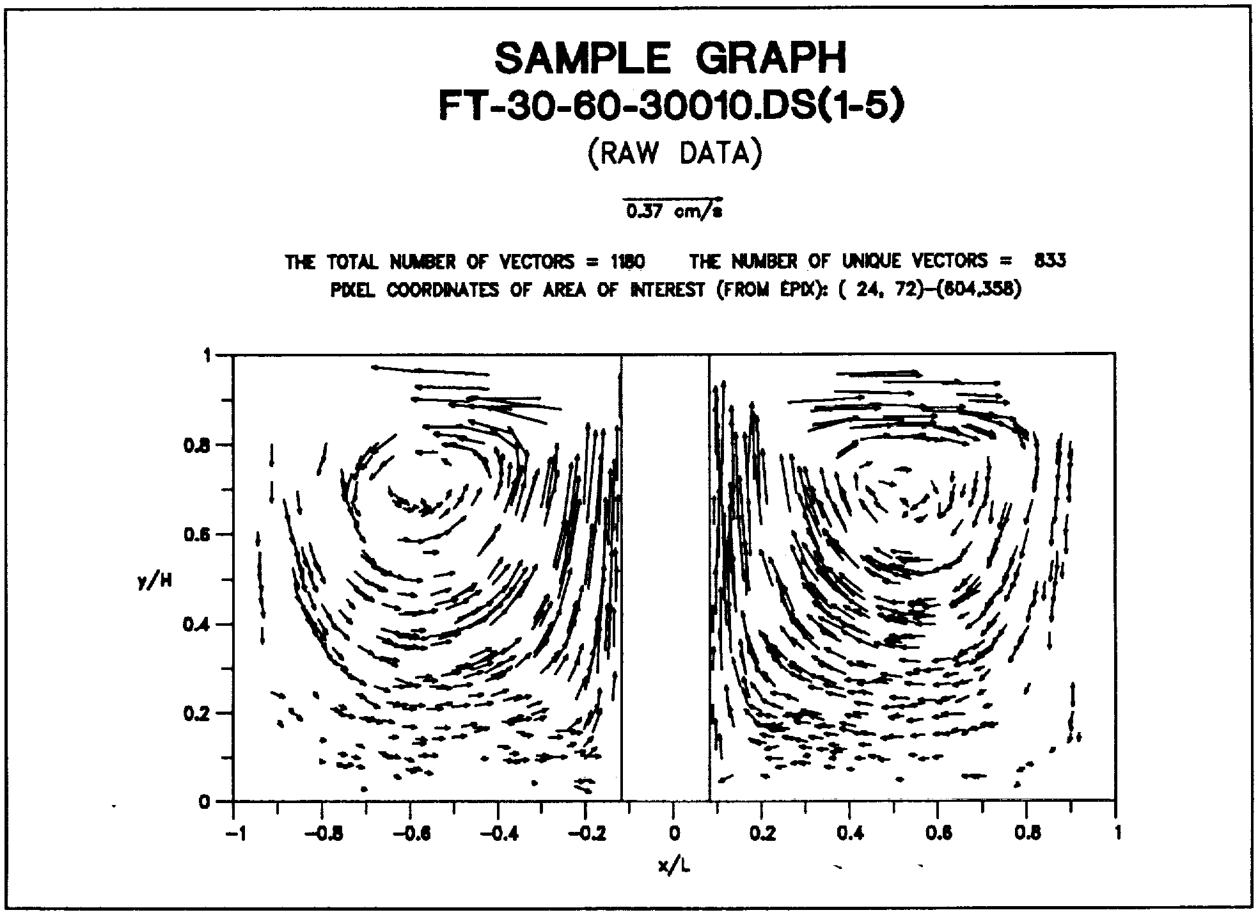


Figure 5: Raw velocity vector data file plotted in the STDC 2:1 aspect graph format with heater.

REMOVE IRREGULAR VELOCITY VECTORS

This operation is used to clean up the data sets obtained using the PDT technique. Not all data sets require the irregular vector removal operation. If the original image quality is high, the particles uniformly distributed and not too dense, and a sufficiently high threshold level is used, then very few incorrect identifications will be made. However, typically a few percent of the velocity vectors obtained will be improperly identified velocity vectors. These improperly identified velocity vectors can be removed by the routine. The main premise used in the irregular vector removal algorithm is that there are a sufficient number of 'good' velocity vectors surrounding the bad velocity vectors. The mean good qualities are used to eliminate the bad velocity vectors.

By selecting the menu option, the operation starts immediately; no user input is required. A banner is displayed on the computer screen showing the initial number of velocity vectors. The percentage completion of the irregular velocity vector removal operation is also displayed in the message banner. At the completion of the operation, the number of remaining vectors is displayed. To return to the main menu the user presses the return key.

The irregular velocity vector removal algorithm is similar to the interpolation algorithm. For each velocity vector, the 10 nearest neighboring velocity vectors from the test vector tail coordinate are found by expediently sorting the velocity vector data. The mean x- and y-velocity components are computed from the 10 nearest neighbors. The mean vector magnitude is also computed. Next, the standard deviations of the x- and y-velocity components are computed. The velocity vector under consideration is then tested by comparing the absolute values of the differences between the mean x- and y-velocities and the x and y velocity standard deviations. If the difference is greater than 3 times the computed standard deviation in either velocity component, the vector under test is rejected. The component tests are essentially insuring that the flow does not turn too sharply. In addition to the component tests, the magnitude of the velocity vector under test is compared to the computed mean vector magnitude. If the vector magnitude is greater than 3 times the mean vector magnitude of the 10 nearest neighbors, the vector is rejected. The final test requires that the nearest neighbor be within 50 pixels from the vector under test.

If the velocity vector passes the first battery of tests, then a second battery of tests is performed. The second series is identical to the first set, with the exception that the tests are performed about the vector head coordinate. Sometimes insufficient information exists about the tail of a velocity vector to make a proper judgement concerning its validity. The secondary test has shown to provide a more rigorous test of the velocity vector quality.

Under most circumstances a single pass of the irregular vector removal routine is required. For extremely erratic data, the routine may be executed a number of times, until no more bad velocity vectors are identified. The user should be cautioned, however, that typically a few good velocity vectors are eliminated along with the bad velocity vectors. The algorithm is not perfect, but does a suitable job, lest the user be inclined to remove the bad velocity vectors manually.

SELECT A DIFFERENT GRAPH TYPE

The change graph type menu option allows the user to change from the default 640×480 pixel velocity vector plot to the 2:1 aspect ratio STDCE graph type. Other graphing options are also selected from the submenus. The first query is for the graph type: 1=FULL GRAPH 640×480, or 2=STDC 2:1 ASPECT. After selecting the graph format, the user is queried for the type of data analysis to perform. The choices are 1=Interpolation (velocity vectors), 2=Velocity Contours, and 3=Flow Streamlines. The interpolation option is the default and has already been discussed above. The randomly spaced velocity vector data are interpolated over a user defined grid. Both options 2 and 3 require an interpolated velocity vector grid as input. The velocity contour operation computes the contour levels for the velocity vector magnitude over the interpolated velocity vector grid. The contouring routine was written in house at NASA LeRC by the Computer Services Division. Given an a 2-D array of x,y,z values, the contouring routine finds the x,y coordinates of lines of constant z-values. A sample velocity contour graph is shown below for the same data plotted in figures 4 and 5.

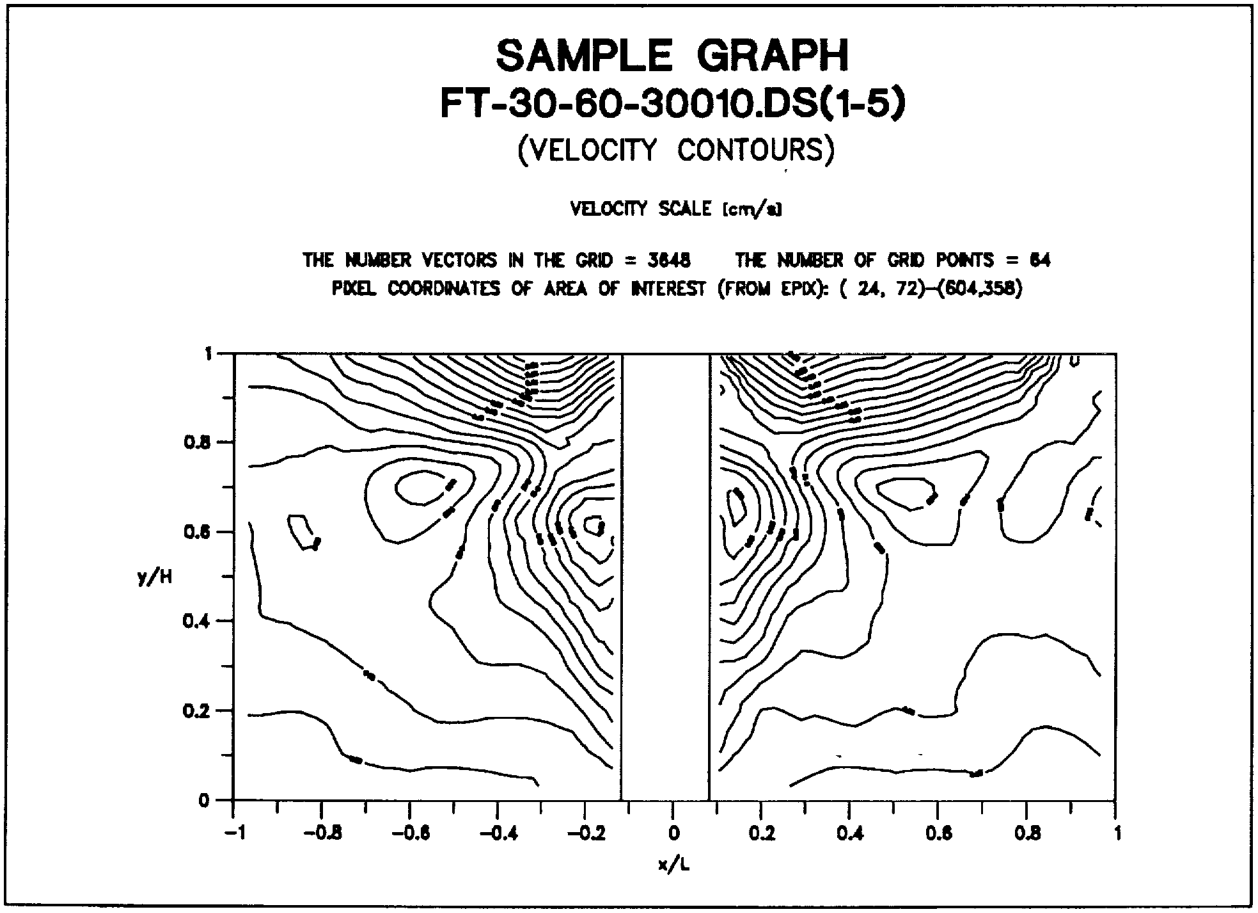
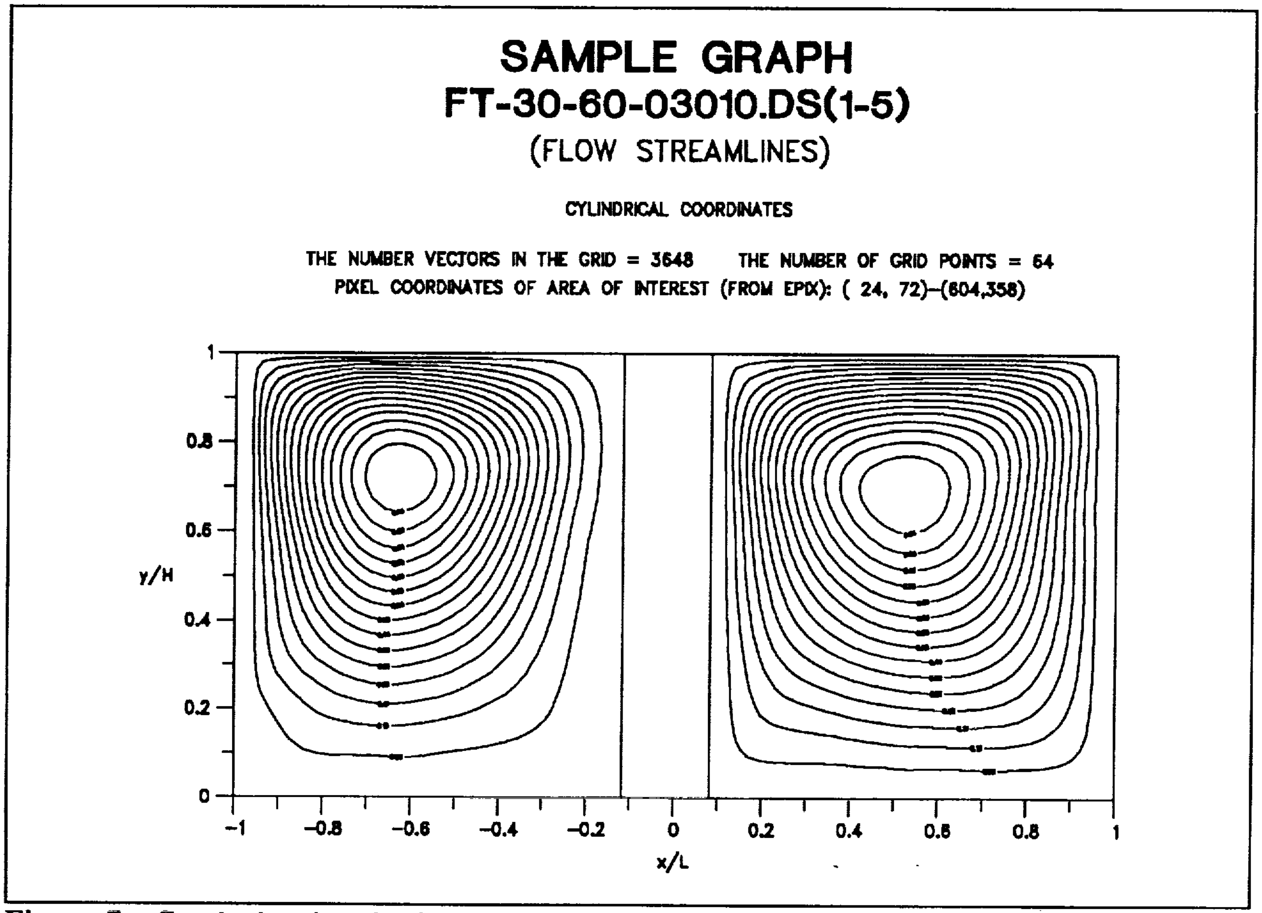


Figure 6: Velocity contour graph in STDC graph format with heater.

The streamline operation first computes the stream function for the flow field from the interpolated grid. The stream function computation uses a Gauss-Seidel integration and assumes that the stream function ψ, is zero at the boundaries of the interpolated velocity vector grid.[18] The $\psi=0$ boundary condition assumes that the flow under study is a closed system, which is appropriate for the STDCE graph type. The flow streamlines are computed using the contouring routine on the calculated stream function. The graph below shows the computed flow stream lines for the same data sets plotted in figures 4 and 5.


SAMPLE GRAPH

FT-30-60-03010.DS(1-5)

(FLOW STREAMLINES)

CYLINDRICAL COORDINATES

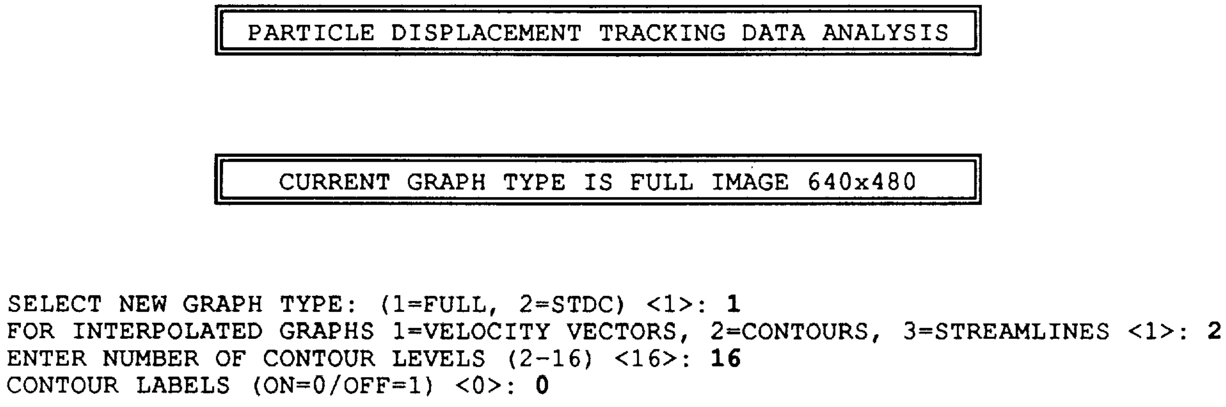
THE NUMBER VECTORS IN THE GRID = 3648 THE NUMBER OF GRID POINTS = 64

PIXEL COORDINATES OF AREA OF INTEREST (FROM EPIX): (24, 72)-(604,358)

Figure 7: Graph showing the flow streamlines in the STDC graph format with heater.

The subsequent number and type of questions presented to the user vary depending on the selected graph format. Screen #5a is shown below for the FULL GRAPH 640×480 graph format. After selecting the graph type the program queries the user for the type of data analysis. The default type of data analysis is the same as the previously selected type. At the start of the program, the default data analysis type is interpolation. If a new type of data analysis is desired select the new type by entering the appropriate option number. If either option 2 or 3 is selected, then two additional questions are asked. The first question is the number of contour levels to display, from 2 to 16. The default number is 16 contour levels. The second question is whether or not to display contour labels. If option 2, velocity contours, has been selected, there are two possibilities for the contour labels displayed. If no velocity magnitude scale has been entered into the program when the data files were read, then the contour labels indicate the velocity vector magnitude in pixels. If a non-zero velocity scale has been entered, then the contour labels are in units of cm/sec. If option 3, flow streamlines, has been selected, then the contour labels are the normalized values of the stream function from 0 to 1, regardless of the velocity scale. After the contour label on/off flag has been set, the program returns to the main menu.

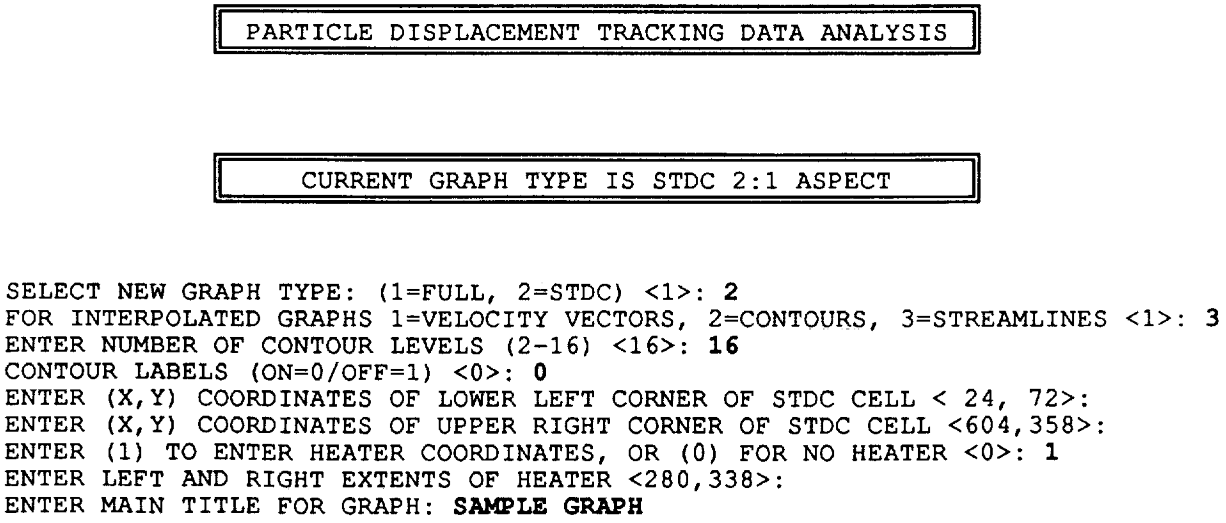
SCREEN #5a



```
PARTICLE DISPLACEMENT TRACKING DATA ANALYSIS

CURRENT GRAPH TYPE IS FULL IMAGE 640x480

SELECT NEW GRAPH TYPE: (1=FULL, 2=STDC) <1>: 1
FOR INTERPOLATED GRAPHS 1=VELOCITY VECTORS, 2=CONTOURS, 3=STREAMLINES <1>: 2
ENTER NUMBER OF CONTOUR LEVELS (2-16) <16>: 16
CONTOUR LABELS (ON=0/OFF=1) <0>: 0
```

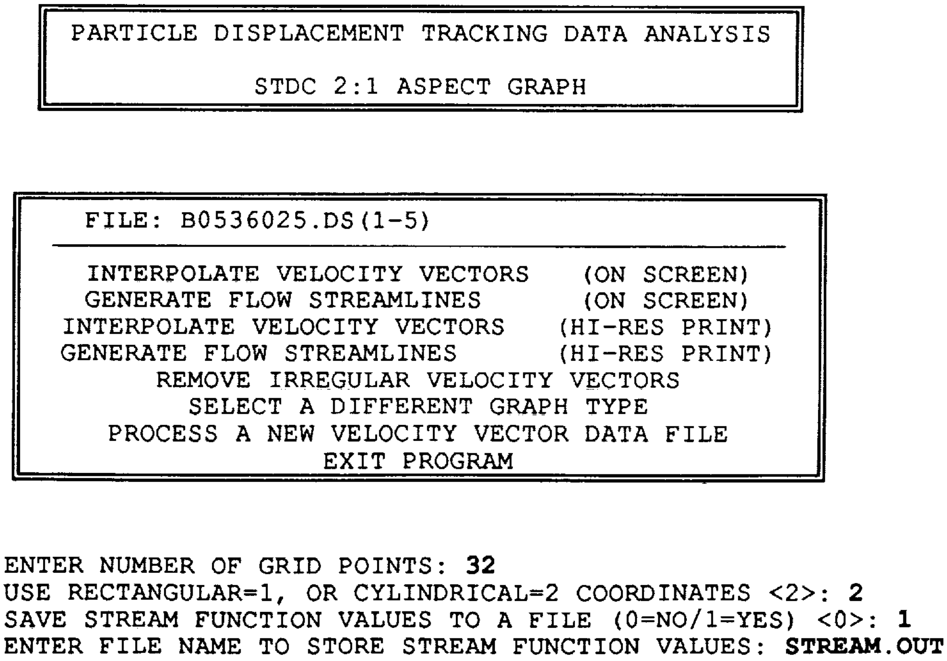
Screen #5b is shown below for the case of the STDCE 2:1 ASPECT graph type. The first four questions to the user are identical to the case above in screen #5a. However, there is additional information required for the STDC graph type. The boundaries of the actual test cell in pixel coordinates must be answered in questions 5 and 6. The default values are from a baseline data set previously stored in the computer. The next query determines whether the heater was used in the STDC setup for the current data set. If the heater was present, the next query requests the horizontal coordinates of the heater (the heater spans the entire height of the test cell). The final question is for a graph title, which appears at the top of the graph above the file name indication title.

SCREEN #5a



```
PARTICLE DISPLACEMENT TRACKING DATA ANALYSIS

CURRENT GRAPH TYPE IS STDC 2:1 ASPECT

SELECT NEW GRAPH TYPE: (1=FULL, 2=STDC) <1>: 2
FOR INTERPOLATED GRAPHS 1=VELOCITY VECTORS, 2=CONTOURS, 3=STREAMLINES <1>: 3
ENTER NUMBER OF CONTOUR LEVELS (2-16) <16>: 16
CONTOUR LABELS (ON=0/OFF=1) <0>: 0
ENTER (X,Y) COORDINATES OF LOWER LEFT CORNER OF STDC CELL < 24, 72>:
ENTER (X,Y) COORDINATES OF UPPER RIGHT CORNER OF STDC CELL <604,358>:
ENTER (1) TO ENTER HEATER COORDINATES, OR (0) FOR NO HEATER <0>: 1
ENTER LEFT AND RIGHT EXTENTS OF HEATER <280,338>:
ENTER MAIN TITLE FOR GRAPH: SAMPLE GRAPH
```

The main menu is affected by the choice of the data analysis type. If the default of just interpolated velocity vectors is selected, then there is no change in the main menu previously displayed in screen #4. If option 2=velocity contours is selected, the INTERPOLATE VELOCITY VECTORS option in the main menu is replaced with GENERATE VELOCITY CONTOURS. Alternatively, if option 3=streamlines is selected, the INTERPOLATE VELOCITY VECTORS option in the main menu is replaced with GENERATE FLOW STREAM LINES. When either option 2 or 3 is selected from the main menu, the user is asked to enter the number of grid points to perform the interpolation. The velocity vector data must be interpolated before the contours or streamlines can be computed. The data interpolation message banner is displayed on the computer screen while the data are being interpolated. At the completion of the interpolation, the velocity contours or flow streamlines are generated. For the GENERATE FLOW STREAMLINES ON SCREEN option, after the user enters the number of grid points, two additional questions are presented. The screen display below shows the flow streamline option activated and the three questions asked when the GENERATE FLOW STREAMLINES (ON SCREEN) option is selected. The second question pertains to the configuration of the test cell. For STDC, the test cell is circularly symmetric, hence, cylindrical coordinates are used to compute the stream function. The alternative option is rectangular coordinates. The last question asks the user if the computed stream function values are to be written to a file. If true, the user is queried for the name of the file to store the data. The first element in the file is the number of data points contained in the file. The data are stored in triplets x,y,ψ, in F12.4 format. The option of writing the stream function values to a file is only available for the (ON SCREEN) option, not for the (HI-RES PRINT) option.



```
PARTICLE DISPLACEMENT TRACKING DATA ANALYSIS

         STDC 2:1 ASPECT GRAPH


  FILE:  B0536025.DS(1-5)
 ------------------------------------------------------
  INTERPOLATE VELOCITY VECTORS      (ON SCREEN)
  GENERATE FLOW STREAMLINES         (ON SCREEN)
 INTERPOLATE VELOCITY VECTORS     (HI-RES PRINT)
 GENERATE FLOW STREAMLINES        (HI-RES PRINT)
      REMOVE IRREGULAR VELOCITY VECTORS
         SELECT A DIFFERENT GRAPH TYPE
    PROCESS A NEW VELOCITY VECTOR DATA FILE
                 EXIT PROGRAM


ENTER NUMBER OF GRID POINTS: 32
USE RECTANGULAR=1, OR CYLINDRICAL=2 COORDINATES <2>: 2
SAVE STREAM FUNCTION VALUES TO A FILE (0=NO/1=YES) <0>: 1
ENTER FILE NAME TO STORE STREAM FUNCTION VALUES: STREAM.OUT
```

PROCESS A NEW VELOCITY VECTOR DATA FILE

This menu option clears all of the data arrays and starts back at screen #1 to query the user for the type of data files to be read. All of the current graph format and data analysis options are preserved.

EXIT PROGRAM

The function of this menu option is to terminate the program execution and return to the DOS prompt.

6). REFERENCES

1) Meynart, R.;"Instantaneous velocity field measurements in unsteady gas flow by speckle velocimetry", *Appl. Opt.* **22**, 535-540 (1983).

2) R. J. Adrian, C. S. Yao,"Pulsed Laser Technique Application to Liquid And Gaseous Flows And The Scattering Power of Seed Materials", *Appl. Opt.* **24**, 44-52 (1985).

3) L. Lourenco, A. Krothapalli,"Particle Image Displacement Velocimetry Measurements of a Three-Dimensional Jet", *Phys. Fluids* **31**, 1835-1837 (1988).

4) R. J. Adrian, C. S. Yao,"Development of Pulsed Laser Velocimetry (PLV) for Measurement of Fluid Flow", in *Proceedings, Eighth Biennial Symposium on Turbulence*, X. B. Reed, Jr., J. L. Kakin, and G. K. Patterson, Eds., 170-186 (U. Missouri Rolla 1983).

5) Adrian, R. J.,"Image Shifting Technique to Resolve Directional Ambiguity in Double-Pulsed Velocimetry", *Appl. Opt.* **25**, 3855-3858, (1986).

6) L. P. Goss, M. E. Post, D. D. Trump, B. Sarka, C. D. MacArthur, G. E. Dunning, Jr.,"A Novel Technique for Blade-to-Blade Velocity Measurements in a Turbine Cascade", *25th Joint Propulsion Conference*, AIAA-89-2691, (1989).

7) Y. C. Cho, B. G. McLachlan,"Personal Computer Based Image Processing Applied to Fluid Mechanics Research", *SPIE Applications of Digital Image Processing* **829**, 253-257 (1987).

8) K. A. Marko, L. Rimai,"Video Recording and Quantitative Analysis of Seed Particle Track Images in Unsteady Flows", *Appl. Opt.* **24**, 3666-3672 (1985).

9) B. Khalighi, Y. H. Lee,"Particle Tracking Velocimetry: An Automatic Image Processing Algorithm", *Appl. Opt.* **28**, 4328-4332 (1989).

10) C. Batur, M. J. Braun,"Measuring Flow With Machine Vision", INTECH, 34-38 (May 1989).

11) B. R. Frieden, C. K. Zoltani,"Fast Tracking Algorithm for Multiframe Particle Image Velocimetry Data", *Appl. Opt.* **28**, 652-655 (1989).

12) Y. C. Cho,"Digital Image Velocimetry", *Appl. Opt.* **28**, 740-748 (1989).

13) Wernet, M. P.,"Particle Displacement Tracking for PIV", NASA TM 103288, November, (1990).

14) EPIX 4MEG VIDEO USER'S GUIDE, Revision 3.1, EPIX Inc, (1989).

15) Wernet, M. P., Edwards, R. V.,"New Space Domain Processing Technique for Pulsed Laser Velocimetry", *Appl. Opt.* **29**, 3399-3417, (1990).

16) Press, W. H., Flannery, B. P., Teukolsky, S. A., Vettering, W. T., *Numerical Recipes*, Cambridge University Press, New York, (1986).

17) Foley, T. A.,"Scattered Data Interpolation and Approximation With Error Bounds", *Computer Aided Geometric Design*, **3**, 163-177, (1986).

18) Carnahan, B., Luther, H. A., Wickes, J. O., *Applied Numerical Methods*, John Wiley & Sons, New York, (1969).

7). APPENDICES

APPENDIX I

Particle Image Boundary Processing Algorithm

The boundary processing routine sequentially processes all 5 fields in the acquired sequence. The images to be processed are compared to the user supplied threshold level as they are read into the computer memory. The threshold level is set just high enough to remove the background noise from the CCD camera. The boundary search is initiated in the upper left corner of the image, searching from left to right for non-zero pixel amplitudes. When a non-zero amplitude pixel is detected, the program searches the particle image boundary in a clockwise manner. Starting at the initially detected non-zero amplitude pixel, the program looks at the pixel on the right, and then looks at the pixel below. If a non-zero amplitude pixel is found to the right, the program moves there. If a non-zero amplitude pixel is found below the current pixel, the program moves there. This search procedure (right-down) is repeated, and the coordinates of the upper right corner of the particle are recorded until the test fails. The program then switches to a look down, look left mode. This search precedence defines the lower right portion of the particle image perimeter, until a null case is found. The lower left corner of the particle image is found by a look left, look up precedence. When this search procedure obtains a null condition, the final search mode begins. The final search mode is a look up, look right, which defines the upper right corner of the particle. The search terminates when the initial search position is encountered. All of the detected non-zero particle image perimeter coordinates are stored.

Next, the program scans the perimeter coordinates to determine the minimum x and y extents, and the maximum x and y extents of the particle image. These limits are used to define a rectangular area which is integrated along the x and then y axes to determine the x and y-projections of the particle image. The particle image's intensity weighted mean x and mean y coordinates are computed from the respective projections. The particle image centroid coordinates, along with the current field number being processed are stored. The pixel amplitudes in the rectangular area delimiting the particle image are then set to zero amplitude. Setting the pixel amplitudes to zero avoids redetecting the particle image on a subsequent pass. The program then resumes searching in a left to right direction for the next non-zero pixel value. The entire bounded region of the image file is searched for particle image centroids.

The particle centroid data is then written to a time history file. The size of the time history file is 640×480 pixels, by 8 bits. The particle images are effectively 1 pixel in diameter, and the amplitude of the pixel indicates on which field in the 5-field sequence the particle was recorded.

The accuracy of the estimated particle image centroids depends upon the size and shape of the particle image. The particle image intensity distributions are assumed to be approximately

Gaussian. The accuracy of the estimate is proportional to the width of the particle image in pixels. A Maximum Likelihood Estimation analysis has been performed for Gaussian shaped particle images and the results are reported in reference 15. In reference 15, a graphical representation of the variance in the particle image centroid estimate as a function of the ratio of the particle image standard deviation to pixel width is presented. For a Gaussian distributed light intensity particle image, the e^{-2} diameter (d_{e-2}) of the particle is defined as 4σ, where σ is the standard deviation of the Gaussian distribution. The pixel width is ε_p. Hence, α_3 can be defined as $\sigma/\varepsilon_p = d_{e-2}/4\varepsilon_p$. For a typical particle image, which is nominally 5 pixels across ($\alpha_3 = 5/4 \cdot 1 = 1.25$), the estimated variance is $0.2 = \mathrm{Log}_{10}\{\alpha_1 Var(\alpha_2)\}$. Hence, $\sigma_{\alpha 2} = \pm 2.8/\alpha_1$, which means that the error in the estimated centroid decreases inversely with the recorded particle image intensity. A conservative estimate for the error in particle image centroid estimate is $\sigma_{\alpha 2} = \pm 0.5$ pixels, which corresponds to a small image brightness. This value for the error estimate is used in all subsequent analysis of the estimates of the error in the measured velocity vectors. For more discussion of the errors in the measured velocity vectors see references 13 and 15.

The conservative estimate for the error in the centroid estimate above is used because, as mentioned previously in the discussion of the boundary processing routines, image fields are acquired by the data acquisition software. The image fields contain only 240 lines, thus each line of video is doubled as the image field files are read into the computer. Doubling the number of rows in the image has a minimal effect on the particle centroid estimates, down to 3 pixel wide particles. For particle images smaller than 3 pixels, the vertical estimate of the particle image centroid may be off by as much as ±1 pixel.

APPENDIX II

The sources for all of the software complilers and frame-grabber hardware and software are given below.

NDP Fortran-386 Version 2.1
Microway Inc.
P.O. Box 79
Kingston, Ma 02364
508-746-7341

Microsoft C Version 5.1
Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, Wa 98073-9717

Microsoft Mouse Programmers Referrence Guide
Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, Wa 98073-9717

4-MEG Video Board
4MDRV Subroutine Library
PXIPL Image Processing Library
4MIP Interactive Image Processing Program
EPIX, Inc.
310 Anthony Trail
Northbrook, IL 60062
708-498-4002

HALO Professional Graphics Kernel System
(Fortran Version)
Media Cybernetics
Silver Spring, Md 20910
800-446-HALO

NASA
National Aeronautics and
Space Administration

Report Documentation Page

1. Report No. NASA TM-103720	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

Software Manual for Operating Particle Displacement Tracking
Data Acquisition and Reduction System

5. Report Date

January 1991

6. Performing Organization Code

7. Author(s)

Mark P. Wernet

8. Performing Organization Report No.

E-5951

10. Work Unit No.

505-62-50

9. Performing Organization Name and Address

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

11. Contract or Grant No.

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, D.C. 20546-0001

13. Type of Report and Period Covered

Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes

Responsible person, Mark P. Wernet, (216) 433-3752.

16. Abstract

This software manual describes the necessary steps required to record, analyze, and reduce Particle Image Velocimetry (PIV) data images using the Particle Displacement Tracking (PDT) technique. The new PDT system is an all electronic technique employing a CCD video camera and a large memory buffer frame-grabber board to record low velocity (≤20 cm/s) flows. Using a simple encoding scheme, a time sequence of single exposure images are time coded into a single image and then processed to track particle displacements and determine 2-D velocity vectors. All of the PDT data acquisition, analysis, and data reduction software is written to run on an 80386 PC.

17. Key Words (Suggested by Author(s)) Particle image velocimetry Digital image processing	18. Distribution Statement Unclassified – Unlimited Subject Category 35

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 44	22. Price* A03

*For sale by the National Technical Information Service, Springfield, Virginia 22161