5. CONCLUSIONS

The goal of this work was to develop optical diagnostic techniques to provide Sandia with advanced flow-measurement capabilities. The first technique is Particle Image Velocimetry (PIV), in which a flow seeded with small particles is photographed repeatedly at short time intervals. Image-processing techniques are applied to find particle velocities from particle positional changes in successive frames, which permits determination of the instantaneous flow field and associated turbulence quantities. Standard PIV uses a light sheet to illuminate a plane of the flow for 2-D measurements, while either volume illumination and multiple cameras, or multiple-exposure holography, are used for 3-D PIV. The second technique uses nonlinear photorefractive optical crystals to measure turbulence quantities more directly. A laser beam passed through the flow acquires a small perturbation signal related to the turbulence, which is superimposed on a large steady informationless signal, rendering interpretation difficult. The laser beam is then passed through one or more crystals to remove most of the steady portion, thereby greatly improving the signal-to-noise ratio.

The two-dimensional PIV system has been fairly well proven on two test flow configurations, a cavity flow and a thermal convection experiment. Both the cross-correlation tracking (CCT) and particle tracking algorithms have been tested and proven. Agreement between PIV, LDV, and computational simulations was reasonable.

The three-dimensional PIV algorithms were developed and tested on a flow in a thermal convection enclosure. However, this technique was unsuccessful due to several unresolved difficulties with the 3-D PIV including strong refractive index gradients in the test section.

The nonlinear photorefractive crystal $BaTiO_3$ has been demonstrated as a significant visualizer of turbulent vs. steady-state flow. However, we have identified two potentially significant limitations in that it (1) appears to visualize gradients differently, depending upon the relative orientation of the gradient to the crystal fanning orientation, and (2) visualizes refractive gradients, not absolute phase change, as previously reported in the literature. The desired quantification of photorefractive measurements proved to be very difficult for the reasons discussed above.

Appendix A- PIV Software Operating Manuals

A.1.Instructions for Using PIV Software for 2D Analysis

1) Obtain the images to be analyzed. The images should be in either a tiff (Tagged Image File Format *.TIF) format or a flat format (8 bit data, XY format, no header). The images can be captured with a film camera and then scanned into a computer to be stored in the proper format, but usually a CCD camera will be used with a frame grabber board and appropriate software to capture the images directly on a computer. The time between images must be recorded for conversion of measured pixel displacements to actual velocity values. Different time intervals can be used within a sequence, but it is easier to use a constant time interval.

2) Evaluate and adjust the images. This can be done with 4MIP if the images were captured with an EPIX board, or can be done using HALO Desktop Imager 2.2. Using the 4MIP software to evaluate the images has the advantage of saving the images directly to an 8 bit XY flat format which is needed later, avoiding saving the images in the tiff format. In order to use HALO, the images must be read in and saved as tiffs, and then converted to flat format using FILECONV.EXE. Despite this, the HALO software seems to be easier to use than the EPIX software. Either way, the images are to be evaluated and adjusted in order to minimize background noise and maximize the spots. This can be done by adjusting the contrast, brightness, and gamma values in HALO, or by using some of the several image processing techniques in 4MIP. The area to be analyzed must be found and recorded as x-start to x-final and y-start to y-final, in pixels. Any areas to be removed should be recorded in the same fashion, as rectangles that block out the unwanted area.

3) Find the spot locations. This is done using FSPOT.EXE, found in the C:\WORK directory. The FSPOT program reads images in flat formats and finds the locations of the spots in the image. The locations are saved to a *.DAT file to be read by the FROM2TO3 program.

4) Convert the spot data from 2D format to 3D format. This is done by FROM2TO3.EXE, in C:\WORK. All this does is read in the XY positions of the spots and save them as XYZ positions with z=0. This allows the 3D tracking programs to read the data.

5) Track the particles through the images. This can be done with either PST3D.EXE or TP3D.EXE, both found within the C:\WORK directory. Both programs read multiple *.DAT files (PST3D reads 2 frames, TP3D needs at least 4 and can track through several) and output a single *.DI3 file that contains the vectors found.

6) Convert the vector file into a TECPLOT file. This is done using MAKETEC.EXE in C:\WORK. It reads in a *.DI3 file and writes a *.TEC file, a data file ready to be read into the TECPLOT plotting program. The program MAKETEC can be used to weed out some of the 'bad' vectors, using the quality numbers printed out by the tracking programs.

7) Display the vectors. This can be done using TECPLOT, found in the TECPLOT directory. TECPLOT can be used to view and print the vectors saved using MAKETEC. In addition, a new zone can be created and TECPLOT will interpolate the vectors onto a regular grid. The interpolated data can be smoothed to remove the effect of bad vectors. Using the time between frames and the distance per pixel, the displacement vectors can be changed into velocity vectors. Streamlines can be obtained from the interpolated data.

A.2. Instructions for Using PIV Software for 3D Analysis

1) Obtain the images to be analyzed. The images should be in either a tiff (Tagged Image File Format *.TIF) format or a flat format (8 bit data, XY format, no header). It is recommended that three cameras be used to capture the data, but 3D analysis can be done with only two. The images can be captured with a film camera and then scanned into a computer to be stored in the proper format, but usually a CCD camera will be used with a frame grabber board and appropriate software to capture the images directly on a computer. The time between images must be recorded for conversion of pixel displacements to actual velocity values. Different time intervals can be used within a sequence, but it is easier to use a constant time interval.

2) Obtain calibration images. In order to combine the 2D images into 3D space, the locations of the cameras must be found. This is done with calibration images. At least six different points must be found with known world and camera points. These points should be within the substance that contains the particles to be tracked in order to compensate for refraction. The cameras cannot be moved until all the data images and the calibration images are captured, since this would give wrong positions for combining the camera images. For the magma chamber experiment, diode lasers were used to find these points. A beam from a laser on the top of the tank with known XZ position was crossed by an orthogonal beam from the side of the tank with known Y position. The place where the beams crossed was used as a calibration point with known world position. An image was taken with the cameras, from which the camera X and Y were found. This was done in 8 different positions (the corners of a cube) to get 8 calibration points (6 needed, 2 extra for checking). Another method is to place a dark card with light points on it at known XY positions in the flow. The card is then placed at different Z positions to get the necessary calibration points. Note: The world axes must be orthogonal to each other. Also, cameras cannot be placed parallel to the Y axis, so avoid putting cameras straight up or straight down. (Due to a quirk in the math, there is no solution for this angle. Of course, one can simply define a different axis to be the Y axis instead.)

3) Calibrate the cameras. This is done using CAMSOL.EXE, found in C:\WORK and on SAHP053 HP Workstation in the JAHENDER user directory. The CAMSOL program reads in the positions of the calibration points and outputs the positions of the cameras to be read by the CAMCOMB program. This step takes a while (order of days).

4) Evaluate the images. This can be done with 4MIP if the images were captured with the EPIX boards, or can be done using HALO Desktop Imager 2.2, which is found in Windows. Using the 4MIP software to evaluate the images has the advantage of saving the images directly to an 8 bit XY flat format which is needed later, avoiding saving the images as tiffs. In order to use HALO, the images must be read in and saved as tiffs, and then converted to flat format using FILE-CONV.EXE, found in C:\IMAGES on the PIV Computer. Despite this, the HALO software seems to be easier to use than the EPIX software. Either way, the images are to be evaluated and adjusted in order to minimize background noise and maximize the spots. This can be done by adjusting the contrast, brightness, and gamma values in HALO, or by using some of the several image processing techniques in 4MIP. The area to be analyzed must be found and recorded as x-start to x-final and y-start to y-final, in pixels. Any areas to be removed should be recorded in the same fashion, as rectangles that block out the unwanted area.

5) Find the spot locations. This is done using FSPOT.EXE, found in C:\WORK. The FSPOT program reads images in flat formats and finds the locations of the spots in the image. The locations are saved to a *.DAT file to be read by the FROM2TO3 program.

6) Combine the 2D spot data into 3D data. This is done by CAMCOMB.EXE, in C:\WORK. This reads in the camera positions from CAMSOL and the spot locations from FSPOT to find the position of the spots in 3D.

7) Track the particles through the images. This can be done with either PST3D.EXE or TP3D.EXE, both found within the C:\WORK directory. Both programs read multiple *.DAT files (PST3D reads 2 frames, TP3D needs at least 4 and can track through several) and output a single *.DI3 file that contains the vectors found.

8) Convert the vector file into a TECPLOT file. This is done using MAKETEC.EXE in C:\WORK. It reads in a *.DI3 file and writes a *.TEC file. The program can be used to weed out some of the 'bad' vectors, using the quality numbers printed out by the tracking programs.

9) Display the vectors. This can be done using TECPLOT, found in the TECPLOT directory. TECPLOT can be used to view and print the vectors saved using MAKETEC. In addition, a new zone can be created and TECPLOT will interpolate the vectors onto a regular grid. The interpolated data can be smoothed to remove the effect of bad vectors. Using the time between frames and the distance per pixel, the displacement vectors can be changed into velocity vectors. Streamlines can be obtained from the interpolated data.

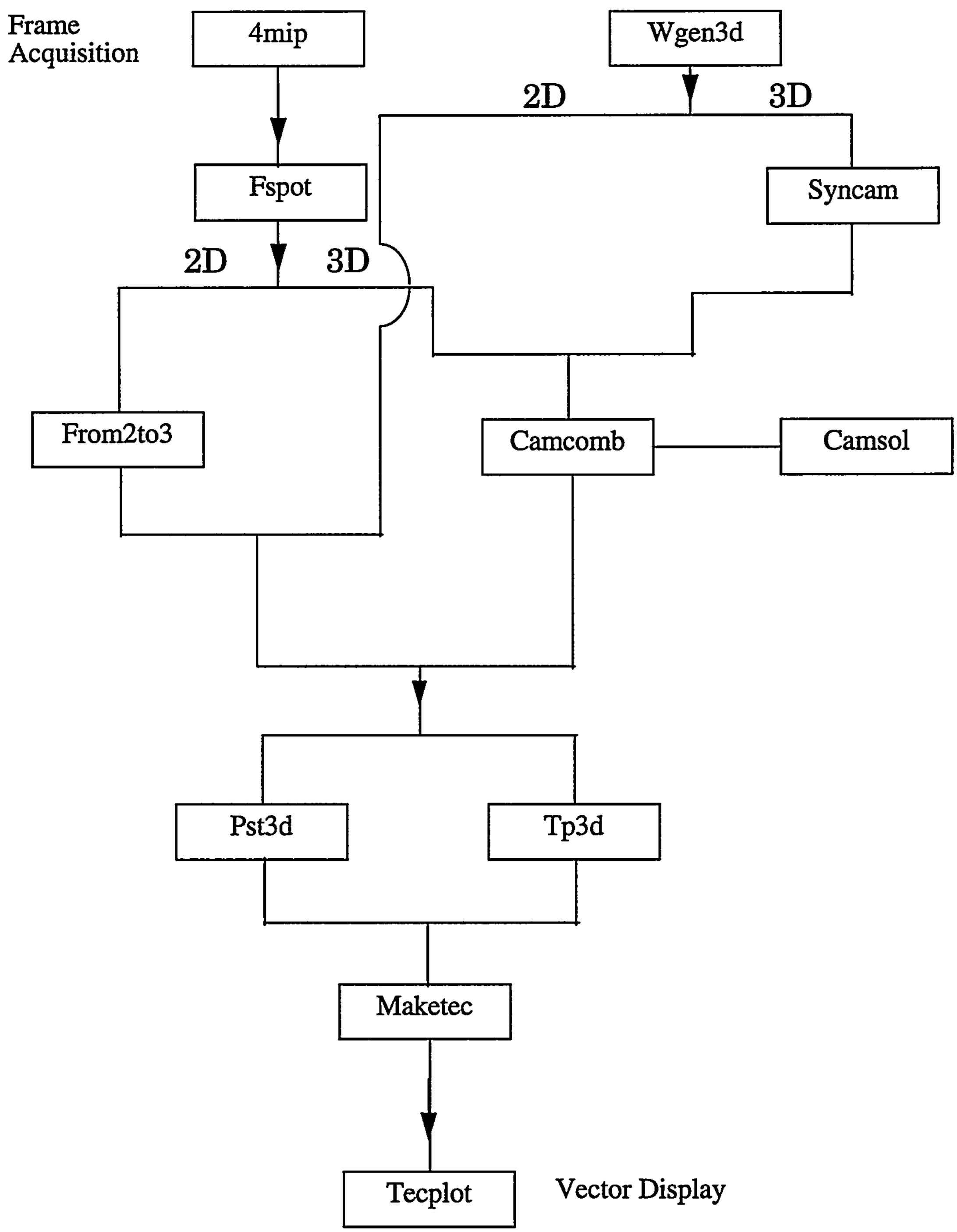

Figure A.1 Basic Outline for PIV software for 2D and 3D analysis

Appendix 2: 2D and 3D PIV Software

Detailed Descriptions

4MIP

4MIP is a commercial image capture/processing software package made to be used with the EPIX frame grabber boards. It has its own manuals, so no details will be given here. In short, 4MIP can be used to capture images from multiple boards, perform image processing on the images, and make measurements of objects in the images.

CAMCOMB(Camera Combination Program)

CAMCOMB is a program that takes the spots found on images taken simultaneously on two or three cameras and combines them into particle locations in three-dimensional space. In order to do this, it requires the locations of the cameras as outputted by the CAMSOL program. The program requires four or five files to be present in the same directory in order to run correctly. The first is the input deck, named CAMCOMB.INP. The input deck simply contains the names of the other files needed, and is structured as follows:

PARAMFILE	(Camera position and combination parameter file)
INPUTFILE#1 (*.DAT)	(Data points file for first camera)
INPUTFILE#2 (*.DAT)	(Data points file for second camera)
INPUTFILE#3 (*.DAT)	(Data points file for third camera, if 3 cameras)

The parameter file contains the number and positions of the cameras. It also contains information on the area of interest and maximum distance between possible spots. If combined spots are outside the area of interest, or are separated by more than the maximum distance, then they are ignored. This file is created by the CAMSOL program and is named PARAM.DAT unless renamed by the user. The input files contain the spot locations as found from each camera, and must be in the same order as the cameras in the parameter file (otherwise the program will use the wrong camera for the spots). These files are created by the FSPOT program and are named by the user when using FSPOT. Typically they have the *.DAT suffix. The program outputs 2 files; an output file and a data file. Both files contain the location of the particles in three-dimensional space. The output file is named CAMCOMB.OUT and is structured for easier reading by humans. The data file is named CAMCOMB.DAT and is structured to be read as a single frame by the tracking programs, TP3D and PST3D. The data file should be renamed for use appropriately; FRAME001.DAT, for example.

CAMSOL(Camera Calibration and Solution Program)

CAMSOL is a program that finds the location of two or three cameras based on the positions of calibration spots in each camera and in the world. Each calibration spot must be measured accurately and must be seen in each camera. The program requires three or four files to be present in the same directory in order to run correctly. The first is the input deck, named CAMSOL.INP. The input deck contains user parameters and the names of the other files needed, and is structured as follows:

NC	(Number of cameras, 2 or 3)
SX1,SX2,SX3	(Scaling factors in X to be used for the cameras, cu/wu)
CAMSTEP,CAMERR	(Initial step size and max. error for camera coordinates, cu)
WORSTEP,WORERR	(Initial step size and max. error for world coordinates, wu)
MINSTEP,MAXIT	(Minimum step size and maximum number of iterations)
XMIN,XMAX	(Area of interest in world coordinates, wu)
YMIN,YMAX	(Area of interest in world coordinates, wu)
ZMIN,ZMAX	(Area of interest in world coordinates, wu)
DMAX	(Maximum distance between possible spots, wu)
INPUTFILE#1 (*.DAT)	(Calibration points file for first camera)
INPUTFILE#2 (*.DAT)	(Calibration points file for second camera)
INPUTFILE#3 (*.DAT)	(Calibration points file for third camera, if 3 cameras)

The various factors are measured in camera units (cu) and world units (wu). These units can be anything consistent, but will usually be pixels and millimeters, respectively. Since there is an infinite number of possible solutions for any set of cameras, the x scaling factors are entered by the user to select a single solution. The program uses a numerical solver that uses partial differentiation to try to correct for error in the calibration point measurements. The user can control the solver using the *STEP, *ERR, and MAXIT variables. If any of these variables (except MINSTEP) is set to zero, then numerical correction will not be performed. The *MIN and *MAX variables are passed on in the output to be used by the CAMCOMB program. The input files contain the calibration spot locations as found in each camera and the world, and are created by the user. One must be created for each camera, and the spot locations must be in the same order in each input file. The same number of points should be used for each camera. The input files are structured as follows:

NS	(Number of calibration spots for this camera)
XC,YC,XW,YW,ZW	(location of first calibration spot on this camera and in world)
.	
.	
XC,YC,XW,YW,ZW	(location of last calibration spot on this camera and in world)

The camera locations (XC,YC) are found in the image, and are measured in camera units. The world locations (XW,YW,ZW) are measured in the experiment in world units. The program needs at least six calibration points to work, and uses any extra points to help check the locations of the camera after solving for them. The program outputs 5 files on the PC, 6 files on the HP. Two of the files are the output file and the data file. Both of these files contain the positions of the cameras. The data file is named PARAM.DAT and is structured to be read by the CAMCOMB program. The output file is named CAMSOL.OUT and it is structured to be read by humans. It also shows the original and shifted positions of the particles, as well as the error distance between them. Along with these, the HP also creates a file named CAMSOL.SCR, this is simply what

CAMSOL on the PC normally prints to the screen. The last three files are named LPARM1.DAT, LPARM2.DAT and LPARM3.DAT. These files contain the shifted position of the calibration particles. They can be used to change the input files to continue iterations where the run that created them stopped. Since these files are recreated after each iteration, even if the program stops because of power failure or program error, the program can be restarted and iterations can be continued. If iterations are continued, be sure to lower the *ERR variables in the input deck, since the spot locations may have already shifted by the initial amount.

FROM2TO3 (From Two-Dimensions to Three-Dimensions Program)

FROM2TO3 is a simple program that takes two-dimensional spot data produced by the FSPOT program and converts it to three-dimensional spot data capable of being read by either of the tracking programs, PST3D or TP3D. Input is interactive, the program asks what the name of the file to be changed is and what to name the file being created. While the data is being converted, a number of other changes can be made. The program will ask for an angle to rotate the data by, enter 0 if no rotation is desired. It will also ask by how much the image is to be shifted, enter 0,0 if no shift is desired. The program will then ask the start and final coordinates of the image. This can be used to break a seed-dense image into two smaller images by giving halfway marks as the final coordinate on the first run and then as start coordinates on a second run. If the entire set of data is to be transformed, then make sure the start and final coordinates are larger than the original image the data was obtained from. Finally, the program asks for a coordinate multiplier, if no multiplication is desired, enter 1. Note that the transformations are performed in the following order:

1) shift position of data
2) rotate data
3) multiply data
4) drop data if outside of start and final coordinates
5) add z coordinate, equal to zero

After transforming data, the program asks if you wish transform some more. Enter 1 to transform a new frame of data, or enter 2 to transform the same data again (if breaking the data into two frames, for example). Any other number will quit the program.

FSPOT (Find Spots Program)

FSPOT is a program that examines an 8-bit flat image and identifies the spot locations on the it. A flat image is simply an image file with no header, with the pixel information in XY order. The 8-bits represent 256 gray-scale levels for each pixel. 8-bit flat images can be created by 4MIP, or by converting another image format using the FILECONV program. The FLATTEN batch file, found in the C:\WORK directory, uses the FILECONV program to convert tiff images in a directory into flat images. To use flatten, type it and the complete directory name where the tiff files are located, i.e., 'C:>FLATTEN \WORK\CAL0801' makes flat images for every tiff image found in the C:\WORK\CAL0801 directory. FSPOT requires the image files and an input deck file to be in its directory to run. Input can be interactive or through the input deck, since FSPOT will edit its input deck if ran interactively. There are two possible command line inputs also. If '-b' is entered after the FSPOT command, i.e., 'C>FSPOT -b', then FSPOT will run in batch mode, using the input deck as is and asking for absolutely no input by keyboard. If -i is entered after the command, and is followed by a colon and a filename, then that file will be used as the input deck. These two commands can be used together, 'C>FSPOT -b -i:TEMP.INP', for example, will use TEMP.INP as the input deck and will run without user input from keyboard. An input deck for FSPOT is structured as follows:

INPUTFILE	(*.FLA)	(Image file to be analyzed)
OUTPUTFILE	(*.OUT)	(Desired name or "DEFAULT") (*1)
DATAFILE	(*.DAT)	(Desired name or "DEFAULT") (*1)
TECFILE	(*.TEC)	(Desired name, "NONE", or "DEFAULT") (*1),(*2)
HISTOFILE	(*.HIS)	(Desired name, "NONE" or "DEFAULT") (*1),(*2)
PICTYPE		("epix", can use just "e")
LOCALSIZE		(Local thresholding region size, cu) (*3)
LEVEL1,LEVEL2		(Zero level and upper level for thresholding, greylevel) (*4)
TOTALX,TOTALY		(Total horizontal and total vertical size, cu)
STARTX,STARTY		(Starting location to use in processing, cu)
FINALX,FINALY		(Final location to use in processing, cu)
WIDTH,HEIGHT		(Width and height of a frame in world units, for reference)
SMTHLO,SMTHHI		(Low and high number of pixels for smoothing calculations)
MINAREA,MAXAREA		(Minimum & maximum number of pixels to keep spot)
TIME		(Time in seconds between frames, for reference)
NBURN		(Number of burnspots to be removed) (*5)
BURNX,BURNY		(Location of burnspot, cu)
NBUB		(Number of bubbles for 1/r^2 thresholding) (*5),(*6)
BUBX,BUBY		(Bubble centroid location, cu)
BUBR,BUBTHRSH		(Bubble radius & threshold within radius, cu, greylevel)
NAREA		(Number of rectangular areas to remove) (*5)
AXS,AXF		(Start x and final x of area to remove, cu)
AYS,AYF		(Start y and final y of area to remove, cu)
ANSWER		(Do another file (y/n)?)

Notes on Inputdeck

(*1) if "DEFAULT" is typed as file name (No quotes, but must be in capital letters) file name is <Inputfilename Prefix><Appropriate Suffix>. The appropriate suffixes are: *.OUT, .*.DAT, .*.TEC, & *.HIS for output, data, tecplot, and histogram files, respectively.

(*2) if "NONE" is typed as file name (Same format as in *1, above) then no file of that type is produced for this set of data

(*3) if 0 is entered, constant thresholding is used

(*4) Only used for constant thresholding, but must be entered

(*5) 0 for none

(*6) Not performed if local thresholding is used

Indented entries are repeated (as a group) a number of times equal to the last non-indented entry above them. If that entry was 0, then nothing should be entered for the indented entries. If the answer is "y" the same format is duplicated until the answer is "n". However, it is recommended that FSPOT be used interactively until the user is familiar with the various options, since the interactive selections have more details on the options (besides, it was a lot of typing). The FSPOT program produces two to four output files. The output file and data file are always created, and both contain information on the spots located. The data file is formatted to be read by the CAMCOMB or FROM2TO3 programs, while the output file is formatted for easy reading. The TECPLOT file, if produced, is structured to be read as a contour plot for TECPLOT. It is simply a binary image showing the spots found. The histogram file, if produced, is simply the number of pixels in the images at each gray level. It can be used with any graphing routine that reads ASCII to show the histogram of the original image

MAKETEC (Make Tecplot File Program)

MAKETEC is a simple program that takes a vector file produced by TP3D or PST3D (typically with a *.DI3 suffix) and creates a file readable by TECPLOT from it. It can also make a 256 level contour plot for TECPLOT from a two-dimensional flat 8-bit image. Input is interactive, the program asks the user for the name of the input file and the type of file it is (output from TP3D, output from PST3D, or a two-dimensional image). It then creates a TECPLOT ASCII file of the same name, but with the *.TEC suffix. Depending on the type of file being analyzed, furtherinput is required. For a TP3D file, a sigma value cutoff is required. A good track produced by TP3D has a sigma value close to zero, so this cutoff will drop any tracks with sigma values greater than it. For a PST3D file, good vectors have a Cij value close to one and a large pair confidence (PC) value. Therefore, the program requires Cij and PC cutoffs, and will drop any vector with lower values than these. It also requires a velocity cutoff, which generally should have the same value as the search region for PST3D. If a two-dimensional flat image is being converted, then the width and height of the image are required, in pixels. The program also allows some simple manipulation of the image gray level, allowing the user to enter multiplication and addition factors to the gray level. Any gray levels greater than 255 are set to 255. Finally, the program will ask if another file is to be transformed. Entering 1 will start the program over while any other number will quit.

PST3D (Pick & Slide Tracking in Three-Dimensions Program)

The PST3D program compares spot location in two frames of data, and finds the three-dimensional vectors between the two frames. It does this using a method based on the two-dimensional method of cross correlation. PST3D requires an input deck and two spot data files as input. The input deck is named PS3.PIN and should be in the same directory as PST3D when ran. It is structured as follows:

NF	(Number of frames, must be 2 for this version)
INPUTFILE1 (*.DAT)	(First spot data file)
INPUTFILE2 (*.DAT)	(Second spot data file)
XS,YS,ZS	(Starting position to be used in processing, wu)
XF,YF,ZF	(Final position to be used in processing, wu)

DX,DY,DZ	(Candidate region half-widths, wu)
DXVS,DYVS,DZVS	(Dynamic region starting half-widths, wu)
IPPN	(Initial possible pair number)

(recall "wu" stands for "world units.") This version of PST3D is not set up for single frame tracking (autocorrelation), e.g., double exposure photography, so the number of frames must always be two. The input files contain the spot locations as produced by either the CAMCOMB program or the WGEN3D program. The starting and final positions, and the half-widths, should be measured in the world coordinates used for the calibration points. The candidate region half-widths are used to find possible pairs between the frames, and should be based on the maximum velocity possible in the volume of interest. If the value is too low, then the correct vectors will not be found; if too large, then the running time is increased, and chances of finding an incorrect vector is increased. The dynamic region starting half-widths are used to determined the comparison volume between the two frames for each spot. This volume will increase in order to encompass at least ten spots in the second frame to compare with. The initial possible pairs number is used for array allocation on the PC, if it's too small the program will increase it and start over, using up more time, but if it's too large the PC may run out of memory, crashing the program. If the program tries to increase the IPPN and crashes, then there may be too many possible vectors to analyze. Either decrease the dx and dy, or split the image into sections for separate analysis. Output is to written to four files. A data file named PS.DI3 is created and contains the information on the best vectors found. It is formatted to be read by the MAKETEC program. The output file, PAIRCON.DAT, is structured for reading by humans. It contains information on all the possible vectors. Two other files, CIJ.DAT and PC.DAT, contain the distribution of the Cij and PC values, respectively. They can be plotted and examined in order to determine the best cutoff values.

SYNCAM (Synthetic Camera Program)

SYNCAM is a program that takes the synthetic three-dimensional data produced by the WGEN3D program and transforms it into two-dimensional data as viewed by a camera. Input is by input deck, which names the data file to be read. The input deck is named SYNCAM.INP, and is structured as follows:

INPUTFILE (*.DAT)	(Data file containing spot locations)
NC	(Number of cameras to be modeled, 1-3)
TH,AL,BE	(Camera rotation angles, for first camera, radians)
DX,DY,DZ	(Shift from world origin to first cameras origin)
SX,SY,SZ	(Scaling factors for first camera)

(Repeat last 3 lines for 2nd and 3rd camera, if necessary)

For each camera modeled, two files are produced. The first is a file named SYNCAM#.DAT (where # is the number of the camera) which contains the two-dimensional data in the same format as a data file created by the FSPOT program. The second is named CALIB#.DAT, and can be used as an input file for the CAMSOL program. In addition, a file named SYNCAM.OUT is produced, which simply contains the positions of all the cameras being modeled for later comparison with the CAMSOL program.

TECPLOT

TECPLOT is a commercial software package made for all sorts of data analysis. Since it has its own manuals, no details will be given here. TECPLOT can be used for making graphs, drawing vectors (2D or 3D) and representing images as contour plots.

TP3D (Track Particle in Three-Dimensions Program

TP3D compares spot locations in four or more frames of data, and finds tracks throughout all the frames. It does this by comparing the possible tracks to paths of constant curvature. It requires an input deck file to tell it the names of the four to fourteen input data files, along with search radius information. The input deck is named TP3D.TIN, and is simply structured as follows:

```
R1,R2,R0            (Search radii)
INPUTFILE1          (First input file)
INPUTFILE2          (Second input file)
   ...
INPUTFILE?          (Last input file)
END                 ("END" or "end", tells TP3D to stop reading input deck)
```

The search radii are used to determine which particles to create tracks with. Similar to the half-widths for PST3D, it's important to get their values in the right range. R1 is the search radius for the second particle in a track, and its value should be the maximum velocity in the frames. R2 is the search radius for the particles after the second frame, and should be the maximum acceleration in the frames. R0 is the minimum search radius for the second particle, and should be the minimum velocity in the frames. The input files contain the locations of the spots in three-dimensional space, as created by either the CAMCOMB program or the WGEN3D program. Output is to three files. The first is TP3D.DI3, which contains the best track information, and is structured to be read by the MAKETEC program. The other two, named TP3D.FNL and TP3D.XYS, contain the track information in different formats.

WGEN3D (Working Generate Three-Dimensional Data Program)

WGEN3D creates a series of frames of synthetic three-dimensional data, which can be tracked to test the other 3DPIV programs. Input is by hand, and no data files are needed for the program. The program can create from 2 to 14 frames of data. From 1 to 10000 spots can be randomly distributed across the first frame of data. These spots are then moved across the other frames by numerical equations representing flow around a sphere, spray flow, or reverse drain flow. A freestream velocity is required. The initial seed placement can be restricted in order to prevent seeds from leaving the area of interest and being lost from later frames. Finally program asks the user to define the area of interest, by asking for a r(ange) and a c(onstant). With this information, the program defines a cube with each side being r units long, shifted along all axes c amount. There is a data file created for each frame, containing the spot locations in a format similar that produced by the CAMCOMB program. These files are named GEN3DF#.DAT, with # being the number of the frame. In addition, two other files are created describing the flow. The first is GENSTATS.DAT, which just describes the user's selections for creating the data. The second is GEN3D.DI3, which is formatted like output from the TP3D program, and can be used to compare with the results of the tracking programs. It can also be used with MAKETEC to create a vector plot in TECPLOT.

References

Adrian, R. J., 1991, "Particle Imaging Techniques for Experimental Fluid Mechanics," *Annual Review of Fluid Mechanics*, **23**, Annual Reviews, Inc., Palo Alto, CA.

Albrecht, G. F., Robey, H. F., and Moore, T. R., 1990, "Optical Technique for the Measurement of Turbulent Spectra Using the Photorefractive Properties of $BaTiO_3$," *Applied Physics Letters*, **57** (9), 864-866.

Blanchat, T. K., 1991, "Development of Pulsed Laser Velocimetry Techniques for Measurement of Two-Phase Interfacial Drag in a Horizontal Stratified Flow," Ph.D. Thesis, Texas A&M University, College Station, TX.

Buchhave, P., and Jakobsen, M. L., 1990, "Optical Correlator for PIV Image Processing," *Proceedings of the 3rd International Conference on Laser Anemometry*, 609-616.

Chu, T. Y., and Hickox, C. E., 1990, "Thermal Convection with Large Viscosity Variation in an Enclosure with Localized Heating," *J. Heat Transfer* **112**, 388-395.

Chu, T. Y., Dunn, J. C., Finger, J. T., Rundle, J. B., and Westrich, H. R., 1990, "The Magma Energy Program," *Geothermal Resources Council Bulletin*, **19** (2), 42-52.

Durst, F., Melling, A. and Whitelaw, J. H., 1981, Principles and Practice of Laser-Doppler Anemometry, Second Edition, Academic Press, New York, p. 369.

Fluid Dynamics International, 1993, *FIDAP Users Manual Revision 7.0*, Fluid Dynamics International, Evanston IL.

Gonzales, R. C., and Wintz, P., 1987, Digital Image Processing, Ch. 2.5, 36-54.

Grant, I., ed., 1994, Selected Papers on Particle Image Velocimetry, SPIE Volume MS99, The Society of Photo-Optical Instrumentation Engineers.

Hassan, Y., and Blanchat, T., 1991a, "Full-Field Bubbly Flow Velocity Measurements by Digital Image Pulsed Laser Velocimetry," *Experiments in Fluids*, **11**, 293-301.

Hassan, Y., and Blanchat, T., 1991b, "Flow Velocity Measurements Using Digital Image Pulsed Laser Velocimetry," *Optical Engineering*, **30** (8), 1220-1227.

Henderson, J. A., Masters thesis, Texas A&M University, 1994.

Hickox, C. E., and Chu, T. Y., 1991, "Developing Natural Convection in a Fluid Layer with Localized Heating and Large Viscosity Variation," in FED Vol. 107, *General Topics in Fluids Engineering*, edited by F. M. White and K. Kamemoto (American Society of Mechanical Engineers, New York), 75-81.

Iwatsu, R., Ishii, K., Kawamura, T., Kawamura, K., and Hyun, J. M., 1989, "Numerical Simulation of three-dimensional flow structure in a driven cavity," *Fluid Dynamics Research*, **5**, 173-189.

Koseff, J. R., and Street, R. L., 1984a, "Visualization Studies of a Shear Driven Three-Dimensional Recirculating Flow," *J. Fluids Engineering*, **106**, 21-29.

Koseff, J. R., and Street, R. L., 1984b, "On End Wall Effects in a Lid-Driven Cavity Flow," *J. Fluids Engineering*, **106**, 385-389.

Koseff, J. R., and Street, R. L., 1984c, "The Lid-Driven Cavity: A Synthesis of Qualitative and Quantitative Observations," *J. Fluids Engineering*, **106**, 390-396.

Lourenco, L. M., Krothapelli, A., and Smith, C. A., 1989, "Particle Image Velocimetry," in Advances in Fluid Mechanics Measurements, Gad-el-Hak, M., editor, Springer-Verlag, Berlin, 128-197.

Marrakchi, A., Petersen, P. M., and Buchhave, P., 1991, "Photorefractive Particle Image Velocimetry," *Optics and Photonics News*, December, 16-17.

Merzkirch, W., 1987, Flow Visualization, Academic Press, Inc., Orlando, FL.

O'Hern, T. J., Torczynski, J. R., Blanchat, T. K., Chu, T. Y., and Tassin, A. L., 1994, "Shear-Driven Flow in a Square Cavity: A Comparative Study Using LDV, PIV, and Computational Simulations," ASME Symposium on Laser Anemometry: Advances and Applications, T. T. Huang and M. V. Otugen, eds. (SAND93-1707C)

O'Hern, T. J., and Tassin, A. L., 1993, "Examination of a Shear-Driven Cavity Flow Using LDV and PIV," *Bull. Amer. Phys. Soc.*, **37** (10).

O'Hern, T. J., Robey, H. F., Torczynski, J. R., Neal, D. R., and Shagam, R. N., 1991, "Measurements of Passive-Scalar Spectra for Grid Turbulence Using Nonlinear Optical Technique," *Proceedings of the ASME Fluid Measurements and Instrumentation Forum*, 29-33.

O'Hern, T. J., d'Agostino, L., and Acosta, A. J., 1988, "Comparison of Holographic and Coulter Counter Measurements of Cavitation Nuclei in the Ocean," *J. Fluids Engineering* **110** (2), 200-207.

Papamoschou, D., Lin, M., Bunyajitradulya, A, and Robey, H. F., 1993, "Non-Intrusive Technique for Measurement of Power Spectra in Compressible Turbulence" AIAA-93-0633, American Institute of Aeronautics and Astronautics 31st Aerospace Sciences Meeting, January 1993, Reno, NV.

Pepper, D. M., Feinberg, J., and Kukhtarev, N. V., 1990, "The Photorefractive Effect," *Scientific American*, October, 62-74.

Reiman, T. C., and Sabersky, R. H., 1968, "Laminar Flow Over Rectangular Cavities," *Int. J. Heat Mass Transfer*, **11**, 1083-1085.

Robey, H. F., 1990, "Spectrum of a Passive Scalar in the Inertial-Convective Subrange of an Anisotropic Flow," *Physical Review Letters,* **65**, 11, 1360-1363.

Robey, H., Albrecht, G., and Moore, T., 1990, "An Optical Technique for the Direct Measurement of the 2-D Spectral Density of a Passive Scalar in Turbulent Flow," AIAA Paper 90-1667 (AIAA, Washington DC).

Siahmakoun, A., and Harrer, T., 1993, "Dynamic Photorefractive Schlieren System," *Applied Optics*, **32**, 7281.

Torczynski, J. R., Henderson, J. A., O'Hern, T. J., Chu, T. Y., and Blanchat, T. K., 1995, "Three-Dimensional Natural Convection of a Fluid with Temperature-Dependent Viscosity in an Enclosure with Localized Heating," National Heat Transfer Conference, HTD-Vol. 305, 81-87.

Torczynski, J. R., and O'Hern, T. J., 1994, "Numerical Simulations of Flow in a Three-Dimensional Cavity-Channel Geometry," ASME Fluids Engineering Division Summer Meeting, General Fluids Engineering, F. M. White, ed. (SAND93-1439C).

Torczynski, J. R., and O'Hern, T. J., 1993, "Numerical Simulations of Three-Dimensional Cavity Flows," *Bull. Amer. Phys. Soc.*, **37** (10).

Distribution

MS0826	Dept. 9111	T. J. O'Hern (2)
MS0826	Dept. 9111	S. N. Kempka, actg.
MS0834	Dept. 9112	J. R. Torczynski (2)
MS0834	Dept. 9112	A. C. Ratzel
MS0825	Dept. 9115	V. A. Amatucci
MS0328	Dept. 2674	R. N. Shagam
MS0328	Dept. 2674	J. A. Wilder
MS1139	Dept. 6423	T. K. Blanchat
MS1139	Dept. 6423	T. Y. Chu
MS1139	Dept. 6423	K. O. Reil
MS0188	Dept. 4523	LDRD Office
U. Michigan		A. L. Tassin-Leger
MS 9018	Central Technical Files, 8940-2	
MS0899	Technical Library, 4414 (5)	
MS0619	Review and Approval Desk, 12690 for DOE/OSTI	