Naval Surface Warfare Center
Carderock Division
West Bethesda, MD 20817-5700

NSWCCD-80-TR–2020/006 March 2020

Naval Architecture and Engineering Department

Technical Report

A METHOD FOR TURBULENT PRESSURE FIELD MEASUREMENT

by

Daniel R. Cadel

NSWCCD

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**			
1. REPORT DATE *(DD-MM-YYYY)* 18-Mar-2020	**2. REPORT TYPE** Final		**3. DATES COVERED *(From - To)*** 01 Oct 2017 – 30 Sep 2018
4. TITLE AND SUBTITLE A Method for Turbulent Pressure Field Measurement			**5a. CONTRACT NUMBER**
			5b. GRANT NUMBER
			5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Daniel R. Cadel			**5d. PROJECT NUMBER**
			5e. TASK NUMBER
			5f. WORK UNIT NUMBER 00T TDIC: 3000001235450010
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES) NAVAL SURFACE WARFARE CENTER CARDEROCK DIVISION (CODE 854) 9500 MACARTHUR BOULEVARD WEST BETHESDA, MD 20817-5700			**8. PERFORMING ORGANIZATION REPORT NUMBER** NSWCCD-80-TR–2020/006
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) NAVAL SURFACE WARFARE CENTER CARDEROCK DIVISION (CODE 854) 9500 MACARTHUR BOULEVARD WEST BETHESDA, MD 20817-5700			**10. SPONSOR/MONITOR'S ACRONYM(S)** NSWCCD
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION / AVAILABILITY STATEMENT Distribution Statement A: Approved for public release; distribution unlimited.			
13. SUPPLEMENTARY NOTES			
14. ABSTRACT A procedure for computing the turbulent pressure field from time-resolved particle image velocimetry (PIV) velocity field data is adapted and implemented. The calculation of pressure is based on the Lagrangian acceleration term in the fluctuating momentum equation coupled with an omnidirectional spatial integration, originally developed by Liu and Katz (2006 Exp. Fluids 41:227-240). Experimental measurements of fluctuating pressure is important in the validation of numerical turbulence models and is also a fundamental source of observed aero/hydro-acoustic noise. The code suite described in this paper yields the fluctuating pressure with Lagrangian fluid acceleration and spatial gradients of velocity. Monte Carlo simulations are performed with synthetic velocity fields to study the uncertainty in the calculations, and sources of uncertainty and data requirements are discussed.			
15. SUBJECT TERMS Particle Image Velocimetry, Pressure, Turbulence			

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Daniel R. Cadel
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED	Same as Report	30	**19b. TELEPHONE NUMBER *(include area code)*** 301-227-4859

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

CONTENTS

Page

CONTENTS iii
FIGURES iv
ADMINISTRATIVE INFORMATION v
ACKNOWLEGEMENTS v
SUMMARY 1
INTRODUCTION 1
Motivation 1
Point-Wise Measurements 2
Surface Pressure Measurements 3
Particle Image Velocimetry 3
PRESSURE CALCULATION VIA PARTICLE IMAGE VELOCIMETRY 4
Pressure from Eulerian Methods 4
Pressure from Lagrangian Methods 4
IMPLEMENTATION 5
Acceleration Processing 6
Data importing 6
Acceleration Processing 6
Spatial Gradient Processing 7
How to Run 7
Output Format 10
Pressure Field Processing 10
Pressure Processing 10
Cropping 10
How to Run 11
Output Format 14
Plotting Tool 14
How to Run 16

UNCERTAINTY ANALYSIS 17
Sources of Uncertainty....17
PIV Uncertainty 17
Numerical Error 18
Temporal and Spatial Resolution Limits 18
Three Dimensionality....18
Pseudo-Tracing 18
Viscous Term....19
Cropping Error....19
Monte Carlo Simulations 19
4.2.1 Noise variance....19
Masking....23
CONCLUSIONS AND OUTLOOK....26
REFERENCES 28

FIGURES

Page

Figure 1. Acceleration Solver GUI Window as Seen upon First Opening 8
Figure 2. Acceleration Solver GUI Window as Seen after a Batch List Has Been Created....9
Figure 3. Pressure Solver GUI Window as Seen upon First Opening 11
Figure 4. Pressure Solver GUI Window as Seen after Batch List Has Been Created and Processing Is in Progress....12
Figure 5. Plotting Tool GUI Window as Seen upon First Opening....15
Figure 6. Plotting Tool GUI as Seen after Data Has Been Loaded and Displayed 16
Figure 7. Contours of |DU/Dt| / (ω2Rmax) from Monte Carlo Simulations at Several SNR Levels 21
Figure 8. Contours of ((p-pcenter)) / (12ρω2rmax2) from Monte Carlo Simulations at Several SNR Levels 22
Figure 9. Resulting Signal Variance of Acceleration (Left Frame) and Pressure (Right Frame) as a Function of the Velocity Field SNR....23
Figure 10. Pressure Field Results from “Framed Case” Mask as Compared with the Full-Field Processing at 40 dB 25
Figure 11. Pressure Field Results from “Corner Case” Mask as Compared with the Full-Field Processing at 40 dB 26

ADMINISTRATIVE INFORMATION

The work described in this report was performed by the Surface Ship Hydromechanics Division (Code 85) of the Naval Architecture and Engineering Department at the Naval Surface Warfare Center, Carderock Division (NSWCCD). The work was funded by the Office of the Chief Technology Officer, Code 00T, as part of the 2018 Technical Director's Innovation Challenge.

ACKNOWLEGEMENTS

This work was funded under the Technical Director's Innovation Challenge at NSWC Carderock Division, program managers Dr. Judy Conley and Garth Jensen. The author is also grateful to Dr. Xiaofeng Liu at San Diego State University for supplying the codes from which the code suite in this work was developed, as well as for his feedback during the development.

THIS PAGE INTENTIONALLY LEFT BLANK

SUMMARY

A procedure for computing the turbulent pressure field from time-resolved particle image velocimetry (PIV) velocity-field data are adapted and implemented. The calculation of pressure is based on the Lagrangian acceleration term in the fluctuating momentum equation coupled with an omnidirectional spatial integration, originally developed by Liu and Katz (2006 Exp. Fluids 41:227-240). Experimental measurements of fluctuating pressure is important in the validation of numerical turbulence models and is also a fundamental source of observed aero/hydro-acoustic noise. The code suite described in this paper yields the fluctuating pressure with Lagrangian fluid acceleration and spatial gradients of velocity. Monte Carlo simulations are performed with synthetic velocity fields to study the uncertainty in the calculations, and sources of uncertainty and data requirements are discussed. A routine is introduced to crop regions of a vector field prior to pressure calculation. This method increases the spatial extent of the valid data although with a higher overall uncertainty in the region around the cropped area.

INTRODUCTION

Motivation

Turbulence modelling is a critical area of development in many fluid mechanic applications owing to its great potential for parametric studies and accelerated design cycles. With the exception of Direct Numerical Simulations (DNS), which are as yet limited to low Reynolds number applications, and Large Eddy Simulation (LES), which simulates large scales of motion while filtering smaller scales into a subgrid-scale stress term, all Reynolds-averaged Navier Stokes (RANS) numerical turbulence models involve simplifications or empirical closure formulations. Experimental measurements are needed both to validate numerical models against benchmark experimental data (Stern et al. 1999) and to motivate the development of new models as new phenomena are observed and understood.

Consider the Reynolds stress transport equation:

$$\frac{D}{Dt}\overline{u_i u_j} = -\frac{\partial}{\partial x_k}\overline{u_i u_j u_k} + \mathcal{D}_{\nu_{ij}} + \mathcal{P}_{ij} + \Pi_{ij} - \varepsilon_{ij} \tag{1}$$

The last four terms on the right hand side are known as the divergence term Equation (2), production tensor Equation (3), velocity-pressure-gradient tensor Equation (4), and dissipation tensor Equation (5), and are defined as follows with tensor notation:

$$\mathcal{D}_{\nu_{ij}} \equiv \nu\nabla^2\overline{u_i u_j} \tag{2}$$

$$\mathcal{P}_{ij} \equiv -\overline{u_i u_k}\frac{\partial \overline{U}_j}{\partial x_k} - \overline{u_j u_k}\frac{\partial \overline{U}_i}{\partial x_k} \tag{3}$$

$$\Pi_{ij} \equiv -\frac{1}{\rho}\overline{\left(u_i\frac{\partial p'}{\partial x_j} + u_j\frac{\partial p'}{\partial x_i}\right)} \tag{4}$$

$$\varepsilon_{ij} \equiv 2\nu\overline{\frac{\partial u_i}{\partial x_k}\frac{\partial u_j}{\partial x_k}} \tag{5}$$

The velocity-pressure-gradient tensor can further be decomposed as (Pope 2000)

$$\Pi_{ij} \equiv \mathcal{R}_{ij} - \frac{\partial T_{kij}^{p}}{\partial x_k} \tag{6}$$

The first term, R_{ij}, is known as the pressure-rate-of-strain tensor, and is defined as

$$\mathcal{R}_{ij} \equiv \overline{\frac{p'}{\rho}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)} \tag{7}$$

In conditions of homogenous turbulence, i.e. the turbulence is independent of position within the flow, the second term in Equation (6) reduces to zero. Under these conditions, the velocity-pressure-gradient tensor can completely be defined with the fluctuating pressure and spatial velocity gradients. Modelling of the pressure-rate-of-strain tensor is the key challenge in Reynolds-stress based RANS models. These formulations are higher fidelity than the more commonly used turbulent-viscosity models, which necessarily include more simplifications.

Pressure fluctuations in a fluid flow are an important metric in describing its turbulent behavior, and are directly responsible for generating observed aero/hydro-acoustic sound. Spatially and temporally resolved planar maps of fluctuating pressure are a fundamental quantity in describing fluid motion, and must be used instead of velocity in formulating the underlying mathematical models in Computational Fluid Dynamics (CFD) (Spalart 2015).

Point-Wise Measurements

Point-wise pressure measurements are widespread and robust. Common instrumentation of this type are generally based on the principle of Pitot-static probes. One advantage of these systems is that by locally measuring both the static and total pressures, the dynamic pressure can be computed as well. To account for flow angularity, shrouded Kiel probes can account for yaw of the probe (Kiel 1935). 5- and 7-hole probes can also measure both pressure and flow angle based on the differential pressures measured at each of their uniquely positioned taps around a typically semi-spherical tip (Dominy and Hodson 1992, Sumner 2002). Surface pressure taps in solid surfaces yield static pressure readings, and when embedded along the chord of a lifting foil, can give estimates of the sectional lift force. More complex arrangements of pressure probes are also common, such as the "wake rake" configuration of tens or hundreds of total pressure tubes closely arranged in a transverse line behind a test article in a wind tunnel. The resulting profile can be used to calculate the wake deficit and, when coupled with a measured or assumed upstream profile, the drag can be computed (Timmer 2008).

Point-wise techniques offer many advantages, including high measurement rate and dynamic response, low cost hardware, real-time processing, and low uncertainties. However, the limitations are twofold: only a single point in space can be measured at a time with these probes, and the probe itself is intrusive to the flow and affects the pressure field. The technology is mature such that a single probe or pressure tap will have minimal impact on the pressure it measures; however, the probe will alter the pressure stream in its own wake. Arrays of closely packed probes may also interfere with one another, limiting the allowable spacing for simultaneous measurements.

Surface Pressure Measurements

Pressure sensitive paint (PSP) has been shown to offer spatially resolved surface pressures in applications where surface pressure taps are not sufficient or cannot be installed (McLachlan et al. 1993). However, this technology is limited to solid surfaces in air. PSP is most commonly applied in high-speed airflow applications, such as transonic wind tunnels (Sellers and Brill 1994, Sellers et al. 2016). The PSP principle fundamentally measures the concentration of dissolved oxygen, which varies with pressure in a compressible fluid. Studies on PSP uncertainty and applicability can be found in Liu et al. (2001) and Nelson (2018).

Particle Image Velocimetry

Particle Image Velocimetry (PIV) is a non-intrusive optical measurement technique, first introduced by Adrian (1984). The technique generally involves laser light to illuminate a 2D plane or 3D volume of a seeded fluid flow (Prasad 2000). The small seed particles, when of sufficiently small Stokes number, are neutrally buoyant and follow the instantaneous motion of the flow (Adrian and Westerweel 2011). One or more cameras image the flow in successive frames synchronized with the pulsing of the laser light. The resulting images are spatially cross-correlated to yield the motion of small clusters of particles within overlapping subdivisions of the field of view, known as "interrogation windows." Therefore, the spatial resolution is defined by the width of an interrogation window, and not by the camera pixel spacing or the ultimate spacing of the data points.

Pulses of laser light can be emitted in paired bursts, or as a high rate time series of evenly spaced pulses; the choice depends on the capabilities of the hardware. The former results in two images with a short time differential (dt) between frames, from which a single instantaneous velocity field is found. While the data are instantaneous velocities, the data cannot be considered time resolved in this configuration since sequential velocity fields are temporally spaced far above the integral time scale of the flow. However, these frames may be considered independent measurements, if the integral time scale is sufficiently small. High-rate laser systems offer the benefit of time series data by correlating successive camera images to produce uniformly sampled velocity fields. However, such lasers systems contain less energy per pulse than double-pulsed lasers and may impose a lower limit on the inter-image time spacing.

Spatial calibration of the camera frame is done with a precision-machined reference plate placed in the field of view. A commonly employed algorithm is that of Tsai (1987). For multi-camera PIV, an iterative process introduced by Wieneke (2005) provides for correction of misalignment between cameras.

Several variants of PIV commonly provide measurements of up to three components of velocity in four dimensions (space and time). Stereoscopic PIV employs two cameras imaging the same 2D field of view but offset at slight yaw angles to each other. The motion of imaged particles is observed differently from the two cameras, allowing for the calculation of the third, in-plane, component of velocity (Willert 1997). Volumetric measurements can be achieved with either holographic PIV (Sheng et al. 2006) or tomographic PIV (Elsinga et al. 2006). High rate measurements about 1 MHz have also been demonstrated by pulse-burst laser systems (Thurow et al. 2013).

PRESSURE CALCULATION VIA PARTICLE IMAGE VELOCIMETRY

Methods for calculating pressure from PIV data can be divided into two categories, those based on the Lagrangian acceleration of the fluid and those based on the Eulerian evolution of the flow. Each approach offers benefits with respect to the minimum data requirements and instrumentation hardware. A review of both methods is given by van Oudheusden (2013). The Navier Stokes equations are the starting point for the derivations of both approaches, given as:

$$\nabla p = -\rho \frac{D\vec{U}}{Dt} - \mu \nabla^2 \vec{U} \tag{8}$$

Pressure from Eulerian Methods

The first use of a Reynolds-averaged approach to solving for the pressure field from PIV data was reported by Gurka et al. (1999). Applying the divergence operator to Equation (8) gives a Poisson equation for the pressure in terms of velocity gradients (assuming incompressible flow).

$$\nabla^2 p = -\rho \nabla \cdot \left(\vec{U} \cdot \nabla\right)\vec{U} \tag{9}$$

This equation includes only spatial derivatives of the velocity and pressure terms. Many researchers have applied this approach, for example Fujisawa et al. (2005), de Kat and Ganipathisubramani (2012), and de Kat and van Oudheusden (2012), among others. The solution to the Poisson equation requires strictly defined boundary conditions along all edges of the region of data, which may require knowledge of the absolute pressure surrounding the measured region and temporal information about the flow.

Pressure from Lagrangian Methods

The Lagrangian approach to PIV pressure calculations involves the spatial integration of the material acceleration throughout the field of view. Acceleration data from PIV were first reported by Jakobsen et al. (1997) from both Lagrangian and Eulerian formulations.

The Navier Stokes equation is solved for the pressure gradient in terms of a material acceleration and viscous term in Equation (8), and the vector notation for the velocity is $\vec{U}$. Liu and Katz (2006) showed that the viscous term $\mu\nabla^2\vec{U}$ could be neglected, when the flow is of sufficiently high Reynolds number and far from boundaries. The remaining term on the right hand side is the Lagrangian acceleration term. The two challenges in this formulation to solve for the pressure field are then the measurement of acceleration and the spatial integration of the pressure gradient.

Material acceleration is computed from particle images with methods of either particle image velocimetry (PIV) or the related principle of particle tracking velocimetry (PTV). By the more traditional PIV approach, “pseudo-tracing” schemes account for motion of the small groupings of particles from one temporal frame to the next by interpolating the velocity at projected positions in the direction of the instantaneous flow at subsequent time steps (Jensen et al. 2003). PTV offers an advantage over PIV in this regard, as the true Lagrangian acceleration can be measured, since the velocity and position of individual particles are uniquely tracked (Novara and Scarano 2013). However, PTV can be a more challenging measurement than PIV, in large part due to seeding requirements.

For a scalar field such as pressure, the integrated value is, ideally, path-independent. For real experimental data, various sources of measurement noise and uncertainty means that the integral path will have an effect on the resulting field. An “omnidirectional” algorithm for integrating the pressure gradient was developed by Liu and Katz (2006) to address this issue. A series of virtual origins outside the field of view of the data are set, and rays emanating from these points are defined along which the pressure gradient field is integrated. The pressure is integrated. The resulting value reported is the weighted average of the passes, and values along the boundary are updated in subsequent passes. The location of the virtual origin is swept around the region of data such that integration paths approach the region from all possible angles.

An improvement to the integration algorithm was presented by Liu at al. (2016) that replaces the rays emanating from virtual origins with parallel rays, effectively placing the virtual origin at infinity. This creates an equal number of integration paths passing through each point in the flow field, and therefore eliminates the weighting that must be done with the Liu and Katz (2006) method.

IMPLEMENTATION

The pressure solver presented in this paper is an adaptation and generalization of that in Liu et al. (2016). The ability to mask out portions of the frame has been newly implemented in the Carderock adaptation introduced here. A Matlab code suite built on this Carderock adaptation of the pressure solver has been developed with corresponding Graphical User Interfaces (GUIs) to process the Lagrangian acceleration, spatial derivatives, and pressure, as well as a tool for interactively viewing the data.

In this section, details of the algorithm are presented with the general architecture of the code suite. Emphasis is placed on aspects that differ from the algorithm as implemented in Liu et al. (2016).

Acceleration Processing

Within the acceleration processing routine, velocities are read in and accelerations are computed in a pseudo-tracing routine, and spatial gradients of velocity are also computed for completeness.

Data importing

The data import functionality is integrated within the acceleration processor. The tool suite is designed for use with LaVision DaVis PIV processing software, and can take data in the native DaVis file formats (e.g. .vc7) or as an exported .dat file from the DaVis software. For the native file formats to work, the “readimx” Matlab tool suite from LaVision must be installed.

Exported .dat files from DaVis should have either four columns (x, y, u, v), or six columns (x, y, z, u, v, w) and three header rows, and must include the data grid size in the third line of the header. A sample file header is as follows:

TITLE = "B00004.dat"

VARIABLES = "x", "y", "Vx", "Vy"

ZONE T="Frame 0", I=101, J=101, F=POINT

Acceleration Processing

Acceleration processing follows a pseudo-tracing approach. At minimum, five temporally sequential image frames are required to compute the Lagrangian acceleration from a central difference scheme. To start, velocities must be recomputed as central differences, rather than the first order forward differences typically reported by LaVision DaVis or similar PIV processors. The second-order central difference (superscript $\mathcal{O}(2)$) of positions X can be reformulated in terms of first-order forward-difference velocities U (superscript $\mathcal{O}(1)$):

$$\mathrm{U}_i^{\mathcal{O}(2)} = \frac{X_{i+1} - X_{i-1}}{2\Delta t} = \frac{1}{2}\left(\frac{X_{i+1} - X_i}{\Delta t} + \frac{X_i - X_{i-1}}{\Delta t}\right) = \frac{1}{2}\left(U_i^{\mathcal{O}(1)} + U_{i-1}^{\mathcal{O}(1)}\right) \tag{10}$$

In the proceeding analysis in this section, only these second order accurate velocities are considered, and the $\mathcal{O}(2)$ subscript is dropped. At each time step i, the convected position and projected velocity must then be determined both forward and backward in time. The hat symbol $\hat{X}$ denotes that a quantity is the convected value. First, the convection forward in time is:

$$\hat{X}_{i+1} = X_i + U_i \Delta t \tag{11}$$

This convected position is then used to interpolate the velocity at time $i + 1$ from the two dimensional interpolation function in Matlab with a cubic spline.

$$\hat{U}_{i+1} = U_{i+1}|_{\hat{X}_{i+1}} \tag{12}$$

The interpolated velocity at time $i - 1$ can similarly be computed stepping backward in time:

$$\hat{X}_{i-1} = X_i - U_i \Delta t \tag{13}$$

$$\widehat{U}_{i-1} = U_{i-1}|_{\hat{X}_{i-1}} \tag{14}$$

Finally the acceleration is calculated as the central difference of these interpolated velocity vectors.

$$\frac{DU_i}{Dt} = \frac{\widehat{U}_{i+1} - \widehat{U}_{i-1}}{2\Delta t} \tag{15}$$

The scheme is analogous for flow in both in-plane dimensions, U and V. Whereas, the first-order velocity reported by the data acquisition software will have a temporal resolution of Δt, the acceleration has a temporal resolution of $4\Delta t$, since five consecutive image frames are needed to compute a single acceleration field.

Spatial Gradient Processing

Spatial gradients are necessary for computing the pressure-rate-of-strain tensor as discussed in the Motivation section. In-plane first and second derivatives of velocity are computed by a fourth-order accurate finite-difference formulation. Although beyond the scope of the present work, these spatial derivatives can be used in the Eulerian formulation of the pressure field solution, i.e. in the Poisson equation.

How to Run

The acceleration processing code is run by building a batch list interactively within the GUI. Figure 1 and Figure 2 are examples of the GUI display before and after setting parameters and building a batch list.

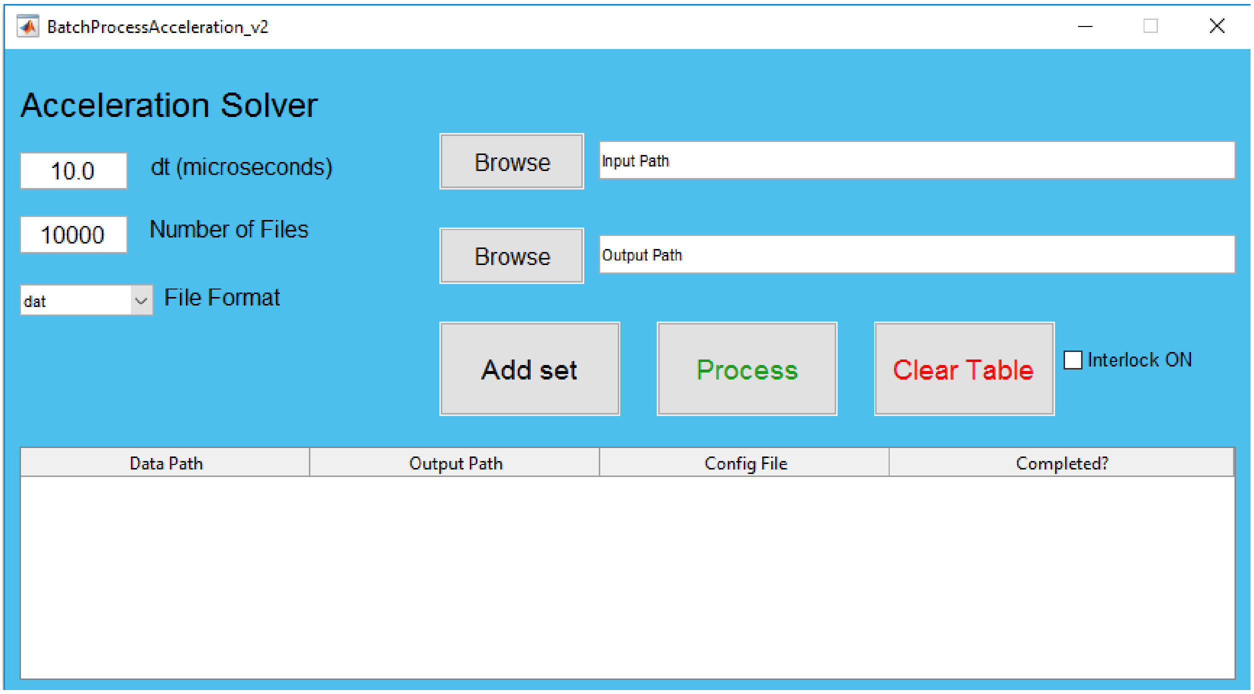

Figure 1. Acceleration Solver GUI Window as Seen upon First Opening

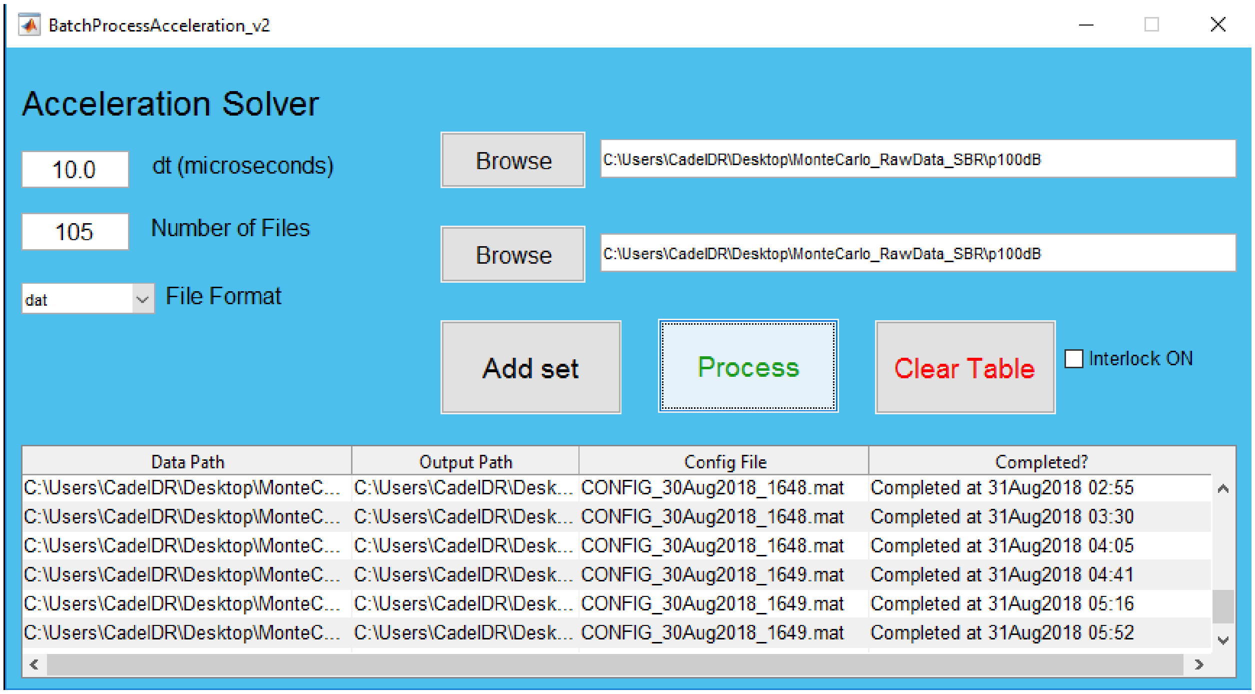

Figure 2. Acceleration Solver GUI Window as Seen after a Batch List Has Been Created

Instructions to run the processor are listed sequentially below.

- Open Matlab and open code named “BatchProcessAcceleration_v2.m”
- Click the “Run” button in the top toolbar
- A warning may pop up indicating that the script “is not found within the current folder or on the MATLAB path.” If so, click “Change Folder.”
- A GUI with a blue background and the words “Acceleration Solver” will open. Inputs are defined as follow:
    - “dt (microseconds)”: time between successive PIV images in time series
    - “Number of Files”: Number of total velocity field files from the input path to process
    - “File Format”: Drop-down menu with options of “dat” or “LaVision.” Use “dat” for exported .dat files, and use “LaVision” for native DaVis formats such as .vc7. The LaVision “readimx” library must be installed in order to import native LaVision formats
    - “Input Path (Browse)”: Click browse to open file browser and select directory containing velocity field files
    - “Output Path (Browse)”: Click browse to open file browser and select directory where acceleration outputs are to be saved
    - “Add set”: Click to add entry into batch list with current values of inputs
    - “Process”: Click to begin batch processing of all files in batch list
    - “Clear Table”: Click to clear all entries from Batch List. Only works if “Interlock” is turned OFF

    - “Interlock”: Radio button that freezes control of “Clear Table” button. Must be checked for “Clear Table” button to work
- Input appropriate values for the input items listed above and click “Add Set.” A new line entry will appear in the batch list
- Update input entries for next set to add to batch list and click “Add Set” again. Repeat as necessary to build batch list
- Click “Process” to begin data processing. For each line item in the list, a wait bar will pop up indicating the time remaining for that set. At the completion of that line item, the final column of the batch list will change to the date and time that processing finished
- In the batch processing, the script will only operate on entries that have not been marked in the table as completed. If processing crashes or is intentionally stopped midway through, processing can be restarted at the current set by clicking again the “Process” button (may be colored pink if processing stopped midway through)
- To clear all entries from the batch list, click the “Interlock” radio button so that it is checked; the label will change to read “Interlock OFF.” Then click the “Clear Table” button to remove all entries. Click to uncheck the interlock button to again disable the “Clear Table” button

Output Format

Two output files are generated from the acceleration solver, both in Matlab .mat format. The first is a configuration file, titled with “CONFIG” and the date and time of its creation. This file contains the values of settings selected in the GUI interface during creation of each run. The second file is titled as “acceleration.mat” and contains a single Data structure titled “Data.” Within this structure are fields for all outputs.

Pressure Field Processing

Within the pressure processing routine, accelerations are read in and pressures are computed from an omnidirectional spatial integration approach.

Pressure Processing

Pressure fields are computed by the method of Liu et al. (2016). The algorithm itself was provided by Prof. Liu through partnership with NSWCCD and ONR, when Prof. Liu collaborated with Carderock through the ONR Summer Faculty Program. The primary modifications to the suite of codes involved accounting for file formats of the inputs and outputs, as the Lagrangian acceleration solver was completely rewritten, as discussed above. A cropping routine has also been incorporated.

Cropping

The cropping routine is newly implemented here building on the algorithm of Liu et al. (2016). The algorithm works by spatial integrating and updating node points as the parallel rays rotate through the domain. At each spatial point within the field, the local material acceleration is integrated and the value updated.

Regions to be masked are those where data are not valid for any reason. For PIV data, this can be due to low seeding density, laser flare, too few valid vectors, or the location physically not being within the flow (such as a solid boundary). With the latter, the algorithm implemented here does not account for the viscous term as discussed above, so care must be taken to crop out the observed viscous boundary layer region when regions near solid surfaces are considered. Viscous forces are non-negligible near solid-fluid interfaces, and even though the spatial gradients are calculated in the preceding acceleration code, the spatial resolution in PIV is typically not fine enough to yield accurate estimates of the viscous term except with advanced hardware and in specialized configurations.

When a region is masked in the image, the regions outside of the valid area are defined as having zero material acceleration. As the integration paths pass through these regions, no change to the value of the integral occurs. For paths that originate in a masked region, the value of the boundary condition is then effectively imposed on the closest non-masked point, while still allowing the value at that location to iterate as it is passed through by subsequent rays. Rays that pass through a masked region at the end of their path will retain the constant value of the last non-masked point passed through. Masked regions that are contained within the middle of an array (such as in the center of a frame) will retain the value of the integral across the masked region to continue once the ray reaches the far end of the masked region. Because of this approach, the pressure is still calculated within these masked regions and is not intended to be accurate. The pressure value at these locations, although allowed to iterate during processing, are removed, before the data are output.

How to Run

The pressure solver code is run by building a batch list interactively within the GUI. Figure 3 and Figure 4 are examples of the GUI display before and after setting parameters and building a batch list.

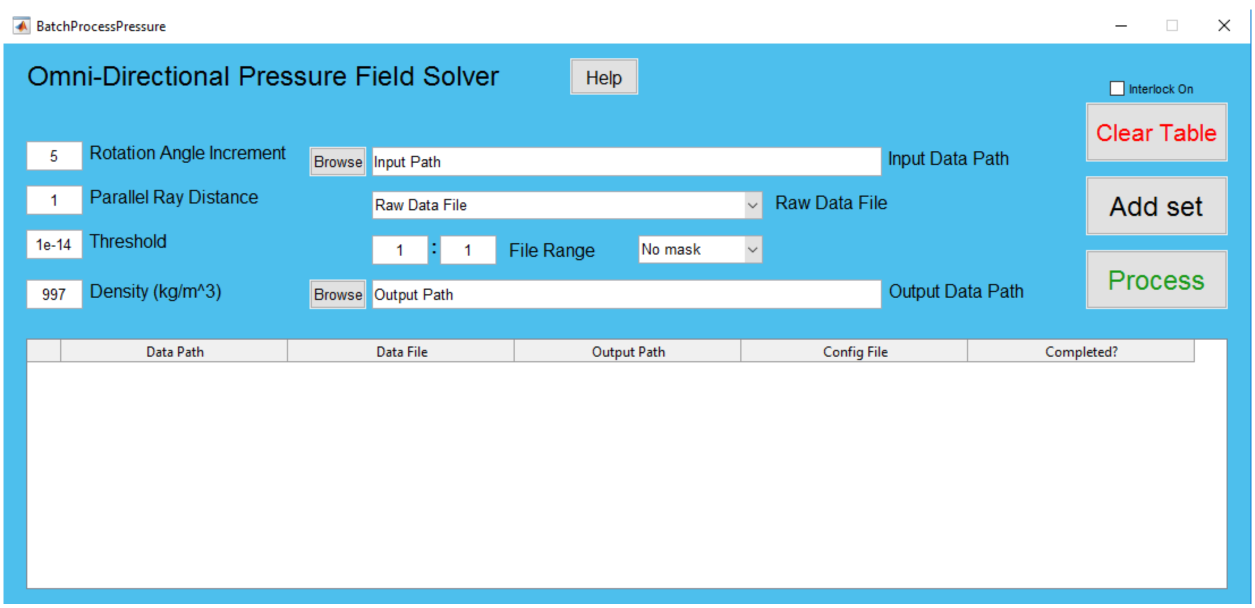

Figure 3. Pressure Solver GUI Window as Seen upon First Opening

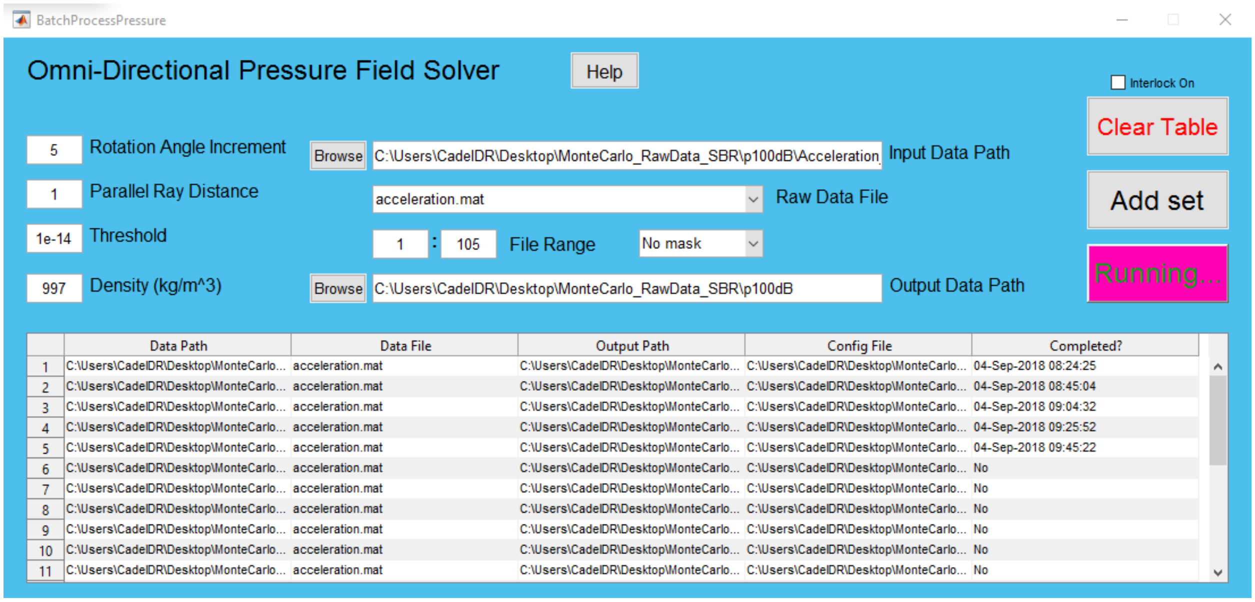

Figure 4. Pressure Solver GUI Window as Seen after Batch List Has Been Created and Processing Is in Progress

Instructions to run the processor are listed sequentially below.

- Open Matlab and open code named “BatchProcessPressure.m”
- Click the “Run” button in the top toolbar
- A warning may pop up indicating that the script “is not found within the current folder or on the MATLAB path.” If so, click “Change Folder.”
- A GUI with a blue background and the words “Omni-Directional Pressure Field Solver” will open. Inputs are defined as follow:
    - “Help”: opens pop-up window describing inputs
    - “Rotation Angle Increment”: Defines the increment (in degrees) through which the omnidirectional rays iterate around the domain. A smaller number will yield more precise results but increase processing time
    - “Parallel Ray Distance”: Defines the spacing between adjacent parallel rays (in pixel units). A smaller number will yield more precise results but increase processing time
    - “Threshold”: Convergence criteria for iterative solver. A smaller number will yield more precise results but increase processing time
    - “Density (kg/m^3)”: Fluid density in SI units
    - “Input Data Path (Browse)”: Select data directory that contains processed acceleration file. The path name must be selected by the “Browse” button, and not by typing in the path directly
    - “Raw Data File”: Dropdown menu listing all .mat files included in selected Input Data Path
    - “File Range”: Left and right boxes indicate first and last index within the individual input path acceleration file on which to operate the pressure processor

  - "No mask": dropdown containing options for masking out data, "No mask," "Define mask," and "Load mask." "No mask" will process pressure in the entire domain, "Define mask" will open the first frame of the acceleration field for the user to interactively draw the mask by clicking points as directed and then save the mask data to a .mat file in the chosen data path, and "Load mask" will open a file browser for the user to select an already drawn mask
  - "Output Data Path (Browse)": Click browse to open file browser and select directory where pressure output is to be saved
  - "Interlock": Radio button that freezes control of "Clear Table" button. Must be checked for "Clear Table" button to work
  - "Clear Table": Click to clear all entries from Batch List. Only works if "Interlock" is turned OFF
  - "Add set": Click to add entry into batch list with current values of inputs
  - "Process": Click to begin batch processing of all files in batch list
- Input appropriate values for the input items listed above and click "Add Set." A new line entry will appear in the batch list
- To define a mask for the acceleration field, the "Define mask" option must be selected from the drop-down menu. Defining a mask in third party software during velocity vector processing is independent from the masking here, a mask would need to be re-drawn over the previously masked area. Upon adding the set to the batch list, the data are loaded into memory (which may take up to several minutes for large data sets) and the Du/Dt field for the first time instance is plotted in a new window. The acceleration field is computed for the fourth time instance forward, due to the nature of the central difference operator in the acceleration solver. Therefore, especially for data sets in which cropping is desired, the file range should start at 4. In the window, interactive crosshairs are automatically activated with on-screen instructions to click within the field to define the valid range of data. After clicking each point, press the mouse again to reactivate the crosshairs to continue defining the cropping range. Press any key (i.e. not click the mouse) to finish selecting the region and close the polygon. The valid region for a mask is everything inside of the polygon, while everything outside of the polygon is considered invalid. The mask information is saved to the defined output data path in a file called "mask.mat" which is then called once the "Process" button is pressed.
- If the "Load mask" option is selected from the dropdown, the user will be prompted to select an existing mask file when the set is added to the batch list. A file browser window is opened, and the user is prompted to select a .mat file as previously output from the "Define mask" option. This option is intended for use when several sets of data under the same optical settings are acquired, and consistency is desired amongst output sets.
- Update input entries for next set to add to batch list and click "Add Set" again. Repeat as necessary to build batch list
- Click "Process" to begin data processing. For each line item in the list, a wait bar will pop up indicating the time remaining for that set. At the completion of that line item, the final column of the batch list will change to the date and time that processing finished

- In the batch processing, the script will only operate on entries that have not been marked in the table as completed. If processing crashes or is intentionally stopped midway through, processing can be restarted at the current set by clicking again the “Process” button (may be colored pink if processing stopped midway through)
- To clear all entries from the batch list, click the “Interlock” radio button so that it is checked; the label will change to read “Interlock OFF.” Then click the “Clear Table” button to remove all entries. Click to uncheck the interlock button to again disable the “Clear Table” button

Output Format

Two output files are produced from the pressure solver, both in Matlab .mat format. The first is a configuration file, titled as “pressure_CONFIG.mat”. This file contains the values of settings selected in the GUI interface during creation of each run. The second file is titled as “pressure.mat” and contains a single variable named “pressure” containing the pressure field data for all time instances. The acceleration output file must be used with this since it contains the x- and y-position data. These data files are intentionally excluded from the “pressure.mat” output so as to eliminate data storage redundancies.

Plotting Tool

A tool for interactively visualizing the results of the processing codes is included in the suite as a Matlab GUI called “PlottingTool.m.” This tool allows for visualizing all input and output fields as well as outputting image files. Figure 5 and Figure 6 are examples of the plotting tool display before and after loading and plotting data.

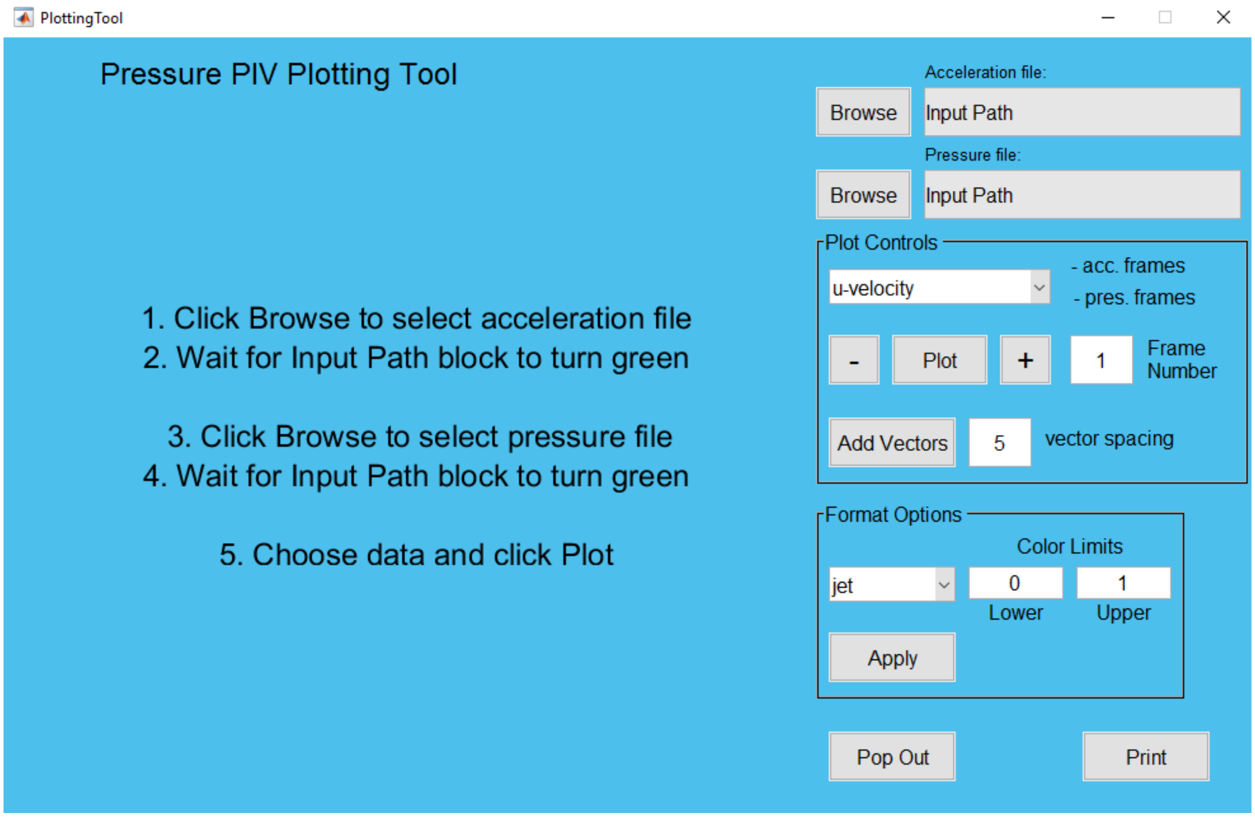

Figure 5. Plotting Tool GUI Window as Seen upon First Opening

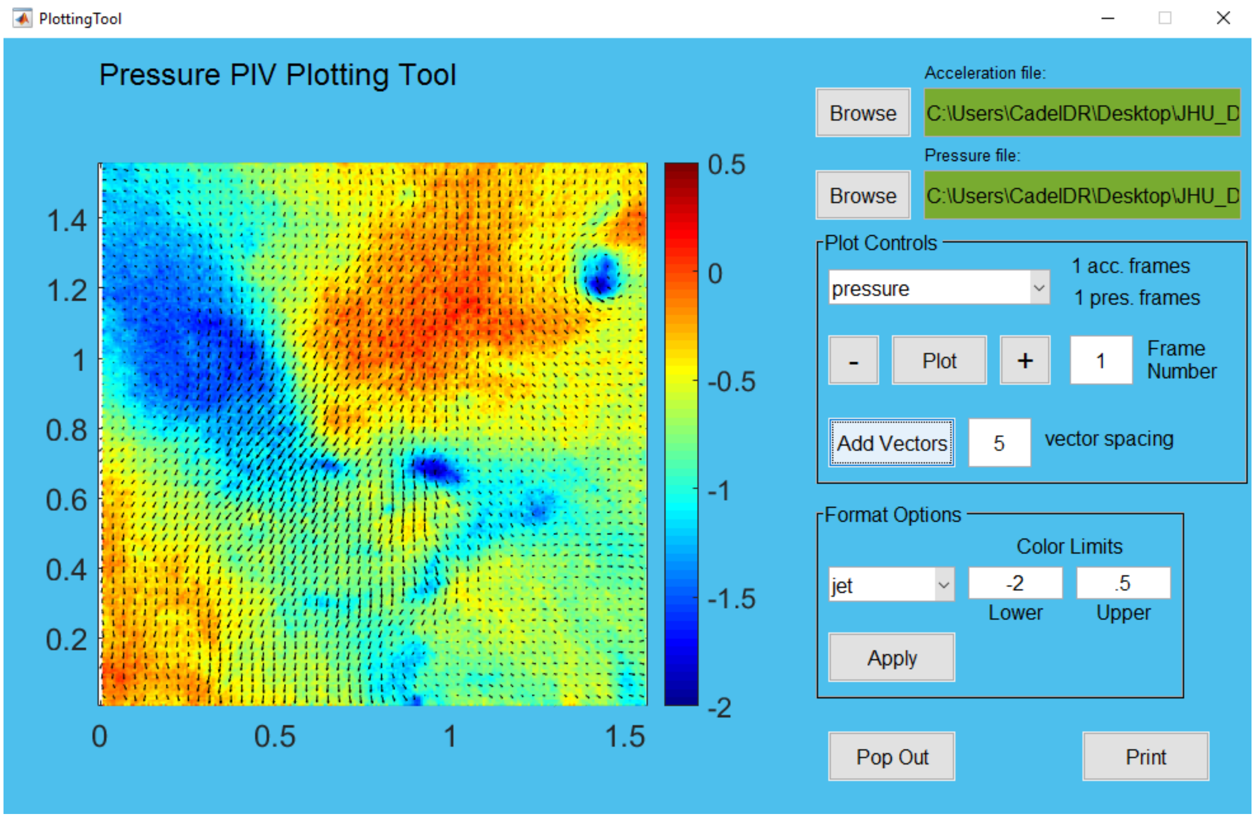

Figure 6. Plotting Tool GUI as Seen after Data Has Been Loaded and Displayed

How to Run

The script is run entirely interactively. Instructions to run the processor are listed sequentially below.

- Open Matlab and open code named “PlottingTool.m”
- Click the “Run” button in the top toolbar
- A warning may pop up indicating that the script “is not found within the current folder or on the MATLAB path.” If so, click “Change Folder”
- A GUI with a blue background and the words “Pressure PIV Plotting Tool” will open with on-screen instructions
- Load in data from the acceleration and pressure solvers. The script will run if only one of the two files is loaded, although for normal operation it is recommended that both files be imported
    - Click the “Browse” button next to the “Input Path” field under the “Acceleration file:” heading. A file browser will open, select the acceleration file to plot. The block containing the input path will turn red upon clicking the “Browse” button, and will then turn green once the file has been successfully loaded. Wait for the green color before pressing any more buttons
    - Click the “Browse” button next to the “Input Path” field under the “Pressure file:” heading. A file browser will open, select the pressure file to plot. The block containing the input path will turn red upon clicking the “Browse” button,

and will then turn green once the file has been successfully loaded. Wait for the green color before pressing any more buttons

- In the "Plot Controls" frame, the number of acceleration and pressure frames will update upon loading each file
- The drop-down menu within the "Plot Controls" frame controls which data field is to be plotted. Select the data to plot, and enter the frame number to display (For higher order fields, the first few and last few frames are not computed.)
- Click the "Plot" button to display the data. Click the "+" and "-" buttons to increment the display in time
- The "Add Vectors" button will overlay a quiver plot on the data. The "vector spacing" field refers to the number of pixels between subsequent vector arrows. Vectors will automatically increment as the "+" and "-" buttons are pressed as well
- Display settings can be changed within the "Format Options" frame. The "Color Limit" fields refer to the maximum and minimum value of the color scaling, and the drop-down menu contains several Matlab color schemes. Click "Apply" to update the plot with new display settings
- Upon clicking the "Plot" button, all settings are reset to the default, and vectors will not appear on the plot
- The 'Pop Out" button creates a figure display in a new window with the same data and plot settings as are currently shown within the GUI. Whereas axis and title labels do not appear in the GUI, they are included in the pop out display. This allows the user to use the suite of Matlab built-in interactive tools such as zoom and the data cursor
- The "Print" button has the same operation as the "Pop Out" button, except the new display window is generated only briefly and the current display is saved to an image file in the current Matlab working directory. A folder is generated and named by the acceleration output file, i.e. "acceleration.mat_plots.png." Within that folder, the data are saved with the output file named and the name of the data field currently being displayed, i.e. "acceleration.matuvel.png"

ASSUMPTIONS AND UNCERTAINTY ANALYSIS

Sources of Uncertainty

Uncertainty in the Lagrangian acceleration, spatial gradient, and pressure fields arise from several sources. Fundamentally, uncertainty from the velocity field itself is propagated through. Uncertainty also comes from numerical error in the computations, and approximations made in the analysis. Some significant sources are discussed in this section.

PIV Uncertainty

Quantification of measurement uncertainty in Particle Image Velocimetry (PIV) is an active field of current research and is beyond the scope of this work. PIV uncertainty itself arises from several sources, such as spatial mapping of the camera images to each other and to the flow, sensitivity of the cameras, statistical errors within the cross correlation of pixel windows, etc. A thorough study of several approaches for quantifying PIV uncertainty is given by Sciacchitano et al. (2015).

Numerical Error

All computer based processing schemes will be subject to numerical truncation error, which at high signal-to-noise ratio (*SNR*) values will cause a noise floor to be reached. In this case, numerical error refers both to the precision of the solver as well as the truncation error of the input data files from third party PIV software.

Temporal and Spatial Resolution Limits

The resolution of the data is critical in the validity of the acceleration solution. As discussed above, the temporal resolution of the acceleration fields is $4\Delta t$, since five successive image frames generate one acceleration field in the Lagrangian scheme. This leads to two related constraints. The temporal resolution must be significantly smaller than the time scale of the flow of interest so that the resulting velocity and acceleration fields can accurately be considered instantaneous snapshots. In addition, the temporal resolution of $4\Delta t$ must allow that a grouping of particles stays within the image field of view for five successive frames so that the pseudo-tracking algorithm is valid. This latter requirement may be at odds with the spatial resolution requirements. Particularly for the spatial gradients, the spatial resolution must be significantly smaller than the length scale of the flow in order to capture the smallest scales of motion. Since cameras have a finite sensor size, the spatial resolution and the field of view will be inversely proportional. Care must be taken in selecting the lens optics based on the expected turbulence and periodicity in the flow to capture adequate spatial extent with sufficient magnification.

Three Dimensionality

All real fluid flows are three dimensional. However, the present algorithms are based on an assumption of 2D flow, as demanded by the constraints of commonly available PIV acquisition hardware. The errors associated with three dimensionality accumulate with successive steps in the pressure solution algorithm. Out of plane velocity cannot be captured by planar PIV systems (notwithstanding stereo-PIV systems, which are not currently compatible with this code suite). Particles that move outside of the laser sheet in subsequent images result in a loss of correlation, and can also lead to errors in the continuity equation. Out of plane acceleration and gradients suffer from the same uncertainty sources as the velocity. From the Navier Stokes equations, Equation (8), the acceleration in the out-of-plane dimension is neglected in the current implementation of the omnidirectional pressure solver, and strong accelerations in this direction will bias the result. Further, the omnidirectional paths refer only to in-plane rays and assumes uniformity in the third dimension. The present code suite should only be applied to flows where out-of-plane motions are expected to be minimal.

Pseudo-Tracing

The Lagrangian acceleration is computed from a pseudo-tracing scheme as described above. The temporal and spatial resolution limits ensure that a group of particles stays within the field of view for a sufficient number of frames. The PIV correlation algorithm itself effectively imposes a low-pass spatial filter on the data based on the width of the interrogation window. Even with routinely implemented anti-peak-locking routines to capture motions smaller than the width of a single pixel, the reported velocity vector is still the single reported value for the entire window representing the dominant energy motion, with other fluctuations filtered out. Overall this limitation dictates the quoted spatial resolution of a reported data set. This error is propagated into the pseudo-tracking scheme, since the instantaneous pathlines that should be in an ideal calculation are instead approximated in this manner. Alternative methods such as Particle Tracking Velocimetry (PTV) should minimize the error associated from the PIV correlation algorithm and pseudo-tracking for the acceleration.

Viscous Term

The viscous term in the Navier Stokes equation, Equation (8), is neglected in the spatial integration of the pressure gradient field. The measurement of this quantity is subject to restrictions from the spatial resolution, as dissipation typically occurs as a high wavenumber phenomenon that is difficult to capture by PIV. As shown by Liu and Katz (2006), in high Reynolds number flows and away from the viscous boundary layer over solid surfaces, the viscous term is orders of magnitude smaller than the acceleration. Care should be taken to mask out properly any regions close to solid boundary.

Cropping Error

The cropping routine developed here necessarily increases the error in the unmasked regions compared to an ideal, noise-free field. Whereas cropping regions of viscous flow is mandated by dropping the viscous term, cropping of regions where poor signal quality occurs means that the spatial integral is effectively over a smaller path length, and less information is considered in the computation. This cropping is still deemed necessary in order to consider the results valid, but the valid regions in the cropped field will have a higher uncertainty than if the masked areas had contained useable data.

Monte Carlo Simulations

Noise Variance

Monte Carlo simulations were performed to quantify overall error propagation arising from noisy velocity fields. A synthetic velocity field representing a solid body rotation was generated following that by Liu and Katz (2006). Whereas in that work the PIV correlation algorithm was also tested by beginning with synthetic particle images, the present work begins with synthetic velocity fields with added Gaussian noise. This added noise is intended to represent uncertainty arising from both the stochastic nature of the flow as well as uncertainty within the PIV correlation algorithm. The added noise is defined based on a prescribed signal-to-noise ratio, *SNR*. The *SNR* is defined by the variance of the signal σ_{sig}^2 and noise σ_{noise}^2 as follows:

$$SNR = 10\log_{10}\frac{\sigma_{sig}^2}{\sigma_{noise}^2} \tag{16}$$

The variance of the signal is readily computed from the noise-free synthetic signal of the solid body rotation, and the *SNR* is defined for each case. The variance of the noise can then be computed for each *SNR* case. A noise field is generated by the Matlab function “randn” with a normal distribution and defined variance, and added to each frame of the synthetic velocity field.

The synthetic velocity is defined as follows with $\omega = 0.0625$.

$$U = \omega \mathrm{R}\sin\theta \tag{17a}$$

$$V = \omega \mathrm{R}\cos\theta \tag{17b}$$

The spatial extent of the field is from $-0.5 \text{ m} < x < 0.5 \text{ m}$ and $-0.5 \text{ m} < y < 0.5 \text{ m}$, and the fluid density is set at $\rho = 997 \text{ kg/m}^3$. The spatial resolution is 0.01 m (since the velocity field is synthetic, the caveat about overlapping windows in the PIV correlation algorithm does not apply), and the temporal resolution of the velocity field is $dt = 10$ µs. A total of 105 sequential fields with non-correlated added noise was employed for each set.

Synthetic data were generated for *SNR* cases in 5 dB increments from -10 to 100. Resulting acceleration fields were normalized as $|DU/Dt| \,/\, (\omega^2 R_{max})$ and pressure as $((p - p_{center})) \,/\, (\frac{1}{2}\rho\omega^2 r_{max}^2)$, as per Liu and Katz (2016). Results are plotted for all *SNR* cases in Figure 7 for the acceleration term and in Figure 8 for the pressure term.

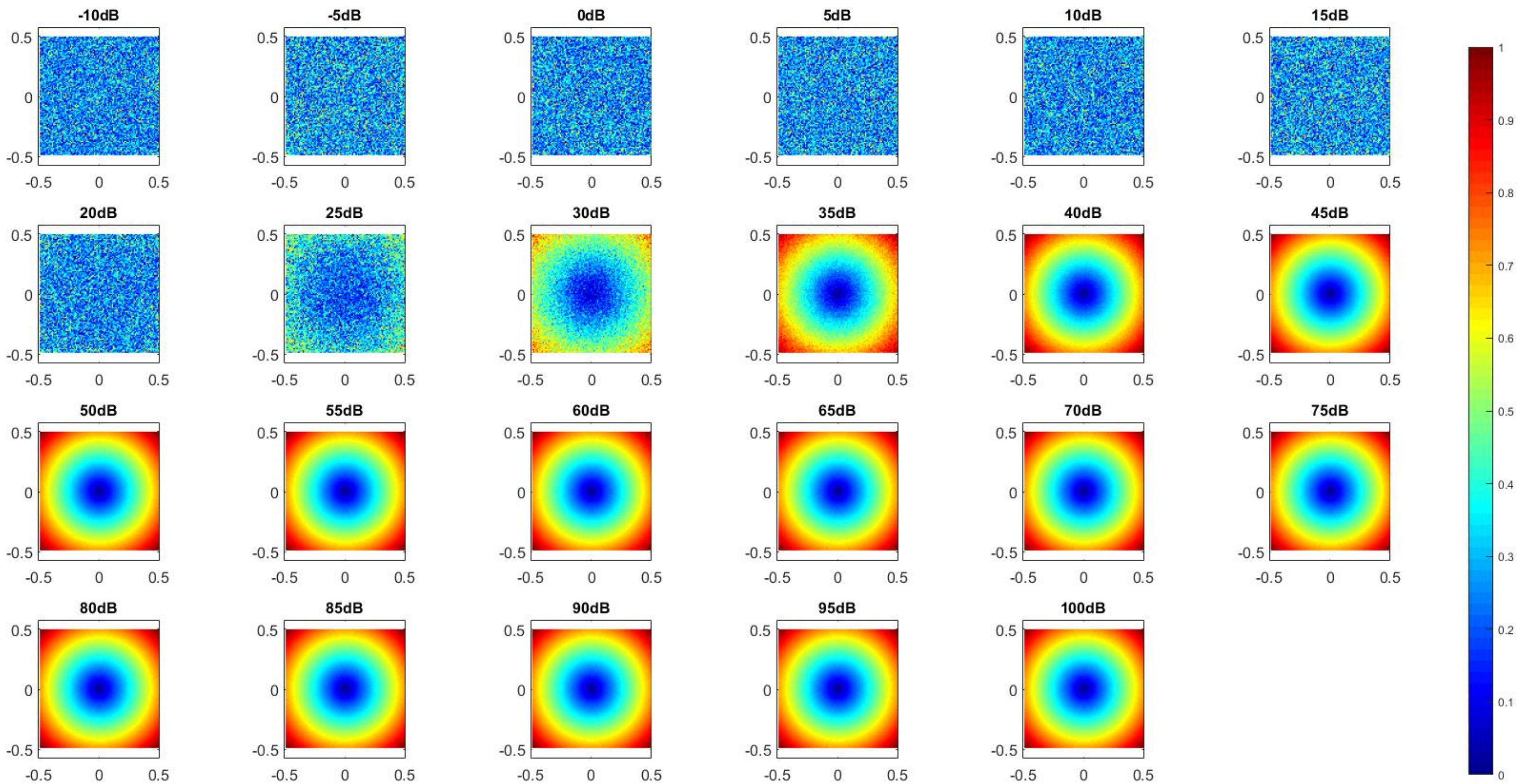

Figure 7. Contours of $|DU/Dt| \,/\, (\omega^2 R_{max})$ from Monte Carlo Simulations at Several SNR Levels

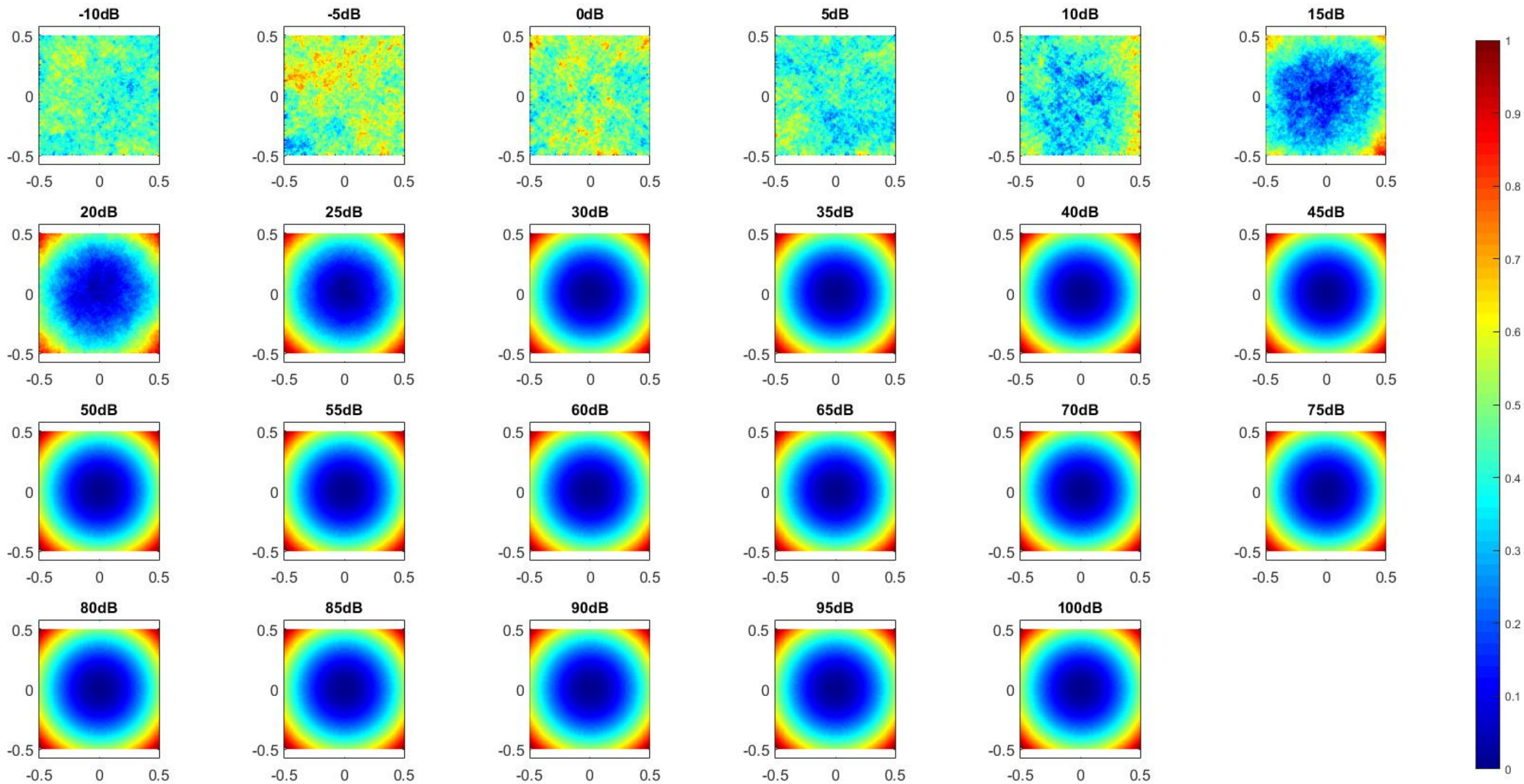

Figure 8. Contours of $((p - p_{center})) \,/\, (\frac{1}{2}\rho\omega^2 r_{max}^2)$ from Monte Carlo Simulations at Several SNR Levels

The temporal variance of the magnitude of acceleration was calculated for each point in space, and the mean variance was computed to yield a single acceleration variance term for the entire case. The pressure variance was similarly calculated, after subtracting the calculated pressure at the [0,0] point from each individual frame. These terms are plotted in Figure 9 against the defined *SNR* of the original velocity field and display a logarithmic trend at lower *SNR* levels, after which each parameter reaches a noise floor. For the acceleration, the noise floor is reached by $SNR \sim 50$, while for the pressure the noise floor is reached by $SNR \sim 35$. This is consistent with the expected propagation of errors as the pressure is a higher order term with additional error sources beyond those that affect the acceleration.

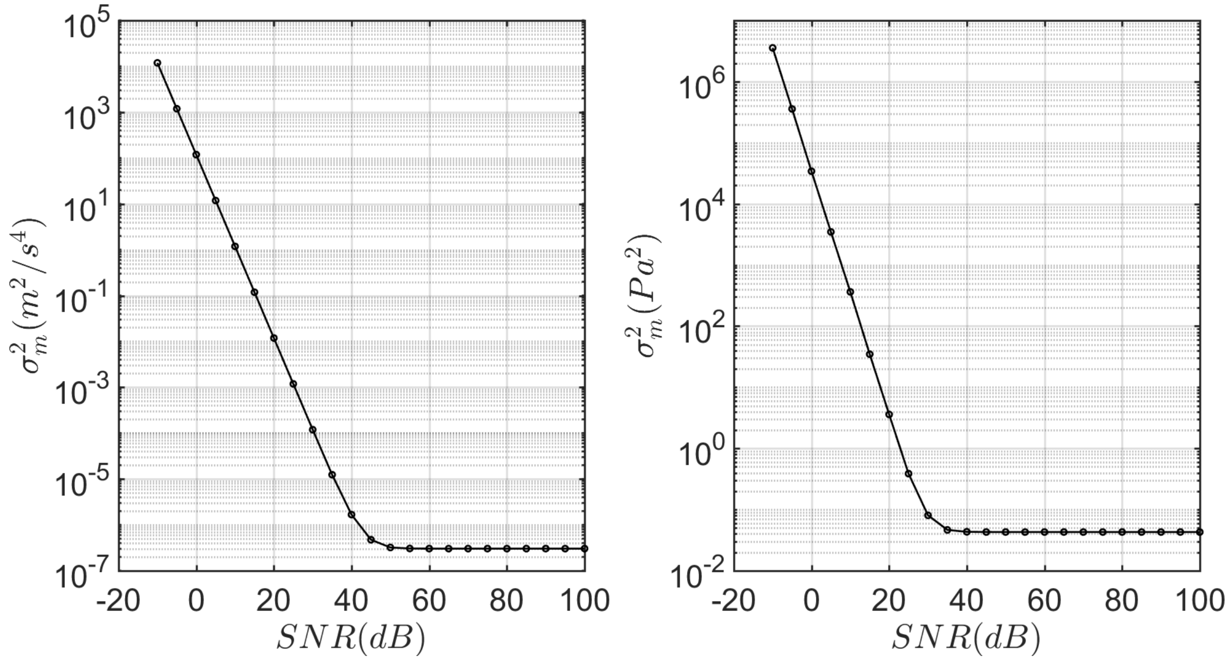

Figure 9. Resulting Signal Variance of Acceleration (Left Frame) and Pressure (Right Frame) as a Function of the Velocity Field *SNR*

Masking

The masking feature of the pressure solver was tested by applying two different masks to the $SNR = 40$ dB acceleration field. The first mask includes a square region of data from -0.3 m $< x < 0.3$ m and -0.3 m $< y < 0.3$ m. This “framed” case represents a field where the outer portions of the frame contain invalid data, i.e. from poor seed density or un-focused optics around the periphery of the frame. The second case excludes a square region of data from -0.5 m $< x < 0$ m and -0.5 m $< y < 0$ m. This “corner” case represents an image where a distortion is present in the frame, perhaps from shadows of the laser sheet or a solid body in the flow field in which viscous forces are non-negligible.

Results are shown in Figure 10 for the framed and Figure 11 for the corner case. The top left frame in each figure is the full, non-masked image shown for reference, while the bottom left shows the masked field result. The top right shows the absolute difference between the full and masked field, while the bottom right is the difference normalized by the local value of the full-field results. As expected, deviations from the full processed field are observed since less information is in integrating the field. Errors are greatest at locations of sharp included angles of the field, i.e. in regions where the mask results in the valid data having an artificial corner. This is attributable to the fact that fewer parallel integration rays exist that pass through these points that contain substantial information from regions within the valid domain, as compared to other points within the field. In both cases, a highly normalized error occurs at the center of the field, due to the small magnitude of the pressure in the normalization. Overall, the cropping routine is shown to produce results that approximate those of an equivalent full-field measurement, although care must be taken to account for higher uncertainties.

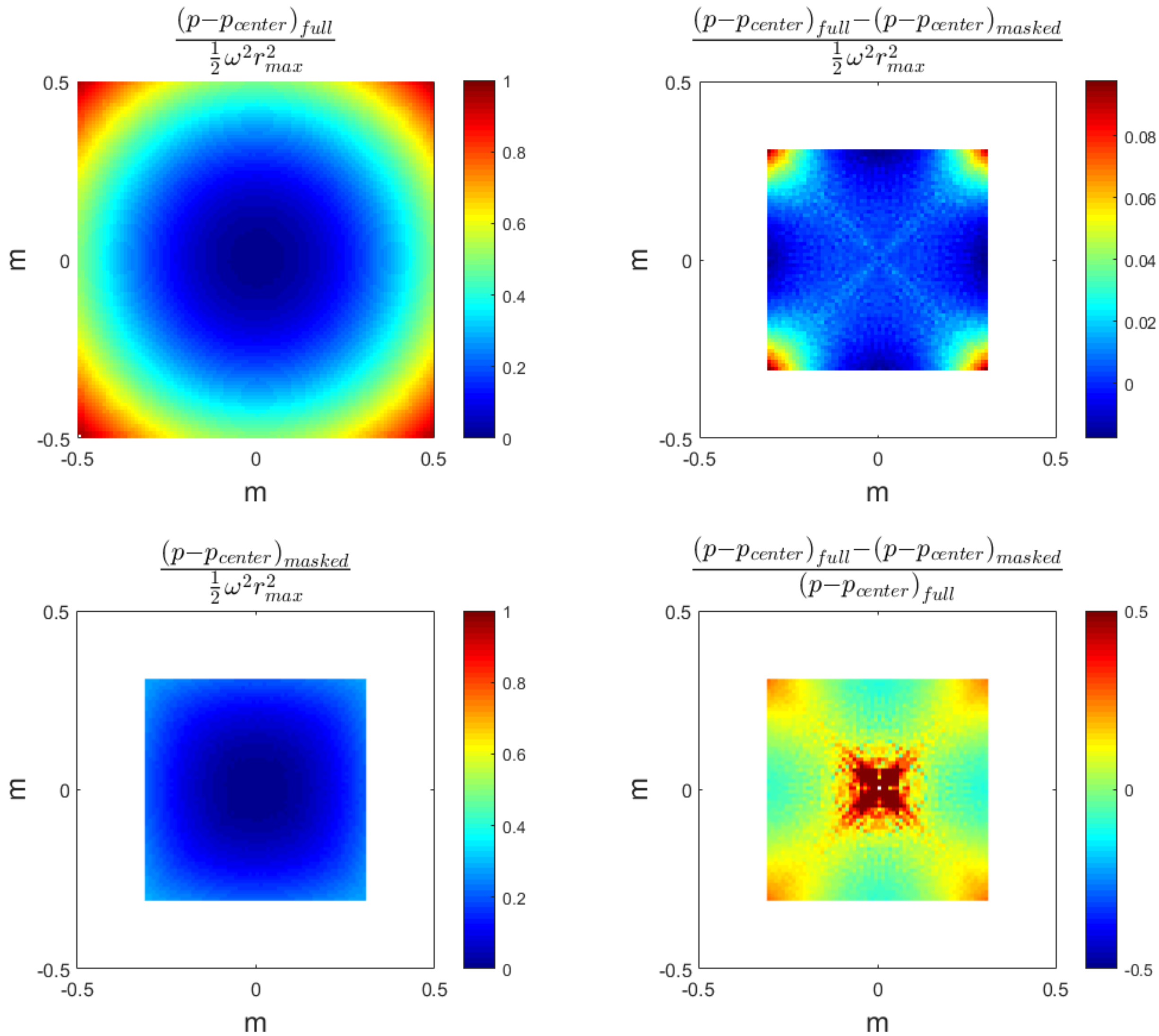

Figure 10. Pressure Field Results from “Framed Case” Mask as Compared with the Full-Field Processing at 40 dB

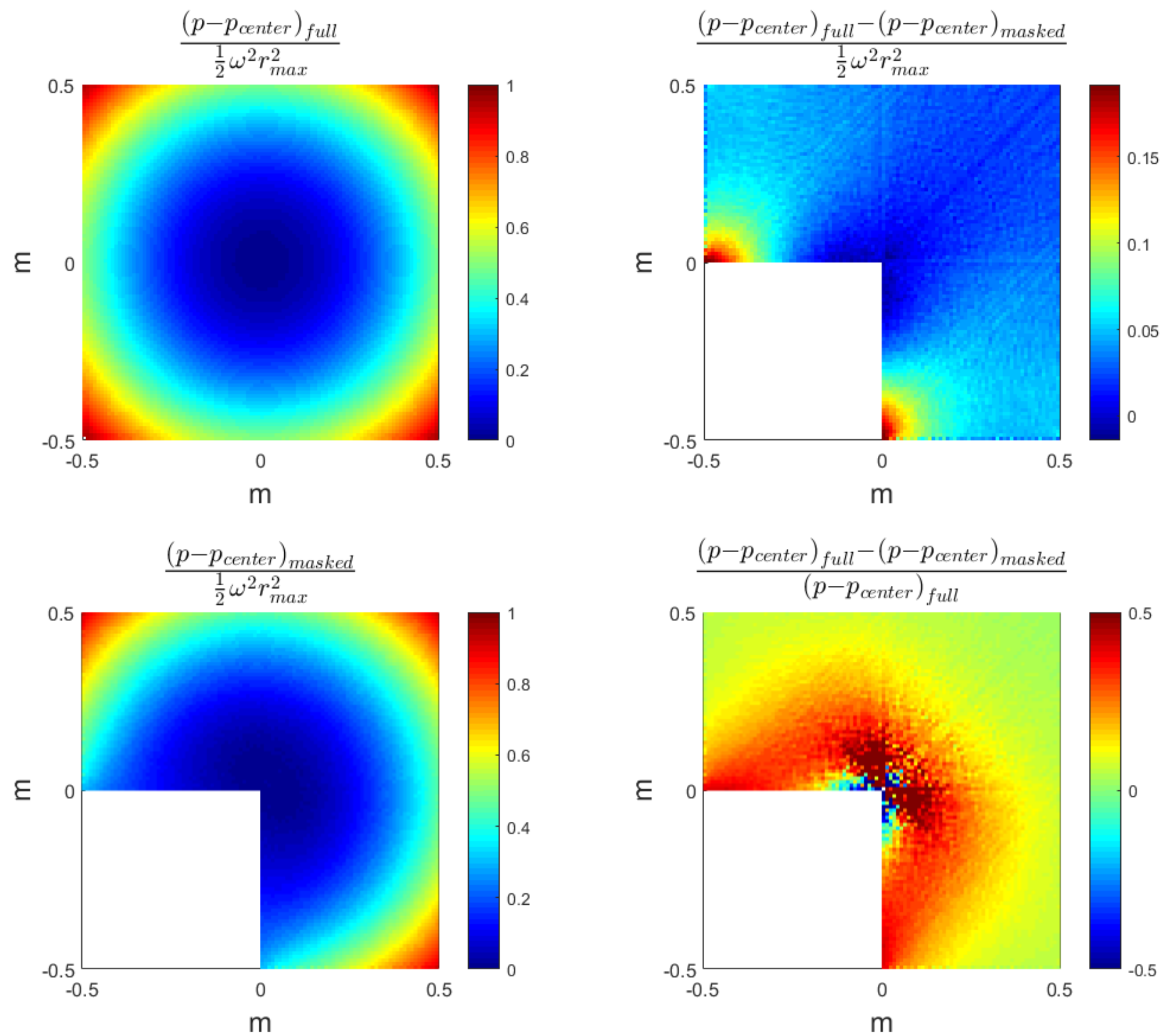

Figure 11. Pressure Field Results from "Corner Case" Mask as Compared with the Full-Field Processing at 40 dB

CONCLUSIONS AND OUTLOOK

A suite of Matlab processing codes has been developed for computing the turbulent pressure field from particle image velocimetry (PIV) data. A review of the theoretical framework is given, including alternate approaches not implemented here. The chosen approach is described mathematically, and the programming architecture outlined. The Lagrangian acceleration is computed from five successive particle images of time resolved data, and this result is in an omnidirectional pressure gradient integration scheme to compute the pressure field at each time instance. The ability to mask regions of the acceleration field is newly implemented here. A graphical user interface for each of the acceleration and pressure solvers is provided to set conditions and build a batch list for processing. A separate graphical user interface (GUI) for interactively visualizing the resultant fields is also described. Noise sources in the data are discussed, and limitations on the acquisition parameters are described, particularly the requirements for high spatial and temporal resolution, while maintaining a sufficiently large field of view for the Lagrangian acceleration pseudo-tracking scheme. Monte Carlo simulations are performed on synthetic velocity fields representing a solid body rotation with added Gaussian noise at varying noise variances. These synthetic data are used to compute the acceleration and pressure in order to study the error propagation through the solvers, and a noise floor is reached for each of the routines beyond which increased *SNR* of the velocity field does not lower the variance of the result. Monte Carlo simulations are also run with a mid-level noise case of the solid-body rotation velocity field with the addition of an applied mask. Two cases are studied, one where the outer region of the frame in all four frame edges is masked, and another where a large square region in the corner of the frame is masked. The resulting pressure field shows the highest deviation from the full-field result in included corners near the mask, with smaller differences elsewhere in the frame.

The completed code suite is a stand-alone package that can be applied to appropriate PIV data. Future work should involve the extension of the current package to three dimensional, three component data, suitable for tomographic PIV velocity data, as the suite is currently limited to sufficiently two-dimensional flows. Dedicated experiments should be designed to validate rigorously the present code suite against a known, canonical flow field with real noise sources and appropriate instrumentation.

REFERENCES

Adrian, R. J. (1984), “Scattering particle characteristics and the effect on pulsed laser measurements of fluid flow: speckle velocimetry vs particle image velocimetry,” *Applied Optics* 23:1690-1691.

Adrian, R. J. and Westerweel, J. (2011), *Particle Image Velocimetry*, Cambridge University Press: New York.

de Kat, R. and Ganipathisubramani, B. (2012), “Pressure from particle image velocimetry fir convective flows: a Taylor’s hypothesis approach,” *Measurement Science and Technology* 24:024002.

de Kat, R. and van Oudheusden, B. W. (2012), “Instantaneous planar pressure determination from PIV in turbulent flow,” *Experiments in Fluids* 52:1089-1106.

Dominy, R. G. and Hodson, H. P. (1992), “An investigation of the factors influencing the calibration of 5-hole probes for 3D flow measurements,” ASME International Gas Turbine and Aeroengine Congress and Exposition, 92-GT-216.

Elsinga, E., Scarano, F., Wieneke, B., and van Oudheusden, B. W. (2006), “Tomographic particle image velocimetry,” *Experiments in Fluids* 41:933-947.

Fujiwasa, N., Tanahashi, S., and Srinivas, K, (2005), “Evaluation of pressure field and fluid forces on a circular cylinder with and without rotational oscillation using velocity data from PIV measurement,” *Measurement Science and Technology* 16:989-996.

Gurka, R., Liberzon, A., Hefetz, D., Rubenstein, D., and Shavit, U, (1999), “Computation of Pressure Distribution Using PIV Velocity Data,” Third International Workshop on Particle Image Velocimetry, Santa Barbara, California.

Jakobsen, M. L., Dewhirst, T. P., and Greated, C. A. (1997), “Particle image velocimetry for predictions of acceleration fields and force within fluid fields,” *Measurement Science and Technology* 8:1502-1516.

Jensen, A., Pedersen, G. K., and Wood, D. J. (2003), “An experimental study of wave run-up at a steep beach,” *Journal of Fluid Mechanics* 486:161-188.

Liu, T., Guille, M., Sullivan, J. P. (2001), “Accuracy of Pressure Sensitive Paint,” *AIAA Journal* 39:1.

Liu, X. and Katz, J. (2006), “Instantaneous pressure and material acceleration measurements using a four-exposure PIV system,” *Experiments in Fluids* 41:227-240.

Liu, X., Moreto, J. R., and Siddle-Mitchell, S (2016), “Instantaneous Pressure Reconstruction from Measured Pressure Gradient using Rotating Parallel Ray Method,” 54th AIAA Aerospace Sciences Meeting 4-8 Jan 2016, San Diego, California, AIAA 2016-1049.

McLachlan, B. G., Kavandi, J. L., Callis, J. B., Gouterman, M., Green, E., Khalil, G., and Burns, D. (1993), "Surface pressure field mapping using luminescent coatings," *Experiments in Fluids* 14:33-41.

Nelson, M. A. (2018), "Uncertainty Quantification in Steady State PSP Using Monte Carlo Simulations at AEDC," AIAA Aerodynamic Measurement Technology and Ground Testing Conference, Atlanta, Georgia.

Novara, M. and Scarano, F. (2013), "A particle-tracking approach for accurate material derivative measurements with tomographic PIV," *Experiments in Fluids* 15:1584.

Pope, S. B. (2000), *Turbulent Flows*, Cambridge University Press, New York.

Prasad, A. K. (2000), "Particle Image Velocimetry," *Current Science* 79:1, 51-60.

Sciacchitano, A., Neal, D. R., Smith, B. L., Warner, S.O., Vlachos, P. P., Wieneke, B., and Scarano, F. (2015), "Collaborative framework for PIV uncertainty quantification: comparative assessment of methods," *Measurement Science and Technology* 26:074004.

Sellers, M., and Brill, J. (1994), "Demonstration test of pressure sensitive paint in AEDC 16-ft transonic wind tunnel using the TST model," 25th Plasmadynamics and Laser Conference, American Institute of Aeronautics and Astronautics, Reston, Virginia, 1994-2481.

Sellers, M., Nelson, M., and Crafton, J. W. (2016), "Dynamic pressure-Sensitive Paint Demonstration in AEDC Propulsion Wind Tunnel 16T," 54th AIAA Aerospace Sciences meeting, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2016-1146.

Sheng, J., Malkiel, E., and Katz, J. (2006), "Digital holographic microscope for measuring three-dimensional particle distributions and motions," *Applied Optics* 45:3893-3901.

Spalart, P. R. (2015), "Philosophies and fallacies in turbulence modeling," *Progress in Aerospace Sciences* 74:1-15.

Stern, F., Wilson, R. V., Coleman, H. W., and Paterson, E. G. (1999), "Verification and Validation of CFD Simulations," Iowa Institute of Hydraulic Research Technical Report IIHR-407.

Sumner, D. (2002), "A Comparison of Data-Reduction Methods for a Seven-Hole Probe," *Journal of Fluids Engineering* 124:523-5.

Thurow, B., Jiang, N., and Lempert, W. (2013), "Review of ultra-high repetition rate laser diagnostics for fluid dynamic measurements," *Measurement Science and Technology* 24:012002.

Timmer, W. A. (2008), "Two Dimensional low-Reynolds number wind tunnel results for airfoil NACA 0018," *Wind Engineering* 32:6, 525-537.

Tsai, R. Y. (1987), "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation* 3:323-344

Van Oudheusden, B. W, (2013), "PIV-based pressure measurement," *Measurement Science and Technology* 24:032001.

Wieneke, B. (2005), "Stereo-PIV using self-calibration on particle images," *Experiments in Fluids* 39:267-280.

Willert, C. (1997), “Stereoscopic digital particle image velocimetry for application in wind tunnel flows,” *Measurement Science and Technology* 8:12, 1465-1479.

Distribution

	Hard Copies	*Digital Copies*
Defense Technical Information Center 8725 John J. Kingman Road Ft. Belvoir, VA 22060-6218	0	1
Dr. Xiaofeng Liu Department of Aerospace Engineering San Diego State University 5555 Canyon Crest Drive San Diego, CA 92182	0	1

NSWC, CARDEROCK DIVISION INTERNAL DISTRIBUTION

Code	Name	*Hard Copies*	*Digital Copies*
83	Technical Data Library	1	1
1033	TIC - SCRIBE	0	1

THIS PAGE INTENTIONALLY LEFT BLANK

NAVSEA

WARFARE CENTERS

Carderock