

A WEB-BASED PYTHON PIV ANALYSIS PROGRAM FOR USE IN UNDERGRADUATE LABORATORY EXPERIMENTS.

JACOB SEPHTON
SUPERVISED BY PROF. JULIO SORIA

1. INTRODUCTION AND AIMS

1.1. Introduction. Many methods exist for measuring velocity, but many commonly used methods, such as pitot tubes, restriction plates, or turbine meters only measure velocity at a point or as an average across a section, and may affect flow both up and downstream of the measurement device (Bruce R. Munson, 2006, pp. 109, 469). These methods have the advantage of simplicity of measurement, but they cannot be used to determine the behaviour of a broader flow field within a flow: another method is required.

Particle Image Velocimetry (henceforth: PIV) is one such method. In PIV, marker objects – usually small spheres – with neutral buoyancy are introduced to the portion of the flow to be studied (Wikipedia contributors, 2021a)¹. When appropriately sized for a specific flow, these particles will track the bulk motion – the advection – of the flow, with a high degree of accuracy. Both the magnitude and direction of the velocity around the particle can then be determined through photography over a known time interval. When many particles are used, the velocity at many points in a flow field can be measured at once, permitting the interpolation of the entire velocity field in a given region of a flow. And, by varying the parameters of the study, like fluid properties, flow rate, particle size, and time interval, flow phenomena at different length and time scales can be modelled.

In light of these potential benefits, undergraduate laboratory experiments to introduce the method represent an opportunity for students to learn about a powerful analytical technique. Since PIV methods require the processing of large amounts of data, computers are used for analysis, and the software used to do so – PIV analysis software – uses a variety of algorithmic and signal processing techniques to do so.

Developing a PIV analysis program for undergraduate use will be the focus of this project. This program will use Dash, a declarative web application framework, and Python code to implement PIV analysis, and make analysing, presenting, and exporting data simple and easy. Dash is a good framework for the development of such an application, as it is open-source and designed to make interface implementation and data display (relatively) painless (Plotly, 2021b). Furthermore, being web-based, Dash provides the capacity for the application to be used remotely on almost any computer. The specific feature requirements and implementation details of the application will be determined as part of the project, and development will be carried out in an iterative fashion to manage complexity.

PIV software is not new, nor is it inaccessible. OpenPIV, for example, is an open-source PIV analysis package available under the GPL² (OpenPIV, 2014). The difference between this software and that which is currently available will be that this project has

¹While not appropriate for a more in-depth analysis, Wikipedia provides reasonably good high level summaries of most concepts and is cited here for this reason.

²Referring to the GNU General Public License, an influential “copyleft” software license that requires derivative software be made available under it’s own terms.

an explicit and narrow target application: that of teaching about PIV in an undergraduate lab setting. Such software may be further made relevant by the potential for this undergraduate learning to take place remotely: while the hardware required to capture PIV images may be expensive, the analysis can be performed anywhere.

1.2. Aims. This project aims to develop a PIV application with a didactic focus. This will involve:

- Determining the specific requirements of the application. Input form, analysis capabilities, and the degree to which the mechanisms of analysis are made visible to the user are all to be determined. This will further involve determining whether physical apparatus needs to be made a part of the development process
- Creating a spec and development timeline for the application which prioritises essential features first to ensure that deadlines are met, and identifies worthy extensions to essential features for continued development.
- Conducting research to understand the theory behind and parameters of PIV analysis and signal processing to the extent necessary to teach the most important concepts behind them in an engaging way. This will further involve researching methods for interpreting output of PIV, identifying error sources, limitations, and data transformations that can provide insight into the flow phenomena studied.
- Developing tests to evaluate the performance of the application. This may involve comparing application output with other packages using widely used benchmark input, such as the PIV images available on PIV Challenge (PIV Challenge, 2016).
- Potentially developing a lab script that could be used in conjunction with the application that makes use of the application's features and develops students' understanding of the PIV process.

2. METHODOLOGY

The methodology for both software development as it pertains to this project and of PIV more generally are discussed in this section. Software development methodologies are a specific domain of project management practices designed to manage the complexities associated with writing software, and will help ensure that major component of this project succeeds. Meanwhile, it is important to discuss the particulars of PIV from a physical perspective as a basis for developing understanding of the topic and informing design decisions to be made.

2.1. Software development. Since software projects grow in complexity quickly with increasing scope, the field of software engineering has arisen to develop and formalise best practices that make writing functional, easily maintainable, and extensible software something that can be done repeatably and efficiently. Since a major portion of this project concerns the delivery of a software application, the adoption of some of these best practices seems prudent. We will consider development methodology and version control systems as part of this proposal.

Development methodologies are principle-guided sets of decisions around how software is written that aim to optimise the quality of software along one or more axes. For this project, principal considerations include hard deadlines that must be met, the fact that the conclusion of development must occur well in advance of these deadlines, and the prospect of requirements changing through the course of the project. With these issues in mind, the methodology used for this project will draw from Agile principles.

Agile software development stresses, among other things, the delivery of working software early and continuously (Agile Alliance, 2021). To do this, software is reduced to its

minimal feature set to begin with, then built upon iteratively. The facts that working software is produced as early as possible, and feature addition decisions do not need to be made until work on that feature actually commences makes this an appropriate school of development to draw from for this project, helping to meet deadlines and deal with requirement flux.

Informing this is the development of a functional specification (spec) early in the process. Per (Spolsky, 2000a), advantages of doing this include the fact that design decisions that have to be made anyway can be made before code is written, avoiding wasted effort, and that a detailed spec allows development work to be scheduled, helping to deliver the best possible application by the deadline. A spec will be developed for this project similar to one described by (Spolsky, 2000b) detailing how the software will work from the user's perspective. This will inform the technical decisions made further along.

Software will be used to assist the development process. Firstly, Git, a version control system, will be used to back-up and track changes to the codebase over time. It permits branched feature development while always having access to the last functional version of the software, and these features can be merged back into the existing main version of the software, even if changes have been made to the main version since the branch occurred (Blischak JD, 2016). Secondly, task tracking software Jira (Atlassian, 2021), which is designed to facilitate an Agile development methodology, will be used for organisation.

2.2. PIV methodologies. PIV typically involves the study of flow in a single plane of a broader system. To accomplish this, a planar strobe lighting setup is common, with a strobe or laser firing through a cylindrical lens, which spreads the beam out along one axis, and then through a spherical lens which focusses the light to a thin sheet at the area of study. Arising from this is the possibility of parallax error in measurement, from lit particles that are closer or further away from the lens than the centre of the light plane. Another error source arising from the planar nature of most PIV analyses is motion perpendicular to the imaging plane, which can lead to incorrect conclusions being drawn if not accounted for. This does mean that PIV is best applied when the imaging plane can be made to closely match the predominant flow directions in the area of study. For more complicated flows, stereo imaging can be used to carry out PIV in three dimensions, albeit with added complexity and cost.

One potential error source in PIV is tracking error between particles and the flow field. This makes particle and flow medium selection an important part of carrying out PIV analysis. In Stokes flow, with particle Reynolds numbers, Re_p (1), less than 1, the Stokes number, Stk , can be given by (2). The Stokes number describes the relationship between the characteristic time of the flow and the response time of a particle in that flow, and thus is highly relevant when considering PIV particle parameters. Higher Stk implies either a high particle response time, or a proportionately high velocity-to-feature size ratio and thus suggests that the motion of the particle will not capture the full velocity profile of the flow (Wikipedia contributors, 2021b).

$$(1) \quad Re_p = \frac{\rho_f u d_p}{\mu}$$

$$(2) \quad Stk = \frac{t_0 u}{l_0} | t_0 = \frac{\rho_p d_p^2}{18\mu}$$

In these equations, ρ is density of the fluid or the particle, u is fluid velocity, d_p is particle diameter, μ is fluid viscosity, t_0 is particle response time, and l_0 is the characteristic length of the flow being studied. Compromise must be made between minimising t_0

through small particle d and difficulty in imaging small particles. Much more goes into the consideration of particle and fluid parameters than can be contained within the scope of this proposal, but this provides an example.

One last point of note are signal processing techniques used in PIV. All PIV analysis requires the interpretation of images to generate velocity fields, and automated methods of doing so vary depending on the images captured. Some PIV, especially using older cameras, relies on two strobes within a single exposure. Here, autocorrelation is used to determine velocity from particle displacement within one image. Newer PIV often uses multiple exposures, with the benefit of being able to identify flow direction: PIV of this sort uses cross-correlation between images. Cutting-edge techniques are investigating the use of machine learning to correlate images with velocity fields, with the benefit of improved velocity field resolution (Wikipedia contributors, 2021a).

3. PROJECT PLAN

The project is broken into stages for planning purposes. Appendix A shows the proposed timeline for the project, detailing when each stage should be started and finished, contextualised within the rest of the project. An elaboration on each stage described therein follows.

3.1. Project proposal. You're reading it! This stage involves not just the creation of the report, but also initial broad research undertaken to gain an understanding of the project and its future direction. Deliverables include this report, a risk assessment for project work, and an understanding of the project's future direction.

3.2. Learning Python and Dash. This stage involves completing the Dash tutorial (Plotly, 2021a) and familiarising myself with the Python programming language. Sub-tasks involve setting up a source control repository for project work and a development environment on my computer. This task is mostly complete, and I'll finish the tutorial over the mid-semester break.

3.3. Requirements determination. In this stage the specific requirements of the app will be defined and clarified. What features must the app have? What non-essential features should be prioritised. How should it work, and how will the design decisions made serve the aim of introducing PIV to undergraduate students. This stage will also involve research into specific algorithms that can be used in PIV, and reading to understand both PIV and the signal processing theory that makes it possible. This stage strongly informs the spec that will be developed in the next stage.

3.4. Application specification. Here, a specification document will be produced for the application. It will define the proposed architecture, and all of the features the app needs to have to be considered complete. It will also define future feature development, prioritising non-essential features that will have the biggest impact on the usefulness of the application. Tests will be developed to measure application performance. This document will inform the implementation of the software through the middle of the project. A lab script may be treated as a feature of the application for project management purposes.

3.5. Progress report. A report detailing findings in the project to this point will be produced. It will contain the findings of research to this point, decisions made around design and features, and the specification document, likely abridged in some fashion. The progress report will also include some information on how implementation is proceeding, and perhaps some demonstration of any software features implemented, if possible.

3.6. Implementing the specification, followed by iterative development. The actual programming will be done in this stage. See methodology section for more detail.

3.7. Final report and research paper. The final report and research paper will be started early, in the mid-year break. The final report will detail all outcomes of the project: the findings of research carried out, how software was implemented, how the finished software performs, and will evaluate the work carried out in terms of the aims of the project. The research paper will investigate an area related to the project, likely informed by research carried out in earlier stages. The aim will be to build both of these documents over the course of semester two, with the first draft of each due at a self-set deadline at the end of week 8 and week 9 respectively. Having these done early will avoid an end-of-semester crunch.

3.8. Shooting and editing project video. The project video will be started as the development phase winds down, and should present the main findings of the final report in an engaging short presentation. A demonstration of the application will feature, and I will talk about the process of research and development that went into it. The self-set deadline at the end of the mid-semester break is the last deadline in the project prior to official deadlines, leaving room for the polishing stage,

3.9. Polishing. At this point, all project deliverables should be complete, at least to the first draft. This stage is where edits, final corrections, and any necessary supplementary information are added to the reports and videos. Two weeks of polishing time is allocated for the video, and three for the two reports. This stage should allow all deliverables to be ready well in advance of their deadline.

REFERENCES

- Agile Alliance (2021). 12 principles behind the agile manifesto. <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>. [Online, accessed 2-April-2021].
- Atlassian (2021). What is jira used for? <https://www.atlassian.com/software/jira/guides/use-cases/what-is-jira-used-for>. [Online, accessed 2-April-2021].
- Blischak JD, W. G., Davenport ER (2016). A quick introduction to version control with git and github. *PLoS Comput Biol*, 12(1): e1004668. doi: 10.1371/journal.pcbi.1004668.
- Bruce R. Munson, T. H. O., Donald F. Young (2006). *Fundamentals of Fluid Mechanics, Fifth Edition*. Wiley.
- OpenPIV (2014). OpenPIV: open source particle image velocimetry. <http://www.openpiv.net>. [Online; accessed 31-March-2021].
- PIV Challenge (2016). PIV Challenge. <https://www.pivchallenge.org>. [Online, accessed 1-April-2021].
- Plotly (2021a). Dash tutorial part 1 : installation. <https://dash.plotly.com/installation>. [Online, accessed 1-April-2021].
- Plotly (2021b). Introduction to Dash. <https://dash.plotly.com/introduction>. [Online; accessed 31-March-2021].
- Spolsky, J. (2000a). Painless functional specifications – part 1: Why bother. <https://www.joelonsoftware.com/2000/10/02/painless-functional-specifications-part-1-why-bother/>. [Online, accessed 2-April-2021].
- Spolsky, J. (2000b). Painless functional specifications – part 2: What’s a spec? <https://www.joelonsoftware.com/2000/10/03/painless-functional-specifications-part-2-whats-a-spec/>. [Online, accessed 2-April-2021].
- Wikipedia contributors (2021a). Particle image velocimetry — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Particle_image_velocimetry&oldid=1012742345. [Online; accessed 30-March-2021].
- Wikipedia contributors (2021b). Stokes number — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Stokes_number. [Online; accessed 30-March-2021].

APPENDIX A. PROPOSED PROJECT TIMELINE

Project wk.	University wk.	Begins	Task			
1	S1 W3	15-Mar-21	Project proposal Due 1st	Learning Python and Dash		
2	S1 W4	22-Mar-21				
3	S1 W5	29-Mar-21				
4	MSB	5-Apr-21			Requirements determination	
5	S1 W6	12-Apr-21				
6	S1 W7	19-Apr-21				
7	S1 W8	26-Apr-21				App specification
8	S1 W9	3-May-21				
9	S1 W10	10-May-21	Progress report Due 28th	Implement software to spec		
10	S1 W11	17-May-21				
11	S1 W12	24-May-21				
12	S1 E0	31-May-21				
13	S1 E1	7-Jun-21				
14	S1 E2	14-Jun-21				
15	S1 E3	21-Jun-21				
16	MB 1	28-Jun-21			Final report	Research paper
17	MB 2	5-Jul-21				
18	MB 3	12-Jul-21		Iterative feature development		
19	S2 W0	19-Jul-21				
20	S2 W1	26-Jul-21				
21	S2 W2	2-Aug-21				
22	S2 W3	9-Aug-21				
23	S2 W4	16-Aug-21				
24	S2 W5	23-Aug-21				
25	S2 W6	30-Aug-21	Shoot video			
26	S2 W7	6-Sep-21				
27	S2 W8	13-Sep-21				
28	S2 W9	20-Sep-21				
29	MSB	27-Sep-21				
30	S2 W10	4-Oct-21	Polish			
31	S2 W11	11-Oct-21			Video: 15th	
32	S2 W12	18-Oct-21			FR: 22nd	RP: 22nd
33	S2 E0	25-Oct-21	Prep for interview		Interview this week	

APPENDIX B. RISK ASSESSMENT

See following pages.

1. Activity or Task Based Risk Assessment [Ref Number: 28240]

Name	MAE FYP S1 2021 Development of PIV application	Current Rating	Residual Rating
		Medium	Low
Location	Off Campus: Office of a private residence		
Business Unit		Last Review Date	Risk Owner
	Dean Faculty of Engineering	30/03/2021	JACOB PATRICK SEPHTON
Risk Assessment Team		Risk Approver	
		Julio Soria	
Additional Notes			
Describe task / use			
	Research and development of software application, primarily computer-based		

1. Activity or Task Based Risk Assessment [Ref Number: 28240]

Risk Factors

Risk Factor

4.3 - Musculoskeletal stress from awkward posture or repetitive movements

Description

Prolonged sitting while writing, using computer, or reading

- 1.0 - Fixed plant (e.g. cool rooms, fume cupboards, safety showers, boilers, lathes, lifts, gas mains, PET scanners) -- No
- 2.0 - Transport and mobile plant (e.g. motor vehicles, forklifts, walky stackers, trolleys and wheelbarrows) -- No
- 3.0 - Powered equipment, tools and appliances (e.g. computers, workshop equipment, kitchen equipment, gas cylinders) -- No
- 4.0 - Non-powered handtools and equipment (e.g. furniture and fittings, ladders, handtools, packing equipment, glassware) -- No
- 5.0 - Chemical materials (e.g. dangerous goods, hazardous substances, poisons and drugs) -- No
- 6.0 - Materials and substances (not otherwise selected from category 5.0) -- No
- 6.3 - Fire and smoke -- No
- 7.1 - Outdoor working environment (e.g. carparks, walkways, outdoor stairs) -- No
- 7.2 - Indoor working environment (e.g. internal rooms, floor surfaces, stairwells) -- No
- 8.5 - Biological materials (e.g. animals, non-living animal materials, microorganisms) -- No
- 8.4 - Personal impairment and/or interaction (e.g. pre-existing medical condition, assisting a patient) -- No
- 9.1 - Psychological (e.g. stressful situations) -- No

1. Activity or Task Based Risk Assessment [Ref Number: 28240]

Medium				
Existing Controls		Proposed Controls		
<ul style="list-style-type: none"> 4 - Engineering control measure: Use of ergonomic task chair with adjustable armrests and lumbar support 4 - Engineering control measure: Use of adjustable desk and full size computer monitor to facilitate correct posture 4 - Engineering control measure: Use of height adjustable desk and computer monitor to facilitate correct posture 		Description	Responsibility	Target Date
		Setting up workspace to align with guidelines – see reference	JACOB PATRICK SEPHTON	10/04/2021
		Standing up at least once an hour	JACOB PATRICK SEPHTON	10/04/2021
		Standing up at least once an hour	JACOB PATRICK SEPHTON	10/04/2021
		Re-evaluating this risk if any pain or discomfort develops	JACOB PATRICK SEPHTON	10/04/2021
		Re-evaluating this risk if any pain or discomfort develops	JACOB PATRICK SEPHTON	10/04/2021

1. Activity or Task Based Risk Assessment [Ref Number: 28240]

Risk Factor	4.3 - Musculoskeletal stress from awkward posture or repetitive movements
Description	
Repetitive strain injuries from typing and mousing	

1. Activity or Task Based Risk Assessment [Ref Number: 28240]

Medium	
Existing Controls	Proposed Controls
<ul style="list-style-type: none"> 4 - Engineering control measure: Use of low-actuation-pressure keyboard 4 - Engineering control measure: Use of ergonomic mouse 	Description
	Use of correct typing technique: neutral posture, floating hands, resting wrists and hands by using armrest when not typing
	Ensuring keyboard is used in a front-and-center location without intervening objects as per guidelines – see reference
	Take a break from typing and mousing every half hour
	Responsibility
	JACOB PATRICK SEPHTON
	JACOB PATRICK SEPHTON
	JACOB PATRICK SEPHTON
	Target Date
	10/04/2021
	10/04/2021
	10/04/2021

1. Activity or Task Based Risk Assessment [Ref Number: 28240]

Risk Factor	0.2 - Slips, trips & missteps (i.e. falls on the same level)
Description	
Trips on level surface from obstacles in work area	<ul style="list-style-type: none"> • 1.0 - Fixed plant (e.g. cool rooms, fume cupboards, safety showers, boilers, lathes, lifts, gas mains, PET scanners) -- No • 2.0 - Transport and mobile plant (e.g. motor vehicles, forklifts, walky stackers, trolleys and wheelbarrows) -- No • 3.0 - Powered equipment, tools and appliances (e.g. computers, workshop equipment, kitchen equipment, gas cylinders) -- No • 4.0 - Non-powered handtools and equipment (e.g. furniture and fittings, ladders, handtools, packing equipment, glassware) -- No • 5.0 - Chemical materials (e.g. dangerous goods, hazardous substances, poisons and drugs) -- No • 6.0 - Materials and substances (not otherwise selected from category 5.0) -- No • 6.3 - Fire and smoke -- No • 7.1 - Outdoor working environment (e.g. carparks, walkways, outdoor stairs) -- No • 7.2 - Indoor working environment (e.g. internal rooms, floor surfaces, stairwells) -- No • 8.5 - Biological materials (e.g. animals, non-living animal materials, microorganisms) -- No • 8.4 - Personal impairment and/or interaction (e.g. pre-existing medical condition, assisting a patient) -- No • 9.1 - Psychological (e.g. stressful situations) -- No

1. Activity or Task Based Risk Assessment [Ref Number: 28240]

Low	Low
Existing Controls	Proposed Controls
<ul style="list-style-type: none">• 3 - Isolation of risk: Ensure cords are routed away from traffic areas• 5 - Administrative control measures: Keep objects off the floor in work area	

Appendix

Documents Referenced

WorkSafe Victoria 2006: "Officewise – A guide to health & safety in the office" at url:
<https://content.api.worksafe.vic.gov.au/sites/default/files/2018-06/ISBN-Officewise-guide-to-health-and-safety-in-the-office-2006-01.pdf> accessed 30 Mar 2021

1. Activity or Task Based Risk Assessment [Ref Number: 28240]

Risk Matrix Level	
Negligible	No additional control measures required
Low	Manage by routine procedures at local management level
Medium	Management responsibility must be specified and response procedures monitored
High	Senior management attention needed and management responsibility specified
Extreme	Immediate action required and must be managed by senior management with a detailed plan