Physics of Fluids

High-speed planar ultrasound blood flow velocimetry using deep learning

L.M. Huntink

Master Thesis
Biomedical Engineering, University of Twente

November 12, 2024

Daily Supervisor: Msc. Julian Suk, Dr. Bram de Wilde
Supervisor:
Assoc. Prof. Dr. Guillaume Jajoinie
Prof. Dr. Michel Versluis
Chair:
Committee member: Assoc. Prof. Dr. Jelmer Wolterink
External Member: Assis. Prof. Dr. Erik Groot Jebbink

Abstract

Cardiovascular diseases (CVD) are diseases affecting the blood flow through arteries. These types of diseases are the leading cause of death globally and require early diagnostics to reduce the number of global deaths. Ultrasound is a technique used to image in vivo and can be used to detect CVD. Echo particle image velocimetry (echoPIV) is currently the most common method used to estimate planar velocimetry out of ultrasound B-mode images. However, the technique lacks spatial and temporal resolution. This thesis proposes an alternative method where a residual neural network estimates a planar velocity field from five consecutive ultrasound frames. The neural network is trained on simulated ultrasound data containing curved pulsatile and parabolic blood flow. The trained neural network is experimentally compared to echoPIV by an in vitro measurement of contrast-enhanced water flow through a circular tube. The results show that the neural network improves on echoPIV in its computation time by an evaluation time twenty times faster than echoPIV. Another improvement over echoPIV is the estimation of near-wall velocities. Future research should focus on in vivo measurements to evaluate the robustness of the neural network.

Contents

1 Introduction

Cardiovascular diseases (CVD) specify all diseases affecting blood vessels and the heart. These diseases are the leading cause of death, resulting in 18.2 million deaths in 2019 [1]. For CVD, it is important to have an early diagnosis, which needs to be done by imaging blood flow. This would reduce the number of global deaths. Most of the deaths are in low- and middle-income countries, where limited access to healthcare, poor infrastructure, and shortage of trained health personnel affect diagnosis, screening, and treatment [2]. These factors make ultrasound a solid alternative to more complex and expensive diagnosing methods like CT and MRI due to its low costs and ease of transport benefits.

Ultrasensitive Doppler and ultrasound particle image velocimetry (echoPIV) are the most common methods for measuring blood flow with ultrasound. Doppler ultrasound extracts the blood flow's velocity by measuring the ultrasound signal's velocity-dependent frequency shift. Ultrasensitive Doppler increases the spatial resolution by using gas-filled microbubbles, decreasing the effects of attenuation [3] and reaching a high sensitivity. This technique has a high spatial and temporal resolution of 100 μm and 1 ms, respectively. The main limitation of ultrasensitive Doppler is that it only predicts a one-dimensional velocity, which makes it impossible to measure multi-directional flows, such as in the artery and the heart.

The alternative to Ultrasensitive Doppler is echoPIV. EchoPIV utilizes high-frame-rate ultrasound B-mode images enhanced by contrast agents. In PIV analysis, the cross-correlation in two consecutive frames extracts the distance from this correlation. It extracts the flow profile from the distance and known sample time, enabling the measurement of planar flow velocity. However, echoPIV is limited in providing accurate near-wall velocity estimation due to the high-velocity gradient of the flow near the walls. Furthermore, the long evaluation time of the PIV analysis makes echoPIV insufficient for real-time blood flow velocimetry. This is important because imaging in real time gives immediate diagnosis. To image in real-time, typical frame times are between 5 and 20 ms [4].

Another method to estimate a velocity field from image pairs is to solve the optical flow problem. Optical flow estimates a moving particle's velocity by analyzing how a pixel's intensity changes between image pairs. This movement of intensity in space and time creates a correspondence between the image pairs, allowing for estimating the velocity field based on the shift in pixel

intensity. This problem is solved iteratively by Horn et al. [5], but showed a limitation in that the method can not accurately measure the flow of noisy measurements.

Recent advances in machine learning have revealed significant potential in the field of image processing, particularly in image segmentation [6], classification [7], and image enhancement [8]. In 2015, Fischer et al. introduced the first convolutional neural network, FlowNet, to predict velocity fields from corresponding image pairs [9].

Following the success of FlowNet, this network has been used for laser PIV measurements, trained on a PIV dataset under perfect light conditions [10]. A comparison of the network to experimental PIV analysis showed similar spatial resolution but an improvement in temporal resolution. Later, the dataset was expanded with non-ideal light conditions, resulting in a more realistic PIV dataset [11], increasing accuracy and robustness. However, this network is only trained on PIV data, where small tracer particles are added in a flow. This network has difficulties in robustness when used on echoPIV data due to different tracer particles (speckles instead of particles) and ultrasound artefacts.

This thesis builds on the success of machine learning methods for regular PIV datasets by training networks on simulated B-mode ultrasound images for application in echoPIV. The network is then validated on a test case containing a flow the neural network has not yet seen before. After this test case, an experimental setup is created to validate the network on physically realistic data.

The main goal of the thesis is to investigate whether a neural network can outperform echoPIV in terms of evaluation time and spatial resolution. This is done by checking the accuracy of the neural network on three different types of data. First on simulated data for which the network is trained on, then on simulated data the network has not been trained on and finally experimental data.

2 Background information

This thesis aims to develop a neural network to estimate a velocity field from ultrasound B-mode images, which will ultimately be used in cardiovascular imaging. To this end, a dataset of physically realistic ultrasound consecutive images and their corresponding flow fields is needed to train this neural network. Two simulators will be used for this purpose, and in subsection 2.2, background information on these simulators and similar ultrasound simulators is provided. The simulators require different flow profiles to simulate blood flow, and subsection 2.1 includes background information on these different flow profiles.

Finally, in subsections 2.3 and 2.4, background information is provided on the two proposed methods for estimating a velocity field from ultrasound images: echoPIV and machine learning.

2.1 Hemodynamics

The ultrasound simulators explained in section 2.2 require different analytical flow profiles inside vessels. These flow profiles are either turbulent or laminar based on the Reynolds number:

$$Re = \frac{\rho u L}{\mu} \tag{1}$$

Where ρ is the density of the fluid $[kg/m^3]$, u is the average flow speed $[m/s]$, L is the characteristic length [m], and μ is the dynamic viscosity of the fluid $[Pa \cdot s]$.

The Reynolds number indicates whether the flow is laminar or turbulent. From a Reynolds number of 2000 to 13000, the flow transitions from laminar to turbulent, where the roughness of the vessel walls determines the amount of turbulence inside a flow [12].

The flow is restricted to only laminar flow profiles to simplify the simulations' hemodynamics. The following subsections will explain the axial flow profiles in detail in this thesis.

2.1.1 Hydrodynamic entrance region

Blood flow at the inlet of vessels slowly develops along a boundary layer until the flow becomes a steady Hagen Posseuile flow:

$$u(r) = \frac{G}{4\mu}(R^2 - r^2) \tag{2}$$

Where u is the axial flow velocity, r is the radial position [m], G is the pressure difference along the pipe $[Pa/m]$, and R is the vessel's radius [m].

The following equation calculates the distance where a laminar flow becomes fully developed:

$$L_h = 0.0575ReD \tag{3}$$

Where L_h is the entrance length [m], Re is the Reynolds number (1), and D is the vessel's diameter.

For large vessel diameters, the entrance length becomes substantial. An entrance length that is too short for a fully developed flow was used for the setup in this thesis, so an axial flow equation is necessary for an undeveloped flow. For this undeveloped flow, the following coupled equation holds:

$$U(\xi,\eta) = \begin{cases} U_c(\xi,\eta) = U_0(1-\eta^2) + \frac{D_U}{32}(1-\eta_\delta^2(1-ln(\eta_\delta^2)))\eta^2 \\ U_w(\xi,\eta) = U_c(\xi,\eta) + \frac{D_U}{32}\left[\eta_\delta^2(1-ln(\eta_\delta^2) - \eta^2 + \eta_\delta^2 ln(\eta^2))\right] \end{cases} \tag{4}$$

Where U is the non-dimensionalized axial flow velocity, U_c, U_w are the flow velocities split by the non-dimensionalized radius η_δ, U_0 is the centerline velocity and the variables, D_U is the gradient of the centerline velocity with respect to the axial distance, and ξ, η are the non-dimensionalized radius, and axial distance, respectively. The derivation of this equation is shown in A.1.

The resulting flow profiles inside the entrance region and fully developed region are shown in Figure 1.

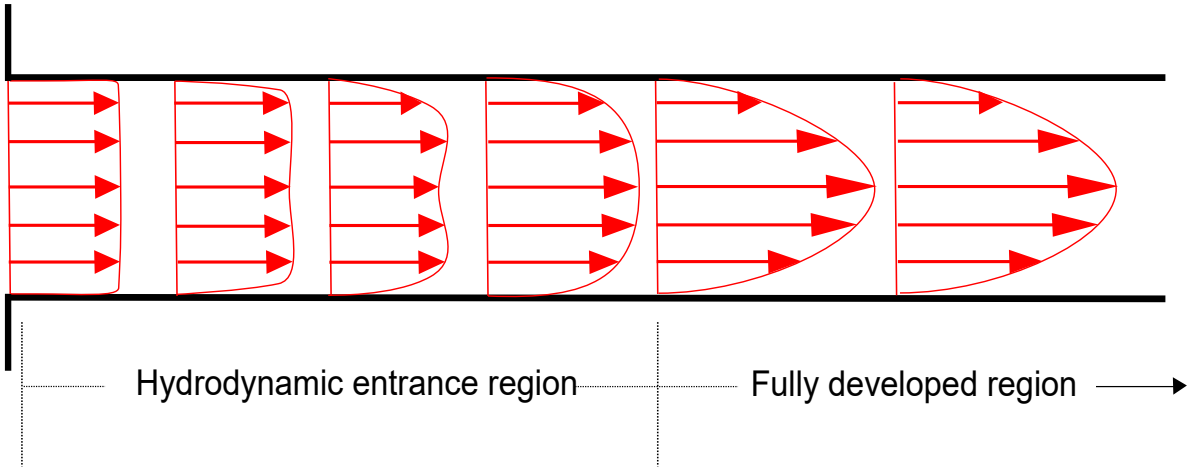

Figure 1: Flow profiles in the hydrodynamic and fully developed flow regions. The flow profile behaves like a plug flow at the tube's inlet. As the flow progresses, it develops into a fully developed flow along the boundary layer. The overshoot at the sides of the flow can be observed in the hydrodynamic entrance region.

2.1.2 Womersley and curved flow

The pressure inside a vessel fluctuates due to the pulsation of the hearth, resulting in a Womersley flow. Additionally, when vessels bend, it can slightly alter the flow. Both of these effects on the axial flow profile contribute additional terms to the Hagen-Poiseuille flow:

$$\begin{aligned} u(r,t) =& \frac{G}{4\mu}(R^2 - r^2) + \mathrm{Re}\left\{\sum_{n=1}^{N} \frac{iP_n}{\rho n\omega}\left[1 - \frac{J_0(\alpha_0 \frac{r}{R})}{J_0(\alpha_0)}\right] e^{in\omega t}\right\} \\ &+ \lambda \mathrm{Re}\left\{\sum_{n=0}^{N} \frac{imP_nR}{2\rho n\omega}\left[\frac{J_1(\alpha_0 \frac{r}{R})}{J_1(\alpha_0)} - \frac{rJ_0(\alpha_0 \frac{r}{R})}{J_0(\alpha_0)}\right] e^{in\omega t}\right\} \end{aligned} \tag{5}$$

Where P_n is the applied pressure [Pa], i is the imaginary number, n is a positive integer ranging over all harmonics, $\lambda = R/R_c$ is a scaling term that is influenced by the curvature, where R_c is the radius of curvature, ω is the frequency of the first order harmonic [rad/s], J_0, J_1 are the zeroth and first-order Bessel functions respectively, t is the time of the pressure wave [s], and $\alpha_0 = \sqrt{i^3\alpha^2}$, where α is the Womersley number. The derivation of this equation is shown in A.1.

The resulting flow profiles with varying Womersley numbers, curvature, and time are shown in Figure 2.

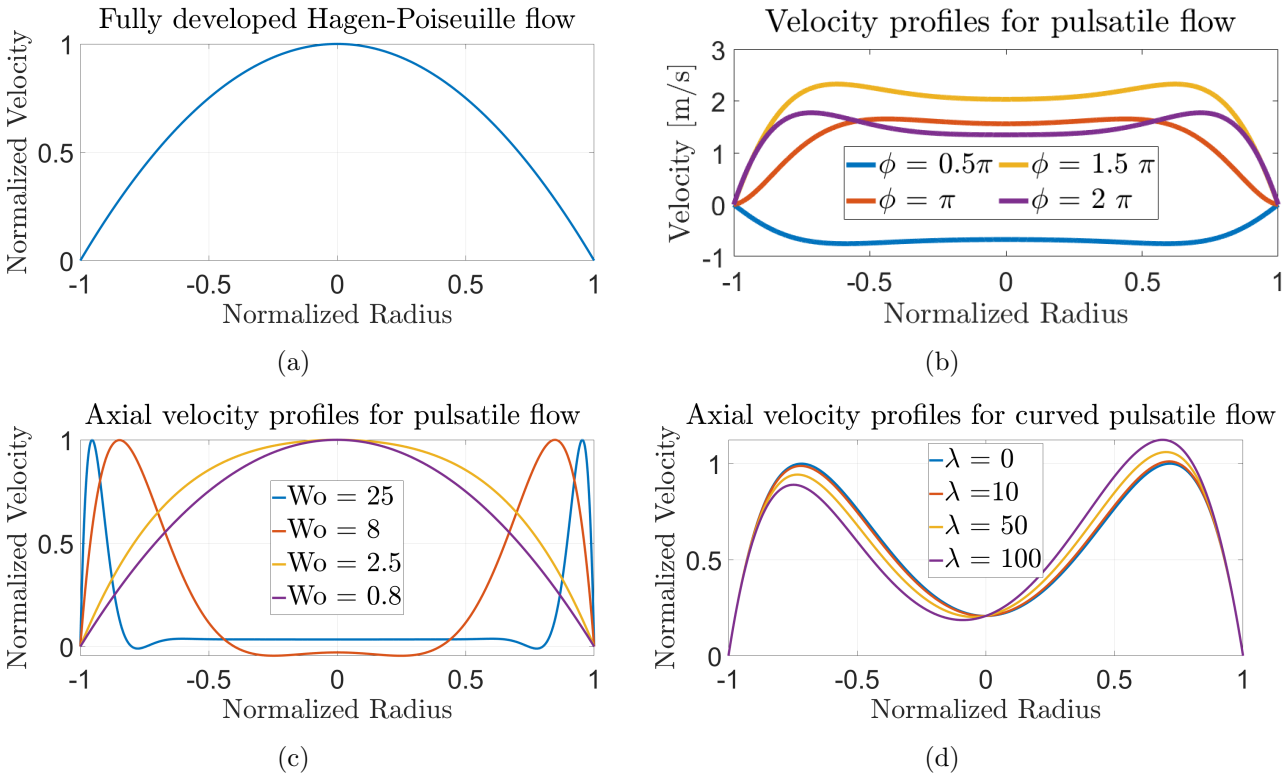

Figure 2: Hemodynamics a) A flow profile according to the fully developed Hagen-Poiseuille flow. b) Different flow profiles for a pulsatile flow, at different phases of the applied pressure. c) Different flow profiles for a pulsatile flow, where it is shown that for large values of the Womersley number, the flow is unsteady. While at low values, the flow becomes steady and behaves like the Hagen Poiseuille flow of a). d) The flow of a curved pulsatile flow. The Womersley number is fixed at five while the curvature influence is increased. The centre of curvature is defined on the left side of the tube, indicating that the curved flow increases the flow on the right side of the profile and decreases the flow on the left.

2.2 Simulators

The thesis aims to develop a neural network to predict blood flow in vivo. This requires a physically realistic dataset. However, acquiring a training dataset consisting of ultrasound reconstructions of blood flow through vessels in an experimental way is difficult due to the lack of a ground truth velocity field. Therefore, a realistic acoustics simulator is necessary. There are a few open-source simulators available for medical ultrasound flow imaging. Heiles, Chavignon, et al. [13] created a simple flow simulation where flow through vessels is simulated by a Hagen-Poisseuille flow (31). In this flow, point-like scatterers are randomly placed inside a tube, and the movement is simulated through the streamlines based on the position of the scatterer in the tube. The radio frequency data from the scatterers are simulated by the linear acoustics Verasonics Research ultrasound Simulator (Verasonics) to get the beamformed images.

Lerendegui et al. [14] developed a simulator called BUbble Flow Field, where blood flow is simulated by a rigid straight tube containing a Hagen-Poisseuille flow. This simulator estimates a vessel network as a system of interconnected rigid tubes. This simulator improves on the previous simulator by simulating the signal from the point-like scatterers as nonlinear microbubbles using a Runge-Kutta solver for a modified Rayleigh-Plesset equation [15].

Belgharbi et al. [16] again uses the steady state parabolic flow for medical ultrasound simulations, but now a realistic vessel network is used for the simulations. Two-photon microscope images of the vasculature network of a mouse brain visualize this network. Even though this is an improvement from the previous two simulators, this simulator does not account for nonlinearity from the bubble signal or the nonlinear wave propagation. The simulator PROTEUS is another alternative and can simulate realistic vasculature environments with nonlinear bubble dynamics and nonlinear ultrasound wave propagation. This simulator is explained in the next section [17].

2.2.1 PROTEUS

PROTEUS is a tool for simulating medical ultrasound imaging with contrast agents. The simulator consists of four modules that generate the ultrasound data. These modules are shown in Figure 3a.

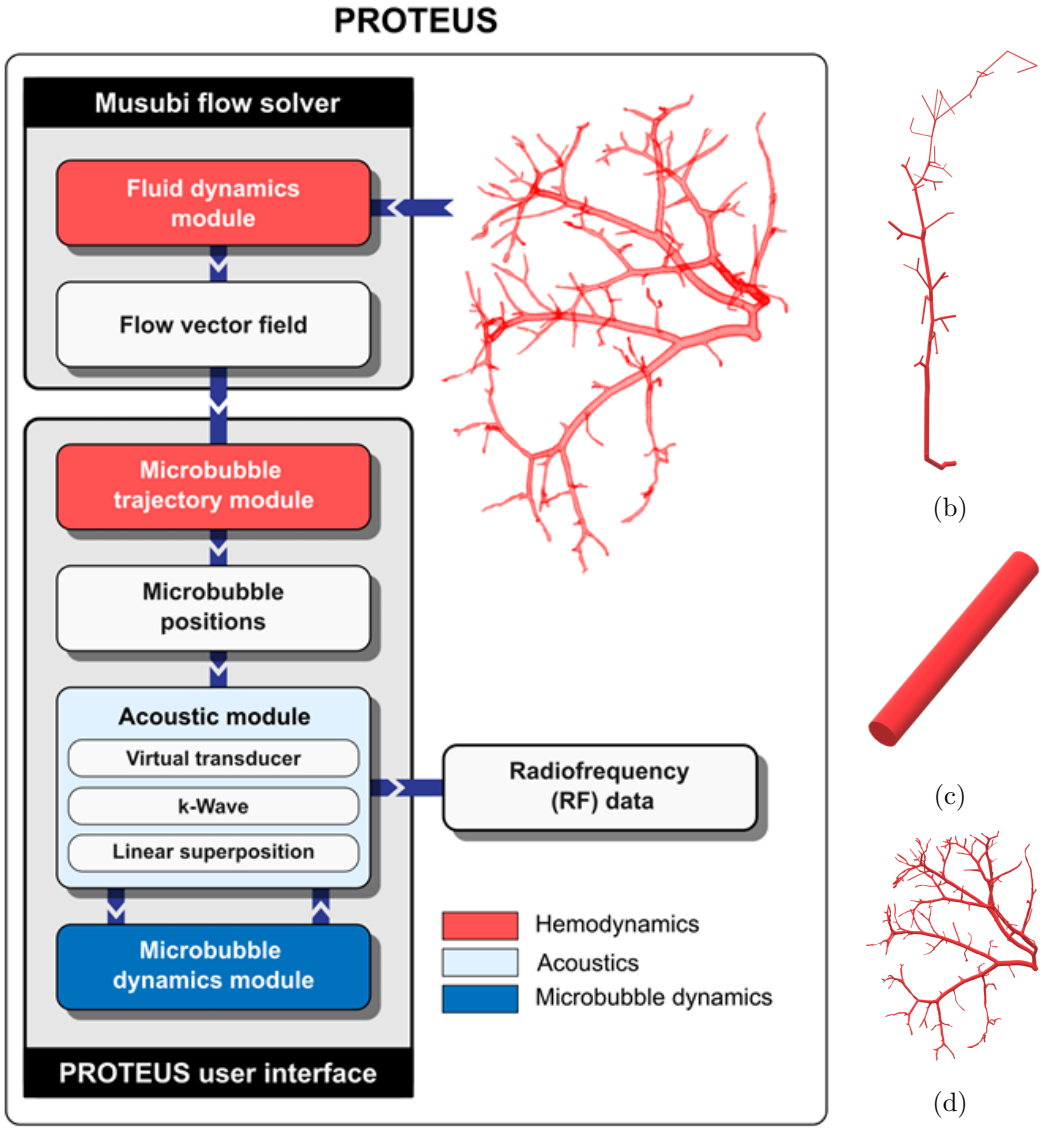

Figure 3: The simulation modules and geometries from Proteus reproduced from [17].a) The architecture showing how the modules interact in the ultrasound environment. b) The geometry of a mouse brain. c) The geometry of a 2cm diameter pipe. d) The geometry of a vasculature network.

The first is the fluid dynamics module, which uses computational fluid dynamics (CFD) simulations to create a flow field. These simulations are done by the Musubi flow solver [18], which solves the Navier-Stokes Equations using the lattice Boltzmann method in various geometries. These CFD simulations are computationally expensive and are thus carried out on the supercomputer Snellius, where three different geometry surfaces are used as boundary conditions. These geometries are a rat renal vasculature network, a mouse brain arteriole, and a rigid cylindrical tube, and are shown in Figure 3b,c and d.

The next module, the microbubble trajectory module, randomly places simulated microbubbles within the flow field and calculates streamlines using the MATLAB ODE23 solver. During the simulation, the microbubbles that flow outside the geometry are randomly repositioned at the inlet of the flow geometry to maintain a consistent number of microbubbles within the geometry.

For the simulations, the P4-1 transducer (Philips ATL) is chosen. The transmit and receive response is experimentally calibrated to model the incoming driving pressure on microbubbles. The acoustic module contains these values and sends the values to the transducer. The microbubble dynamics module calculates the radial response of these microbubbles by solving the modified Rayleigh-Plesset equation using the MATLAB ODE45 solver. The radial response is then returned to the acoustic module to obtain the final radiofrequency data, which the Delay And Sum algorithm can reconstruct to produce ultrasound B-mode images.

PROTUES lacks available flow geometries and is computationally expensive. Hence, a new simulator is presented in section 3 that adds on the previous simulators by incorporating curved pulsed flow profiles to simulate the hemodynamics.

2.3 EchoPIV analysis

2.3.1 SVD filtering

In echoPIV analysis, it is necessary to distinguish the signal from microbubbles, from the static signal of the surrounding tissue. This enhances the contrast and makes cross-correlation more reliable. After ultrasound image reconstruction on a series of frames, a singular value decomposition (SVD) filter can distinguish between the signals from moving scatterers and the static reflections from vessel walls.

The individual frames are arranged in a Casorati matrix, where the pixels over time are stacked in a single 2D matrix with dimensions n_p, n_t. Here, n_p represents the total number of pixels in a frame (the features), and n_t represents the total number of frames (the observations). SVD is then performed on the Casorati matrix, decomposing it into three matrices: a spatial matrix U, a diagonal matrix S, and a transposed spatial matrix V, as illustrated in Figure 4.

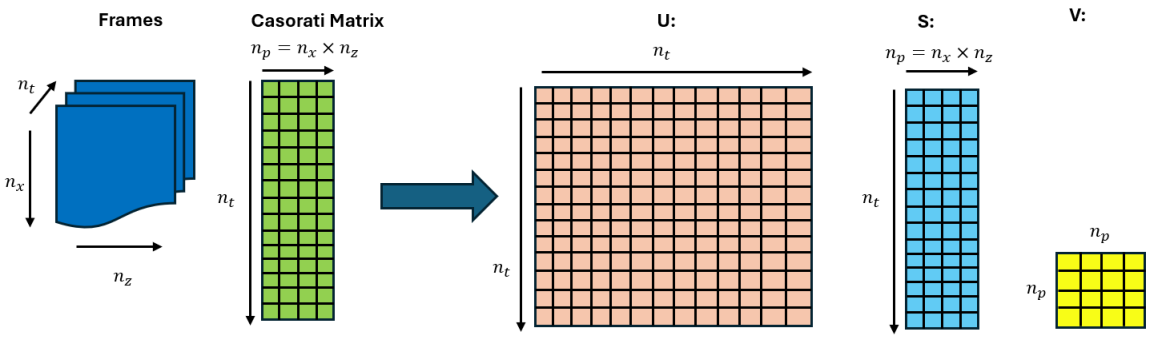

Figure 4: Schematic of the different matrix in SVD. The pixels of the initial frames get stacked in the Casorati matrix by the observations (over time) and the features (all pixels). The Casorati matrix gets decomposed into three matrixes, U, S, and V: $Casorati = USV^T$. The matrix S contains the principal components in a diagonal matrix where the variance of the observations is ordered for every feature in decreasing order.

The Casorati matrix gets decomposed in such a way that the following equation holds:

$$\text{Casorati Matrix} = USV^T \tag{6}$$

Where U is a purely temporal matrix $[n_t, n_t]$, V is a strictly spatial matrix $[n_p, n_p]$, and S is a diagonal matrix that contains the principal components in the order of their strength $[n_t, n_p]$. These principal components are arranged based on their variance, with stronger components representing lower variance in the observations and weaker components representing higher variance.

In ultrasound measurements, the strong components are associated with static regions in an image, such as vessel walls, while the weak components correspond to noise. We can obtain a signal containing the moving scatterers by filtering out these components and recalculating the Casorati matrix by equation (6). This Casorati matrix can be transformed back into a video by resizing the matrix to the original video size.

2.3.2 Offline temporal filtering

An alternative to the SVD filter is to filter on the envelope extracted pixel intensities over time. Pixel intensities over time will have different characteristics in different image regions. The pixels outside the vessel only contain noise. In contrast, the pixels in the vessel walls include a constant intensity alongside noise, and the pixels inside the vessels contain a fluctuating signal corresponding to the bubbles flowing over the pixel. The different signals can be seen in figure 5.

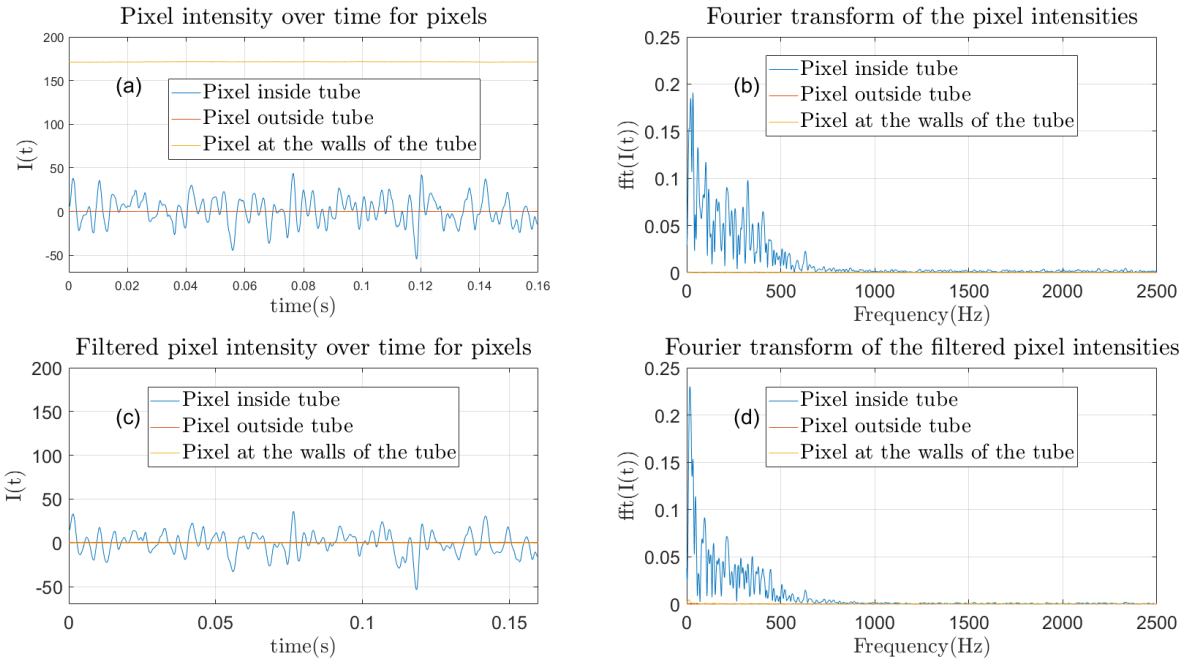

Figure 5: Example of offline temporal filtering. a) Unfiltered pixel intensities over time in three different regions, at a static intensity over time (at the walls of the tube), at a region that only contains noise (outside the tube), and at the intensity of moving scatters (inside the tube). b) Fourier plot of the pixel intensities of the three different regions. c) Filtered signal over time of the intensities of the three different regions. d) Fourier plot of the filtered intensities

It is wanted to remove the signal from the static regions, which can be done by first taking the average of this signal. After subtracting the average of the signal, the Fourier transform can be taken for a pixel over time. Figure 5b shows the Fourier transform of the pixels in different regions. Here, high frequencies correspond to high temporal movement, while low frequencies correspond to low temporal movement. There is almost no movement for the static parts of an ultrasound image (the tissue), corresponding to low frequencies. High frequencies correspond to noise where the intensity rapidly changes. Using a bandpass filter on the high and low frequencies results in only the signal of the microbubbles. The exact frequency boundaries are chosen by knowing that a moving speckle crosses a pixel in a certain amount of time T_b related to its velocity v. Signals that are longer than this amount of time correspond to static regions and smaller signals than this time corresponds to noise. The equation to relate the frequency to the moving speckles is:

$$f_s = \frac{1}{T_b} = \frac{v_{speckle}}{s_{speckle}} \tag{7}$$

Where f_s is the cutoff frequency [Hz], $v_{speckle}$ is the velocity of the moving speckle [m/s], and $s_{speckle}$ is the size of the speckle [m].

Knowing the lower and higher bounds for the velocities give the cutoff frequencies.

2.3.3 PIV analysis

After post-processing of the ultrasound data, the frames are ready for velocimetry. A technique used to get a velocity field out of two consecutive frames is particle image velocimetry (PIV). Figure 6 shows a schematic of the different steps involved in PIV analysis.

With PIV, the frames are divided into smaller subsections of only a few pixels. A subsection of frame two then slides over the corresponding subsection of the first frame, and at every position, the following equation calculates the cross-correlation:

$$C(s_x, s_y) = \sum_x \sum_y I_1(x, y) I_2(x + s_x, y + s_y) \tag{8}$$

Where C is the cross-correlation for a given displacement, I_1 is the subsection of the first frame, I_2 is a small subsection of the second frame, and s_x and s_y are the displacements in the x and y directions, respectively.

This function for the cross-correlation has its maximum value on s_x, s_y when the smaller sub-image of frame two overlaps as well as possible with frame one. The local velocity is then given by:

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} s_x \\ s_y \end{bmatrix} \times \frac{1}{\Delta t} \tag{9}$$

Where u_x, u_y is the velocity at the centre of the subsection of frame 1, and Δt is the time between the two consecutive frames.

After extracting the velocity for the subsection, the window is shifted, and a new window is chosen. The cross-correlation again extracts the velocity, which is repeated for both frames.

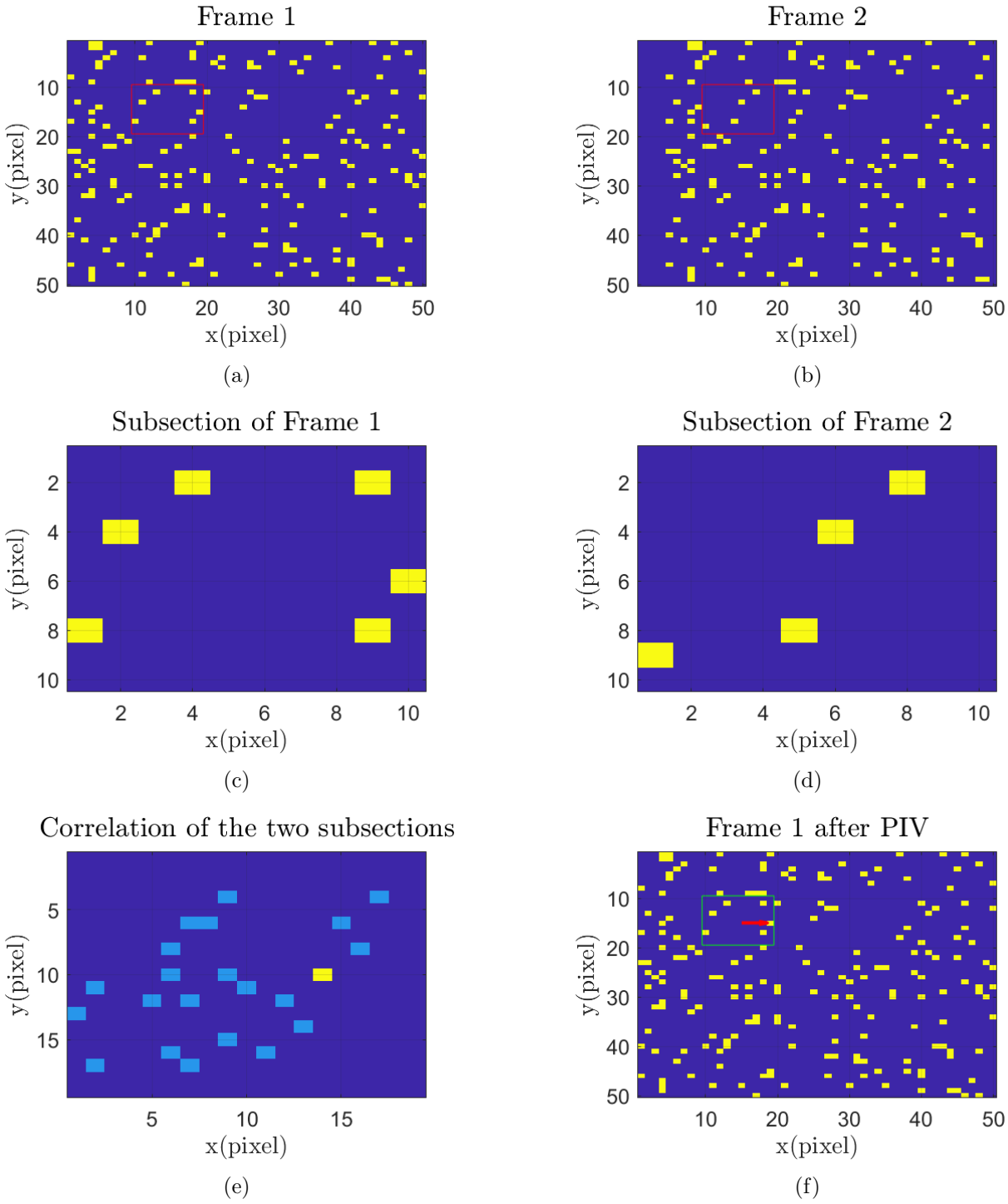

Figure 6: For PIV analysis, two consecutive frames shown in a and b are analyzed for the velocity profile. c), d) show a subsection of frames one and two for which the cross-correlation is calculated. e) Shows the resulting cross-correlation matrix where the maximum cross-correlation is given as a high value, here in yellow. f) From the cross-correlation, a vector is drawn and placed on frame one, showing the velocity of the small subsection. Doing this analysis for different windows gives the full velocity profile.

2.4 Machine Learning

An alternative to extracting the velocity field out of ultrasound flow images instead of echoPIV is to use machine learning. For this thesis, it was chosen to restrict to convolutional neural networks. These types of networks have succeeded in all kinds of image classification challenges. The most common type of a convolutional neural network (CNN) is a deep CNN. The original deep CNN is called AlexNet [19]. This network aims to map an image towards a classification vector probability distribution. Consecutive convolutional and max pooling layers downsize the image towards a single array of probabilities. This network is improved using a visual geometry group CNN (VGG) [20]. This network again downsamples an image by convolution and max pooling layers. This network gradually decreases an image into a linear array containing classification probabilities. The VGG network is also used for image segmentation applications [21]. The network uses two VGG networks, where the first network downsamples an image into channels. These channels are then passed by a reversed VGG network, where deconvolution and unspooling layers upsample the channels towards different network features. The resulting representation contains the original image size and channels containing different segmentation classes' probabilities.

The VGG segmentation network was enhanced by Ronneberger et al. [22] through the addition of skip connections between the convolutional (encoder) and deconvolutional (decoder) layers, which resulted in the development of a structure known as U-net. This network outperforms previous networks because the skip connections allow a much deeper network to be trained because there is an identity path to go back through [23]. An alternative to the U-net is a stacked hourglass network [22]. This network consists of different encoder decoder blocks, similar to the U-net. The difference with the U-net is that the skip connections do not concatenate the input towards the output, but sums both representations.

The network used in this thesis is called a Residual U-net (ResU-net) [22]. This network contains the same architecture as the U-net but improves stability by changing the convolutional layers into residual blocks. These blocks contain the convolution layers with an overlapping skip connection. These extra skip connections improve performance and stability by smoothening the loss surface [23].

3 Methods

The previous section explained the background on blood flow, simulators, offline filters, and the velocimetry techniques echoPIV and machine learning. This section uses the background to show how the neural network is implemented and validated on experimental data. The method to obtain the simulated data to train the neural network is elaborated on in subsection 3.1.1, and the architecture of the network itself is explained in subsection 3.2. Furthermore, the methods of the in vitro experiments are presented in subsection 3.2.5. Finally, the methods to estimate the velocity fields from ultrasound data for both echoPIV and machine learning are presented in subsection 3.2.9 and 3.2.10.

3.1 Training and validation dataset

3.1.1 PROTEUS

A new script was added to the PROTEUS source code to create a diverse training dataset by modifying various parameters for simulating blood flow through a rigid tube. The flow geometry involves a macroscopic cylindrical pipe with an inner diameter of 2 cm [24]. The flow through this pipe is a fully developed Hagen-Poiseuille flow (31) with a Reynolds number of 5000. Even though the Reynolds number suggests the flow should be in the laminar to turbulent transition, the CFD simulations show that the flow is still a fully developed laminar flow as long as there are no distortions in the geometry of the pipe.

To change the simulations' parameters, an initial parameters file is saved using the PROTEUS GUI, and then specific values are randomly adjusted in the geometry and GUI parameters file. This includes changing the velocity magnitude, inner diameter of the pipe, starting depth, rotation of the pipe, number of microbubbles, and flow direction.

These changes to the pipe and flow are related to different vessel types, ensuring the network is more robust to in vivo measurements. The other rotations and directions of the flow ensure that the transducer does not have to be perfectly aligned with the vessel, which again increases robustness. Table 1 lists the fixed and changed parameters. 250 different PROTEUS simulations are performed on an external server with randomly chosen values from the parameter space. Each simulation contains five different frame couples with their corresponding velocity field. This results

in a total of 1250 frame pairs.

Fixed Parameters	Value	Randomly Adjusted Parameters	Parameter Space
Transducer	P4-1	Maximum Velocity	(0.4-1.6) m/s
Center Frequency	2.5 MHz	Inner diameter pipe	(7.5-25) mm
Number of cycles	2	Starting Depth	(10-50) mm
Focus	Inf	Number of microbubbles	10.000-15.000
Type Bubbles	SonoVue	axial orientation	(-50,50)°
Frame Rate	500 Hz	elevational orientation	(-15,15)°
Number of Frames	20		
Surrounding Tissue	General Tissue		
Medium inhomogeneity	2%		
Pipe flow	blood flow		
Sampling Rate	52 MHz		
axial domain	(-1.2,80) mm		
lateral domain	(-15.4,15.4) mm		
elevational domain	(-9.2,9.2) mm		

Table 1: This Table shows the fixed and changeable parameters of PROTEUS.

3.1.2 2D Simulator

Most vessels are not perfectly straight and do not have a perfect Hagen-Poiseuille flow. More complex flow profiles and geometries are necessary to get a diverse training set to train a neural network. These flow geometries can be made using CFD simulations and implemented in PROTEUS. However, setting up new CFD simulations is time-consuming. The solution to bypass the CFD simulations is to create a new simulator that implements the analytical solution to more complex flows, namely curved pulsatile flow profiles. Another issue of PROTEUS is that the simulator has a significant computational time due to the simulator acting in a 3D environment. Hence, the new simulator acts in a 2D environment, trading physical accuracy for computational speed.

Static RF lines

The general pipeline of the simulator is shown in Figure 7. The first step of this simulation is to define a grid where linear scatters are placed randomly. The grid has a width of 30.70 mm and a depth of 80 mm, with a pixel size of 0.1232 mm in both the lateral and axial directions. The scatterers start at a depth of 1.8 mm to ensure that the scattered signal is still visible on the RF lines.

From this static image, an ultrasound reconstruction is made by simulating how the P4-1 transducer (Philips ATL) images these linear scatterers. For this, the time it takes for an ultrasound

plane wave to travel from the transducer toward the scatterer and back to an element is calculated using the following equation:

$$dt = \frac{z_p + \sqrt{z_p^2 + (x_{el} - x_p)^2}}{c} \tag{10}$$

Where dt is the return time of the scatterer [s], z_p, x_p is the axial and lateral position of the scatterer respectively [m], x_{el} is the axial position of the element [m], and c is the speed of sound in the medium [m/s].

For every element, an empty array with a time of 0 to 3×10^{-3} s is made, which acts as an empty RF line. These lines are filled with the reflected pulse signals by adding the inverse transmit pulse from the P4-1 transducer, simulated by PROTEUS [17], to these lines for every scatterer.

Moving scatterers

When the RF lines are filled for all scatterers, they are saved in a matrix for every frame. Then, the scatterers are moved by a flow field acting on the entire grid. In this flow field, a tube contains a laminar (31) or pulsatile (48) axial velocity profile. For every point outside this tube, the velocity is set to 0. The tube is made around a center line on the grid. From this line, a radius and radius of curvature are defined. The u and v components of the velocity field are calculated on every grid point inside the tube. This is done by extracting the angle and direction of a grid point in contrast to the center of the curvature, as shown in Figure 7. The magnitude of the axial flow is calculated based on the relative position within the tube's curvature. This magnitude is then converted by calculating the relevant u and v components from the point's angle to the curvature's centre.

Since the tube is simulated as an inner and outer circle, the circles may cross the imaging grid twice, or the circles may not cross the upper or lower edge of the grid at all. To prevent this from happening, there is a limit on the radius of curvature to make sure it is of sufficient length and that it falls inside the grid. The inlet conditions of the field are also extracted from the tube by knowing the boundaries of the inner and outer circles at the upper and lower edges of the grid.

For every scatterer, the displacement is calculated by extracting the velocity from the flow field, where linear interpolation is used for velocities between grid points. The new grid point from the original position and velocity follows from the fourth-order Runge-Kutta method, which accuracy ensures that scatterers stay on their streamlines.

After the first full rotation over all scatterers, the RF contribution of the stationary scatterers is saved as a background signal. Only the RF contribution for the moving scatterers is calculated and added to the background RF signal for the next frames. This decreases the computational time significantly and allows the user to choose whether to use the background signal in the final reconstruction.

Image reconstruction

After the RF lines are calculated for all elements and frames, these lines can be reconstructed in a B-mode image. First, the signal envelope is extracted from the RF lines by applying the absolute Hilbert transform over the signal per element. Then, the Delay And Sum algorithm (DAS) converts the RF lines from a temporal space to a spatial image. This algorithm calculates the time delays from equation (10) for every pixel and every element. The values for the time delays can be looked up in the RF signals for all elements and these values are added to get the final value. The algorithm used for this simulator is based on the faster DAS method from [25], where equation (10) is used to construct a large sparse DAS matrix containing the hyperbolas resulting from equation (10). This DAS matrix is multiplied by the envelope-extracted RF data to get the beamformed RF signals:

$$RF_{bf} = M_{DAS} \times abs(Hilbert(RF)) \tag{11}$$

Where the RF data is stacked by its columns to get a $[n_s \cdot N_e]$ array, where n_s, N_e are the number of samples and elements, respectively. The DAS matrix contains a shape of $[M \cdot n_s \times N_e]$, where M has a size of $[n_x \times n_z]$, the total number of pixels in the image. The resulting matrix multiplication gives a stacked column matrix. Rearranging this matrix in its n_x, n_z dimension gives the final beamformed image. The final step is to log compress the image using the equation below, resulting in the final image of Figure 7.

$$IMG = 20 * Log_{10}\left[\frac{RF_{bf}}{max(RF_{bf})}\right] \tag{12}$$

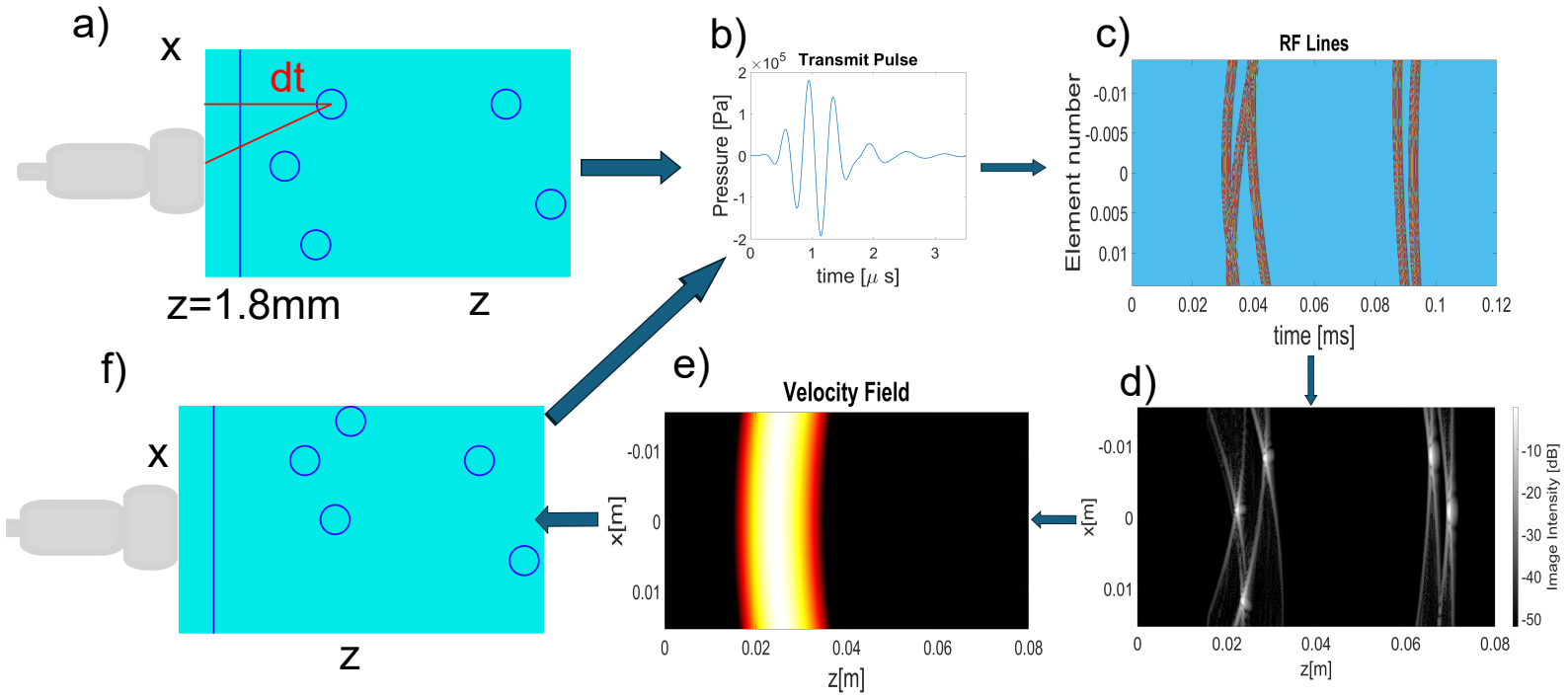

Figure 7: The pipeline of the 2D simulator for a single sample. a) First, linear scatterers are randomly placed on a 2D grid. For all these scatterers, the echo arrival times are calculated. At these time differences, the transmit pulse signal of the P4-1 transducer is placed. b) The transmit pulse of a P4-1 transducer. c) All pulse signals are summed for every element, resulting in the RF lines. d) The calculated RF lines are transformed into a B-mode image using the Delay and Sum algorithm. e) A velocity field is applied on the grid, where every grid point contains a velocity with an u and v component. f) The scatterers are translated by the fourth-order Runge-Kutta method according to the velocities of the flow field. After the translation, the simulation calculates the new RF lines and continues in a loop until all frames are reconstructed.

Simulator dataset 1000 simulations are done on data where both the RF lines of the moving and static scatterers are saved, and 1000 samples are done where only the RF data of the moving scatterers are saved. Five different frame couples, alongside their velocity field, are saved for each sample. The chosen parameters for each sample are randomly chosen out of Table ??.

3.1.3 Data splitting

From PROTEUS, 250 samples are saved, whereas, for the 2D simulator, 2000 samples are saved. Both samples from the simulators are split equally among the validation and training sets such that the network does not overfit to any of the simulators while validating on the other. The training set itself consists of 75 % of all samples, and the validation set contains 25 % of all samples.

Fixed-Parameter	Value	Randomly adjusted Parameter	Parameter space
Transducer	P4-1	Maximum Velocity	(0.4-1.6) m/s
Center Frequency	2.5 MHz	Inner diameter pipe	(7.5-25) mm
Number of cycles	2	Starting Depth	(23.1-56.9) mm
Focus	Inf	Number of microbubbles	4500 - 7000
Type Bubbles	Linear Scatterers	Lateral center of curvature	(-7.68,7.68) mm
Frame Rate	500 Hz	Radius of Curvature	(3.5,100) mm
Number of Frames	10	Probability for Pulsatile flow	50%
Surrounding Tissue	Linear Scatterers	Probability for Hagen-Poiseuille flow	50%
Sampling Rate	52 MHz	Pulsatile phase	(0,2pi) rad
axial domain	(-1.2,80) mm		
lateral domain	(-15.4,15.4) mm		
elevational domain	(-9.2,9.2) mm		
Type of flow	water flow		
speed of sound	1540		
Dynamic viscosity	0.0045		

Table 2: This table shows the tabled and changeable parameters of the 2D simulator.

3.2 Machine Learning

3.2.1 Architecture of the Residual U-net

In this thesis, the chosen network architecture is a Residual U-net (ResU-net) style network, whose architecture is shown in Figure 8. This choice favoured the common U-net due to the enhanced stability provided by residual connections [23].

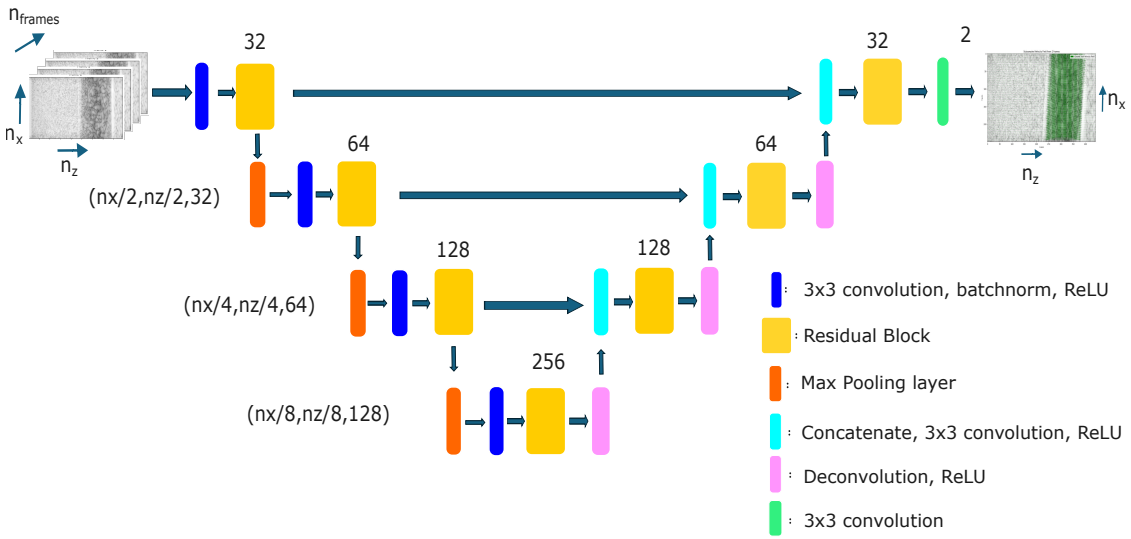

Figure 8: Architecture of the ResU-net. A stack of input frames of dimension $[n_x, n_z, n_{frames}]$ can be chosen by the user, where n_z are the number of pixels in the axial direction, n_x are the number of pixels in the lateral direction, and n_{frames} are the number of input frames. These frames get downsampled in an encoder to extract the features. These features are then upsampled by a decoder towards the u and v components of the corresponding velocity field. Between the decoder and encoder, there are different skip connections where the representation x from the encoder is concatenated with the features of the decoder.

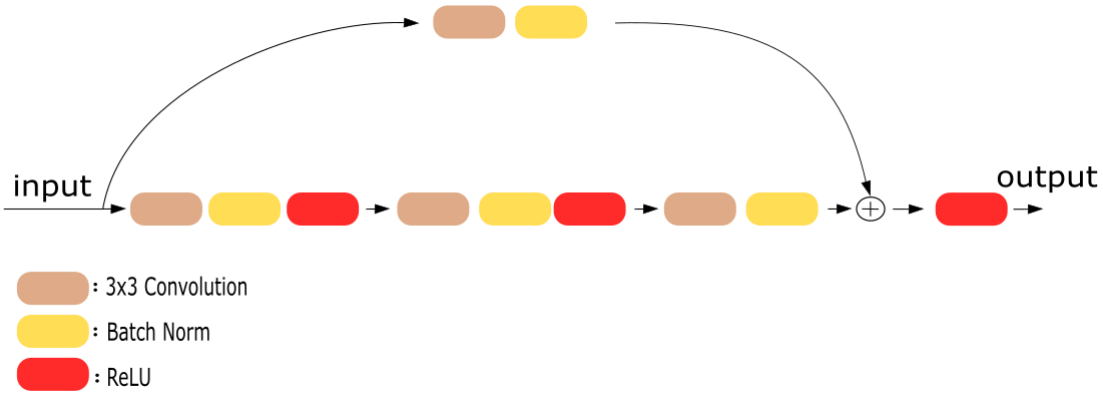

Figure 9: A sketch of the residual block. Here, the input goes through different convolution, batch norm, and ReLU layers with a convolution-based skip connection at the top.

The network has an encoder-decoder type of architecture involving various pooling and convolution layers to downsample input frames. The number of input ultrasound frames is user-selected and is concatenated as input features. By a 3x3 convolution layer, the features are increased to a number of 32 features, after which the features are passed through a residual block, whose architecture is shown in Figure 9. Within this block, the input goes through three convolution layers, with a convolutional skip connection over the block. A batch normalization layer is added after each convolution to improve stability [23]. The output is then downsampled using a max pooling

layer, reducing the input dimensions by half. From here, the features go through the same structure until they reach 256 features. The downsampled frames are passed through a decoder, where a deconvolution layer upsamples the frames. After every deconvolution, a skip connection combines the input from the corresponding encoder layer with the upsampled frames. This combination is done by concatenating and using a 3x3 convolution layer. These steps are repeated until the output reaches the exact input dimensions. A final convolution layer then estimates the planar velocity field consisting of the u and v components of the field.

3.2.2 Loss Functions

The neural network aims to predict velocity fields from ultrasound frames that are as similar as possible to a ground truth dataset. This training is done with multiple loss functions, which the neural network can use to minimize the difference between ground truth and prediction. Three different loss functions are used to train the network. First, a Mean Squared Error (MSE) loss function is used:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left((u_{pred,i} - u_{true,i})^2 + (v_{pred,i} - v_{true,i})^2\right) \tag{13}$$

Where N is the total amount of data in a single batch, $u_{true,i}, u_{pred,i}$ are the ground truth and predicted axial components of a vector, and $v_{true,i}, v_{pred,i}$ are the ground truth and predicted lateral components of a vector respectively.

The MSE loss function minimizes the vectors' angle and magnitude components. It is also interesting to focus only on the angle component of the vectors so that the field will generally be more aligned. Focusing on the angle loss is done by the second loss function, which is a loss function that calculates the difference in angle between the predicted and ground truth velocity field vectors. This can be done by the cosine similarity:

$$\textit{Cosine Similarity} = \cos(\theta) = \frac{\mathbf{y_{pred}} \cdot \mathbf{y_{true}}}{||\mathbf{y_{pred}}||\ ||\mathbf{y_{true}}||} \tag{14}$$

Where θ is the angle between a vector from the prediction (y_{pred}) and ground truth (y_{true}), the cosine similarity gives a value between -1 and 1, where a value of -1 is for two vectors that are opposite to each other, 0 is for vectors that are orthogonal to each other. A value of 1 is for vectors

that are aligned with each other. The goal for the neural network is to have all vectors aligned, which results in the following loss function:

$$L_\theta = \frac{1}{N}\sum_{i=1}^{n}\left(\frac{1-\cos\theta}{2}\right)\cdot||\mathbf{y_{true}}||^{1/5} \tag{15}$$

Here, the loss value of aligned vectors is 0, while unaligned vectors give values between [0,1]. The loss of individual terms is multiplied by the magnitude of the ground truth vectors to the power 1/5. This is done so that the angles of smaller vectors influence the total loss function more, which increases accuracy at near-wall velocities.

The final loss function used in this thesis is based on the continuity equation of an incompressible fluid.

$$\frac{du}{dx}+\frac{dv}{dy}+\frac{dw}{dz}=0 \tag{16}$$

Where $\frac{du}{dx},\frac{dv}{dy},\frac{dw}{dz}$ are the velocity gradients in axial, lateral and elevational directions, respectively. This equation is used to construct the continuity-based loss function, where the term in the elevational direction is taken as an error since the velocity in this direction is not given from the prediction.

$$L_{con}=\frac{du}{dx}+\frac{dv}{dy} \tag{17}$$

This loss function contains small values when vectors are correctly aligned, so it is expected to smooth the velocity field.

3.2.3 Total Loss

The total loss function is the weighted sum of the three individual Loss functions:

$$L_{tot}=4*L_{MSE}+L_\theta+\frac{1}{3500}*L_{con} \tag{18}$$

The MSE and continuity loss are scaled such that the initial loss value is approximately the same as the initial value for the angle loss. This results in all loss functions having values between

zero and one.

3.2.4 Metrics

Two different methods were used to evaluate the results. First, a Root Mean Squared Error metric is used, which is the square root of the MSE loss function:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((u_{pred,i} - u_{true,i})^2 + (v_{pred,i} - v_{true,i})^2)} \tag{19}$$

This RMSE results in an error containing the same units as the output y. Since the output y contains the u and v vectors of a velocity field, the result from the RMSE gives an average estimate of the velocity error in [m/s].

The second metric is a metric that calculates the average angle deviation between the ground truth and predicted vectors. These angles are found by applying the arccosine over the cosine similarity of equation (20):

$$\theta_i = \cos^{-1}\left[\frac{\mathbf{y_i} \cdot \hat{\mathbf{y}}_\mathbf{i}}{||\mathbf{y_i}||\ ||\hat{\mathbf{y}}_\mathbf{i}||}\right] \tag{20}$$

The mean over all these errors gives the average angle deviation:

$$Average\ Angle\ Deviation = \frac{1}{N}\sum_{i=1}^{N}(\theta_i) \tag{21}$$

3.2.5 Experimental setup

The neural network is trained and validated on simulations. However, these simulations still have limitations, like neglecting artefacts and vessel walls. To validate the network's robustness, a setup is made that replicates the conditions of the simulations as well as possible. This setup is shown in Figure 10.

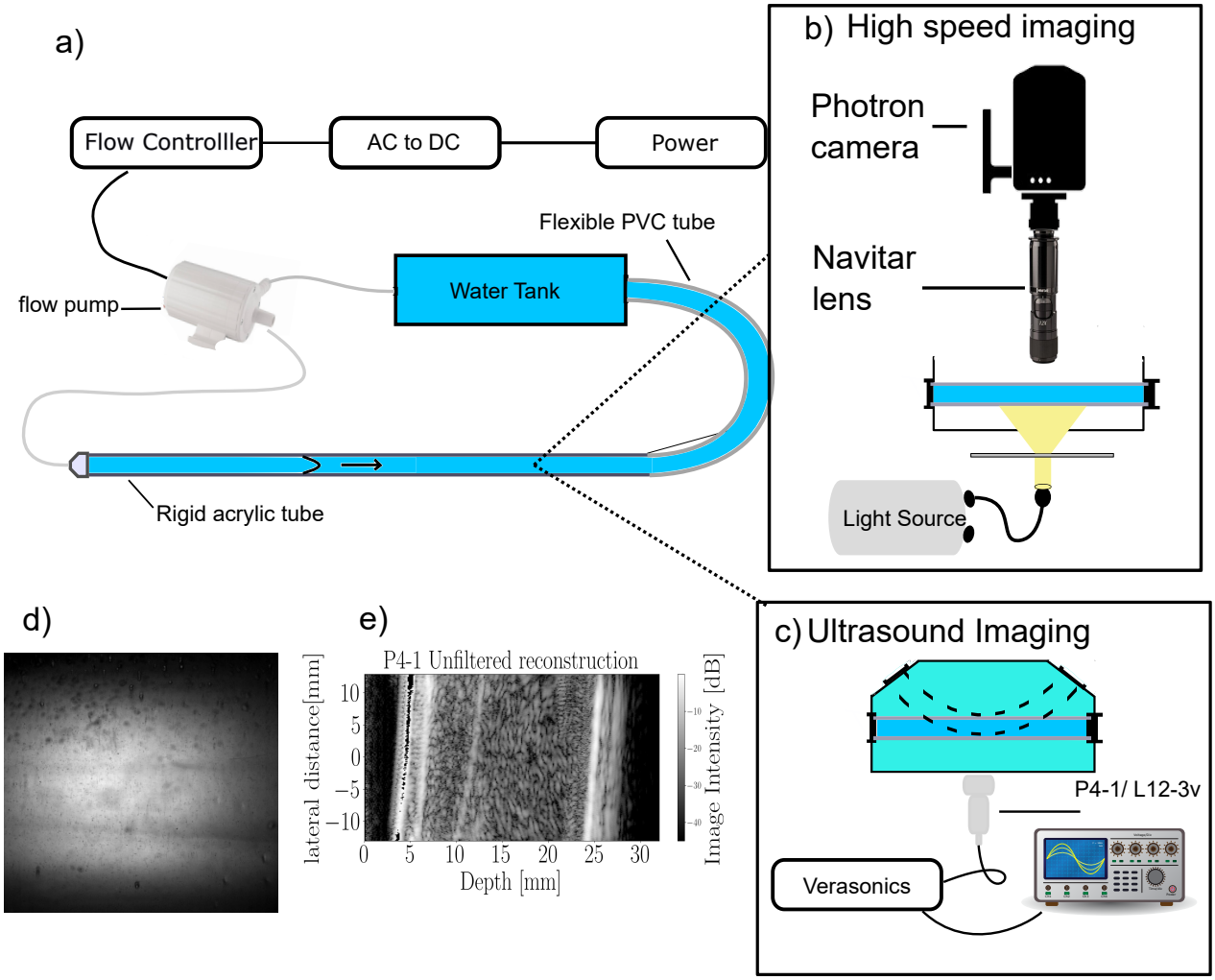

Figure 10: a) Schematic of experimental setup. b) Schematic of the imaging tank used for high frame-rate imaging. c) Schematic of the imaging tank used for ultrasound imaging. d) Image of the microbubbles inside the tube by the high-speed camera. e) ultrasound reconstruction from microbubbles inside the tube by the P4-1 transducer.

The setup consists of a transparent flexible PVC tube [20 x 26]mm immersed in a degassed water reservoir [350x200x140]mm made of acrylic. Degassing of the water is necessary to neglect the effects of surrounding gas bubbles. The PVC tube is connected to an open water tank [350x120x140]mm made of acrylic, where contrast agents can be added to the flow through the tube. The contrast agents that can be used are microbubbles produced in-house by the Physics of Fluids department (C4F10 gas contained in a phospholipid coating) or glass beads. The water tank is also connected to a Barwig 02 water pump (with a maximum flow rate of 20L/min) by a [5.5x8]mm PUR tube. The flow pump circulates the water through the setup controlled by a Barwig flow controller that sets a constant pressure (maximum of 1 bar). The water flow then flows towards a rigid acrylic tube [20x24] mm with a length of 80mm. In this entrance length, the flow develops towards a fully

developed Hagen Poisseuille flow.

The imaging reservoir can be used for ultrasound measurements and high frame-rate imaging, as shown in Figure 10c. An imaging window is placed at the long side of the imaging reservoir for the ultrasound measurements by taping a thin cling film over a rectangular hole in the acrylic reservoir. The transducer can be placed on this cling film held by a clamp on a Newport mount. An Aquasonic 100 ultrasound transmission gel is put between the transducer and film to prevent reflection from the air. The distance between the cling film and the centre of the PVC tube is 60mm. The transducer can also be placed above the tube on the water-air interface. This way, the distance between the transducer and the centre of the tube can be as small as 15mm. The P4-1 (Philips ATL) and L12-3v transducer (Philips ATL) were chosen to image the water flow. The transducer is connected to a Verasonics Vantage 256 system, where the user can select the imaging parameters using a MATLAB script. The system then controls the transducer and saves the imaging data. During imaging, the Verasonics hardware system sends a signal towards the Tektronix TBS 2000 digital oscilloscope, triggered on the transmit pulse. Figure 10e shows a reconstruction after imaging with the P4-1 transducer.

If water is left out of the imaging reservoir, the tank can be used for high-frame-rate imaging. This setup is shown in Figure 10b. Here, the tank is positioned with the top side towards a Photron NOVA S16 camera containing a frame rate of 16000 frames/s. The camera is connected to a Navitar lens with a 12x magnification. A continuous light source by a Schott KL 2500 LED is placed at the back side of the tank, emitting a beam towards a light diffuser, ensuring contrast of the contrast agents. Figure 10d shows a frame from the camera.

The imaging reservoir has four different entrances. Moving the PVC tube through different entrances creates a bent flow, allowing the measurement of curved flow profiles.

3.2.6 Average Flow

The exact flow rate through the tube is found empirically by filling a beaker for a fixed amount of time. The equation for the flow rate is shown below:

$$Q = \frac{V}{t} \tag{22}$$

Where Q is the flow rate $[m^3/s]$, V is the volume collected in the beaker $[m^3]$, and t is the total time for filling the beaker [s].

After retrieving the flow rate, the average flow velocity is calculated by the following equation:

$$\bar{v} = \frac{Q}{A} = \frac{Q}{\pi R^2} \tag{23}$$

Where $\bar{v}$ [m/s] is the average axial flow velocity, A is the cross-sectional surface $[m^2]$, and R is the radius of the tube [m].

If the velocity profile is known, this average axial flow velocity can also be extracted from the integral over the tube cross-section:

$$\bar{v} = \frac{1}{\pi R^2} \int_0^R 2\pi r v \, dr \tag{24}$$

Where v is the velocity [m/s] at a location r [m], and R is the full radius of the tube [m].

3.2.7 Data Acquisition

The Verasonics Vantage 256 ultrasound system is used to set the transducer parameters. The chosen parameters are set to match the simulations as closely as possible. These parameters are shown in Table 3.

To image at a high frame rate, the frames are not directly saved to the Verasonics system. The imaged frames are collected as buffer frames, put into a superframe, and then sent to Verasonics. This allows for a higher framerate of 5000 frames per second.

3.2.8 Data Processing

The data is stored as raw RF lines in Verasonics. This raw data is then reconstructed towards a B-mode image with a Matlab script. For this reconstruction, different parameters are chosen. These parameters are shown in Table 4.

The RF lines are reconstructed in an image for both transducers using a Hilbert transform over the RF lines. This extracts the envelope of the signal. Then, the same delay and sum algorithm is used in the 2D simulator. 3.1.2. The data is then log-compressed and cut off at the amplitude

Chosen Parameter	**P4-1 Measurment**	**L12-3v Measurement**
Image width	30mm	30mm
Image Depth	80mm	80mm
number of Super Frames	16	16
Number of buffer Frames	50	50
Total Number of Frames	800	800
Frame Time	200 μs	200 μs
Number of elements	96	128
Transmit Transducer Frequency	2.5 MHz	5 MHz
Speed of sound	1480	1480
elevation focus	80mm	20mm
Sampling Frequency	5.2e-8 Hz	5.2e-8 Hz
Transducer focus	infinite	infinite

Table 3: Table showing the chosen parameters for the P4-1 and L12-3v measurements.

Chosen Parameter	**P4-1 Measurment**	**L12-3v Measurement**
Number of Reconstructed frames	800	800
Near Field saturation	2300 Pa	500 Pa
Maximum Angle	45 degree	90 degree
pixel size	0.1232 mm	0.0423 mm
SVD lower limit	40	40
SVD higher limit	200	200
Velocity based filter lower limit	10 Hz	10 Hz
Velocity based filter higher limit	250 Hz	250 Hz
Amplitude Range	[0,-45] dB	[0,-55] dB
Width of the domain	26mm	20mm
Depth of the domain	50 mm	50 mm

Table 4: Table showing the input parameters for the ultrasound measurements.

shown in Table 4, which gives an unfiltered B-mode image.

The user can choose between a velocity-based filter and an SVD filter to filter the images. The velocity-based filter acts on the raw IQ data, while the SVD filter acts on the RF lines. The velocity-based filter's higher and lower bounds for the velocities are chosen. The expected profile is an undeveloped laminar flow, which means that the speckles have a velocity range between 0 and at max twice the average velocity of equation (23). Since it is still wanted to remove static signals, the lower cutoff frequency is chosen for a velocity of 0.05 m/s, where equation (7) gives the frequency boundary. A Butterworth filter with order four is chosen for the bandpass filter. This is a soft filter, meaning that the frequencies from a velocity of 0 m/s are damped instead of entirely removed. This ensures that speckles with lower velocity do not completely vanish from the resulting images.

3.2.9 PIV analysis

The PIV analysis is done on the reconstructed images from the previous section. The open-source PIVLab tool in MATLAB [26] does the PIV. In this tool, the reconstructed frames are loaded in a GUI, and the user can choose specific PIV settings from the GUI, such as the calibration, window size, and region of interest. The calibration is done by knowing the pixel size from the reconstruction script, and the following rule of thumb is used for choosing the window size:

$$\begin{aligned} N_I &> 10 \\ |\Delta X| &< D_I/4 \end{aligned} \tag{25}$$

Where N_I are the number of scatterers in a window, ΔX is the displacement of the scatterers, and D_I is the size of the window. This average displacement is found by multiplying the average velocity of the flow (23) by the frame time.

PIVlab returns the subpixel-accurate u and v components of the velocity field, which can be saved as a MAT file. These components are then loaded in a script, and a post-processing algorithm is applied to the velocity field. The post-processing is done by calculating the mean over the velocity field in the lateral direction. The local mean replaces every vector if the magnitude of the vector deviates from this mean by a threshold of 1.5 times the mean. Replacing these vectors removes unwanted vectors, ensuring a stable velocity profile over time.

After post-processing, the average velocity field in its depth is saved by taking the mean over all frames in the lateral+ direction.

3.2.10 AI velocimetry

The trained neural network directly outputs the velocity field of the reconstructed frames. This network is trained with a frame interval time of 2ms. If the experimental frames contain a different sample time, the velocities can be scaled to the actual values by the following equation:

$$S = \frac{2}{t_{exp}} \tag{26}$$

$$\begin{bmatrix} u_{exp} \\ v_{exp} \end{bmatrix} = S \times \begin{bmatrix} u_{NN} \\ v_{NN} \end{bmatrix} \tag{27}$$

Where S is the scaling factor, t_{exp} is the frame time of the experimental frames [ms], u_{exp}, and v_{exp} are the velocity components of the experiments [m/s], and u_{NN},v_{NN}, are the velocity components given directly by the neural network.

4 Results

The neural network is validated on experimental and simulated data after training on the data by the method explained in section 3.1.1. First, the results of the validation dataset are presented. Then, the network is tested on a simulated case containing a flow field not yet seen in the training or validation dataset. Finally, the neural network is tested on experimental data obtained in vitro.

4.1 Training Results

As explained in section 3.1.1, the 2d simulator and PROTEUS are used for training and validation samples. The user selects the number of input frames, learning rate, and batch size, which are optimized to minimize the validation loss. Figure 11a shows the lowest validation loss for runs with different amounts of input frames. Increasing the number of input frames results in a lower validation loss, with five frames chosen as the maximum. This limit is set because using more input frames extends the time between the first and last frame, which makes the results more susceptible to unsteady flow changes and can lead to inaccurate estimations in experimental data. Figure 11b shows the training and validation loss for a run with five input frames. This figure shows that the validation loss stabilizes around 200 epochs. There are still a few spikes that can be seen in the loss indicating a rough loss surface. Figure 11c and d show the average angle metric and RMSE metric with 5 input frames. the Training converges to an RMSE of around 0.08 m/s and an average angle error of 10 degrees. Since five input frames showed the highest accuracy, the remainder of this thesis uses the trained neural network of five different input frames.

4.2 Validation Results

Figures 12a and b show the ground truth and predicted velocity field of a validation example from the 2D simulator with a background filled with linear scatterers. Here, the flow is shown as a velocity magnitude on top of a single simulation frame. Figure 12c shows the velocity magnitude difference between the ground truth and the prediction, indicating that the highest velocity differences occur at the walls, inlet, and outlet of the flow, while there is almost no velocity magnitude difference at the centre of the flow. Figure 12d shows the angle deviation between ground truth and prediction, with the most significant angle deviations occurring at the inlet and outlet of the flow. Figure 12e shows the axial velocity profile of the prediction and ground truth. This axial profile is adjusted

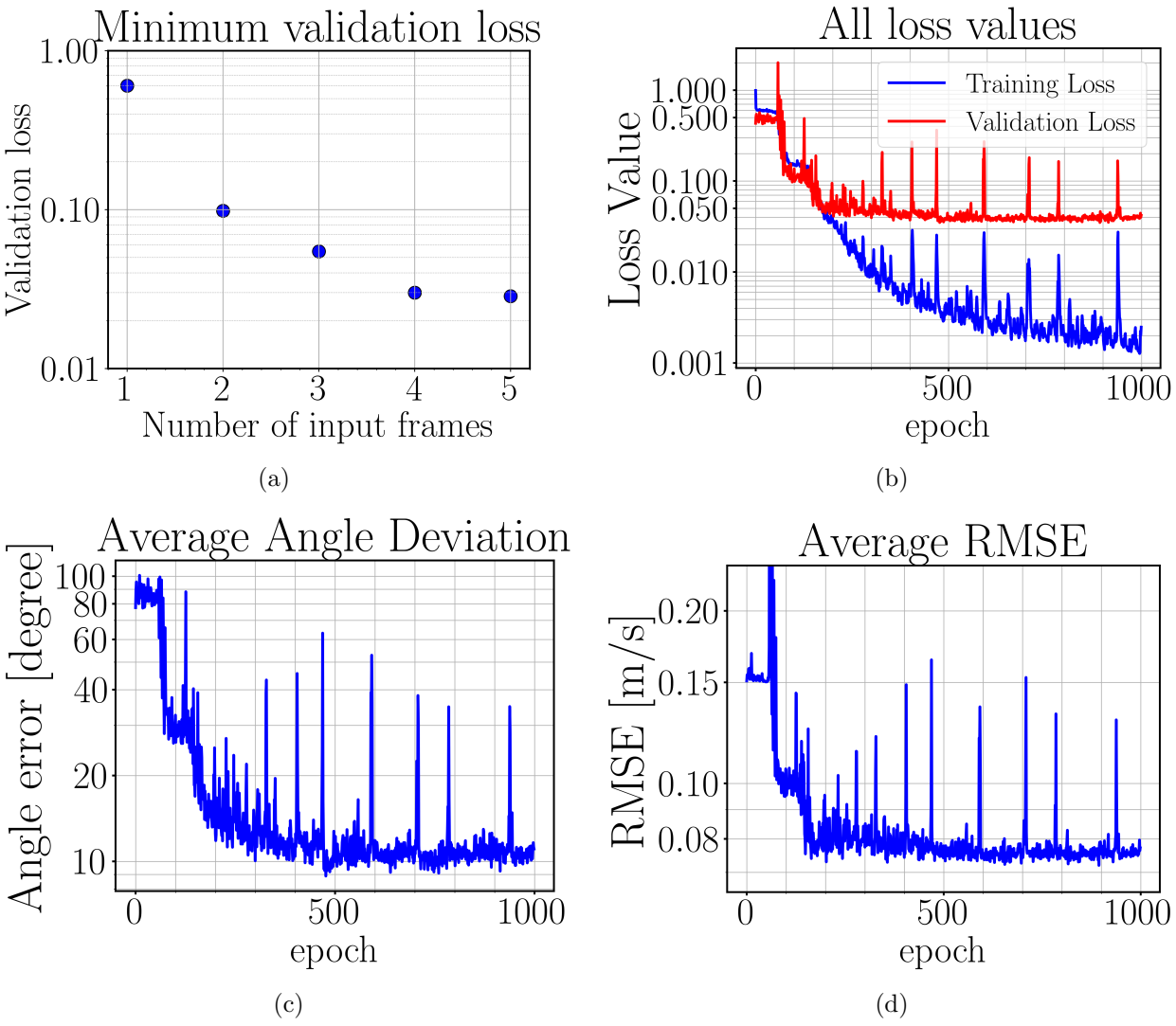

Figure 11: a)The minimum validation loss against different numbers of input frames. b) The training and valdition loss for a run with 5 input frames. c) The average angle deviation for a run with five input frames. d) The average root mean squared error for a run with five input frames.

for the curvature of the flow so that the profile is orthogonal to the flow direction. This profile shows that the network can accurately predict near-wall velocities. Finally, Figure 12f presents the angle and velocity errors as a histogram, with the deviation normalized to the maximum possible deviation. The maximum angle deviation is 180 degrees, and the maximum velocity deviation is the maximum absolute velocity of both prediction and ground truth velocity. The angle deviation does not exceed 6%, while the velocity magnitude deviation rapidly decreases after 10%.

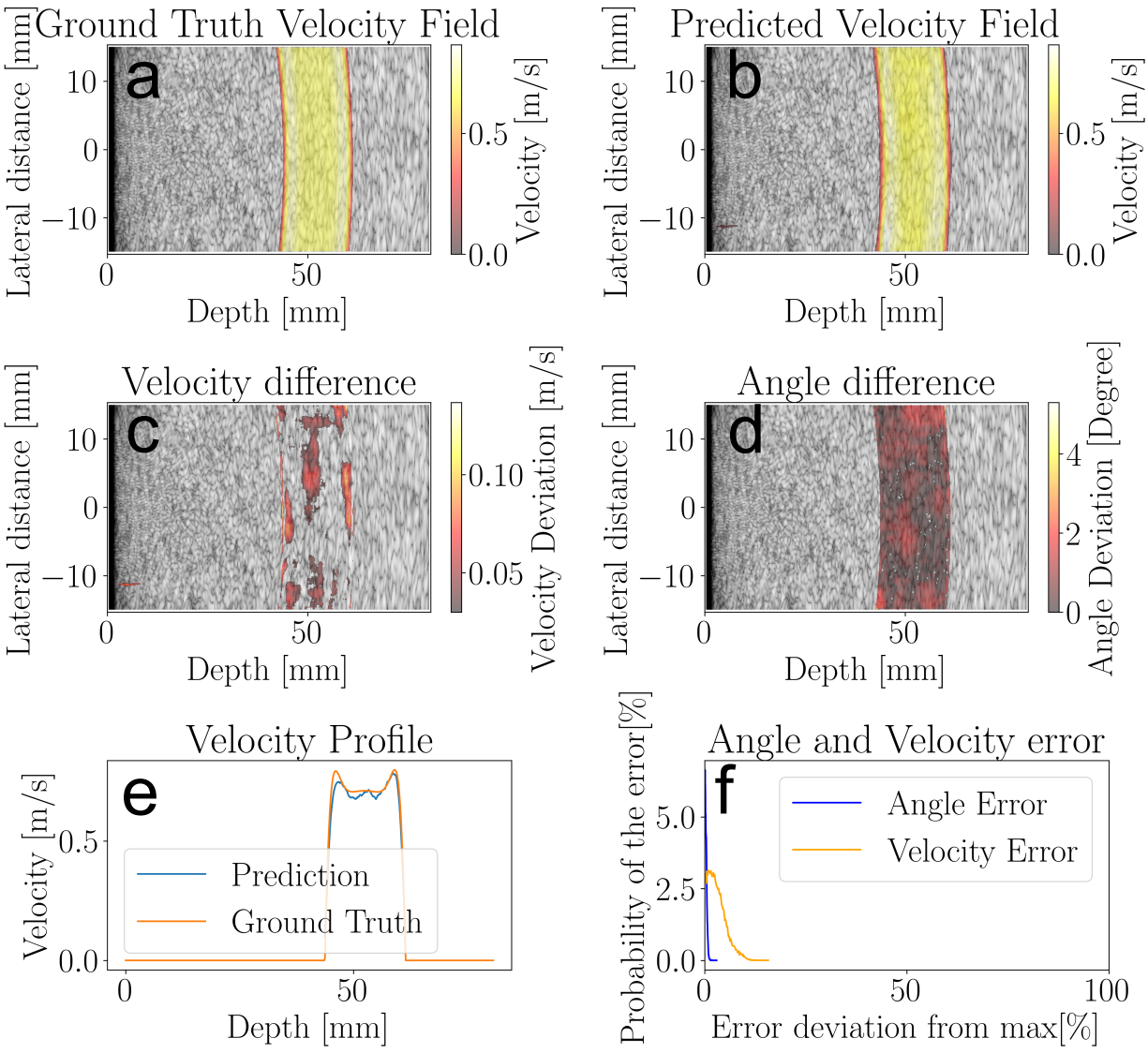

Figure 12: a) Ground truth Velocity field of a sample from the 2D simulator, where the profile is plotted on top of the first frame. b) The predicted velocity field is for the same validation sample, where the profile is again plotted on top of the first frame. c) The velocity magnitude difference between prediction and ground truth. d) Angle difference between ground truth and prediction. e) Axial velocity profiles for the prediction and ground truth. f) Normalized histogram for both the velocity and angle errors. Both errors are scaled such that the maximum possible error is at 100%

Figures 13a,b and 14a,b show a validation sample from the 2D simulator where the background

signal is left out and a validation example from Proteus. In both cases, the network accurately predicts the location of the flow, whereas it does not predict any substantial velocity vectors outside the pipe. Additionally, Figures 13c,f and 14c,f show that the angle deviation between prediction and ground truth is no more than 10 % of the maximum angle deviation of 180 degrees. However, the velocity magnitude deviations are up to 25 % for the 2D simulator, and 50 % for Proteus, as shown in Figures 13c,f and 14d,f. Furthermore, the axial velocity profiles of Figures 13c,f and 14e, indicate a systematic velocity error. This suggests that the neural network struggles with accurately predicting the velocity scale.

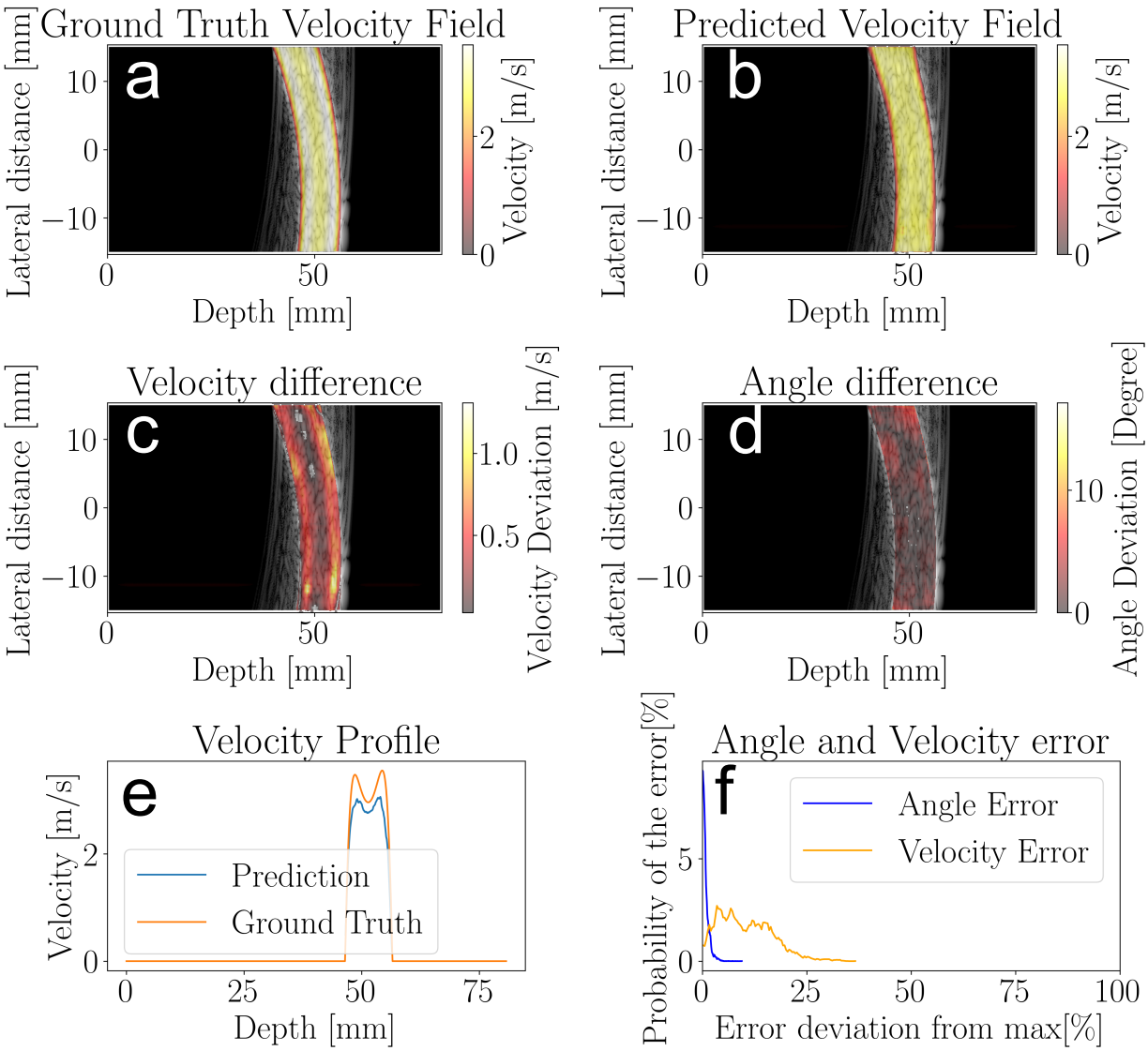

Figure 13: a) Ground truth Velocity field of a sample from the 2D simulator, where the profile is plotted on top of the first frame. b) The predicted velocity field is for the same validation sample, where the profile is again plotted on top of the first frame. c) The velocity magnitude difference between prediction and ground truth. d) Angle difference between ground truth and prediction. e) Axial velocity profiles for the prediction and ground truth. f) Normalized histogram for both the velocity and angle errors. Both errors are scaled such that the maximum possible error is at 100%

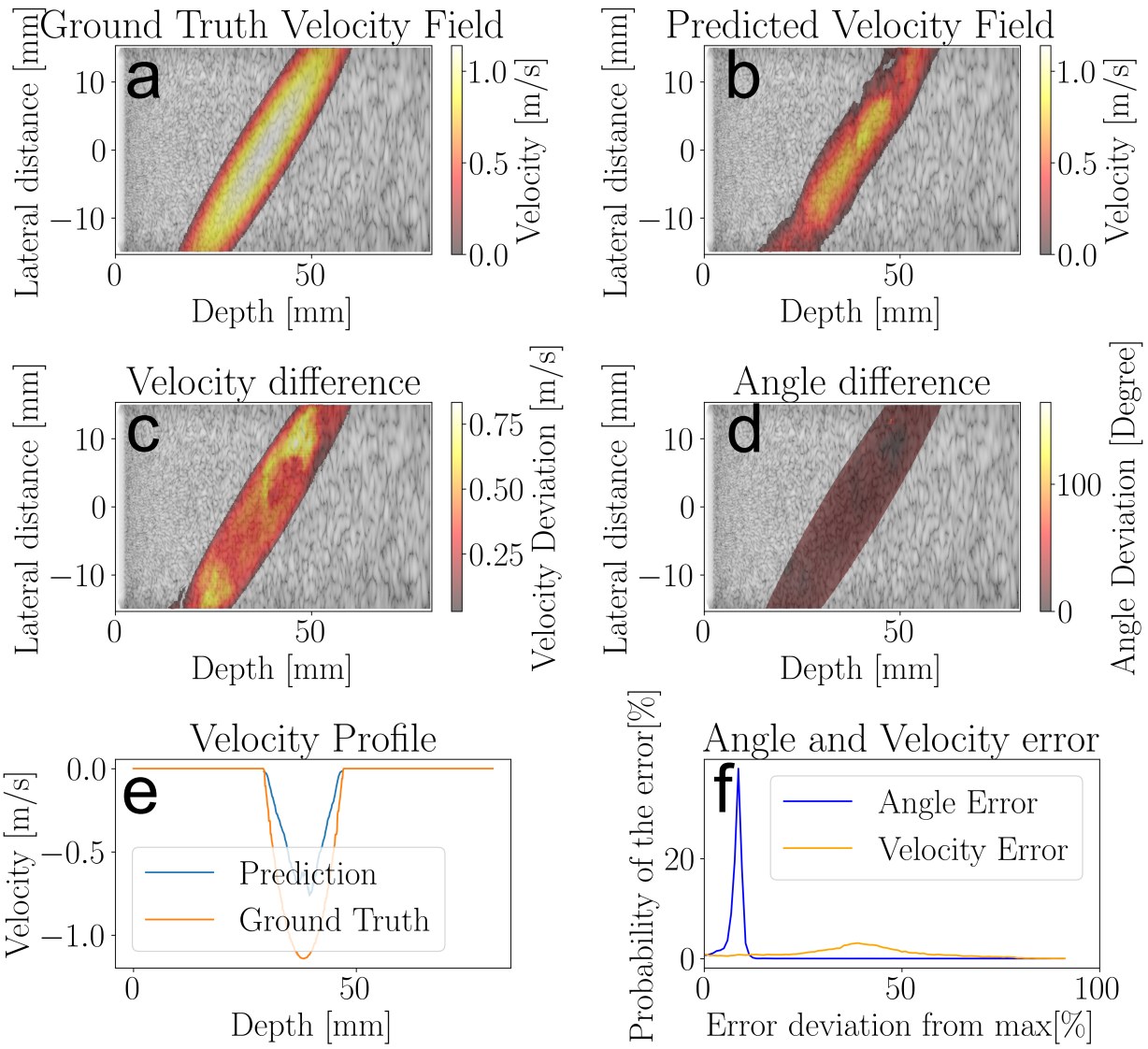

Figure 14: a) The ground truth velocity field of a sample from Proteus is where the profile is plotted on top of the first frame. b) The predicted velocity field is for the same validation sample, where the profile is again plotted on top of the first frame. c) The velocity magnitude difference between prediction and ground truth. d) Angle difference between ground truth and prediction. e) Axial velocity profiles for the prediction and ground truth. f) Normalized histogram for both the velocity and angle errors. Both errors are scaled such that the maximum possible error is at 100%

4.3 Test case

Alongside validating the neural network on simulation parameters the network has already seen before in the training dataset. The network is also tested on a flow profile that the neural network has not seen before. This indicates the robustness of the neural network. The flow profile used for the simulation is shown in Figure 18, and is used for the 2D simulator 3.1.2. The neural network has not yet seen this flow profile and thus falls outside the training and validation set. The simulation

has been run for a hundred different frames, and the remaining parameters for this simulation are shown in Appendix B.

The simulation is then used to compare the accuracy of the estimation from echoPIV with the estimation from the trained neural network. The average velocity profiles of these estimations are shown in Figure 15b. The profile from PIV shows a small flat overshoot while the neural network estimates a small curved undershoot. Figure 15d shows the relative error of the profiles from the ground truth solution. Here it is shown that the neural has trouble with estimating the overall shape of the flow, but it correctly estimates the flow at the sides of the flow. PIV has trouble with these velocities due to the small movement at the outer side of the flow. The average error of the estimation from the neural network is given by 0.014 m/s, while the average error of PIV gives an error of 0.019 m/s.

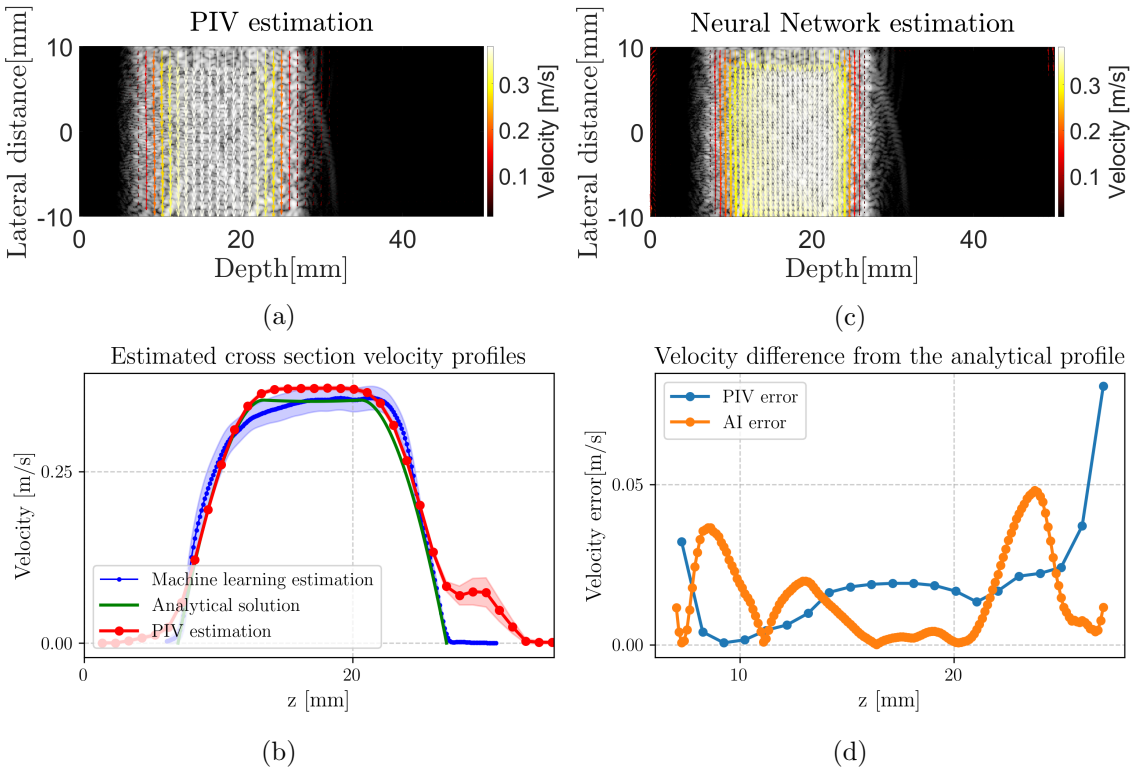

Figure 15: FLow estimation on an simulated undeveloped flow. a) Average PIV estimation on the undeveloped flow averaging over 100 frame pairs. b) Average neural network estimation on the undeveloped flow averaging over 100 frame pairs.

4.4 Experimental validation

As a more challenging alternative to validating the trained neural network on a simulated dataset, the network is also validated on the experimental setup described in section 3.2.5. Three different

B-mode reconstructions measured by the P4-1 transducer are shown in Figure 16, alongside a reconstruction of the 2D simulator. Figure 16a shows an unfiltered frame, 16b shows a frame filtered with the velocity-based filter 2.3.2, 16c shows a frame filtered with an SVD filter 2.3.1, and 16d shows a simulated frame from the 2D simulator. For this simulation, the tube location, radius, flow rate, and fluid properties are all set to match the experiments as well as possible. The unfiltered frame shows a high static intensity from the walls alongside a static reflection displayed as a line inside the tube. The velocity-based and SVD-filter can filter these static signals, leaving only the bubbles as the signal. This is done such that the experiments look similar to the simulated ultrasound images.

The dynamic range of the bubble signals between filtered measurements and simulations and the speckle size match. However, the speckles from the measurement are a bit more elongated than the round speckles from the simulations. Additionally, behind the tube, a moving speckle signal is observed. This artefact is likely due to additional reflection effects inside the tube wall and cannot be filtered out by the SVD or velocity-based filter.

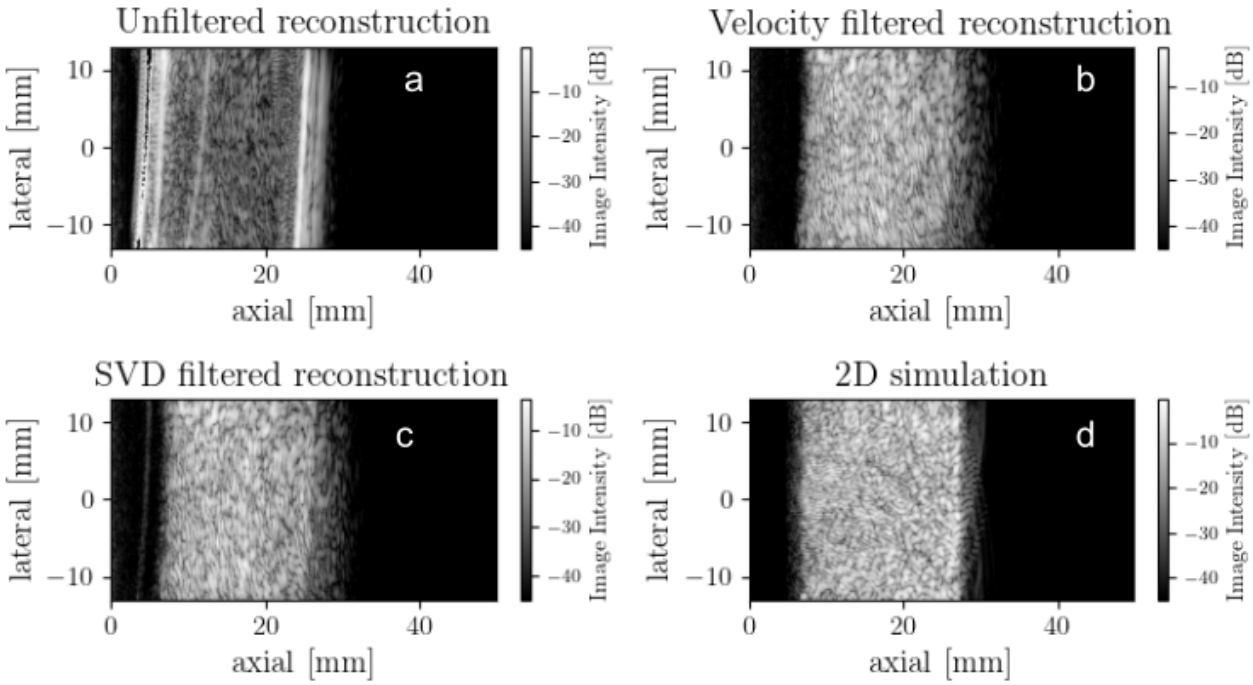

Figure 16: a) Reconstruction of a pipe with flowing microbubbles measured by the P4-1 transducer. b) A velocity-based filter is applied on the frames, resulting in this new reconstruction where only the bubble signal remains. c) An SVD-based filter is applied to the frames, resulting in this new reconstruction where only the bubble signal remains. d) A frame from the 2D simulator with matching parameters to a, b and c.

The same measurement was also done with the L12-3v transducer (Philips ATL.). Figure 17a shows a reconstructed frame of this measurement. Similar to the measurement of the P4-1, the static signal of the wall dominates the signal from the bubbles. Figure 17b shows the velocity-based filter applied to the reconstructed frame, showing the removal of the static intensity. As a result, the bubbles are now clearly visible and more easily distinguishable compared to the P4-1 measurement.

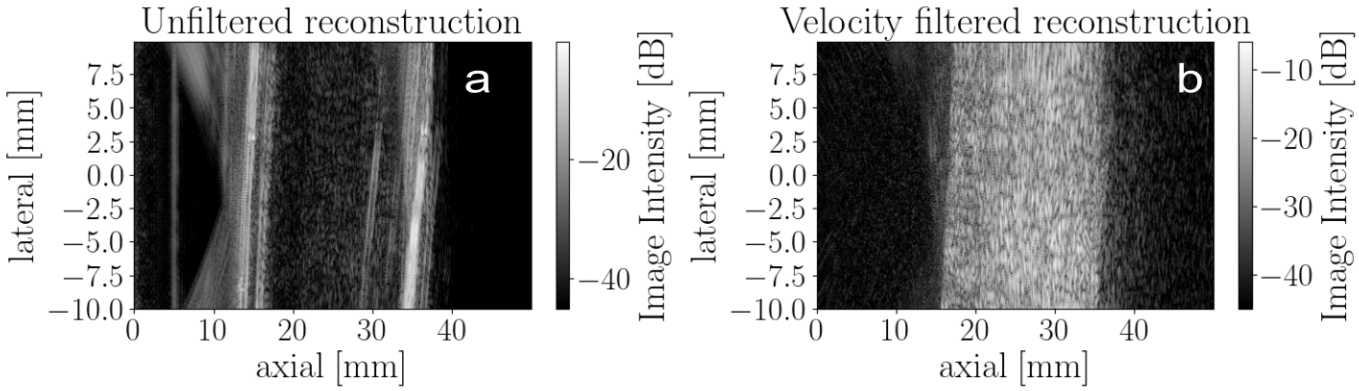

Figure 17: a) Reconstruction of a pipe with flowing microbubbles measured by the L12-3v transducer. b) A velocity-based filter is applied on the frames, resulting in this new reconstruction where only the bubble signal is left over.

4.5 Flow estimation

The flow rate inside the tube is determined using the method described in section 3.2.6. This resulted in filling a beaker up to 2200mL in 30 seconds. With equations (22) and (23), the average velocity is calculated to be 0.23m/s. The Reynolds number for this flow is calculated by equation (1), and gives a value of 4670. This value shows that the flow is in the laminar-turbulent regime [27]. The smoothness of the tube determines the turbulence of the flow. The straight acrylic and PVC tubes are expected to be smooth enough for the flow to remain laminar.

Even though the flow is expected to be laminar, the entrance length is too short for a fully developed flow. Equation (34), together with the setup parameters, approximate the undeveloped laminar flow through the tube. Figure 18 shows this axial flow profile.

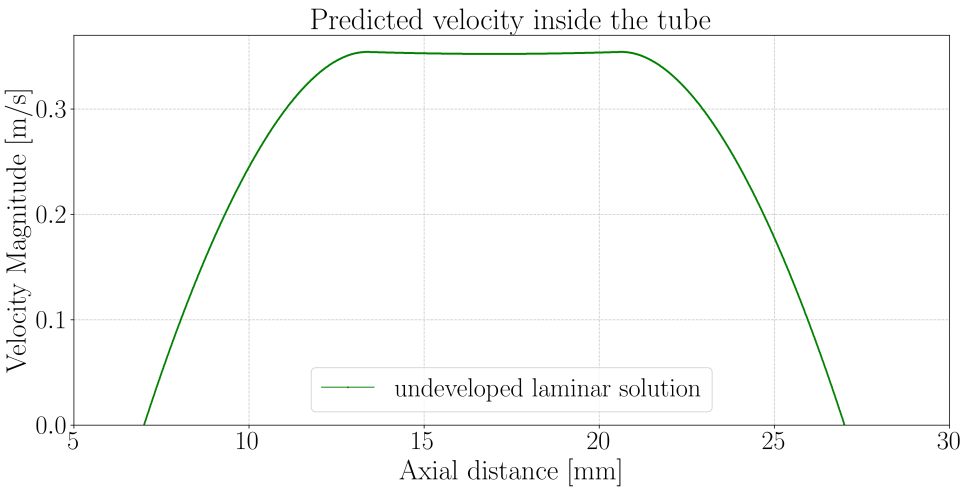

Figure 18: Analytical solution to an undeveloped laminar flow for a Reynolds number of 4670 and an entrance length of 90cm.

4.5.1 PIV results

The velocity-based filtered reconstructions of the P4-1 and L12-3v measurements and the high frame/rate images are analyzed by the PIV method described in section 3.2.9. For this method, the parameters are set by the rule of thumb described by equation (25) and shown in Table **??**.

Chosen Parameter	**P4-1 Measurment**	**L12-3v Measurement**	**High frame rate imaging**
Window size pass 1	128	128	256
Window size pass 2	64	64	128
Window size pass 3	32	32	64
Step pass 1	32	32	64
Step pass 2	32	32	64
Step pass 3	16	16	32
pixel-size	0.1232 mm	0.0423 mm	0.01597 mm
Number of interval frames	10	10	1
Total Frame time	2ms	10	1

Table 5: Table showing the chosen parameters for the PIV analysis

The average velocity profile is taken over all 80 frame pairs for the P4-1, L12-3v, and 400 frame pairs for the high frame-rate measurements. Figures 19a, b and c show these resulting velocity profiles. These figures show a uniform velocity field and that the velocity magnitudes match the estimated values of figure 18. It is also shown that echoPIV predicts velocities outside the tube due to the reflections inside the outer wall.

4.5.2 AI results

The filtered reconstructions can also be passed through the network to validate how well the neural network can predict a physical velocity profile. For this pass, five frames with an interval of ten frames are chosen from the P4-1 measurement. Choosing this interval of ten frames ensures the interval time is 2ms, the same interval time used in the simulations. Figure 19 c shows the average velocity profile over 600 frames. Here, it can be seen that even though the vectors are all aligned, the velocity magnitude fluctuates over the lateral direction.

At the opposite side of the tube, between an axial distance of 25 and 30 mm, a moving pattern can be seen where there should be no movement. This movement is most likely due to some scattering effects inside the outer wall. The neural network correctly predicts no flow field inside this region. However, the velocities just before this region are higher than expected, indicating that this movement does affect the general flow prediction inside the tube.

4.5.3 AI and PIV comparison

The PIV algorithm and neural network are tested on evaluation speed and accuracy. First, the difference in speed is analyzed.

Evaluation speed

For the PIV analysis, the speed of PIVLab is tested by analyzing 157 different frame couples. A stopwatch times the time it takes for the complete analysis of all frame couples. This timing is done ten times, from which the average frame time is taken. This average time is given by (14.5 ± 1.34)s. Dividing this total run time by the amount of frame couples gives an average evaluation time of 92.3 ± 8.50 ms.

The neural network's evaluation time is also tested for speed. This is done by taking the average time of a forward pass over 500 different frame couples. A GPU with 48 GB of memory is used to increase the speed for this forward pass. This timing resulted in an average of (5.2 ± 0.5)ms, where the error is the standard deviation over all 500 frame couples. This evaluation time is 20 times faster than the evaluation time of PIVLab.

The time between five frames for the constructed experiments is 100ms, indicating that the network can image in real time.

Accuracy

From the velocity fields of Figures 19a, b and c, an axial velocity profile is taken by averaging the flow along the lateral region of [-5,5]mm. This way, the velocity changes in the lateral direction do not influence the final profile. Figure 19 d shows the resulting profiles.

This figure shows that the average PIV and neural network velocities match the P4-1 and L12-3v ultrasound measurements. The PIV analysis on the high-frame images shows a slightly higher velocity profile. The neural network estimates the near-wall velocities well and only slightly deviates from the wall at the far side of the tube. The high frame rate measurements seem to follow the analytical solution quite well near the walls, but since a too-short imaging window was chosen, the full tube size could not be imaged. The PIV estimates cannot measure these near-wall velocities due to the outside speckle movement of the tube.

Figure 19f displays the gradients of the velocity profiles from part e. The gradients of the analytical solution reveal a linear trend along the sides of the tube, with a zero gradient at the centre. The transition between a steady gradient and a linear gradient occurs at varying depths for different estimations. This transition aligns precisely with the analytical estimation for the ultrasound PIV measurements. In contrast, the transitions for the neural network estimation and the high frame rate imaging PIV estimation occur closer to the tube walls, suggesting a flatter profile in these cases. High gradients occur at the tube walls, which the ultrasound PIV measurements cannot capture. However, the neural network does account for these high gradients, although it tends to overshoot due to the differing transition locations. This indicates that the neural network provides a more accurate estimation of the near-wall velocity differences.

Since the neural network estimates a full velocity field, equation (24) is used to calculate the average velocity of the flow. This gives back an average velocity of 0.233 ± 0.047m/s, where the average velocity from the standard deviation of the velocity field calculates the error. The expected velocity of 0.23 m/s from the beaker measurement falls in between the range of the computed velocity field.

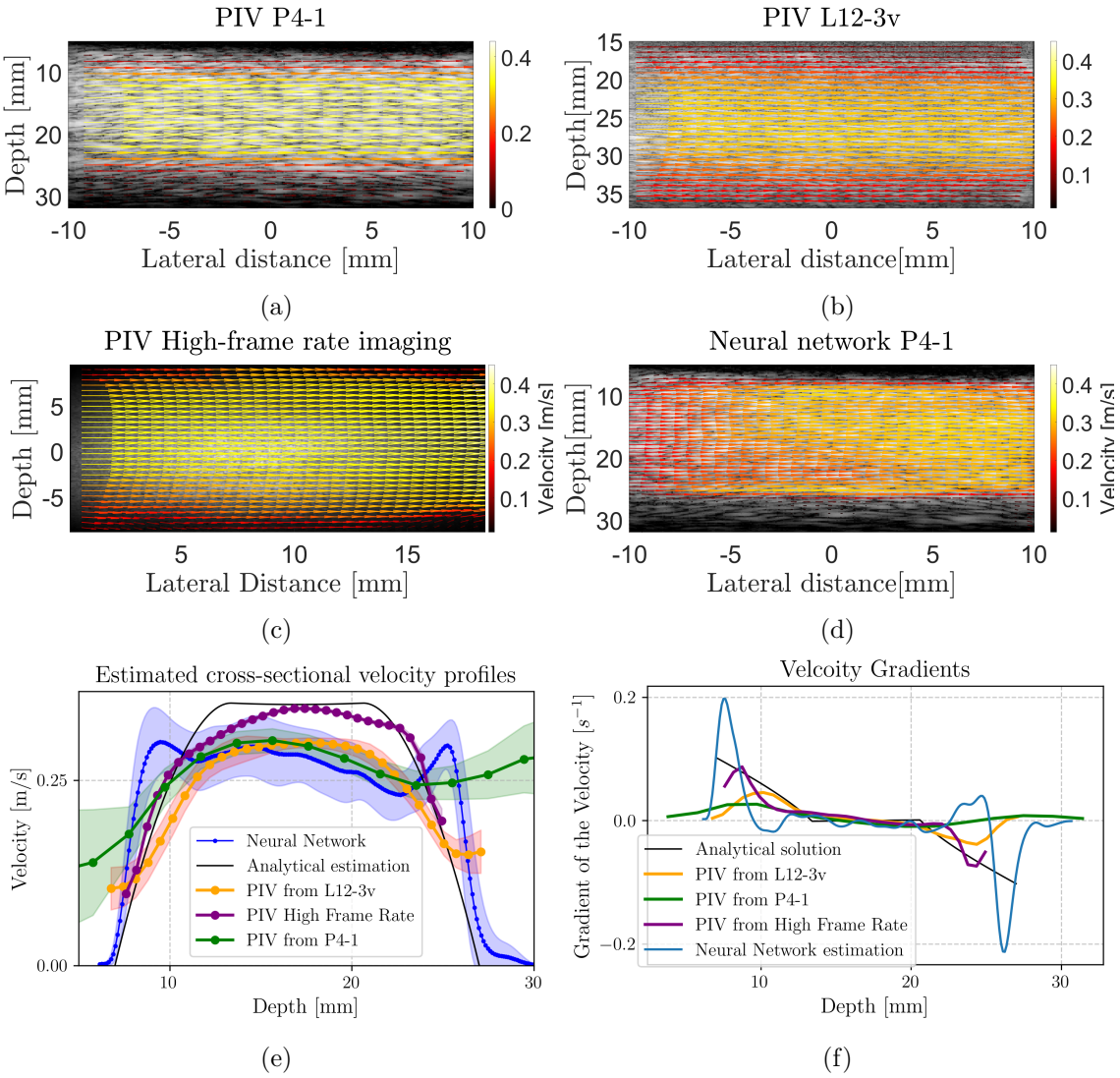

Figure 19: ultrasound Velocimetry plots. a) The average planar velocity profile from the PIV analysis on the P4-1 reconstructions over 80 different frame couples. b) The average planar velocity profile from the PIV analysis on the L12-3v reconstructions over 80 different frame couples. c) The average planar velocity profile from the PIV analysis on the L12-3v reconstructions over 400 different frame couples.d) The average planar velocity profile from the neural network estimation on the P4-1 reconstructions over 600 frames. e) Average cross-section velocity profiles for the PIV analysis on the L12-3v and P4-1 reconstructions and High frame rate measurement, neural network estimation on the P4-1 reconstructions, and the estimated analytical profile. f) Velocity gradients of the cross-sectional velocity profiles from e).

5 Discussion and Recommendations

5.1 Machine Learning and PIV

The results section compared the use of the neural network and the PIV analysis. First, the evaluation time is tested, showing that the neural network has an evaluation time of (5.2 ± 0.5)ms, and the PIV analysis contains an evaluation time of 92.3 ± 8.50ms. The evaluation time of the

neural network is twenty times faster than PIV, showing the benefit of using a neural network instead of the PIV algorithm. For real-time imaging, typical frame times of 5 to 20 ms per frame are used. These times are too fast for echoPIV, but with these times, the neural network should estimate the velocity field in real time if the times for filtering Delay and Sum are neglected.

Figure 19d showed the average axial velocity profile for the PIV, neural network, and estimated analytical solution. From these results, it became clear that PIV lacks accuracy at near-wall velocities. The machine learning algorithm significantly improved at these near-wall velocities, indicating another benefit of using the trained neural network instead of PIV. This result is also seen in Figure 15, for the simulated test case. This indicates that the neural network not only learns the velocity magnitudes but also the location of the flow boundaries.

5.2 Experimental setup

This thesis used a too-short entrance length for a fully developed laminar flow. This makes it challenging to know the flow inside the tube precisely. This is now done by estimating an undeveloped flow from equation (34). However, this equation expects a uniform entrance flow. This is not the case for the used setup, where the entrance flow comes from a smaller tube. It is unclear how much this difference changes the estimated flow, but this could explain the differences between analytical and experimental flow patterns.

So, further research should create a setup with a smaller entrance length so that a fully developed laminar flow can be expected. This entrance length (3) scales linearly on the Reynolds number and diameter of the tube. This implies that choosing a lower velocity, shorter tube diameter, and higher fluid viscosity should all contribute to a shorter entrance length.

Furthermore, the average flow velocity is estimated by calculating the average velocity over 30 seconds. This average velocity neglects flow fluctuations, which can significantly influence the final flow profile on the small time scales used for ultrasound imaging, even if the flow is averaged over multiple frames. Future research could utilize a flow meter that measures the flow at all times to account for small fluctuations.

5.3 Robustness of the network

The neural network is trained on simulated data from a P4-1 transducer. It is shown that the neural network performs well on a validation set inside this parameter space and a test case with a new type of flow. However, blood flow typically involves turbulent flows containing complex, non-steady flow profiles. Future research should test whether the neural network can estimate accurate vectors from these complex flows, indicating the network's robustness. This test can, for example, be done by using a programmable piston pump (Super Pump, Vivitro, Victoria, Canada) to create a pulsatile flow.

Also, the neural network is only trained with simulations of the P4-1 transducer. Transducers drastically change the appearance of the reconstructed speckles. Future research should test whether the transducer type influences the flow's accuracy.

Finally, the neural network has not been trained on artefacts like shading or reverberations. Future research should test whether these artefacts influence the neural network's accuracy velocimetry. For example, testing on shading can be done by placing an object in front of the tube.

5.4 Physics of simulations

The 2D simulator and Proteus cannot simulate a perfect physical environment. The main issue is with ultrasound artefacts. For example, the moving signal outside the tube of Figure 16. A more physically realistic simulator increases the robustness of the neural network and provides better implementation of experimental data. Also, the simulated linear and non-linear scatterers now flow along the streamlines of the flow. Generally, many forces act on the microbubbles, such as drag, Bjerknes, bubble-bubble, and wall interaction forces. These different forces ensure that the scatterers do not perfectly flow along the streamlines, and not including these imperfections might make implementing the neural network in experimental data more challenging. Future research should thus improve the simulators to increase the robustness of the neural network.

5.5 ultrasound filtering

An SVD and velocity-based filter extract only the moving scatterers from the experimental reconstructions. These filters can filter out the static signal but lack processing speed. Even more so, the velocity-based filter is more accurate when many frames are used. Only a few frames are available

for imaging in real-time, making the velocity-based filter less applicable. Further research must provide a fast and accurate filter to make real-time velocimetry available.

5.6 In vivo imaging

The final goal of this research is to implement a neural network for in vivo imaging. The neural network has been experimentally validated on an ideally experimental setup, neglecting tissue movement or small movements caused by the clinician. These extra movements might decrease the neural network's accuracy. Future research should thus investigate whether the trained neural network generalizes to in vivo data.

6 Conclusion

This thesis aimed to train a neural network to predict a velocity field from ultrasound frames and compare this to the state-of-the-art technique PIV. A residual U-net is implemented and trained on ultrasound-simulated data. Two different simulators were used to execute these simulations. First, the simulator PROTEUS was used, which acts in a three-dimensional, physically accurate environment. The second simulator was made part of this thesis and is a 2D simulator, but it has the benefits of more complex flow geometries. The simulations are executed for a P4-1 transducer in a fixed parameter space. Training the neural network with varying input frames demonstrated that increasing the number of frames enhances accuracy. A maximum of five input frames was selected for practical reasons. Using more frames would extend the imaging time, which could lead to unwanted effects, such as slight deviations in an unsteady flow, leading to less accurate results.

Validating the neural network on the simulated data resulted in an average angle error of 10 degrees and a root mean squared error of 0.08 m/s on an average velocity scale of 1 m/s. This neural network was then tested on an undeveloped laminar flow profile that the neural network had not seen before. The neural network accurately predicted the flow with an average error of 0.014 m/s. These results were compared to echoPIV, where the PIV analysis got an average error of 0.019 m/s. Even though these errors are similar, the comparison showed that the neural network outperforms echoPIV at estimating near-wall velocities.

Furthermore, an experimental setup is made to validate the neural network experimentally. This setup consisted of a straight tube containing an undeveloped water flow, imaged by a P4-1 and L12-3v transducer and a high frame-rate camera. It was successfully observed that a velocity-based filter removed the static signal of the tube walls. However, it was impossible to filter out the unwanted moving reflections of the outer wall. Even though this moving signal influenced the PIV and neural network analysis, it was still possible to estimate the velocity profile inside the tube. For these measurements, the velocity magnitudes slightly deviated from the expected analytical solution even though the profile of the neural network estimation did estimate the correct average flow velocity. However, the neural network was more accurate in correctly assessing the near-wall velocities than the PIV analysis.

Future research should investigate the generalizability and robustness of the neural network. This can be tested by experimentally measuring more complex flow profiles with a pulsatile pump

or comparing PIV and neural network estimations on in vivo data.

7 Acknowledgements

Foremost, I would like to thank my supervisor, Guillaume, for his continuous support during my assignment. His guidance and knowledge helped me in all the research for this thesis.

In addition to my supervisor, I would like to thank my daily supervisors, Bram and Julian, for their help, ideas, and motivation. I would also like to thank Jelmer and Michel for the feedback during the weekly meetings and Erik for participating in my examination committee.

My sincere thanks also go to Margot, who helped me during her internship by contributing to the design and planning of the experimental setup and conducting the experiments. Also, thanks to the technicians Gert-Wim, Thomas, and Martin for helping me build the setup.

I thank the POF PhD students, Hermen, Rienk, Fleur, Jelle, Coen, and Anass, for the stimulating discussions and fun during the lunch and tea breaks. I also thank all other POF members for the fun Friday evening drinks and activities.

Last but not least, I would like to thank my family, friends and Denise for always supporting me.

References

[1] Gregory A. Roth, George A. Mensah, and Valentin Fuster. The global burden of cardiovascular diseases and risks: A compass for global action, 12 2020.

[2] Ana Olga Mocumbi. Cardiovascular health care in low- and middle-income countries. *Circulation*, 149:557–559, 2 2024.

[3] Claudia Errico, Bruno-Félix Osmanski, Sophie Pezet, Olivier Couture, Zsolt Lenkei, and Mickael Tanter. Transcranial functional ultrasound imaging of the brain using microbubble-enhanced ultrasensitive doppler. *NeuroImage*, 124:752–761, 2016.

[4] James C. Lacefield Aaron Fenster. *Ultrasound Imaging and Therapy.* CRC Press,Boca Raton, 2015.

[5] Berthold K P Horn and Brian G Schunck. Determining optical flow, 1981.

[6] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. 11 2014.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84–90, 6 2017.

[8] Yiwei Fan, Chunyu Guo, Yang Han, Weizheng Qiao, Peng Xu, and Yunfei Kuai. Deep-learning-based image preprocessing for particle image velocimetry. *Applied Ocean Research*, 130, 1 2023.

[9] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. 4 2015.

[10] Shengze Cai, Shichao Zhou, Chao Xu, and Qi Gao. Dense motion estimation of particle images via a convolutional neural network. *Experiments in Fluids*, 60, 4 2019.

[11] Chang Dong Yu, Yi Wei Fan, Xiao Jun Bi, Yang Han, and Yun Fei Kuai. Deep particle image velocimetry supervised learning under light conditions. *Flow Measurement and Instrumentation*, 80:102000, 8 2021.

[12] Y. C. Fung. *Motion, Flow, Stress, and Growthn.* Springer New York, NY, 2013.

[13] Baptiste Heiles, Arthur Chavignon, Vincent Hingot, Pauline Lopez, Eliott Teston, and Olivier Couture. Performance benchmarking of microbubble-localization algorithms for ultrasound localization microscopy. *Nature Biomedical Engineering*, 6:605–616, 5 2022.

[14] Marcelo Lerendegui, Kai Riemer, Bingxue Wang, Christopher Dunsby, and Meng-Xing Tang. Bubble flow field: a simulation framework for evaluating ultrasound localization microscopy algorithms. 11 2022.

[15] Philippe Marmottant, Sander van der Meer, Marcia Emmer, Michel Versluis, Nico de Jong, Sascha Hilgenfeldt, and Detlef Lohse. A model for large amplitude oscillations of coated bubbles accounting for buckling and rupture. *The Journal of the Acoustical Society of America*, 118:3499–3505, 12 2005.

[16] Hatim Belgharbi, Jonathan Porée, Rafat Damseh, Vincent Perrot, Léo Milecki, Patrick Delafontaine-Martel, Frédéric Lesage, and Jean Provost. An anatomically realistic simulation framework for 3d ultrasound localization microscopy. *IEEE Open Journal of Ultrasonics, Ferroelectrics, and Frequency Control*, 3:1–13, 1 2023.

[17] Nathan Blanken, Baptiste Heiles, Alina Kuliesh, Michel Versluis, Kartik Jain, David Maresca, and Guillaume Lajoinie. Proteus: A physically realistic contrast-enhanced ultrasound simulator—part i: Numerical methods. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, July 2024. Publisher Copyright: Authors.

[18] Manuel Hasert, Kannan Masilamani, Simon Zimny, Harald Klimach, Jiaxing Qi, Jörg Bernsdorf, and Sabine Roller. Complex fluid simulations with the parallel tree-based lattice boltzmann solver musubi. *Journal of Computational Science*, 5:784–794, 2014.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 9 2014.

[21] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. 5 2015.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. 5 2015.

[23] Simon J.D. Prince. *Understanding Deep Learning*. The MIT Press, 2023.

[24] Maresca D. Jain K. Blanken N. Heiles B. Versluis M. Lajoinie, G. Cfd models for the proteus simulator [data set]. zenodo., 2024.

[25] Vincent Perrot, Maxime Polichetti, François Varray, and Damien Garcia. So you think you can das? a viewpoint on delay-and-sum beamforming. *Ultrasonics*, 111, 3 2021.

[26] William Thielicke and René Sonntag. Particle image velocimetry for matlab: Accuracy and enhanced algorithms in pivlab. *Journal of Open Research Software*, 9:1–14, 2021.

[27] Meinhard T. Schobeiri. *Laminar-Turbulent Transition*, pages 225–259. Springer International Publishing, Cham, 2022.

[28] Taig Young Kim. Analytical solution for laminar entrance flow in circular pipes. *Journal of Fluid Mechanics*, 979, 1 2024.

[29] Lance Jonathan Myers and Wayne Logan Capper. Analytical solution for pulsatile axial flow velocity waveforms in curved elastic tubes, 2001.

A Hemodynamics

The following subsections show the derivations of the used axial flow profiles of the background section.

A.1 Fully developed Hagen-Poiseuille flow

Blood flow profiles are derived from the radial part of the cylindrical Navier-Stokes equation:

$$\rho(\delta_t u_z + u_r\delta_r u_z + \frac{u_\phi}{r}\delta_\phi u_z + u_z\delta_z u_z) = \\ -\delta_z p + \mu(\frac{1}{r}\delta_r(r\delta_r u_z) + \frac{1}{r^2}\delta^2_{phi}u_z + \delta^2_z u_z) \\ + \frac{1}{3}\mu\delta_z(\frac{1}{r}\delta_r(ru_r) + \frac{1}{r}\delta_{phi}u_{phi} + \delta_z u_z)\rho g_z \quad (28)$$

Where ρ is the density of the fluid $[kg/m^3]$, $u_{r,z,\phi}$ is the velocity of the fluid for the radial, lateral, and angular directions respectfully [m/s], p is the applied pressure on the fluid [Pa], μ is the dynamic viscosity $[Pa \cdot s]$, and g are the externally applied forces [N].

This equation can be simplified by a few assumptions for a laminar flow inside a straight vessel. These assumptions are that the flow is a steady ($\delta_t = 0$) axisymmetric ($\delta_\phi = 0$) , and fully developed ($\delta_z u_z = 0$). It is also assumed that no external forces are acting on the fluid (g=0) and that the radial and angular part of the velocity are 0 ($u_{r,\phi} = 0$). These assumptions result in the following equation:

$$\frac{1}{r}\delta_r(r\delta_r u_z) = \frac{1}{\mu}\delta_z p \quad (29)$$

To get the Hagen-Poiseuille equation, the pressure difference between both ends of the vessel is assumed to be constant:

$$-\delta_z p = G \quad (30)$$

Assuming a no-slip boundary condition $u(0) = u(R) = 0$ at the pipe walls gives the following

solution to equation (29):

$$u(r) = \frac{G}{4\mu}(R^2 - r^2) \tag{31}$$

Where r is the radial position [m], and R is the entire radius of the vessel [m].

A.2 Hydrodynamic entrance region

The flow is not yet fully developed at the vessel's inlet, as shown in Figure 1. The term $\delta_z u_z = 0$ of equation (28) can no longer be set to 0, making an analytical solution more complex. Kim [28] solves the Navier stokes equation for the velocity profile in the hydrodynamic entrance region by splitting the flow into two regions:

$$U(\xi, \eta) = \begin{cases} U_c(\xi, \eta), & 0 < \eta < \eta_\delta \\ U_w(\xi, \eta), & \eta_\delta < \eta < 1 \end{cases} \tag{32}$$

Where η_δ is the radius at the boundary layer that splits the two flow profiles, and the variables U, ξ, η are the non-dimensionalized velocity, radius, and axial distance, respectively, which are non-dimensionalized by:

$$U = \frac{u}{u_b}, \xi = \frac{x}{DRe}, \eta = \frac{r}{D/2} \tag{33}$$

Where u, x, r are the axial velocity [m/s], radius [m], and axial distance respectively [m], u_b is the average velocity [m/s], D is the diameter of the tube [m], and Re is the Reynolds number ($Re = \frac{\rho u_b}{\nu}$), where ρ is the density $[kg/m^3]$, and ν is the kinematic viscosity of the fluid $[m^2/s]$.

The equations for the two different flow regimes are solved for the radial and axial distances and have the following equations:

$$\begin{aligned} U_c(\xi, \eta) &= U_0(1 - \eta^2) + \frac{D_U}{32}(1 - \eta_\delta^2(1 - ln(\eta_\delta^2)))\eta^2 \\ U_w(\xi, \eta) &= U_c(\xi, \eta) + \frac{D_U}{32}\left[\eta_\delta^2(1 - ln(\eta_\delta^2) - \eta^2 + \eta_\delta^2 ln(\eta^2))\right] \end{aligned} \tag{34}$$

Where, U_0, D_U, η_δ are unknown variables depending on ξ. The following coupled equations can numerically solve these variables:

$$D_U(\xi) = \frac{dU_0^2}{d\xi} = \frac{32(U_0 - 2)}{\eta_\delta^2(1 - \eta_\delta^2 + ln(\eta_\delta^2))} \rightarrow \frac{dU_0}{d\xi} = \left(\frac{32(U_0 - 2)}{\eta_\delta^2(1 - \eta_\delta^2 + ln(\eta_\delta^2))}\right)/2U_0 \tag{35}$$

$$\frac{d\Theta(\xi)}{d\xi} = \frac{8(U_0 - 2))}{1 - \eta_\delta^2 + ln(\eta_\delta^2)} \tag{36}$$

$$\Theta(\xi) = \frac{2}{3} - \frac{(U_0 - 2)(1 - \eta_\delta^2)^2}{6(1 - \eta_\delta^2 + ln(\eta_\delta^2))} + \frac{(U_0 - 2)^2(2 + 3\eta_\delta^2 - 6\eta_\delta^4 + \eta_\delta^6 + 6\eta_\delta^2 ln(\eta_\delta^2))}{12(1 - \eta_\delta^2 + ln(\eta_\delta^2))^2} \tag{37}$$

Equations (35), (36) are numerically solved by the Runge-Kutta method, resulting in the variables U_0, Θ for the next time step. The bisection method extracts η_δ from equation (37) from these variables.

A.3 Pulsatile flow

The pressure inside a vessel fluctuates due to the pulsing of the hearth, leading to a Womersley flow. The assumption of a steady flow can then no longer be accepted, which changes equation (31) into:

$$\delta_r^2 u_z + \frac{1}{r}\delta_r u_z = \frac{1}{\mu}\left(\delta_z p + \rho \delta_t u_z\right) \tag{38}$$

The pressure term can be decomposed into different harmonics, leading to a Fourier series. The zeroth order harmonic gives the Hagen-Poiseuille equation (31), and the other harmonics are added to the profile to form an unsteady solution that gets added to the steady flow profile [29]:

$$u(r,t) = \frac{P_0}{4\mu}(R^2 - r^2) + \mathrm{Re}\left\{\sum_{n=1}^{N} \frac{iP_n}{\rho n\omega}\left[1 - \frac{J_0(\alpha_0 \frac{r}{R})}{J_0(\alpha_0)}\right] e^{in\omega t}\right\} \tag{39}$$

Where P is the applied pressure [Pa], i is the imaginary number, n is a positive integer ranging over all harmonics, ω is the frequency of the first order harmonic [rad/s], J_0 is the zeroth order Bessel function, t is the time of the pressure wave [s], and $\alpha_0 = \sqrt{i^3\alpha^2}$, where α is the Womersley number:

$$\alpha = \sqrt{\frac{\omega R^2}{\nu}} \tag{40}$$

Where ν is the kinematic viscosity $[m^2/s]$.

A.4 Pulsatile flow through curved vessels

Vessels are usually not only straight but can also bend. The axial flow velocity then gets an extra term being influenced by the curvature of the tube:

$$u = u_0 + \lambda u_1 \tag{41}$$

Where u_0 is the pulsatile flow from equation (39), u_1 is the correction to the velocity profile due to the added curvature, and $\lambda = R/R_c$ is a scaling term that is influenced by the curvature, here R is given by the radius of the tube [m], and R_c is the radius of curvature [m].

Myers, Capper [29] found the expression for the pulsatile axial flow velocity in curved elastic tubes. The addition to the curvature was found to be:

$$u_1 = \mathrm{Re}\left\{\sum_{n=0}^{N} \frac{imP_nR}{2\rho n\omega}\left[\frac{J_1(\alpha_0\frac{r}{R})}{J_1(\alpha_0)} - \frac{rJ_0(\alpha_0\frac{r}{R})}{J_0(\alpha_0)}\right]e^{in\omega t}\right\} \tag{42}$$

Where m is given by:

$$\begin{aligned} m &= \frac{2 + x(2\sigma - 1)}{x(F_{10} - 2\sigma)}; \\ F_{10} &= \frac{2J_1(\alpha_0)}{\alpha_0 J_0(\alpha_0)} \\ x &= \frac{Eh}{(1-\sigma^2)R\rho c^2} \end{aligned} \tag{43}$$

Here, E is the Young's modulus, σ is the Poisson's ratio, h is the wall thickness [m], and c is the solution to the frequency equation [29].

This addition is a general term that accounts for elastic walls. For the simulations done in this thesis, a rigid wall is assumed. For a rigid wall, the Young's modulus can be expected to be infinite,

which makes the variable x go to infinity as well. This changes the variable m into:

$$m = \lim_{x\to\infty} \frac{2 + x(2\sigma - 1)}{x(F_{10} - 2\sigma)} = \lim_{x\to\infty} \frac{\frac{2}{x} + (2\sigma - 1)}{F_{10} - 2\sigma} = \frac{\frac{2}{\infty} + (2\sigma - 1)}{F_{10} - 2\sigma} = \frac{2\sigma - 1}{F_{10} - 2\sigma} \tag{44}$$

Simplifying this equation even further can be done by finding the limits of the first and zeroth order Bessel functions for small Womersley numbers:

$$\begin{aligned} J_0(\alpha_0) &= 1 + \mathcal{O}(\alpha_0^2) \\ J_1(\alpha_0) &= \frac{\alpha_0}{2} + \mathcal{O}(\alpha_0^3) \end{aligned} \tag{45}$$

These approximations changes the value of F_{10} into:

$$F_{10} = \frac{2J_1(\alpha_0)}{\alpha_0 J_0(\alpha_0)} \approx \frac{2\frac{\alpha_0}{2}}{\alpha_0 1 - \frac{\alpha_0^2}{4}} = \frac{\alpha_0}{\alpha_0} = 1 \tag{46}$$

And m into:

$$m = \frac{2\sigma - 1}{1 - 2\sigma} = -1 \tag{47}$$

This results in the following equation for a curved pulsatile flow:

$$\begin{aligned} u = \frac{P_0}{4\mu}(R^2 - r^2) + \mathrm{Re}\left\{\sum_{n=1}^{N} \frac{iP_n}{\rho n\omega}\left[1 - \frac{J_0(\alpha_0\frac{r}{R})}{J_0(\alpha_0)}\right] e^{in\omega t}\right\} \\ + \lambda \mathrm{Re}\left\{\sum_{n=0}^{N} \frac{imP_nR}{2\rho n\omega}\left[\frac{J_1(\alpha_0\frac{r}{R})}{J_1(\alpha_0)} - \frac{rJ_0(\alpha_0\frac{r}{R})}{J_0(\alpha_0)}\right] e^{in\omega t}\right\} \end{aligned} \tag{48}$$

B Test case simulation parameters

Chosen Parameter	**Value**
Image width	30mm
Image Depth	80mm
Total Number of Frames	100
Frame Time	200 μs
Number of elements	96
Transmit Transducer Frequency	2.5 MHz
Speed of sound	1540
elevation focus	80mm
Sampling Frequency	5.2e-8 Hz
Transducer focus	infinte
Radius	10mm
Starting Depth	25mm
number linear scatters	1000