User Manual: PWA Matlab App

Lukas Weissinger[*]

July 15, 2024

1 Introduction

This manual provides comprehensive instructions for installing and using the MATLAB-based application, "PWA-Tool". Further, all mathematical methods utilized in the App are shortly explained here.

1.1 Purpose of the App

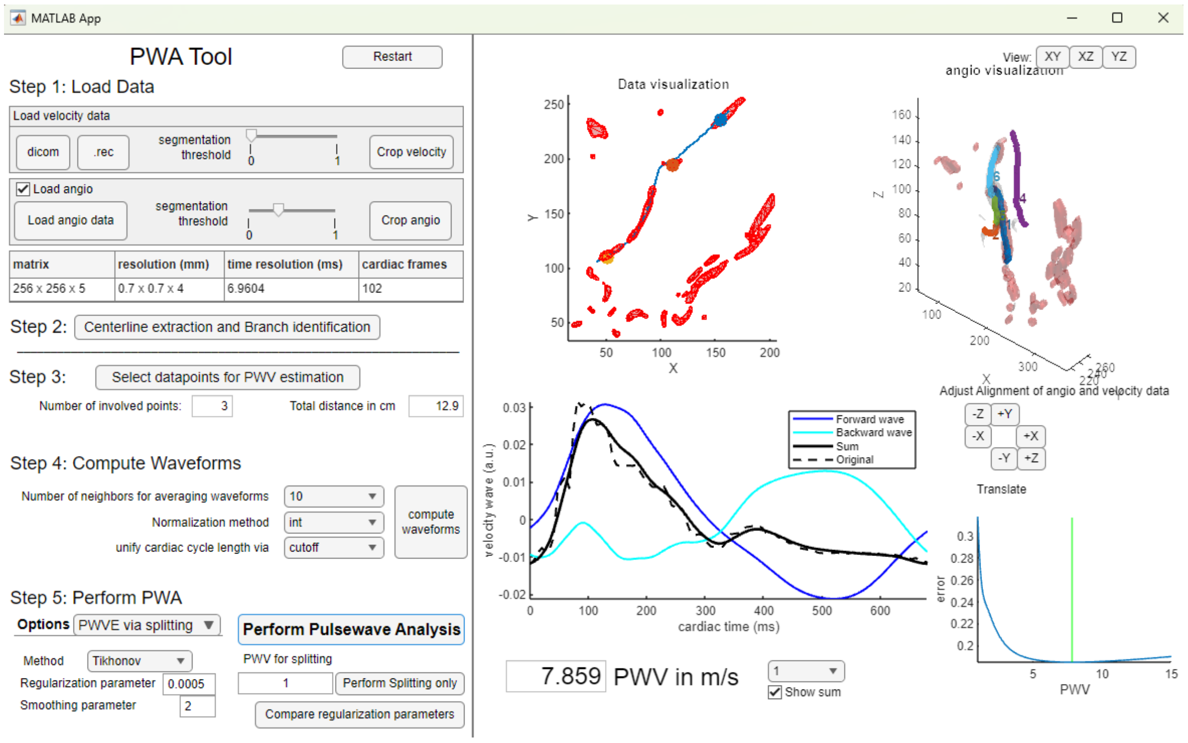
The ultimate goal of the app is to provide a complete tool for PWV-estimation (and pulse wave splitting - see Section 5.4) for MRI based methods, using blood flow velocity data, in the following settings:

- If continuous artery data is available and distance computation can be performed on the velocity data itself.
- In the case that only limited velocity data is available, but additionally 3D-TOF data is available for distance computation.
- The following data formats are currently available:
  - .rec format (tested for data obtained by Phillips scanner)
  - .dcm format (tested for data obtained by GE scanner)
- Several methods for MRI-based PWV-estimation can be performed, see Section 5.

2 Installation

The PWA-Tool is available both as source-code and .exe installation file. Matlab Runtime is required for running the app, available via `https://de.mathworks.com/products/compiler/matlab-runtime.html`. Note that the app installation file also

[*]Johann Radon Institute Linz, Altenbergerstraße 69, A-4040 Linz, Austria, (lukas.weissinger@ricam.oeaw.ac.at).



☑ Load angio

matrix	resolution (mm)	time resolution (ms)	cardiac frames
256 x 256 x 5	0.7 x 0.7 x 4	6.9604	102

Number of involved points: 3 Total distance in cm 12.9

Number of neighbors for averaging waveforms: 10
Normalization method: int
unify cardiac cycle length via: cutoff

Options: PWVE via splitting
Method: Tikhonov
Regularization parameter: 0.0005
Smoothing parameter: 2
PWV for splitting: 1

7.859 PWV in m/s

1

☑ Show sum

Figure 3.1: Main Layout after appearance of all features.

includes Matlab Runtime installation if necessary. The PWA-Tool was created on Matlab version R2022b. Once installed, the PWA-Tool .exe can be executed from its installation folder directly. Example files are available from the author upon request.

3 User Interface Overview

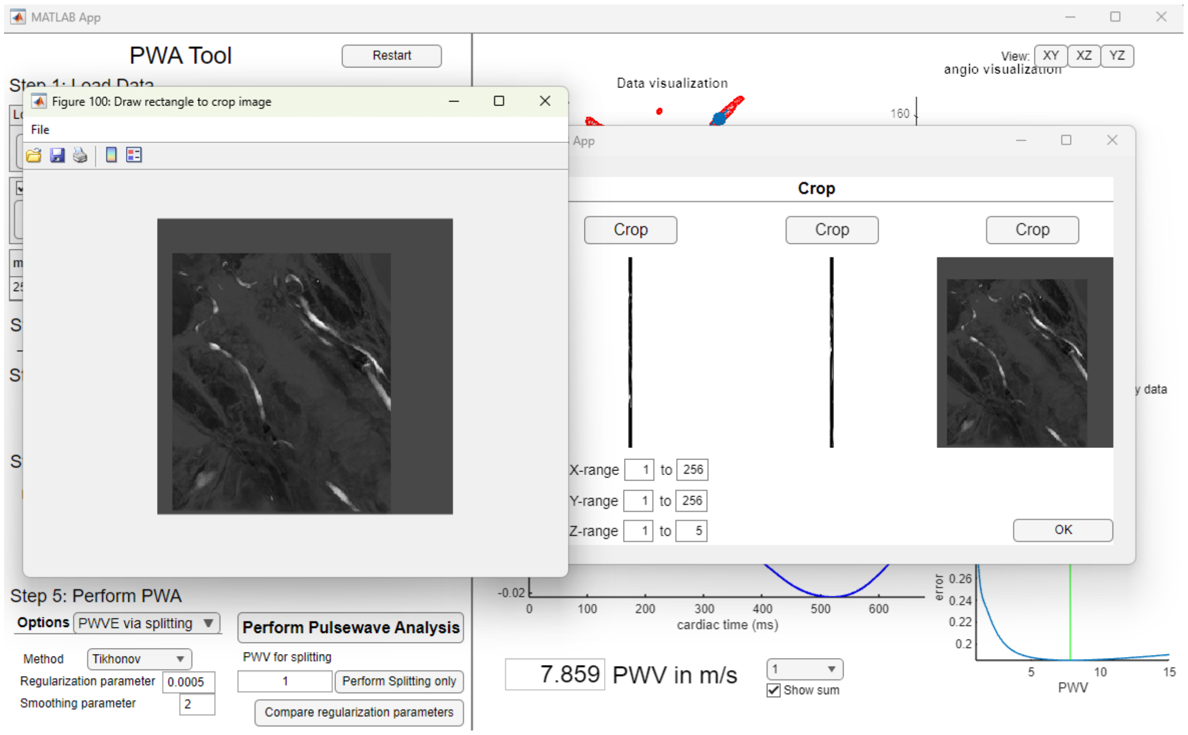
The main window of the App is separated in two panels: the options-panel on the left and visualization panel on the right side, as depicted in Figure 3.1. For the complete Pulse-wave-velocity estimation there are 5 steps that need to be performed. All options and parameters can be tuned in the options-panel, while the visualization panel shows results.

Note that the steps 1-5 depicted in the App have to be performed in order, which is why parts concerning later steps are only appearing when the preceding steps have been performed.

4 Functional Guide

Step 1: Prepare MRI data

The PWA-Tool has been designed to successfully load 4D-velocity MRI-data (obtained by phase contrast) either in .dcm (obtained by a GE scanner) or .rec (obtained by



X-range 1 to 256
Y-range 1 to 256
Z-range 1 to 5
Options PWVE via splitting
Method Tikhonov
Regularization parameter 0.0005
Smoothing parameter 2
PWV for splitting 1
7.859 PWV in m/s
1
☑ Show sum

Figure 4.1: Pop-up window to crop the data.

Phillips scanner) data format. Make sure all data of a specific data-set is stored in the same folder, where no other files of the same data format are stored and click any file to load all files of the specified data format in that folder. A segmentation into flowing/non-flowing media (voxels) is performed for the data, the segmentation threshold can be manually changed. The higher the threshold, the more media is identified as flowing media, depicted as isosurface in red in the top left plot of the visualization panel. Subsequently, you can crop the volume to the specific region of interest. For better visualization we advise to always crop data to the smallest possible region.

Optionally, 3-dimensional Time of Flight (TOF) data can be loaded, which is used for the length-computation of involved artery segments. Similarly to the 4D-velocity data, the TOF data can be cropped and the segmentation threshold can be changed. This feature is particularly helpful if the available velocity data does not cover the whole artery segment under investigation. Segmentation of the TOF data is shown in the top right plot of the visualization panel

Crop App: When a crop button is pushed for either velocity or TOF data, a pop-up window (see Figure 4.1) appears, showing maximum intensity projections (MIPs) of the data from different view angles. Above each MIP there are crop buttons, which open an additional window, in which a rectangle can be drawn to crop the data parts outside the rectangle.

Step 2: Center-line extraction and branch identification
When the data is loaded and prepared, step 2 can be performed, where a center-line

of the arteries is computed for the available segmented data. This also includes the separation of the data into several branches. If it has been loaded before, the center-line extraction will be performed on the TOF data. Furthermore, this step includes a visual overlay of TOF and velocity data, which is done automatically. The overlay and computed branches are shown in the top right plot of the visualization panel. The view angle can be changed via the buttons above the plot.

If the TOF and velocity data are slightly misplaced (e.g., if the patient moved between acquisition of TOF and velocity data), the overlay can be manually adjusted with the translate buttons in each axis direction. Please use this function with care, as the alignment should generally be close to exact and manual changes can lead to inaccurate alignments. The used algorithm is based on [11] and reused from [5].

Step 3: Select data-points for PWV estimation
Clicking on the "Select data-points for PWV estimation"-button opens a pop-up window where you can manually select points on the center-lines of the corresponding branches (see Figure 4.2). Branch numbers and time dependent wave-forms in the chosen spatial point are shown. Additionally, the branches that make up the total used artery segment (i.e., all branches that lie between the chosen points) have to be selected. Note that the branches need to be selected in correct order and orientation and make up a connected artery segment in order for the PWV-estimation methods to work.

Alternatively, the "manual selection" checkbox can be unchecked, in that case just the desired branches need to be selected (in correct order and orientation, making up a connected artery segment) and all data-points available along these branches are selected automatically. If necessary, branches can be flipped, if the orientation is wrong. Therefore, the proximal point of the branches are highlighted in the plots.

After data-points are selected, the distances between the points are calculated by a Bezier-curve approximation via De Casteljau's algorithm (see [6]) to achieve sub-pixel accuracy, which assumes certain smoothness of the arteries. The smoothed artery segment obtained by De Casteljau's algorithm is shown in the bottom right plot of the visualization panel, which can be also used to visually check if branch orientation was chosen correctly.

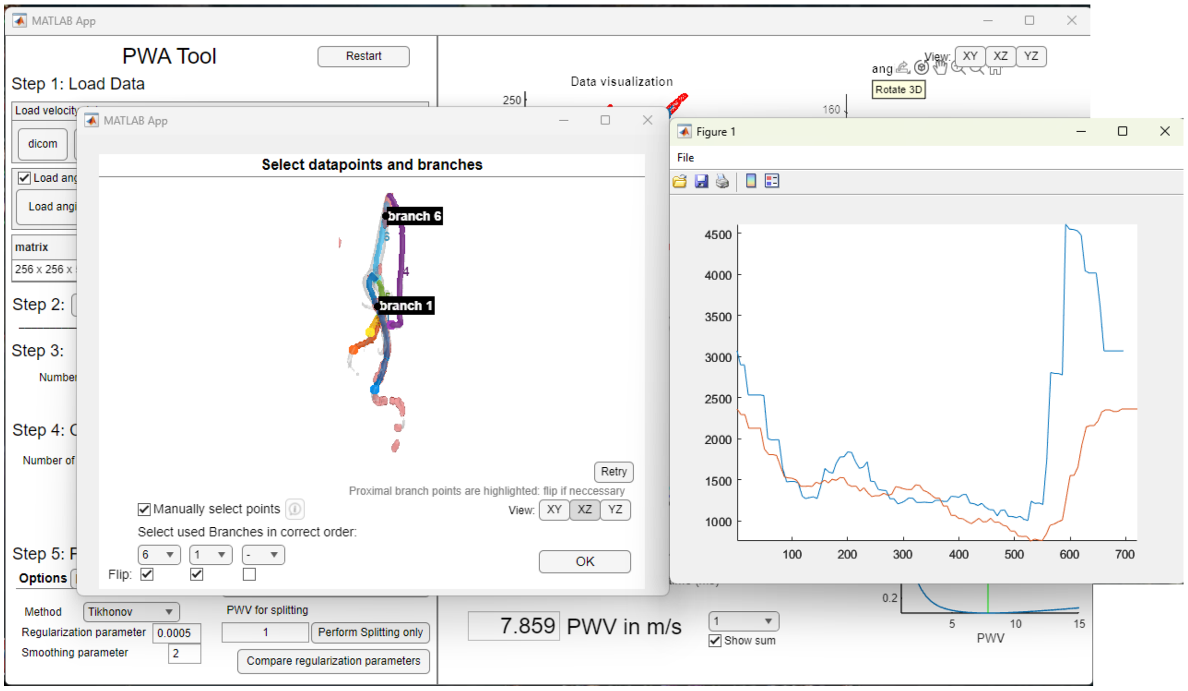
Step 4: Compute wave-forms
The wave-forms selected in step 3 might need preceding processing steps before performing PWV estimation. In particular, velocity data is unsuitable for PWV estimation if the diameter of the artery changes across the considered segment. Furthermore, the partial voluming effect (see [9]) might result in reduced amplitude of the wave-forms. As a remedy, several methods are available to normalize the given wave-forms.

Available normalization methods:

- "int": Recommended method, simulates blood flow in arbitrary unit. The sum of values for each wave are equal, equivalent to L1-normalization.

- "range": scales all waves onto the interval [0,1].

- "L2": normalizes via the euclidean norm.



☑ Manually select points

Select used Branches in correct order: 6, 1, -

Flip: ☑ ☑ ☐

Method: Tikhonov

Regularization parameter: 0.0005

Smoothing parameter: 2

PWV for splitting: 1

7.859 PWV in m/s

Figure 4.2: Window to select data-points used for PWV-estimation. An additional window shows the pulse-wave in the chosen location.

- "scale": waves are scaled by its standard deviation.

Details and further information are available in the documentation of the MATLAB function `normalize`.

To account for noise, averaging wave-forms over a certain number of neighboring points is possible and recommended as well.

Finally, variations in the length of the cardiac cycles in different voxels are possible. This can be accounted for by the following methods:

- "cutoff": Recommended method, shifts the foot of the wave to the beginning of the cardiac cycle and cuts off values exceeding the minimal available cardiac cycle length. Interpolation is used for wave-forms with cut off values, such that vector length of all pulse waves is equal.

- "average": Cardiac cycle lengths are averaged, the data-points are not changed.

The computed wave-forms are depicted in the bottom left plot of the visualization panel.

Step 5: Perform PWA

Several methods are available for PWV estimation, each one explained in detail in the next section. Note that the chosen method should also influence how data-points are selected! Visualization in the bottom plots of the visualization panel differs for each method and are explained in the corresponding section below!

5 Mathematical methods for PWV estimation

PWV estimation has been studied extensively in the past. Several methods have been proposed for MRI-data based PWV estimation. The methods that are included in the PWA-Tool are based on: Cross-correlation, wavelet cross-spectrum analysis, Maximum likelihood estimation and an inverse problems approach based on pulse wave splitting. each method is shortly explained below, further references are provided.

5.1 Cross-correlation method

This method is based on the work [7], where it is described as: "The waveform at each location at the centerline is compared to the waveform at the first location using a cross-correlation function. The cross-correlation function applies a time-shift to the more distal waveform until the highest correlation value between the two waveforms is obtained. This effectively returns a time shift for each location relative to the first location. A line is then fitted to the plot of location vs. time shift and the inverse of the slope is the PWV, this is shown in the bottom left plot of the visualization panel.

5.2 Wavelet method

This method is based on the work [1]. Similarly to the cross-correlation method, cross correlation is performed here, but for wavelet-transformed signals. Benefits of transforming into frequency domains have been shown in [8], and the wavelet-transformation has the additional benefit of time-localization, to restrict the signal in time, which is here done to restrict the signal to the systolic upstroke only, in order to reduce reflected (backwards travelling) parts. Details can be found in [1], the corresponding source-code for the method is from [11, 5]. Similarily to the cross-correlation method, the bottom left plot of the visualization panel shows the plot of location vs. time shift and the fitted line, with the inverse of the PWV as slope.

5.3 Maximum-likelihood estimator

This method is based on the work [3]. Denote the (normalized!) velocity wave-form obtained at position i as $p_i(t)$, the pulse wave velocity as u and the arterial distance between the first measurement point and point i as L_i, where $i = 1, \ldots, N$. Then this method assumes that the velocity wave-forms are identical but phase shifted, at delays given by L_i/u. Note that the cardiac cycle is characterized via the time-points t_j, $j = 1, \ldots, M$ such that the solution for p_1, u depending on the data p_{ij} is given by

$$(p_1, u) = \arg\min_{p,v} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(p(t_j - L_i/u) - p_{ij}\right)^2 . \tag{5.1}$$

Detailed description and results of the method can be found in [3]. The bottom left plot in the visualization panel shows the initial guess for the pulse-wave, which is all pulse-waves of the data averaged, and the final reconstruction of the pulse-wave.

5.4 Pulse Wave Splitting

This method is based on the work [12]. In contrast to the methods introduced before, this method takes account of the fact, that the measurable pulse waves p_i in each spatial point i consist of a forward and backward travelling part, such that $p_i(t) = p_{if} + p_{ib}$. This split waves are computed along with the PWV as well. Here, the assumption of identical but shifted pulse-waves holds for the component waves, i.e., $p_{if}(t) = p_{1f}(t - L_i/u)$ and $p_{ib}(t) = p_{1b}(t + L_i/u)$. Using this equations together with the Fourier-transform, one obtains the operator equation $F(\hat{p}_{1f}(\omega), \hat{p}_{Nb}(\omega), u) = (\hat{p}_1, \ldots, \hat{p}_N)$ for the searched values $\hat{p}_{1f}, \hat{p}_{Nb}, u$ and the operator F defined via

$$F(x_1(\omega), x_2(\omega), u) := (F_k(x_1, x_2, u))_{k=1}^N := \left(x_1(\omega)e^{i\omega L_k/u} - x_2(\omega)e^{i\omega(L_N - L_k)/u}\right)_{k=1}^N$$

This nonlinear Operator equation leads to an ill-posed problem and can be solved using methods from regularization theory. For the solution of the problem, the popular methods Tikhonov regularization [4, 10] and FISTA [2] are available in the PWA-Tool. Additionally, a constraint can be added to the reconstruction methods, stating that the Fourier-transform of the backward wave has to be smaller or equal in each component than the Fourier-transform of the forward wave.Further information on this topic can be found in [12]. Changing the regularization and smoothing parameters will both change the appearance of the reconstructed split waves: The higher the parameters, the smoother the waves. If you are insecure in how to choose the parameters, we advise to keep the default values. In FISTA, the regularization parameter equals the percentage of cut-off frequencies.

Here, the bottom left plot shows the separate forward and backward part of the wave, optionally the resulting total wave compared to the original data, for a specific data-point. The bottom right plot shows the functional values for different PWVs and the computed minimum as green line.

Compare regularization parameters: This additional feature allows to compare regularization parameters, via computing the residual error of the method and the reconstructed PWV for a specified range of regularization parameters to help choose parameters and check for robustness of the method.

While the other methods work best by maximizing the available data-points, i.e., pulse-waves at different locations, the splitting approach has not shown significant improvements by adding more data-points to the data, and that the minimum amount of 3 points appears to be sufficient already. Still, averaging over neighbouring points (see step 4) is recommended to reduce influence of noise.

6 Disclaimer

The here described Matlab tool is an extension of the "4D Flow PWV Tool" developed by Eric Schrauben [11, 5], specifically tailored to the needs of the methods developed in [12] and problems arising in PWV estimation in the human brain. This includes that in contrast to Schrauben's tool, no Flow computation is performed, which includes the computation of the area of the arterial cross-sections. Particularly for small artery diameters covered by only a small number of voxels, this would result in inaccurate computations and has been resolved by the normalization method “int”, which leads to a pulse-wave equivalent to the flow-wave. Further extensions include the manual selection of data-points and the possibility to include TOF data for distance computation, which allows for the computation of the PWV on artery-segments with no continuous velocity data available. Lastly and most importantly, the method based on the inverse problems approach for pulse wave splitting is included here, which was the main motivation for developing this app in the first place.

If you find any error, please contact the author under lukas.weissinger@ricam.oeaw.ac.at.

References

[1] Ioannis Bargiotas et al. “Estimation of aortic pulse wave transit time in cardiovascular magnetic resonance using complex wavelet cross-spectrum analysis”. In: *Journal of Cardiovascular Magnetic Resonance* 17.1 (Jan. 2015), p. 65. ISSN: 1097-6647. DOI: `https://doi.org/10.1186/s12968-015-0164-7`.

[2] A. Beck and M. Teboulle. “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems.” In: *SIAM J. Imaging Sci.* 2.1 (2009), pp. 183–202. DOI: `10.1137/080716542`. URL: `http://dx.doi.org/10.1137/080716542`.

[3] Cecilia Björnfot et al. “Assessing cerebral arterial pulse wave velocity using 4D flow MRI”. In: *Journal of Cerebral Blood Flow & Metabolism* 41.10 (2021). PMID: 33853409, pp. 2769–2777. DOI: `10.1177/0271678X211008744`. URL: `https://doi.org/10.1177/0271678X211008744`.

[4] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems.* English. Dordrecht: Kluwer Academic Publishers, 1996, pp. viii + 321. ISBN: 0-7923-4157-0.

[5] PhD Eric Schrauben. *4D Flow PWV Tool.* URL: `https://github.com/schrau24/4DFlowPWVTool`.

[6] G. Farin and D. Hansford. *The Essentials of CAGD.* CRC Press, 2000. ISBN: 9781439864111. URL: `https://books.google.at/books?id=ODFRDwAAQBAJ`.

[7] Samuel W. Fielden et al. “A new method for the determination of aortic pulse wave velocity using cross-correlation on 2D PCMR velocity data”. In: *Journal of Magnetic Resonance Imaging* 27.6 (May 2008), pp. 1382–1387. ISSN: 1522-2586. DOI: `10.1002/jmri.21387`.

[8] Antonella Meloni, Heather Zymeski, Alessia Pepe, Massimo Lombardi, and John C. Wood. “Robust estimation of pulse wave transit time using group delay”. In: *Journal of Magnetic Resonance Imaging* 39.3 (2014), pp. 550–558. DOI: `https://doi.org/10.1002/jmri.24207`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/jmri.24207`.

[9] SA Röll, Alan CF Colchester, Paul E Summers, and Lewis D Griffin. “Intensity-Based Object Extraction from 3D Medical Images Including a Correction of Partial Volume Errors.” In: *BMVC*. 1994, pp. 1–10.

[10] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational Methods in Imaging*. Applied Mathematical Sciences. Springer New York, 2008. ISBN: 9780387692777.

[11] Eric Schrauben et al. “Fast 4D flow MRI intracranial segmentation and quantification in tortuous arteries”. In: *Journal of Magnetic Resonance Imaging* 42.5 (Apr. 2015), pp. 1458–1464. ISSN: 1522-2586. DOI: 10.1002/jmri.24900.

[12] Lukas Weissinger, Simon Hubmer, Ronny Ramlau, and Henning Uwe Voss. *An Inverse Problems Approach to Pulse Wave Analysis in the Human Brain*. 2024. URL: https://arxiv.org/abs/2402.09803.