A comparison of phase unwrapping methods in velocity-encoded MRI for aortic flows

Miriam Löcke, Jeremías Garay, Pamela Franco, Sergio Uribe, Cristóbal Bertoglio

A comparison of phase unwrapping methods in velocity-encoded MRI for aortic flows

Miriam Löcke Jeremías Garay Pamela Franco Sergio Uribe
Cristóbal Bertoglio

April 9, 2022

Abstract

Purpose: The phase of a MRI signal is used to encode the velocity of blood flow. Unfortunately, phase unwrapping artefacts may appear when aiming to improve the velocity-to-noise ratio (VNR) of the measured velocity field. This study aims to compare various unwrapping algorithms at hand of ground-truth synthetic data generated using CFD simulations.

Methods: We compare four different phase unwrapping algorithms on two different synthetic datasets of 4D MRI flow as well as 26 datasets of 2D PC-MRI acquisitions including the ascending and descending aorta. The synthetic datasets are constructed using CFD simulations of an aorta with a coarctation, with different levels of spatio-temporal resolutions and noise. The error of the unwrapped images was assessed by comparison against the ground truth velocity field in the synthetic data and dual-VENC reconstructions in the in-vivo data.

Results: Using the unwrapping algorithms, we were able to remove aliased voxels in the data almost entirely, reducing the L2-error of the data compared to the ground truth by 50%-80%. Results indicated that the best choice of algorithm depend on the spatio-temporal resolution and noise level of the dataset. Temporal unwrapping is most successful on datasets with a high temporal and low spatial resolution ($\delta t = 30ms$, $h = 2.5mm$), reducing the L2-error by 70%-85%, while Laplacian unwrapping performs better with a lower temporal resolution or better spatial resolution ($\delta t = 60ms$, $h = 1.5mm$), especially for SNR 12 as opposed to SNR 15, with an error reduction of 55%-85% compared to the 50%-75% achieved by the Temporal method. The differences in performance between the different methods were found to be statistically significant.

1 Introduction

Phase contrast MRI is an important technique for evaluating vascular hemodynamics [1, 2]. This technique encodes velocity information in the data phase, so all values are restrained to the range $(-\pi, \pi)$, where the maximum encoded velocity V_{enc} is mapped to π. If V_{enc} is

set lower than the actual velocity, an image error known as "phase wrapping" occurs as the phase value is wrapped back into the range $(-\pi, \pi)$.

Conversely, choosing a too high V_{enc} decreases the velocity-to-noise ratio (VNR), thus lowering the quality of the data [3]. Since the range of measured velocities in MRI can be very large, the selection of the proper V_{enc} may involve repeated scans. This makes it attractive to post-process the acquired velocity data to correct the wraps introduced by using a low V_{enc} by phase unwrapping techinques. This can be performed in a number of different ways by assuming regularity in the spatial [4, 5, 6] or the temporal dimension [7, 8, 9] or both [10, 11].

However, so far, there has been no thorough comparison of these different algorithms. For the vast majority of the algorithms, the code is also not publicly available and hence methods cannot be compared with the same data sets. Moreover, ground truth data in subjects or experimental work would involve for instance multi-venc acquisitions – in order to get high VNR perfectly unwrapped images – which require advanced scan protocols not yet available in clinical practice.

Therefore, the aim of this article is therefore to compare four different phase-unwrapping methods using synthetic data generated using CFD simulations where a ground truth is available. We establish the advantages and drawbacks of the various methods and the cases where they are appropriate. Additionally, the code we used for each of the algorithms will be made publicly available. Moreover, we confirm some of the findings on 2D PC-MRI data sets published previously and acquired with multiple V_{enc} values.

The remainder of the paper is structured as follows. In Section 2 the phase wrapping problem is introduced and we present the different unwrapping methods. In Section 3 we describe the details of the synthetic and subject datasets as well as the implementation details of the algorithms. Section 4 contains the results of the unwrapping algorithms for the different datasets, which are then compared and discussed in Section 5. Finally, Section 6 gives some conclusions.

2 Theory

2.1 Problem Statement

In PC-MRI, the velocity information is encoded into the phase of the transverse component of the magnetization such that there is a linear relationship between phase contrast ϕ and velocity v

$$\phi = \frac{v \cdot \pi}{V_{enc}} \tag{1}$$

where the parameter V_{enc} is set by the user of the MRI scanner. Since the phase can only be measured in the interval $[-\pi, \pi)$, the V_{enc} parameter decides the range in which the velocity data can be encoded. As a result, when the true velocity exceeds the V_{enc}, *velocity aliasing* occurs, i.e. the estimated velocity will be $u_{true} - 2n \cdot V_{enc}$ instead of u_{true}. Here $n \in \mathbb{Z}$

depends on the relation between u_{true} and V_{enc}. Analogously, the estimated phase will be $\phi_w = \phi_{true} - 2n \cdot \pi$ where the true phase may now exceed the interval $[-\pi, \pi)$.

Out of the available methods, we review the three main algorithms solving the 4D phase unwrapping problem: one relying on temporal smoothness, one relying on spatial smoothness for solving a Laplacian equation over the entire domain, and one using both temporal and spatial smoothness to iteratively unwrap the picture pixel-by-pixel by assigning each pixel a probability of being wrapped.

2.2 Phase unwrapping methods

2.2.1 Temporal unwrapping

Temporal phase unwrapping has first been introduced in [8]. This method is based on the assumption that at each voxel the phase only varies slowly in time or that the temporal resolution is high enough. Inter-voxel dependency is neglected.

Given a time series of N phase maps $\phi_w(x, t_j), j = 1, \cdots, N$, the differential phase maps are computed as:

$$D_j(x) = \phi_w(x, t_j) - \phi_w(x, t_{j-1}), \quad j = 2, ..., N. \tag{2}$$

According to the aforementioned assumption, these differential maps cannot contain any phase wraps of their own. Therefore any value $|D_j(x)|$ greater than π has to be the result of a phase wrap occurring in one of the phase maps. To regain the "correct" differential value, it is sufficient to wrap the differential phase maps into the range $(-\pi, \pi)$. Once the wrapping-free differential phase maps have been computed, the unwrapped phase maps can be calculated by integrating over the differential maps, starting at a reference time frame. This requires the existence of a timeframe that does not have any wrapped pixels, which will have to be manually selected. When trying to measure blood velocities, the cardiac cycle contains times with very low velocities, e.g. in diastole, so such a reference timeframe should exist when the temporal resolution is high enough [9]. The unwrapped phase ϕ_u with a reference timeframe $\phi_w(x, t_{ref})$ at time t_{ref} can then be described as

$$\phi_u(x, t_j) = \begin{cases} \phi_w(x, t_{ref}) + \sum_{i=ref+1}^{j} D_i(x) \text{ for } t_j > t_{ref} \\ \phi_w(x, t_{ref}) - \sum_{i=j+1}^{ref} D_i(x) \text{ for } t_j < t_{ref} \\ \phi_w(x, t_{ref}) \text{ for } t_j = t_{ref} \end{cases} \tag{3}$$

This algorithm is simple to implement and efficient in terms of computational effort. However, it is requires a sufficiently high temporal resolution as well as a wrap-free reference frame, which has to be manually selected. The method can unwrap multiple wraps (which occur if the V_{enc} is less than 50% of the maximal velocity), as long as they occur in separate timesteps, i.e. a wrapped pixel in one timeframe being wrapped again in a later timeframe. This method is sensitive to noise and may falsely unwrap pixels due to noise. Movement of the aorta due to the contraction of the heart also negatively impacts the results for this method as it breaks the continuity of the pixels between the timeframes.

2.2.2 4D gradient-based unwrapping

In gradient-based unwrapping methods [10] each pixel is assigned a probability that it is wrapped based on the gradients with the neighbouring pixels, and all pixels with a probability higher than a certain threshold are unwrapped. This method evaluates the data pixel-by-pixel and often requires several consecutive iterations. The method assumes smoothness in both space and time. The probability that a pixel is wrapped is estimated by the gradients of the neighboring pixels to the evaluated pixel via

$$P(x,t_j) = \frac{\sum_{i\in N(x)} w_s(\phi_w(i,t_j) - \phi_w(x,t_j)) + \sum_{t_i\in N(t_j)} w_t(\phi_w(x,t_i) - \phi_w(x,t_j)}{2\pi \cdot (\#N(x)w_s + \#N(t_j)w_t)} \tag{4}$$

where $N(x)$ is the set of the spatial neighbours of the pixel x and $N(t_j)$ is the set of the temporal neighbours of the time position t_j. w_s, w_t are weighting factors. It is recommended to set w_s to 1 for the spatial dimension and w_t to 2.5 for the temporal dimension. Each step is detailed in Algorithm 1.

It is suggested to use a dual threshold approach, where the algorithm first completes a number of iterations with a low threshold in order to break up larger regions of wrapped, and then uses a higher threshold for further iterations to accurately unwrap the remaining pixels. The suggestion for the optimal thresholds are $r_{low} = 0.32$ and $r_{high} = 0.75$, for 50 iterations each. This algorithm is very simple to implement and requires little computational effort per iteration. However, depending on the number of iterations, the time required may increase above that of other methods. This method can also only unwrap single wraps and therefore requires that $V_{enc} > 50\%$ of the maximal velocity to unwrap successfully.

Algorithm 1: 4D gradient based unwrapping algorithm

Given a wrapped phase image ϕ_w, a set of thresholds r_{low} and r_{high}, and iteration numbers n_{low}, n_{high}:
Set $\phi = \phi_w$
Repeat n_{low} times:
foreach $(x,j) \in \Omega \times \{1,\cdots,N\}$ **do**
    compute $P(x,t_j)$
    if $P(x,t_j) > r_{low}$ **then**
        $\phi(x,t_j) = \phi(x,t_j) + 2\pi$
if $P(x,t_j) < -r_{low}$ **then**
    $\phi(x,t_j) = \phi(x,t_j) - 2\pi$
Repeat again n_{high} times with r_{high} instead of r_{low}
Return ϕ

2.2.3 Laplacian unwrapping

Laplacian unwrapping is a popular single-step technique that can be used with 2D, 3D, and 4D Flow MRI data. Starting from the relationship

$$\phi(x,t) = \phi_w(x,t) + 2\pi n(x,t) \tag{5}$$

the method assumes the continuity and differentiability of ϕ and ϕ_w and aims to first find the Laplacian of the true phase. Specifically, the method is obtained by first showing the relation:

$$\Delta e^{i\phi(x,t)} = -e^{i\phi(x,t)}\|\nabla\phi(x,t)\|_2^2 + ie^{i\phi(x,t)}\Delta\phi(x,t),$$

with Δ and ∇ the spatio(-temporal) Laplacian and gradient operators, respectively. By combining it with Equation (5), the following equation is obtained:

$$\Delta\phi(x,t) = \mathrm{Im}(e^{-i\phi_w(x,t)}\Delta e^{i\phi_w(x,t)}). \tag{6}$$

Simplifying the right hand side leads to

$$\Delta\phi(x,t) = \cos\phi_w(x,t)\Delta\sin\phi_w(x,t) - \sin\phi_w(x,t)\Delta\cos\phi_w(x,t) \tag{7}$$

Therefore, the unwrapped phase is computed by solving the problem:

$$\phi_u(x,t) = \tilde{\Delta}^{-1}\left(\cos\phi_w(x,t)\Delta\sin\phi_w(x,t) - \sin\phi_w(x,t)\Delta\cos\phi_w(x,t)\right)$$

where $\tilde{\Delta}$ denotes the Laplacian operator but applying some set of boundary conditions. In the case of a spatio-temporal Laplacian operator for 4D datasets, it's necessary to introduce a scaling constant for the temporal dimension. In practice, this scaling is quite robust and was simply set to $1s^2/cm^2$, as it was in [4]. Note that, in discrete data, the Laplacian operation on the right-hand-side's terms is approximated by the sum of the second derivatives along each dimension.

The results differ depending on the chosen boundary conditions – Periodic, Dirichlet or Neumann. In this work, periodic boundary conditions were used on all boundaries, as the implementation is provided by [4].

The Laplacian method can be computationally expensive, depending on the method used to apply the inverse Laplacian and the size of the input data, but it can unwrap nested wraps and it is reported to handle noise well since it provides a simultaneously smoothing field.

3 Methods

3.1 Synthetic data

A computational fluid dynamics simulation in physiological regimes was generad as detailed in Appendix A. Let u_{true} be the velocity field generated by this simulation. Then the MRI magnetization measurements were modeled in the following way:

$$M^u_{meas} = C\exp(i(\phi^0 + \pi u_{true}/V_{enc})) + \epsilon^u \tag{8}$$

where ϕ^0 is the background phase, C corresponds to the magnitude, and ϵ is a Gaussian measurement noise with a mean of zero. PC-MRI requires an additional measurement to recover an estimate of u_{true} since the background phase is unknown. This can by done by turning off the motion encoding, resulting in the magnetization

$$M^0_{meas} = C\exp(i(\phi^0)) + \epsilon^u. \tag{9}$$

Now the velocity can be estimated from the two magnetizations via the equation

$$u_{meas} = \frac{\angle\exp(i(\angle M^u_{meas} - \angle M^0_{meas}))}{\pi} V_{enc}. \tag{10}$$

These synthetic measurements are interpolated into a tetrahedral box mesh placed around the aorta geometry. We consider two different resolutions for the box mesh: $1.5\times1.5\times1.5mm^3$ and $2.5\times2.5\times2.5mm^3$. The reference solution is also undersampled in time with $\tau_1 = 0.03s$ or $\tau_2 = 0.06s$, leading to a total of 28 respectively 14 time steps. Two different noise levels of 15dB or 12dB in the complex magnetization was applied, and two V_{enc}s were set to 120% and 60% of the maximum velocity, resulting in $192cm/s$ and $96cm/s$, respectively. Only the velocity in the foot-head direction was acquired.

Next to the 4D Flow aorta measurements, we also generated synthetic 2D "slice" measurements by selecting a single 2D slice perpendicular to the foot-head direction and interpolating the velocity into a rectangular hexahedral mesh with a spatial resolution of $2.5\times2.5mm^2$. The sub-sampling in time as well as the noise levels remained the same. The chosen V_{enc}s were $153cm/s$ and $77cm/s$ corresponding to 120% and 60% of the maximum velocity respectively.

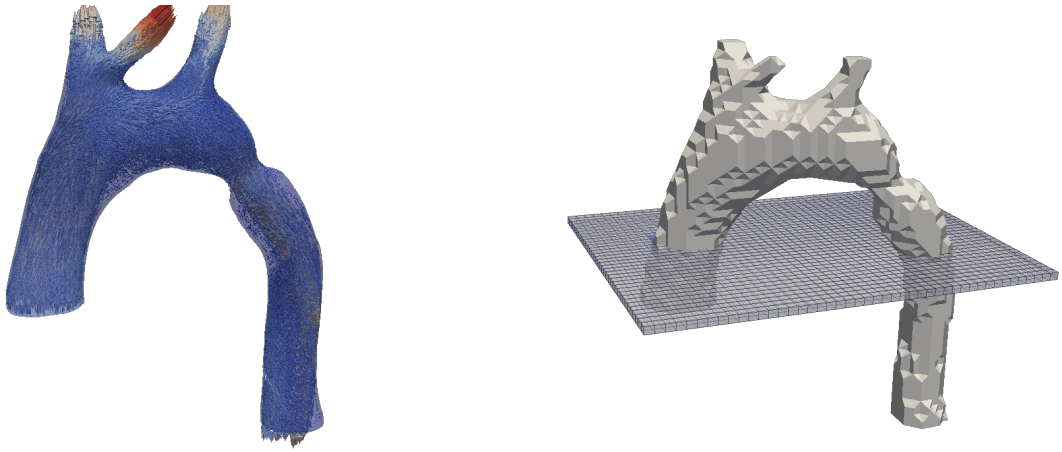
Fifty independent realizations of each noise level per V_{enc} and spatial resolution were made. The results in Section 4 therefore show the averages and standard deviations over these fifty realizations.

3.2 Subject's data

We used a database of 2D PC-MRI of 26 subjects originally reported in [13]. Measurements were acquired in a 1.5T Achieva scanner with a 5 elements cardiac coil. The following scan parameters were used: FA = 15°, TR = 5.5ms, TE = 3.7ms, matrix size = (320, 232). The acquired data are slices perpendicular to the ascending aorta just above the valsalva sinus, in 2D plus time. The spatial resolution was $1mm\times1mm$ and the temporal resolution ranged from 35ms to 48ms depending on the patient's heart rate. Two different V_{enc}s were used: 150cm/s and 75cm/s.

The MRI images were segmented using MATLAB so that non-zero entries are only present in the ascending and descending aorta.

To investigate the noise sensitivity, the experiments were repeated with a Gaussian noise with a zero mean and a standard deviation of 0.15π added to the phase of the segmented measurements.



(a) CFD velocity solution at $t = 0.24s$.

(b) 3D aorta and slice mesh used for the measurements

Figure 1: Reference solution of the blood flow from the CFD simulation and meshes used for simulating measurements

3.3 Implementation

All algorithms were implemented in both Python and Matlab. The codes are included as supplementary material, including the input CFD data and scripts generating the results.

For the Temporal method, the reference timeframe was selected at diastole, where the flow velocity is low enough to ensure to be aliasing-free. For the synthetic data, this frame was the first time frame. For the volunteer data, the last time frame was used instead. The selection of the reference timeframe does not change the results if it is aliasing-free.

For the Probability method, we used the threshold values suggested in [10], i.e. $t_{low} = 0.32$ and $t_{high} = 0.75$. Their work suggests using 50 iterations of either threshold, but we adjusted this value empirically to 10 iterations of either threshold.

The implementation of the Laplacian method is a translation of the code provided with [4] from MATLAB to Python. This provides both an implementation in 3D as well as 4D.

3.4 Error assessment

We compare the unwrapping methods on the error norm

$$e = \frac{\|u_{unwrapped} - u_{ref}\|}{\|u_{ref}\|} \tag{11}$$

where u_{ref} is a reference solution. For the synthetic data, u_{ref} is the ground truth, i.e. the result of the CFD simulation without noise or aliasing. For the in-vivo data, the reference solution is gained by unwrapping the data with the multi-V_{enc} algorithm OMME [12], which is an extension of the dual-V_{enc} unwrapping algorithm presented in [13], with $V_{enc1} = 150cm/s$ and $V_{enc2} = 75cm/s$.

To assess the statistical significance of the difference between the unwrapping results, a Friedman test with a post-hoc Wilcoxon Signed-Rank test was used, with $p < 0.05$ being considered statistically significant after a Bonferroni correction to account for multiple comparisons. The statistical analysis was performed in Python.

4 Results

4.1 Computing times

The processing times of the studied algorithms, running on a notebook with 16GB RAM and an Intel Core i5 processor, are shown in Table 1.

Algorithm	Average time			
	Aorta ($h = 2.5mm$)	Aorta ($h = 1.5mm$)	2D slice	PC-MRI
Temporal	0.131s	0.247s	0.019s	0.011s
Spatial Laplacian	0.605s	1.344s	0.076s	0.123s
Spatiotemporal Laplacian	0.933s	1.966s	0.074s	0.247s
4D Gradient	81.915s	163.001s	8.081s	0.307s

Table 1: Processing times of the algorithms for one realization

4.2 Results

4.3 Synthetic data

4.3.1 3D data

Figures 2 and 4 show the errors and standard deviation for each unwrapping method for all considered spatial and temporal resolutions for the Aorta mesh and the slice mesh respectively. Also included are the errors for the wrapped data and unwrapped with OMME [12] using $V_{enc1} = 120\%$ and $V_{enc2} = 60\%$ of V_{max}. The reference solution used to compute the error is the ground truth of the fluid simulation without noise.

In Figure 2(a), it can be seen that all unwrapping methods lead to an improvement in the error over the original data. The Temporal method is very close to the baseline OMME solution, whereas the other methods perform worse. The 4D gradient method has a significantly higher error with a larger standard deviation than the other methods. The pattern persists with the increase in noise for SNR 12.

Using the finer grid in 2(b), all methods except 4D gradient perform significantly better, with the Spatiotemporal Laplacian now achieveing similar errors to the Temporal method for both noise levels. Additionally, the standard deviation of the errors has decreased.

For SNR12, the differences between the Temporal results and the OMME results were not statistically significant ($p = 0.5$ for $h_1 = 2.5mm$ and $p = 0.11$ for $h_3 = 2.5mm$),

indicating that the Temporal method performs at the same level as OMME with regards to both means and variance in these cases. All other methods were pairwise statistically significantly different ($p < 0.05$).

For the lower temporal resolution of $\delta t = 60ms$ in 2(c), the Temporal method has a significantly higher error and larger standard deviation as compared to $\delta t = 30ms$. Also, the method's sensitivity to noise appears to have increased. The other methods are not impacted significantly by the change in temporal resolution. As a result, the 4D Laplacian now has similar errors as the Temporal method and lower ones for SNR 12.

This pattern persists in 2(d) with the higher spatial resolution. This improves the error values for all methods but the Temporal method, leading to the 4D Laplacian having a significantly lower error and standard deviation. For SNR 12, the 3D Laplacian outperforms the Temporal method as well.

For the lower temporal resolution, the differences in the performance of the methods were in each case found to be statistically significant for each pairwise combination of methods.

Figure 3 shows the results of the unwrapping methods on the Aorta mesh on an example 2D slice including the ascending and descending aorta. It can be seen that the Temporal method and both Laplacian methods unwrap perfectly or nearly perfectly, whereas the 4D Gradient only leads to a very partial unwrap.

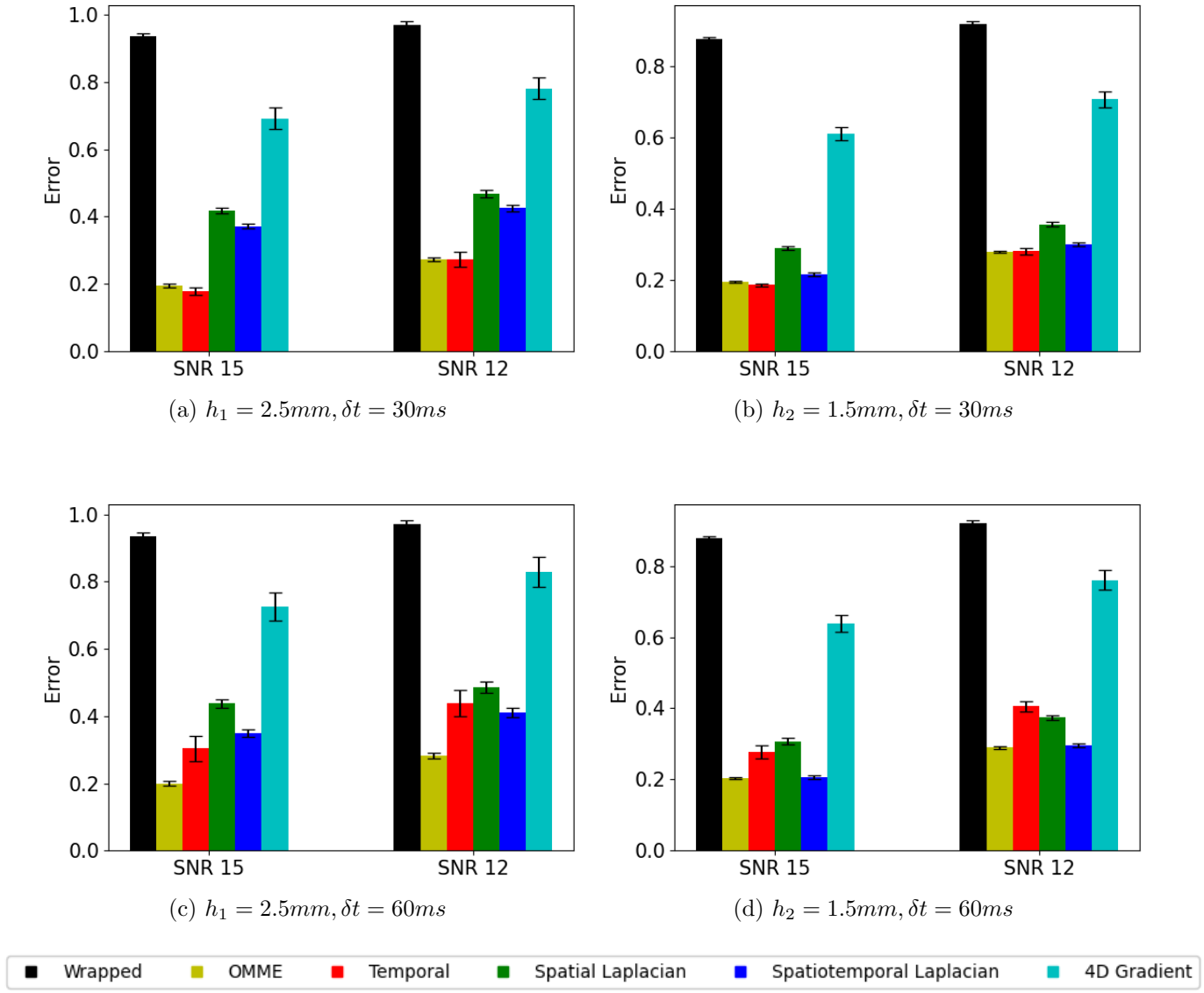
4.3.2 2D data

The results on the 2D slices with hexagonal elements are depicted in Figure 4. Here, all methods except for the 4D gradient method perfom very similarly for $\delta t = 30ms$. The Spatiotemporal Laplacian has a much lower standard deviation than the other methods, however, especially for SNR 12. For the coarser temporal resolution, the Temporal method does significantly worse than the Laplacian methods, with an error twice as large as that of the Spatiotemporal Laplacian and a much larger standard deviation. The 4D gradient method does not lead to a significant reduction in the error. The only methods without statistically significant differences were OMME and Temporal for $\delta t = 30ms$ ($p = 1$ and $p = 0.17$, respectively) as well as OMME and 4D Laplacian for SNR15, $\delta t = 30ms$ ($p = 0.28$), Temporal and 4D Laplacian also for SNR15, $\delta t = 30ms$ ($p = 1$), and Temporal and 3D Laplacian for SNR 15, $\delta t = 60ms$ ($p = 0.13$).

Examples for the unwrapping results on the 2D slice data can be seen in Figure 5, with a temporal resolution of $\delta t = 30ms$ and a noise level of SNR 12. It can again be observed that the Temporal method unwraps most wrapped voxels, but also generates a falsely wrapped voxel with a very high velocity. The Spatial Laplacian leads to a good unwrap with only a few wrapped pixels remaining, while the Spatiotemporal Laplacian unwraps perfectly. However, the 4D gradient method only leads to a partial unwrap, with many aliased voxels remaining.

4.4 Subject's data

The errors of the unwrapping methods on each of the volunteer MRI acquisitions are depicted in Figure 6. The reference solution used here is acquired by applying OMME to the MRI



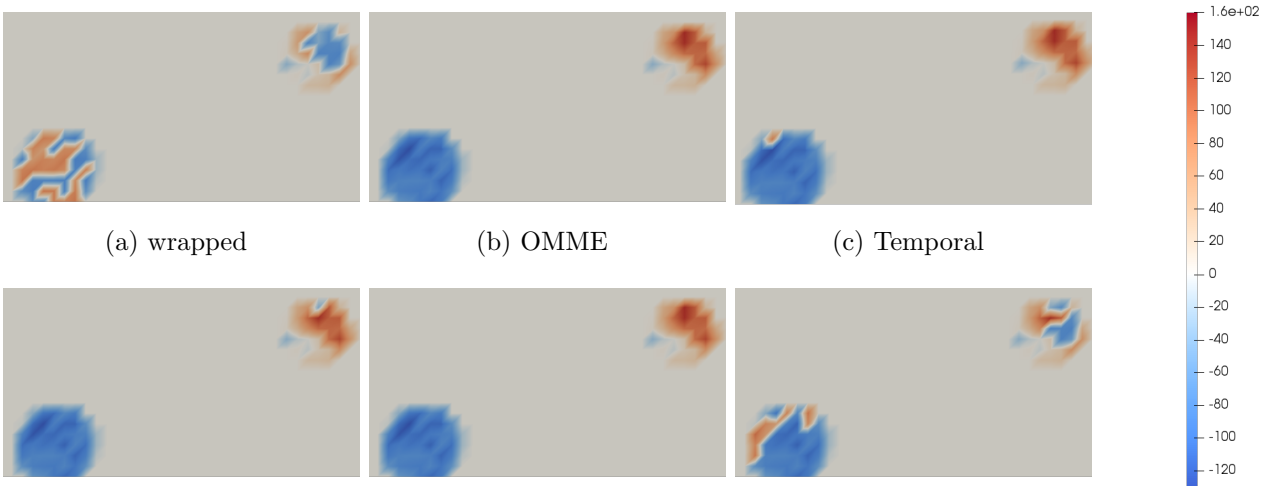
(a) $h_1 = 2.5mm, \delta t = 30ms$

(b) $h_2 = 1.5mm, \delta t = 30ms$

(c) $h_1 = 2.5mm, \delta t = 60ms$

(d) $h_2 = 1.5mm, \delta t = 60ms$

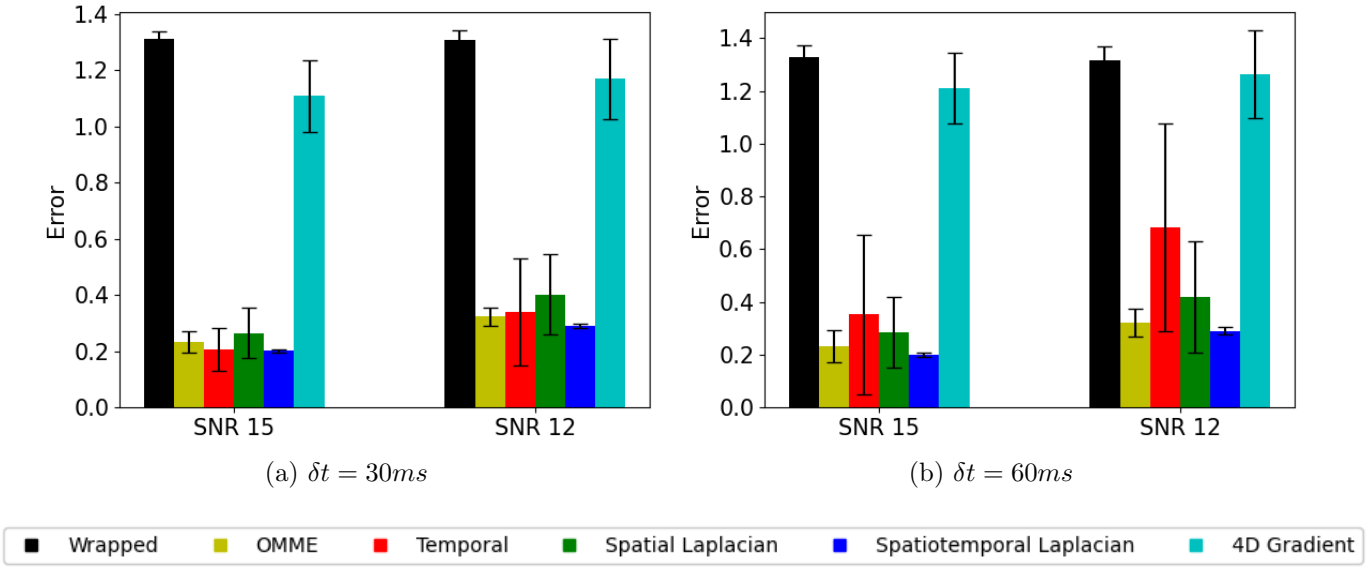
Figure 2: Error by each method for different mesh sizes and δt on the Aorta mesh



(a) wrapped (b) OMME (c) Temporal

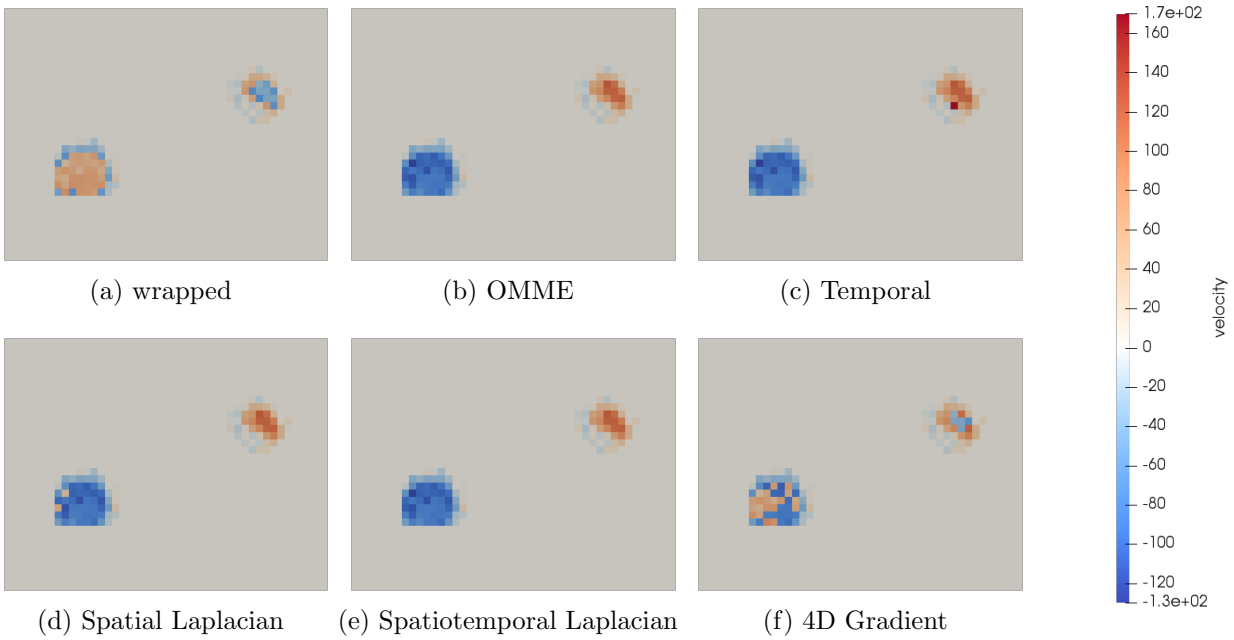
(d) Spatial Laplacian (e) Spatiotemporal Laplacian (f) 4D Gradient

Figure 3: Examples of the unwrapped velocity on the Aorta dataset, sliced at $z = 5.7cm$, with SNR 12, $\delta t = 60ms$, at $t = 0.18$, on the coarse mesh with spatial resolution $h_1 = 2.5mm$.



(a) $\delta t = 30ms$ (b) $\delta t = 60ms$

Figure 4: Error by each method for different δt on a slice taken at $z = 5.7cm$.



(a) wrapped (b) OMME (c) Temporal

(d) Spatial Laplacian (e) Spatiotemporal Laplacian (f) 4D Gradient

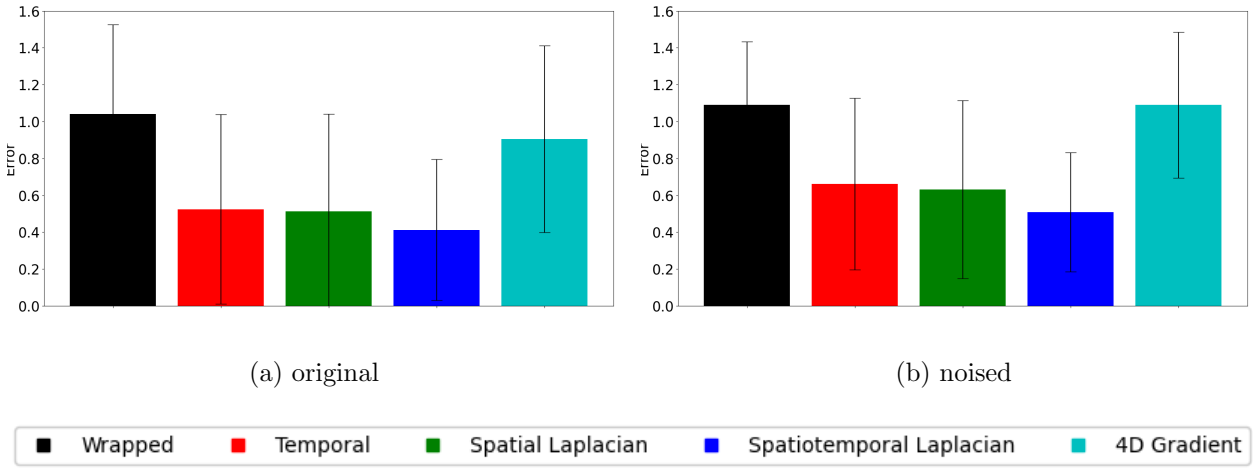
Figure 5: Examples of the unwrapped velocity on a slice taken at $z = 5.7cm$, with SNR 12, $\delta t = 30ms$, at $t = 0.18$

data with $V_{enc1} = 150cm/s$ and $V_{enc2} = 75cm/s$. Therefore, the OMME results are no longer displayed in the graph.

It can be seen that the reduction in error is lesser than for the synthetic data Still the error is on average reduced by unwrapping by more than 50%. The Temporal method and Spatial Laplacian method perform similarly on this dataset, while the Spatiotemporal Laplacian has the lowest error and standard deviation of all methods. The 4D gradient method, as with the synthetic data, only marginally reduces the error.
For the errors on the noised data in Figure 6 show that the errors for all methods increased slightly, but the comparative performance of the methods remained the same.

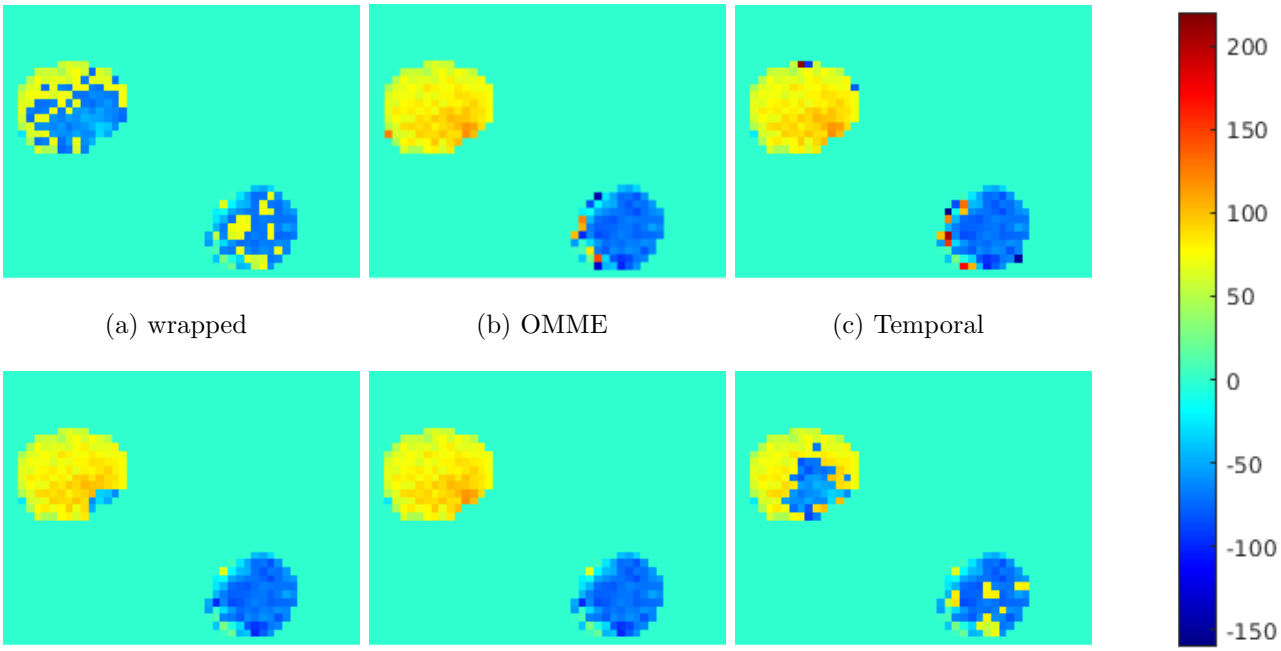
For this type of data, the differences between Temporal and 3D Laplacian were not statistically significant for both the original and noised data ($p = 0.79$ and $p = 0.35$, respectively). Additionally, for the noised data, there were no statistically significant differences between the input data and the results of the 4D Gradient method ($p = 0.88$), confirming that no effective unwrapping was done. All other methods were statistically significantly different from both the input data and each other.

Figure 7 shows example pictures of the unwrapping results from volunteer 7. The Temporal method unwraps very well except for several voxels on the edges of the ascending and descending aorta, where the movement of the aorta disrupts the temporal continuity of the voxels between one timestep and the next. The spatial Laplacian method fails to unwrap a small area on the edge of the ascending aorta, but unwraps everything else successfully .The spatiotemporal Laplacian methods removes all aliased voxels. In contrast, the 4D gradient



(a) original (b) noised

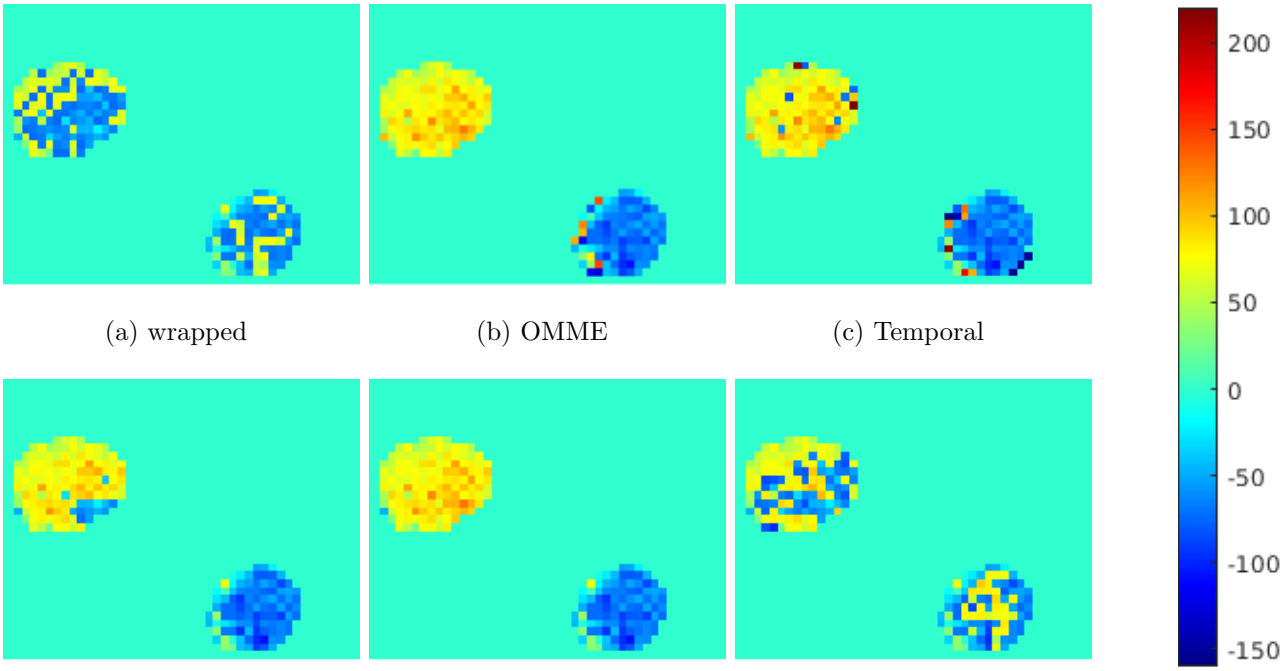
Figure 6: Average error and standard deviation by each method on the 2D PC-MRI data from 26 different subjects.



(a) wrapped (b) OMME (c) Temporal

(d) Spatial Laplacian (e) Spatiotemporal Laplacian (f) 4D Gradient

Figure 7: Examples of the unwrapped velocity on MRI data acquisition from volunteer 7 at $t = 4$



(a) wrapped (b) OMME (c) Temporal

(d) Spatial Laplacian (e) Spatiotemporal Laplacian (f) 4D Gradient

Figure 8: Examples of the unwrapped velocity on MRI data acquisition with added noise from volunteer 7 at $t = 4$

method removes only a subset of aliased voxels, leading to a partial, fractured unwrapping. The results of the data with additional noise are shown in Figure 8. It can be seen that all the methods except 4D Gradient deliver similar results to the original data except for a few additional falsely wrapped voxels, while the unwrapping results of 4D Gradient are significantly worse, with no effective unwrapping done.

5 Discussion

The results on the different datasets are consistent with each other in regards to the relative performance of the different unwrapping methods. This shows that the performance of the uwrapping algorithms on the synthetic data used here is likely a good indicator of their performance on in-vivo data.

On each of the datasets, the Temporal method and the Spatiotemporal method generally outperform the Spatial Laplacian and the 4D gradient method. The 4D gradient method fails to perfom adequately, as the resulting images are still heavily aliased, and is additionally more computationally expensive than the other methods due to the number of iterations. The Spatial Laplacian offers no advantage over the Spatiotemporal Laplacian and performs worse than the Temporal method in almost all cases as well, with the exception of the PC-MRI acquisitions, where there is no statistically significant difference between the Spatial Laplacian and the Temporal method. It is only slightly less computationally expensive than using the higher-dimensional Laplacian and significantly more expensive than the Temporal method.

With a small timestep, the Temporal method achieves equal or better results than the Spatiotemporal method, especially on a coarser mesh. It is also significantly cheaper to compute than the 4D Laplacian. However, it is more sensitive to noise, and has a much larger standard deviation, especially on the 2D datasets. As a result, the Spatiotemporal Laplacian outperforms the Temporal method on datasets with a finer grid, a larger timestep, or more noise. It is overall the most reliable method and often generates completely unaliased images.

An interesting aspect is that the Spatiotemporal method does not drastically outperform the Temporal method on the full 4D Aorta datasets, despite having the advantage of using an additional spatial dimension as compared to the PC-MRI slice datasets.

A limitation of this study is that we have not covered here the performance of the algorithms of data with a lower V_{enc} ($< 50\%$ of the maximum velocity). That scenario is however, relatively unrealistic in clinical practice.Moreover, the algorithms were also not tested on full 4D Flow in-vivo MRI acquisitions, nevertheless the synthetic data presented here is realistic in terms of spatio-temporal resolution and noise.

6 Conclusion

In this work we compared four different algorithms to correct the aliased data and retrieve correct velocity values. Using two different synthetic datasets and in-vivo data, we have shown the advantages and disadvantages of the algorithms and have come to the conclusion that the Spatiotemporal Laplacian described in [4] outperforms the others in terms of performance and reliability with acceptable computational cost. The Temporal method is computationally cheaper, but more sensitive to noise and low temporal resolution, and is therefore only preferable if the temporal resolution is high compared to the spatial resolution or computational capacities are highly limited. The other two algorithms (Spatial Laplacian and 4D gradient) offer no advantages compared to these two.

Acknowledgements

C.B. and M.L. acknowledge the funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 852544 - CardioZoom).

A Reference solution setup

The synthetic flow data is generated on the domain Ω, which is the lumen of the aortic geometry shown in Figure 9. The boundary $\partial\Omega$ of the geometry is partitioned as

$$\partial\Omega = \Gamma_{in} \cup \Gamma_w \cup (\cup_{\ell=1}^{4} \Gamma_\ell) \tag{12}$$

where Γ_{in} is the inflow boundary, Γ_w the aortic wall, and Γ_ℓ the outlet boundaries. On this domain we consider the incompressible Navier-Stokes equations

$$\begin{cases} \rho\dfrac{\partial \vec{u}}{\partial t} + \rho\big(\vec{u}\cdot\nabla\big)\vec{u} - \mu\Delta\vec{u} + \nabla p = 0 \quad \text{in} \quad \Omega\times[0,T] \\ \nabla\cdot\vec{u} = 0 \quad \text{in} \quad \Omega\times[0,T] \\ \vec{u} = \vec{u}_{inlet} \quad \text{on} \quad \Gamma_{in}\times[0,T] \\ \vec{u} = \vec{0} \quad \text{on} \quad \Gamma_w\times[0,T] \\ \mu\dfrac{\partial \vec{u}}{\partial n} - p\vec{n} = -P_\ell(t)\vec{n} \quad \text{on} \quad \Gamma_\ell\times[0,T], \quad \ell = 1,\dots,4, \end{cases} \tag{13}$$

for the velocity $\vec{u}(x,t)$ and the pressure $p(x,t)$ and with the density $\rho = 1.2gr \cdot cm^2$, the dynamic viscosity $\mu = 0.035P$, and with $P_\ell(t)$ given by the *three-element Windkessel model*:

$$\begin{cases} P_\ell = R_{p,\ell}\, Q_\ell + \pi_\ell \\ Q_\ell = \displaystyle\int_{\Gamma_\ell} \vec{u} \cdot \vec{n} \\ C_{d,\ell} \dfrac{d\pi_\ell}{dt} + \dfrac{\pi_\ell}{R_{d,\ell}} = Q_\ell. \end{cases} \tag{14}$$

Here $R_{p,\ell}$ and $R_{d,\ell}$ represent the resistance of the vasculature proximal and distal to Γ_ℓ, respectively, while $C_{d,\ell}$ models the compliance of the distal vessels. $\vec{n}$ is the exterior normal vector of $\partial\Gamma$, and the initial conditions $\vec{u}(x,0) = \vec{0}$, $\pi_1(0), \ldots, \pi_4(0) = 85mmHg$ are given.

The discretized model is solved using a modified semi-implicit Chorin-Temam scheme with a backwards Euler scheme for the time derivative.

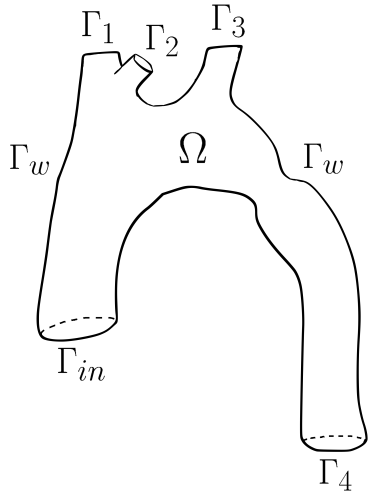


Figure 9: 3D aortic model geometry with boundaries

The boundary condition at the inlet is a pulsatile flow described by

$$\vec{u}_{inlet} = -U f(t) \vec{n}, \tag{15}$$

where $U = 75cm/s$ is a constant amplitude and $f(t)$ is defined as

$$f(t) = \begin{cases} sin(\frac{\pi t}{T} \text{ if } t \leq T \\ \frac{t}{T}(t-T)e^{-k(t-T)} \text{ if } T_c > t > T \end{cases} \tag{16}$$

with $T = 0.36s$ being the opening time of the valve, $T_c = 0.80s$ the duration of the cardiac cycle, and $1/k = 1/70s$ the typical time for the closing of the aortic valve. The parameters of the Windkessel boundary conditions are given in Table 2.

	Γ_1	Γ_2	Γ_3	Γ_4
R_p $(dyn \cdot s \cdot cm^{-5})$	480	520	520	200
R_d $(dyn \cdot s \cdot cm^{-5})$	7200	11520	11520	4800
C $(dyn^{-1} \cdot cm^5)$	$4 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$4 \cdot 10^{-4}$

Table 2: Numerical values of the three-element Windkessel parameters for every outlet

The reference solution for the velocity was established by solving the described model on a mesh consisting of 2,752,064 tetrahedrons and 510,755 vertices, resulting in an average element diameter of 1.1mm. The time step is set as $\tau = 0.001$.

References

[1] M. Markl, A. Frydrychowicz, S. Kozerke, M. Hope, and O. Wieben. "4D flow MRI". In: Journal of Magnetic Resonance Imaging 36.5 (2012), pp. 1015–1036.

[2] M. Markl, F. P. Chan, M. T. Alley, K. L. Wedding, M. T. Draney, C. J. Elkins, D. W. Parker, R. Wicker, C. A. Taylor, R. J. Herfkens, and N. J. Pelc. "Time-resolved three-dimensional phase-contrast MRI". In: Journal of Magnetic Resonance Imaging 17.4 (2003), pp. 499–506.

[3] T. E. Conturo and G. D. Smith. "Signal-to-noise in phase angle reconstruction: Dynamic range extension using phase reference offsets". In: Magnetic Resonance in Medicine 15.3 (1990), pp. 420–437.

[4] M. Loecher, E. Schrauben, K. Johnson, and O. Wieben. "Phase unwrapping in 4D MR flow with a 4D single-step Laplacian algorithm". In: Journal of Magnetic Resonance Imaging 43 (Sept. 2015), n/a–n/a.

[5] A. Bhalerao, C.-F. Westin, and R. Kikinis. "Unwrapping phase in 3D MR phase contrast angiograms". In: CVRMed-MRCAS'97. Ed. by J. Troccaz, E. Grimson, and R. Mösges. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 191–202.

[6] J. Dong, F. Chen, D. Zhou, T. Liu, Z. Yu, and Y. Wang. "Phase unwrapping with graph cuts optimization and dual decomposition acceleration for 3D high-resolution MRI data". In: Magnetic Resonance in Medicine 77.3 (2017), pp. 1353–1358.

[7] M. Untenberger, M. Hüllebrand, L. Tautz, A. A. Joseph, D. Voit, K. D. Merboldt, and J. Frahm. "Spatiotemporal phase unwrapping for real-time phase-contrast flow MRI". In: Magnetic Resonance in Medicine 74.4 (2015), pp. 964–970.

[8] K. Itoh. "Analysis of the phase unwrapping problem". In: Applied optics 21 (July 1982), p. 2470.

[9] Q.-S. Xiang. "Temporal phase unwrapping for cine velocity imaging". In: Journal of Magnetic Resona
5.5 (1995), pp. 529–534.

[10] M. Loecher, K. Johnson, B. Landgraf, and O. Wieben. “4 D Gradient Based Phase Unwrapping for PC-MR Flow Data”. In: 2010.

[11] E. Barnhill, P. Kennedy, C. L. Johnson, M. Mada, and N. Roberts. “Real-time 4D phase unwrapping applied to magnetic resonance elastography”. In: Magnetic Resonance in Medicine 73.6 (2015), pp. 2321–2331.

[12] H. Herthum, H. Carrillo, A. Osses, S. Uribe, I. Sack, and C. Bertoglio. “Multiple motion encoding in Phase-Contrast MRI: A general theory and application to elastography imaging”. working paper or preprint. July 2021.

[13] H. Carrillo, A. Osses, S. Uribe, and C. Bertoglio. “Optimal Dual-VENC (ODV) Unwrapping in Phase-Contrast MRI”. English. In: Ieee transactions on medical imaging 38.5 (May 2019), pp. 1263–1270.