NTNU

Norwegian University of
Science and Technology

Intraventricular Vector Flow Mapping — An In Silico and In Vivo Evaluation

Magnus Sælensminde

Master of Science in Physics and Mathematics
Submission date: June 2018
Supervisor: Catharina de Lange Davies, IFY
Co-supervisor: Lasse Løvstakken, ISB

Norwegian University of Science and Technology
Department of Physics

Abstract

Modern cardiology research aims to encompass sophisticated analysis of intraventricular blood flow in the clinical practice. Such analysis relies on robust and effective means of measuring blood flow fields, an approach also known as vector flow imaging (VFI). Because cardiovascular disease is the leading cause of death globally, and ultrasound is the most used imaging modality in cardiology, finding new and improved echocardiographic VFI techniques has huge potential benefits.

This master thesis concerns the development and evaluation of a Doppler based VFI technique called intraventricular vector flow mapping (iVFM). iVFM is a two-dimensional VFI method by Assi et al. that formulates and solves a regularized linear minimization problem with Doppler and wall data as inputs.

The iVFM method was implemented in the visualization pipeline of the ultrasound viewer named PyUSView, which was used as a platform for in silico and in vivo studies that covered a comparison between iVFM, speckle tracking (ST) and an a combined method. Two proposed iVFM improvements were also subject to analysis: One that restricts artificial radial flows by addition of a new regularization term and another that shows how it is possible to include out-of-plane-flow in iVFM.

The results outline some of the differences between ST and iVFM, and shows that a combination of the two methods produces results that harvest strengths from both. The new regularization term proved effective in the suppression of artificial radial flows outside the color Doppler mask, and the out-of-plane flow gradient analysis showed a large potential for error reductions.

Sammendrag

Moderne kardiologiforskning har en målsetning om å inkludere sofistikerte analyser av intraventrikulær blodstrøm i klinisk praksis. Slike analyser avhenger av robuste og effektive metoder for måling av strømningsfelt i blod, en tilnærming også kjent som *vector flow imaging* (VFI). Fordi kardiovaskulære sykdommer er den ledende dødsårsaken i verden, og fordi ultralyd er den mest brukte avbildningsmodaliteten i kardiologi, har nye og forbedrede ultralydbaserte VFI-teknikker stor potensiell nytteverdi.

Denne masteroppgaven tar for seg utvikling og evaluering av en Doppler-basert VFI-teknikk kalt *intraventricular vector flow mapping* (iVFM). iVFM er en todimensjonal VFI-metode utviklet av Assi et al. som setter opp og løser er regularisert lineært minimaliseringproblem med Dopplerdata og veggdata som input.

iVFM ble implementert i visualiseringsdelen av ultralydprogrammet PyUSView, som ble brukt som plattform for in silico og in vivo studier som inkluderte en sammenlikning av iVFM, *speckle tracking* (ST) og en kombinert metode. To foreslåtte iVFM-forbedringer ble også analysert: En som begrenser kunstige radielle strømninger ved å legge til et nytt regulariseringsledd og en annen som viser hvordan det er mulig å inkludere ut-av-planet-strømning i iVFM.

Resultatene illustrerer flere av forskjellene mellom ST og iVFM, og viser at en kombinasjon av de to metodene produserer resultater som utnytter stryker fra begge. Det nye regulariseringsleddet viste seg effektivt til å begrense de kunstige radielle strømningene utenfor fargedopplermasken, og analysen av ut-av-planet-strømning viste et stort potensial for reduksjon av feil.

Preface

This dissertation marks the completion of the degree of Master of Science at the Department of Physics at the Norwegian University of Science and Technology (NTNU), Trondheim. The work that culminated in this master thesis took place at the Department of Circulation and Medical Imaging. It builds directly upon the writer's project thesis that was completed fall 2017. As a consequence, the introductory, theory and methodology parts of this master thesis contain some sections, either used directly or revised, from the project thesis.

Acknowledgements

Special thanks and acknowledgements goes to my supervisor Professor Lasse Løvstakken for excellent feedback and directions during the work process. Thank you, not only for sharing your knowledge and experience in the world of ultrasound, but also for being an enthusiastic and supportive person. Moreover, I wish to give PhD student Thomas Grønli well deserved credit for his role in the project. Thank you for enlightening me with new insights, for guiding me in the right direction and for many cheerful moments. Thanks also to my supervisor at the Department of Physics, Professor Catharina de Lange Davies for taking care of the administrative duties of the master project.

I also want to direct endless gratitude towards my parents for highly appreciated encouragements and support, and for always being there for me.

Contents

Chapter 1

Introduction

This chapter provides an introduction to Doppler vector flow reconstruction and the current state of the art, plus the motivation behind the master project and its main objectives. Section 1.2 and 1.3 originate from the author's project thesis.

1.1 Doppler based flow reconstruction

Doppler based ultrasound techniques as a means to measure blood flow velocities has been a part of clinical cardiology for decades. Their direct application in intraventricular cardiac flow imaging is however restricted by the fact that a Doppler velocity estimate only contains the velocity component pointing in the the same direction as the ultrasound beam. This makes Doppler based flow reconstruction in cases where the direction of the flow is unknown a matter that requires additional measurements, constraints or assumptions.

1.2 Motivation

Blood flow patterns in the heart chambers, and especially vortex formation in the left ventricle, have in recent years attracted the attention of clinical cardiology research. Several intraventricular flow characteristics have been recognized as possible precursors for heart disease [1], but further research and possible future diagnostic implementation relies heavily on robust and cost-effective means of blood flow measurements. Several approaches have been investigated and developed to find the blood flow velocity components in two or three dimensions, a branch of ultrasound imaging called vector flow imaging (VFI) [2]. One approach involves using crossing ultrasound beams, another uses tracking of speckle patterns between frames, and yet another uses directional beamforming to estimate velocity components perpendicular to the ultrasound beam direction. The development has

been ongoing for years, but it is fairly recently that commercial VFI implementations have started to become available.

1.3 State of the art VFI

A possible way to utilize ultrasonic Doppler data in intraventricular flow imaging is to formulate a regularized minimization problem that includes the Doppler data as well as other physical constraints. This is fairly different from other, more analytically founded Doppler VFI methods, such as the method based on integrating the continuity equation along the scan lines that has been developed by Damien Garcia [3]. This method has been implemented on an Hitachi ultrasound scanner [4] and is currently the clinical state of the art Doppler VFI. The regularized minimization approach has, however, resulted in what the authors claim to be an improved Doppler VFI method called intraventricular vector flow mapping (iVFM) [5].

1.4 Thesis goals

The main objective of this thesis has been to develop and implement an improved version of iVFM and evaluate its performance. This master thesis is a continuation of the writer's project thesis [6], in which the iVFM performance was studied in silico under scenarios related to walls, noise and dropouts of small velocities. The project thesis identified challenges related to aliasing, overestimation of velocities in dropout regions and out-of-plane flow that violates the two dimensional non-divergence assumption. The working hypothesis has been that the overestimation can be improved by an extra regularization term, and that out of plane flow is an important error source that can be reduced by including out-of-plane flow measurements in the divergence term in iVFM.

In addition to implementing and evaluating these two iVFM improvements, an effort to implement iVFM in a practical pipeline for in vivo ultrasound analysis has been made. This pipeline contains a speckle tracking (ST) based aliasing correction method, the possibility to use V_x estimates from ST as iVFM input and an iterative conjugate gradient solver sped up by preconditioning. The pipeline was used to produce the main results in the thesis, which are in silico quantitative evaluation of the developed iVFM additions, in vivo iVFM analysis and comparison to ST.

Chapter 2

Background

This chapter contains the most important theory needed to follow the rest of the thesis. It starts with sections describing ultrasound, Doppler and ST. Then follows a part about Tikhonov regularization, preconditioning of linear algebra systems, before the chapter ends with a description of Doppler based flow field reconstruction. In this chapter, section 2.1.1, 2.1.2 and most parts of 2.2 and 2.4 originate from the author's project thesis.

2.1 Theory of ultrasonic imaging

2.1.1 History of ultrasound

Ultrasound imaging has become a valuable diagnostic tool in modern medicine practice. Inspired by research done on the physiology behind bats' echolocation by Lazzaro Spallanzani [7] in the 18th century, the ultrasound development accelerated after the discovery of the piezoelectric crystal by Jacques and Pierre Curie in 1880 [8]. Ultrasound was suggested as basis for an iceberg detection system after the sinking of Titanic in 1912, and armed forces soon understood the value of a technique especially applicable in submarine detection. Since its first medical application in 1942 by the neurologist Karl Dussik [9], the ultrasonic scanner has developed into a versatile instrument used by physicians in many medical specialties.

2.1.2 Physics of Doppler ultrasound

Ultrasound imaging is based on the sending, receiving and interpretation of longitudinal acoustic waves. Changes in mass density and compressibility of the medium in which an ultrasound wave is propagating will cause scattering, reflection and re-

fraction. By analyzing the echo from an ultrasound wave using models founded on wave physics, one can approximate the mass density and compressibility of the medium, and use this information to form what we call an ultrasound image.

Ultrasound transducers are typically composed of arrays of piezoelectric crystals that through individually controlled delays offer endless possibilities for beamforming and receiving schemes. The ultrasound images in this master thesis are recorded with a linear array and a phased array, which are the two most common transducer types in cardiac imaging. This results in images with different geometric properties: The linear array has parallel scan lines and the image is stored in Cartesian coordinates, while the phased array scans in a sector that results in polar coordinate images.

Doppler ultrasound is based on the Doppler effect observed in all wave phenomenons: The frequency of a signal emitted by a wave transmitter moving towards or away from a stationary observer changes as a function of the wave transmitter's velocity. By analyzing the frequency shift f_d and knowing the speed of sound in the medium, c, the velocity, v, of for instance blood flowing towards the ultrasound probe can be determined by the relation

$$v = \frac{c}{2f_0 \cos\theta} f_d \,. \tag{2.1}$$

Here, θ is the angle between the velocity vector and the ultrasound beam and f_0 is the ultrasound center frequency.

In practice, the received ultrasound signal is converted into so-called IQ data, using quadrature demodulation. This is done mainly to reduce the amount of data without losing essential information [10], and will not be discussed in further detail here. The Doppler velocity for pulsed wave (PW) Doppler can from the IQ data be determined by

$$V_d = \frac{V_{ny}}{\pi} \text{ angle } \left(\left\langle \overline{IQ_i} \cdot IQ_{i+1} \right\rangle \right) \,, \tag{2.2}$$

where V_{ny} is the Nyquist velocity, angle(z) returns the angle of a complex number z, and $\left\langle \overline{IQ_i} \cdot IQ_{i+1} \right\rangle$ is the mean of the autocorrelation of the IQ frames in the data packet, but shifted by one frame.

PW Doppler is a technique where the Doppler velocities in different regions can be determined. This is done with a number of short pulses that are sent and received by the same device. The pulses are too short to pick up the Doppler shift individually, which is why two or more IQ frames are needed in equation (2.2) to get a velocity estimate. Although PW Doppler gives spatial Doppler information,

its major disadvantage is the inherent upper limit of measurable velocities, V_{ny}, defined by

$$V_{ny} = \frac{c \cdot \text{PRF}}{4f_0}, \tag{2.3}$$

where PRF is the pulse repetition frequency. Radial velocities larger than this limit will be sampled in the wrong end of the velocity spectrum, a sampling phenomenon known as aliasing. The result of this for color Doppler images can be observed in figure 2.1, where the color Doppler is shown before and after aliasing correction.

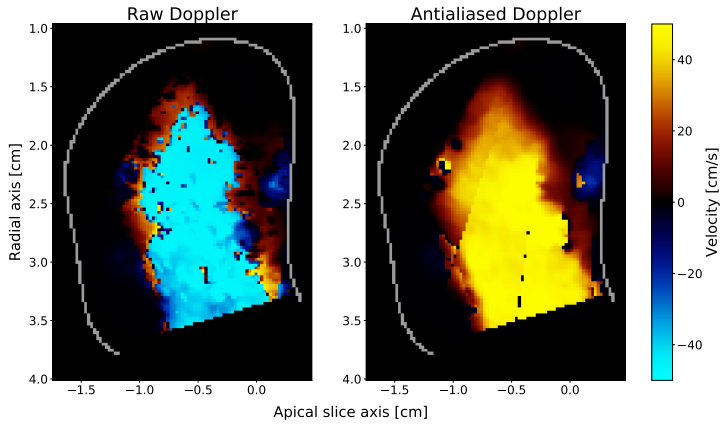

Figure 2.1: Color Doppler images from patient 4002 frame 46. The large blue area of seemingly negative velocities in the left image are in reality aliased positive velocities. The alias corrected image is shown on the right.

2.1.3 ST and aliasing correction

The idea that velocity estimates can be obtained from the tracking of speckle patterns from frame to frame in ultrasound images was first reported by Robinson et al. [11] in 1982. As the ST method has matured over the years, it seems to overcome the two most important challenges in Doppler velocity estimation, namely aliasing and the beam direction dependent velocity measurement. In ST, the backscattered echos are tracked under the assumption that speckle signatures remain relatively constant as tissues or liquids move, and that their motion can be followed using pattern matching algorithms. The procedure can be done in one, two or three dimensions, depending on the available ultrasound data.

As explained in an overview article by Bohs et al. [12], the ST process takes a kernel region from one acquisition and finds the best match within a search region from a later acquisition. This process is called block matching, and is illustrated

crudely in figure 2.2. The vector describing the motion of a kernel can then be acquired from the position of the kernel, the position of the best match and the time between the two acquisitions. The whole vector field can be found by performing repeated block matching with different kernels. There are several ways to define the best match in the search region, as well as many other algorithmic nuances in the ST process that lie well beyond the scope of this master thesis. The process used for comparison in this thesis uses the sum of squared differences (SSD) to define the best match.

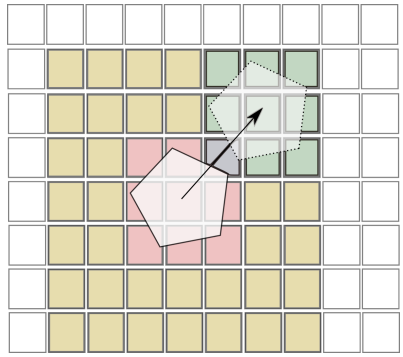

Figure 2.2: Outline of the block matching process of ST. The procedure tracks a speckle (hexagon) by finding the area in the search region (yellow) that best matches the kernel (red). The best matching region (green) defines the displacement of the speckle. The figure is used with the author's permission [13].

For the aliasing correction process used to correct the Doppler velocity estimates in this thesis, a two-dimensional version of the aliasing correction described in [14] was used. In this algorithm, the spatial correlation was calculated for the positions corresponding to the aliased positions. This resembles how the ST works in general, but the dealiasing procedure restricts its search area to the aliased positions. An overview of the process is viewed in figure 2.3, where we see the alias candidate with the best correlation (green) the other aliased positions (red).

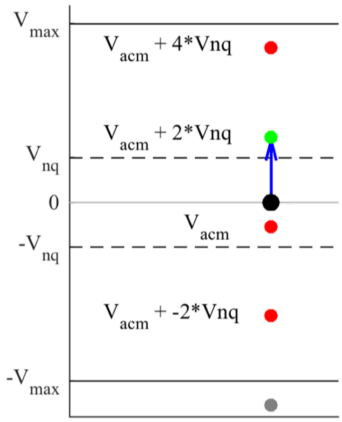

Figure 2.3: Overview of the aliasing correction process. The alias candidate with the best correlation (green) defines the alias corrected Doppler estimate. The other aliased positions as well as the initial Doppler estimate, V_{acm}, are shown in red. The gray alias candidate outside the predefined range $[-V_{max}, V_{max}]$ is neglected. The figure is used with permission of Morten Smedsrud Wigen.

2.2 Method of least squares and Tikhonov regularization

The least squares method is used to find the best fit solutions to overdetermined problems in regression analysis. In the system

$$A\mathbf{x} = \mathbf{b} + \epsilon\,, \tag{2.4}$$

where the matrix A and the vector $\mathbf{b}$ are known, we wish to find values for $\mathbf{x}$ that minimizes the residual error $\epsilon^T \cdot \epsilon = (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b})$. This can be done by differentiation of the residual error with respect to $\mathbf{x}$ and setting equal to zero, which leads to the expression for the least square estimator $\mathbf{x} = (A^TA)^{-1}A^T\mathbf{b}$.

In the physical world, we sometimes want to approximate solutions to ill-posed problems, that is problems without a unique solution or with a solution that does not change continuously when the initial conditions are changed. A popular approach to solving ill-conditioned problems is the Tikhonov regularization [15]. By adding one or more regularization terms to the residual error, the ill-posed system is replaced with a closely related, but well-posed system. With only one simple regularization term, the expression we seek to minimize is $(A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}) + \lambda\mathbf{x}^T\mathbf{x}$, where $\lambda > 0$ is the regularization parameter. We call this functional the cost functional. The solution for this simple regularization is known, but in the

more general case where the cost functional is on the form

$$J(\mathbf{x}) = (A\mathbf{x} - \mathbf{b})^T (A\mathbf{x} - \mathbf{b}) + \sum_k \lambda_k C_k(\mathbf{x})^T C_k(\mathbf{x}) \,, \tag{2.5}$$

one possible approach is to differentiate with respect to $\mathbf{x}$ and set equal to zero. Whether or not this gives results depends on the constraints $C_k(\mathbf{x})$. If they can be expressed as $C_k(\mathbf{x}) = Q_k\mathbf{x}$ for a known matrix Q_k, this approach leads to

$$\Big(A^T A + \sum_k \lambda_k Q_k^T Q_k\Big)\mathbf{x} = A^T \mathbf{b} \,, \tag{2.6}$$

which is a new linear system on the form $A'\mathbf{x} = \mathbf{b}'$. If instead only i of the k constraints can be expressed as $C_i(\mathbf{x}) = Q_i\mathbf{x}$ for a known matrix Q_i, and the other j constraints can only be expressed as $C_j(\mathbf{x}) = Q_j\mathbf{x} - \mathbf{d_j}$ for a known matrix Q_j and vector $\mathbf{d_j}$, the resulting linear system becomes

$$\Big(A^T A + \sum_k \lambda_k Q_k^T Q_k\Big)\mathbf{x} = A^T \mathbf{b} + \sum_j \lambda_j Q_j^T \mathbf{d_j} \,. \tag{2.7}$$

2.3 Preconditioning in linear algebra solvers

When solving the general linear algebra system on the form

$$A\mathbf{x} = \mathbf{b} \,, \tag{2.8}$$

where A is a sparse matrix, direct solvers tend to become computationally costly as we increase the size of A [16]. Such a problem can often be solved much more efficiently with an iterative solver, for example the conjugate gradient (CG) solver. An iterative solver is an algorithm that, given an initial solution guess $\mathbf{x_0}$, iteratively provides new solution guesses $\mathbf{x_k}$, that for each iteration comes closer to the true solution $\mathbf{x} = A^{-1}\mathbf{b}$. We are however not guaranteed that the solution guess $\mathbf{x_k}$ converges towards the true solution or that it comes satisfyingly close to the true solution in a practical amount of iterations. To ensure this, one can transform the linear problem (2.8) into

$$MA\mathbf{x} = M\mathbf{b} \,, \tag{2.9}$$

where the iterative solver will have a better starting point for iteration the closer MA lies to the identity. In other words, we wish M to approximate the inverse of A. The matrix M is what we call a preconditioner, and can be taken as an additional argument by many iterative solvers.

2.4 Flow field reconstruction based on Doppler ultrasound

Reconstruction of a flow field solely from Doppler velocity estimates seems infeasible. Considering the Doppler data only provides vector components parallel to the ultrasound propagation direction, further data or new assumptions about the flow field are required for a reconstruction scheme to work. By including wall information and assuming non-divergence and a smooth flow field, Assi et al. establish the global quadratic minimization problem that plays the main part in the most recent version of their reconstruction method, iVFM [5].

It is physically consistent to require that the reconstructed flow should not cross the cardiac wall and that it should be somewhat smooth. One can also assume incompressible flow and mass conservation by demanding the divergence of the vector field to be zero. Although non-divergence is a valid assumption in 3D blood flow, it is generally not valid for a 2D slice through a 3D flow. Since such a 2D field can have hidden flow components perpendicular to the slice surface, one should be cautious when applying iVFM to 2D cardiac views with potential flow out of the plane. However, by measuring the perpendicular flow component and using it with the result in equation (2.7), we will in section 3.1.6 see how the non-divergence assumption can become valid.

The cost functional for the global minimization problem solved in iVFM, $J(\mathbf{v})$, can be expressed as

$$J(\mathbf{v}) = \int_\Omega (v_r - v_D)^2 \, dV + \lambda_1 \int_\Omega (\nabla \cdot \mathbf{v})^2 \, dV + \lambda_2 \int_{\delta\Omega} (\mathbf{v} \cdot \mathbf{n})^2 \, d\sigma + \lambda_3 \int_\Omega S(\mathbf{v}) \, dV \,, \tag{2.10}$$

which is the sum of three constraint terms in addition to a Doppler fit term. In a least square sense, we want the solution, $\mathbf{v} = v_r \cdot \mathbf{e_r} + v_\theta \cdot \mathbf{e}_\theta$, to fit well to the Doppler data, v_D, that lies inside the domain, Ω. We also want the solution to not flow through the domain boundary, $\delta\Omega$, to have small divergence, $\nabla \cdot \mathbf{v}$, and to be relatively smooth. In addition to the three constraint terms in equation (2.10), we add a fourth constraint

$$\lambda_4 \int_{\Omega_4} v_r^2 \, dV \,, \tag{2.11}$$

where Ω_4 is the dropout regions inside the domain. The motivation behind the penalization of radial velocities in these regions is to prevent the tendency for iVFM to overestimate velocities in dropout regions. This was seen in the dropout tests on the CFD phantom in the writer's project thesis [6], and also from the earliest in vivo results in the master project.

To prevent an ill-posed minimization problem that can result in overfitting and in worst case infinitely many solutions, we include the four constraints as Tikhonov

regularization terms. The best fit solution is the one that minimizes the functional J.

Chapter 3

Methodology

In the work that lead up to the writer's project thesis [6], two versions of iVFM were implemented: An angular and a linear version, iVFM_{ang} and iVFM_{lin}. These were written in Python 3 with the scientific computing tool SciPy. The angular version was implemented using the procedure proposed by Assi et al [5]. It works in a polar coordinate system, while the linear version, iVFM_{lin}, works in a Cartesian coordinate system, and was derived from the angular version through a coordinate transformation.

Even though this master thesis uses iVFM_{lin} as its starting point, this methodology chapter has been made more accessible to the new reader by starting with two revised sections from the project thesis, section 3.1.1 and 3.1.2. These sections contain descriptions of how iVFM_{ang} and iVFM_{lin} were implemented. The sections that follow focus on the methods that have been applied to produce and validate an improved iVFM version in this master thesis. Section 3.2.1 and 3.2.2, describing the in silico phantom and the ultrasound simulations, as well as the last part of section 3.3, describing the clutter filtering, are also from the author's project thesis.

3.1 iVFM implementation

3.1.1 iVFM_{ang}

To implement the minimization of the cost functional (2.10), one needs to choose a smoothness function, S, before discretizing and performing the actual minimization that leads to the linear system described in equation (2.6). Assi et al. chose $S(\mathbf{v}) = \sum_{m\in\{r,\theta\}}(r^2\partial_r^2 v_m)^2 + 2(r\partial_{r\theta}^2 v_m)^2 + (\partial_\theta^2 v_m)^2$, which contains second degree partial derivatives that favor a spatially smooth solution. This term has no

other physical foundation than the tendency for cardiovascular flow fields to be smooth, and other smoothness functions could have been chosen. To get a more convenient form in polar coordinates, $r\nabla \cdot \mathbf{v} = r\partial_r v_r + v_r + \partial_\theta v_\theta$, is used as the divergence term instead of $\nabla \cdot \mathbf{v}$. Since we are minimizing the cost functional by differentiation with respect to $\mathbf{v}$ later, this choice does not affect the final solution.

To discretize the cost functional, the following $(M \times N)$ matrices were defined, where N is the number of scanlines and M is the number of samples per scanline: V_d is the matrix containing the negative Doppler velocities, V_r and V_θ contain the estimated radial and angular velocities, R contains the radial coordinates of the grid nodes, N_r and N_θ contain the radial and angular component of the domain normal vector and the binary matrix Δ defines the domain. For the Δ matrix, elements inside the domain are set to 1, while elements outside are set to 0. For the N_r and N_θ matrices, only the border elements contain the r and θ component of the normal vector in that position, all other elements are set to zero. All the matrices are vectorized into $(MN \times 1)$ column vectors such that $\text{vec}(V_d) = \mathbf{v_d}$, $\text{vec}(V_r) = \mathbf{v_r}$, $\text{vec}(\Delta) = \boldsymbol{\delta}$ and so on. The constant radial and angular step size are h_r and h_θ. The differential operators are approximated with the $(q \times q)$ three point stencils $\dot{D}_q$ and $\ddot{D}_q$, defined as

$$\dot{D}_q = \begin{bmatrix} -0.5 & 0.5 & & & \\ -0.5 & 0 & 0.5 & & \\ & \ddots & \ddots & \ddots & \\ & & -0.5 & 0 & 0.5 \\ & & & -0.5 & 0.5 \end{bmatrix}, \quad \ddot{D}_q = \begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{bmatrix}. \tag{3.1}$$

The $(q \times q)$ identity matrix is denoted I_q, and $\mathbf{I_q}$ represents a column vector with ones of size $(q \times 1)$.

The Doppler term, divergence term, wall term and smoothing term in the cost functional (2.10) can now be discretized, which changes $J(\mathbf{v})$ to

$$J(\mathbf{v}) = (Q_0\mathbf{v} - \mathbf{v_D})^T(Q_0\mathbf{v} - \mathbf{v_D}) + \sum_{k=1,2,3} \lambda_k \mathbf{v^T} Q_k^T Q_k \mathbf{v}\,, \tag{3.2}$$

where

$$
\begin{aligned}
Q_0 =& [1,\, 0] \otimes \mathrm{diag}(\boldsymbol{\delta}),\\
Q_1 =& \left[\frac{1}{h_r}(\mathbf{r}\,\mathbf{I}^{\mathbf{T}}_{\mathbf{MN}}) \circ (I_N \otimes \dot{D}_M) + I_{MN},\, \frac{1}{h_\theta}\dot{D}_N \otimes I_M\right],\\
Q_2 =& [\mathrm{diag}(\mathbf{n_r}),\, \mathrm{diag}(\mathbf{n}_\theta)]\,,\\
Q_3 =& \begin{bmatrix} I_2 \otimes \frac{1}{h_r^2}\left((\mathbf{r} \circ \mathbf{r}\,\mathbf{I}^{\mathbf{T}}_{\mathbf{MN}}) \circ (I_N \otimes \ddot{D}_M)\right) \\ I_2 \otimes \frac{2}{h_r h_\theta}\left((\mathbf{r}\,\mathbf{I}^{\mathbf{T}}_{\mathbf{MN}}) \circ (\dot{D}_N \otimes \dot{D}_M)\right) \\ I_2 \otimes \frac{1}{h_\theta^2}\left(\ddot{D}_N \otimes I_M\right) \end{bmatrix}.
\end{aligned}
\tag{3.3}
$$

Here $\otimes$ is the Kronecker product, $\circ$ is the Hadamard product and diag() denotes the diagonal matrix. The full details of the discretization can be viewed in the Assi et al. appendix [5].

The resulting cost functional is now on the same form as equation (2.5), and can therefore be minimized to form a new linear system that can be solved with a linear algebra solver. The resulting system is

$$
\left(Q_0^T Q_0 + \sum_{k=1,2,3} \lambda_k Q_k^T Q_k\right)\mathbf{v} = Q_0^T \mathbf{v_d}\,. \tag{3.4}
$$

3.1.2 Transition from iVFM$_{\text{ang}}$ to iVFM$_{\text{lin}}$

The coordinate transformation from iVFM$_{\text{ang}}$ to iVFM$_{\text{lin}}$ only affects the divergence and the smoothing term, represented by Q_1 and Q_3 in the linear system in equation (3.4). Since $\nabla \cdot \mathbf{v} = \partial_z v_z + \partial_x v_x$ in Cartesian coordinates, Q_1 becomes

$$
Q_1 = \left[\frac{1}{h_z} I_N \otimes \dot{D}_M,\, \frac{1}{h_x}\dot{D}_N \otimes I_M\right]. \tag{3.5}
$$

The smoothing function for the linear variant of iVFM was chosen to contain second degree partial derivatives with cross terms. With $S(\mathbf{v}) = \sum_{m\in z,x}(\partial_z^2 v_m)^2 + 2(\partial_{zx}^2 v_m)^2 + (\partial_x^2 v_m)^2$, the linear smoothing function resembles the angular, but is not equivalent. With this as the smoothing function, Q_3 simplifies to

$$
Q_3 = \begin{bmatrix} I_2 \otimes \frac{1}{h_z^2}\left(I_N \otimes \ddot{D}_M\right) \\ I_2 \otimes \frac{2}{h_z h_x}\left(\dot{D}_N \otimes \dot{D}_M\right) \\ I_2 \otimes \frac{1}{h_x^2}\left(\ddot{D}_N \otimes I_M\right) \end{bmatrix}. \tag{3.6}
$$

Notice that the system no longer has an $\mathbf{r}$ dependence when working in Cartesian coordinates, so the iVFM function does not need the R matrix. The rest of the

terms involved in the system in equation (3.4) are changed from $\mathbf{n_r}$ to $\mathbf{n_z}$, $\mathbf{n}_\theta$ to $\mathbf{n_x}$ and so on, as the velocity field is now described by $\mathbf{v} = v_z \cdot \mathbf{e_z} + v_x \cdot \mathbf{e_x}$.

3.1.3 λ_4 – A constraint on radial flow outside the color flow mask

In a less ideal iVFM setting with uncertain Doppler estimates and signal dropouts, one does not necessarily wish to give the same weight to all Doppler velocity estimates within the domain. Different weighing can be achieved in iVFM by letting Δ consist of floating point values between 0 and 1, instead of binary values. The measurements we trust the most get the highest weight, and the ones we trust the least get zero weight. The areas that get very small weights typically reside within dropout regions that in practice only contain noisy measurements due to the clutter filtering.

To prevent iVFM's observed tendency to construct solutions containing high radial velocities in areas with a very small Δ value, a new constraint was imposed on radial velocities. The physical reasoning behind this constraint goes as follows: A region with high radial velocities will always provide good Doppler estimates, and therefore high radial velocities in dropout regions are simply non-physical. Even though all this tells for certain is that the real radial velocity must be somewhere below the cutoff velocity of the clutter filter, the approach taken in this thesis is to constrain radial velocities linearly by adding a new regularization term, Q_4 and control its influence through tuning of the regularization parameter λ_4. The matrix Δ_4 was introduced to regulate in which regions this new regularization term is allowed to function. Δ_4 was constructed by taking $B - \Delta$ in positions within both the domain and the Doppler region of interest (ROI) and setting all other elements to zero. B is here a matrix of ones with the same shape as Δ. Q_4 takes the form

$$Q_4 = [1,\ 0] \otimes \mathrm{diag}(\boldsymbol{\delta_4})\,, \tag{3.7}$$

and was included in the global problem by letting $k = 1, 2, 3, 4$ in equation (3.4).

3.1.4 The Θ matrix

In cases where the input Doppler data is recorded with phased ultrasound array and later scan converted to Cartesian coordinates, iVFM_{lin} would not be directly applicable. The Doppler estimates will then contain velocity components in both z and x direction, whereas iVFM_{lin} assumes they all point along $\mathbf{e_z}$. To handle this, the matrix Θ was given to iVFM_{lin} as an additional argument containing the angle between every input velocity estimate and and $\mathbf{e_z}$. By including Θ in the definition of Q_0 and Q_4, we can effectively impose any angle on the Doppler estimates that

are included in the global minimization problem. Q_0 and Q_4 then becomes

$$\begin{aligned} Q_0 &= [\cos\boldsymbol{\theta},\, \sin\boldsymbol{\theta}] \otimes \mathrm{diag}(\boldsymbol{\delta})\,, \\ Q_4 &= [\cos\boldsymbol{\theta},\, \sin\boldsymbol{\theta}] \otimes \mathrm{diag}(\boldsymbol{\delta_4})\,. \end{aligned} \tag{3.8}$$

3.1.5 Preconditioning

Sparse linear systems on the form $A\mathbf{x} = \mathbf{b}$ can be solved using for instance the $spsolve$ function from the Scipy library. However, computation times can be drastically improved by using an iterative solver such as cg from the same library. The latter uses the conjugate gradient method to approach a solution until an error tolerance level is reached. cg also has the ability to take a preconditioner as an argument to improve the computation time. As explained in section 2.3, the preconditioner for the matrix A should approximate the inverse of A. The iVFM procedure usually consists of setting up and solving the linear system for several consecutive frames that have a somewhat similar domain, and therefore somewhat similar A matrices. The approach taken to speed up iVFM computation was to calculate the exact inverse of A for the first frame and feed this into the cg solver as a preconditioner for all the following frames in the same sequence. The piece of code used to do this can be found in code line 66-71 in section 3.1.7.

3.1.6 Inclusion of out-of-plane flow

To address the problem of the non-divergence assumption in the 2D flow, the divergence regularization term was changed to include the out-of-plane flow. Since the assumption is generally not valid unless all three spatial dimensions are included in the divergence term, we simply change the constraint from $\nabla\cdot\mathbf{v}$ to $\nabla\cdot\mathbf{v}+\frac{\partial}{\partial y}v_y$, where $\mathbf{v} = v_z\mathbf{e_z} + v_x\mathbf{e_x}$. If we assume $\frac{\partial}{\partial y}v_y$ is known, the new constraint is now on the form $C_j(\mathbf{x}) = Q_j\mathbf{x} - \mathbf{d_j}$ as described in section 2.2, and the result presented in equation (2.7) is directly applicable. Only the right hand side of the resulting system from equation (3.4) is affected, and becomes

$$Q_0^T\mathbf{v_d} - \lambda_1 Q_1^T\frac{\partial}{\partial y}v_y\,. \tag{3.9}$$

3.1.7 Full Python implementation

The implemented Python version of the new iVFM is viewed below. It imports sparse matrix functionality from Numpy and Scipy to set up and solve the linear system, while the derivative operators are constructed by the imported functions Dp_1 and Dp_2, that construct the derivative operators (3.1) as sparse matrices. The new iVFM function has three different Doppler input options. It can 1) take

Doppler velocity estimates in the z direction just like iVFM$_{\text{lin}}$, 2) take Doppler velocity estimates together with a Θ matrix specifying the direction of each estimate or 3) take both initial v_z and v_x estimates. In addition to this, one can input $\frac{\partial}{\partial y} v_y$ if that is available for any of the three modes.

```
import numpy as np
from scipy.sparse import csr_matrix
from scipy.sparse import kron, identity, vstack, hstack, diags, block_diag
from scipy.sparse.linalg import spsolve, cg

def iVFM(Vd, h_z, h_x, N_z, N_x, Delta, lambda_1, lambda_2, lambda_3,
    Vx=None, Theta=None, tol=None, maxiter=None, M_p=None, Delta_4=None,
    lambda_4=None, Vy_grad=None, h_y=None):

    order = 'F'
    M = Vd.shape[0]
    N = Vd.shape[1]

    if Theta is None:
        d = diags(Delta.ravel(order), format='csr', dtype=np.float64)
        d4 = diags(Delta_4.ravel(order), format='csr', dtype=np.float64)
        if Vx is None:
            Q0 = hstack((d, csr_matrix((M * N, M * N), dtype=np.float64)))
            Q4 = hstack((d4, csr_matrix((M * N, M * N),
                        dtype=np.float64)))
        else:
            Q0 = block_diag((d, d))
            Q4 = block_diag((d4, d4))

    else:
        d_0 = diags(Delta.ravel(order) * np.cos(Theta.ravel(order)),
                    format='csr', dtype=np.float64)
        d_1 = diags(Delta.ravel(order) * np.sin(Theta.ravel(order)),
                    format='csr', dtype=np.float64)
        Q0 = hstack((d_0, d_1))

        d4_0 = diags(Delta_4.ravel(order) * np.cos(Theta.ravel(order)),
                    format='csr', dtype=np.float64)
        d4_1 = diags(Delta_4.ravel(order) * np.sin(Theta.ravel(order)),
                    format='csr', dtype=np.float64)
        Q4 = hstack((d4_0, d4_1))

    Q1 = hstack(((1 / h_z) * kron(identity(N), Dp_1(M)),
                 (1 / h_x) * kron(Dp_1(N), identity(M))))

    Q2 = hstack((diags(N_z.ravel(order), format='csr', dtype=np.float64),
                 diags(N_x.ravel(order), format='csr', dtype=np.float64)))

    P1 = kron(identity(N), Dp_2(M))
    P2 = kron(Dp_1(N), Dp_1(M))
    P3 = kron(Dp_2(N), identity(M))
    Q3 = vstack((kron(identity(2), (1 / (h_z * h_z)) * P1),
                 kron(identity(2), (2 / (h_z * h_x)) * P2),
                 kron(identity(2), (1 / (h_x * h_x)) * P3)))

    A = (Q0.T.dot(Q0)) + (lambda_1 * Q1.T.dot(Q1)) +
```

```
        (lambda_2 * Q2.T.dot(Q2)) + (lambda_3 * Q3.T.dot(Q3)) +
        (lambda_4 * Q4.T.dot(Q4))

    if Vx is None:
        if Vy_grad is None:
            b = Q0.T.dot(Vd.ravel(order))
        else:
            b = Q0.T.dot(Vd.ravel(order)) -
                (1 / h_y) * lambda_1 * Q1.T.dot(Vy_grad.ravel(order))
    else:
        b = Q0.T.dot(np.hstack((Vd.ravel(order), Vx.ravel(order))))

    if tol is None:
        x = spsolve(A, b)
    else:
        if M_p is None:
            from scipy.sparse.linalg import splu, LinearOperator
            M2 = splu(A)
            M_x = lambda x: M2.solve(x)
            M_p = LinearOperator((A.shape[0], A.shape[0]), M_x)
        x,info = cg(A, b, M=M_p, maxiter=maxiter, tol=tol)

    return np.reshape(x, (2, N, M)).transpose(0, 2, 1), M_p
```

The new iVFM function was fitted into the visualization pipeline of the ultrasound analysis program PyUSView, which can be used to analyze clinical ultrasound data. In this analysis scheme, the domain needs to be defined manually for a chosen number of frames and then the domain definition for the remaining frames are interpolated linearly. This domain definition was used to define the domain mask both for the in vivo and FUSK analysis in this master thesis. The Δ matrix was defined by taking the arbitration mask within the domain multiplied with the Doppler power. This proved to give a reasonable weighing of the Doppler velocity estimates, prioritizing the higher velocities. The arbitration in PyUSView is a sophisticated procedure for separating flow and tissue regions in the image based on the combined properties of the IQ and B-mode signals. The normal matrices, N_z and N_x, were defined from the domain border positions by using ConvexHull from scipy.spatial. The vectors were then normalized and interpolated to fill the whole domain border, except at the basis, where they were set to zero.

3.2 CFD, FUSK and in vivo data

3.2.1 CFD phantom

The computational fluid dynamics (CDF) model used in the evaluation of iVFM is based on simulations carried out with a commercial CFD solver (ANSYS Fluent) applied to a generic 3D model of a neonatal left ventricle [17]. The simulation is a time varying flow inside a truncated prolate spheroid with an imposed volume change to simulate the full heart cycle. The heart rate of the ventricle is set to

120 beats per minute and the sequence contains 100 frames, which gives a total sequence length of 0.5 seconds and 0.005 seconds between each frame. Figure 3.1 shows the model flow and domain walls for four frames used in the later analysis. The plots demonstrate ejection (frame 18), filling (frame 50) and late diastole (frame 70 and 90) for three different cardiac views. See section 3.4 for an explanation of how the cardiac view slicing was done.

3.2.2 FUSK – Simulated ultrasound

While ultrasound can be simulated by solving the acoustic wave equation, it can be extremely expensive in terms of memory and computational cost when we are dealing with anything more advanced than the simplest ideal models [18]. The choice of approximation method used for ultrasound simulation is therefore a trade off between achieved precision, and computational cost and memory demands. Many methods exist in the spectrum ranging from the slow accurate to the faster less accurate, and the simulated ultrasound used to evaluate iVFM belongs to the second group. The fast ultrasound simulation in k-space (FUSK) uses a Fourier convolution technique to convolve the point spread function with the field of sparsely distributed ultrasound scatterers [19]. FUSK was used to simulate ultrasound on the CFD phantom, and the simulated Doppler data was used as input for iVFM. In this ultrasound simulation, the pulse repetition frequency (PRF) was 5 kHz, the center frequency f_0 was 3 MHz, the speed of sound c was set to 1540 m/s, and the packet size was 12. This gives a Nyquist velocity of $V_{ny} = \frac{c}{4f_0}\text{PRF} = 0.64$ m/s.

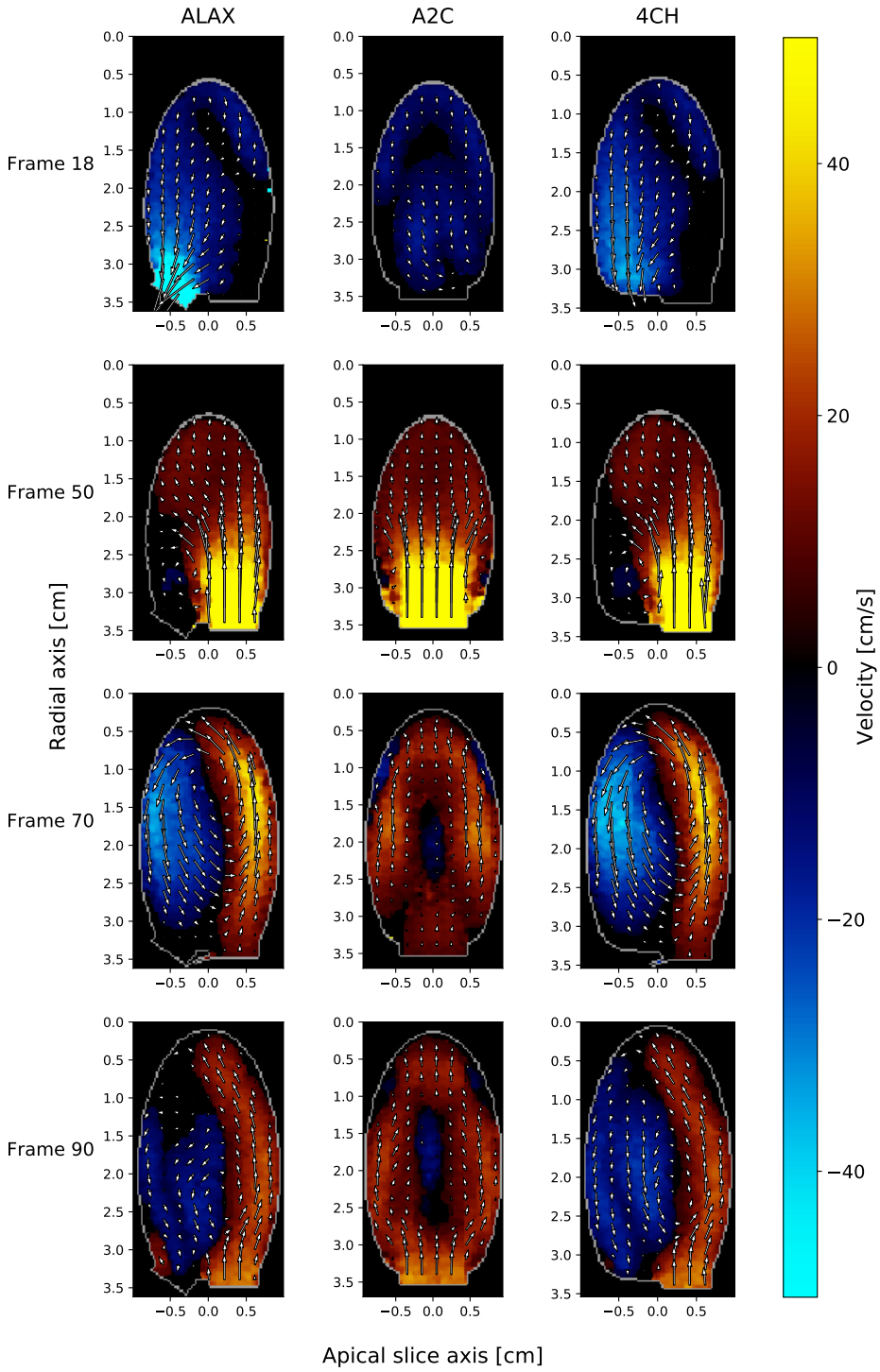

Figure 3.1: Ground truth phantom flow (arrows), true domain walls (gray) and the alias corrected Doppler estimates from FUSK (yellow/red/blue). The columns mark the cardiac views and the rows mark the frame.

3.2.3 In vivo analysis

The in vivo data subject to iVFM analysis was acquired from two healthy patients, 4002 and 4018, and was recorded in 2016 by a cardiologist with a GE E95 ultrasound scanner. For 4002, the PRF was 7 kHz, f_0 was 5.9 MHz, the packet size was 16 and V_{ny} was 0.464 m/s. For 4018, the PRF was 5 kHz, f_0 was 3.7 MHz, the packet size was 16 and V_{ny} was 0.53 m/s. Since the data was acquired with a phased ultrasound array, a Θ matrix had to be constructed and fed to the iVFM function together with the Doppler estimates. The Θ matrix was constructed based on the positional ROI data of the recordings. This was done with the following few lines of code:

```
z = numpy.linspace(ROI_e[1], ROI_e[3], res[0])
x = numpy.linspace(ROI_e[0], ROI_e[2], res[1])
xv, zv = numpy.meshgrid(x, z)
Theta = numpy.arctan2(xv,zv) .
```

Here, ROI_e is a list containing the coordinates of the bounding box of the ROI and res contains the resolution of the iVFM Doppler input matrix. Apart from the dependence on Θ to get the correct direction on the Doppler estimates, the in vivo iVFM analysis was identical to the in silico analysis.

3.3 Aliasing correction and clutter filtering

In the analysis done in this master thesis, ST was always performed prior to iVFM. This is because the aliasing correction method that was used here utilizes the ST library developed at ISB, NTNU. The library has the ability to return the aliasing corrected Doppler estimates in a Cartesian coordinate space, which can be sent directly into the iVFM function, in addition to the more conventional ST algorithms. The library also contains the SSD ST algorithm, and a fairly new hybrid algorithm that combines Doppler estimates with lateral ST.

In practice, all the ST settings were kept at default values, except the output data density parameter, which was tuned to give output arrays of about 100 datapoints in each direction. A finite impulse response (FIR) clutter filter was also applied to the ultrasound data prior to the actual tracking. The frequency response for the FIR clutter filter that was applied to the FUSK and in vivo data is shown in figure 3.2. As shown, this is a highpass filter, and its function is to filter out the strong echoes from the slowly moving heart tissue that disturbs the blood echo signal. An inevitable effect of clutter filtering is the signal attenuation of small Doppler velocities. The regions where the Doppler velocities are attenuated by the clutter filter are known as dropout regions.

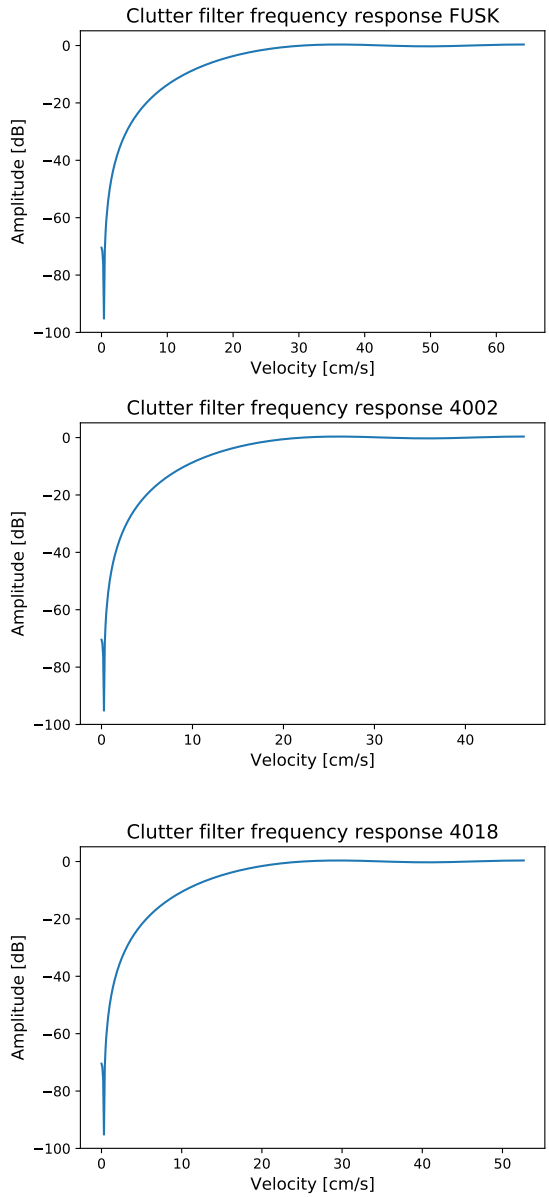

Figure 3.2: The frequency response of the clutter filter plotted up to the Nyquist velocity for the FUSK (top), 4002 (middle) and 4018 (bottom) data.

3.4 Analysis and validation

For all the analysis and validation, values of the regularization parameters were found by trial and error. Since it was established early that λ_1 and λ_2 variations gave only small changes in the resulting field, different smoothing parameters, λ_3, and λ_4 values were tested to find an optimal tuning. The following regularization parameters were found to give satisfying results and were used as default settings throughout the analysis and validation: $\lambda_1 = 5$, $\lambda_2 = 5$, $\lambda_3 = 5 \times 10^{-13}$ and $\lambda_4 = 0.5$.

To analyze the FUSK data in PyUSView, the FUSK data were loaded into an empty dummy HDF file that was exported from PyUSView. By loading FUSK IQ data, positional data, pulse repetition and center frequency info into the same pipeline used for in vivo data, we ensure as equal treatment as possible for the in vivo and in silico analysis. Since the FUSK IQ data is a three-dimensional ultrasound recording, the data must be sliced into two-dimensional data before it can be analyzed by a two-dimensional method such as iVFM. Prior to the slice, the coordinate space was rotated and shifted according to three predefined roll, pitch, yaw and offset settings that result in three slices that resemble the three standard cardiac clinical views apical long axis (ALAX), apical 2 chamber (A2C) and 4 chamber (4CH). The settings for rotation and shifting can be found in table 3.1. The rotations are given in degrees and the offsets in pixels. The offset defines the displacement of the middle slice that is performed after the rotation of the coordinate space, and is valid for the dimensions of the CFD phantom data, which are [178, 95, 95] for z, y, and x respectively. Both the FUSK and CFD ground truth data were subjected to the same rotation, offset and slicing, to get the same views in the ground truth comparisons and the FUSK analysis.

Table 3.1: Table showing the settings used to obtain the three cardiac views for the FUSK analysis.

View	Roll	Pitch	Yaw	Offset
ALAX	-20°	-2°	0°	3
A2C	90°	5.34°	0°	-8
4CH	20°	0°	0°	0

3.4.1 Analysis of the effects of λ_4

To study the influence of λ_4 on the iVFM results from the FUSK data, an iVFM analysis was carried out for all the three FUSK cardiac views with $\lambda_4 = 0$ and $\lambda_4 = 0.5$. All other parameters were kept at default levels. The results were then compared to the ground truth from the CFD phantom through root-mean-square

(RMS) plots for all frames and flow plots for a selection of frames showing the result flow and absolute error. The effect of λ_4 on in vivo results was studied in the exact same manner as the FUSK λ_4 analysis, although the result presentation is restricted to a side by side qualitative comparison.

3.4.2 Comparison of iVFM and ST

iVFM and ST with default settings were run on the FUSK and in vivo data. The ST results were smoothed with a Gaussian kernel of 4 mm using the function gaussian_filter from scipy.ndimage.filters. The FUSK results were compared to ground truth and the in vivo results were compared qualitatively side by side. The FUSK results were presented in RMS plots for all the views and frames as well as difference plots to study the degree of over- and underestimation of velocities. Only points within the color flow mask were included in the RMS and difference plots. Differences in flow characteristics between iVFM and ST were studied in flow plots, showing the flow field for the in vivo results in addition to the absolute errors for the selected FUSK frames and views.

3.4.3 iVFM with V_x input from ST

The output from the hybrid ST, which combines Doppler estimates with lateral SSD ST, was fed as input to the iVFM. iVFM then worked with both V_z and V_x estimates. The results were compared to conventional iVFM and ST in the same RMS error plot and compared qualitatively to iVFM through flow plots showing the the absolute errors for the same FUSK frames and views as used in the comparison of iVFM and ST described in section 3.4.2. In vivo comparisons were also performed for some of the same frames as for the ST vs iVFM comparison. This makes it easier to compare iVFM with V_x input from ST to both the Doppler iVFM version and the SSD ST version.

3.4.4 iVFM with V_y gradient on FUSK data

The impact of flow out of the plane on iVFM was studied by loading the exact out-of-plane spatial gradient, the V_y gradient, into the HDF file loaded into PyUSView for the three views. This gradient was obtained from the CFD data in the slicing process by applying the Numpy gradient function to the velocity perpendicular to the slice plane. The iVFM results with default settings were compared to the iVFM results with default settings and the out-of-plane gradient. The two versions were analyzed through RMS plots and flow plots showing the absolute errors for selected frames and views for the FUSK data only.

Chapter 4

Results

This result chapter is divided into four sections. First the radial flow constraint and the effect of λ_4, secondly a comparison between iVFM and ST, thirdly iVFM with V_x estimates from ST, and lastly the inclusion of out-of-plane flow via the V_y gradient. The first three sections show analysis of both FUSK and in vivo data, while the V_y gradient test is done for the FUSK data only. An overview of all the results can be viewed in table 4.1.

The following pages also contain two FUSK figures displaying the frames and views that are used in the rest of the analysis. Figure 4.1 shows the domain and the Δ_4 mask used in the iVFM analysis, while figure 4.2 shows the true domain together with the walls used in the iVFM analysis. Worth noting in these figures is the relatively small extent of the Δ_4 mask in figure 4.1 and the mismatch between the manually drawn domain walls and the true domains in figure 4.2.

Table 4.1: Overview of the figures in the results chapter.

FUSK domain and Δ_4 mask	4.1
FUSK true domain and walls	4.2
λ_4 analysis: RMS error	4.3
λ_4 analysis: FUSK ALAX frame 18	4.4, 4.5
λ_4 analysis: FUSK 4CH frame 50	4.6, 4.7
λ_4 analysis: FUSK ALAX frame 90	4.8, 4.9
λ_4 analysis: In vivo example and Δ_4 mask	4.10, 4.11
iVFM vs. ST: RMS error	4.12
iVFM vs. ST: FUSK ALAX frame 18	4.13, 4.14
iVFM vs. ST: FUSK A2C frame 70	4.15, 4.16
iVFM vs. ST: FUSK 4CH frame 90	4.17, 4.18
iVFM vs. ST: FUSK difference plots	4.19, 4.20, 4.21, 4.22, 4.23, 4.24
iVFM vs. ST: In vivo examples	4.25, 4.26, 4.27, 4.28, 4.29
iVFM with V_x input from ST: FUSK A2C frame 70	4.30, 4.31
iVFM with V_x input from ST: FUSK 4CH frame 90	4.32, 4.33
iVFM with V_x input from ST: In vivo examples	4.29
iVFM with V_y gradient: RMS error	4.35
iVFM with V_y gradient: FUSK A2C frame 50	4.36, 4.37
iVFM with V_y gradient: FUSK ALAX frame 60	4.38, 4.39
iVFM with V_y gradient: FUSK 4CH frame 70	4.40, 4.41

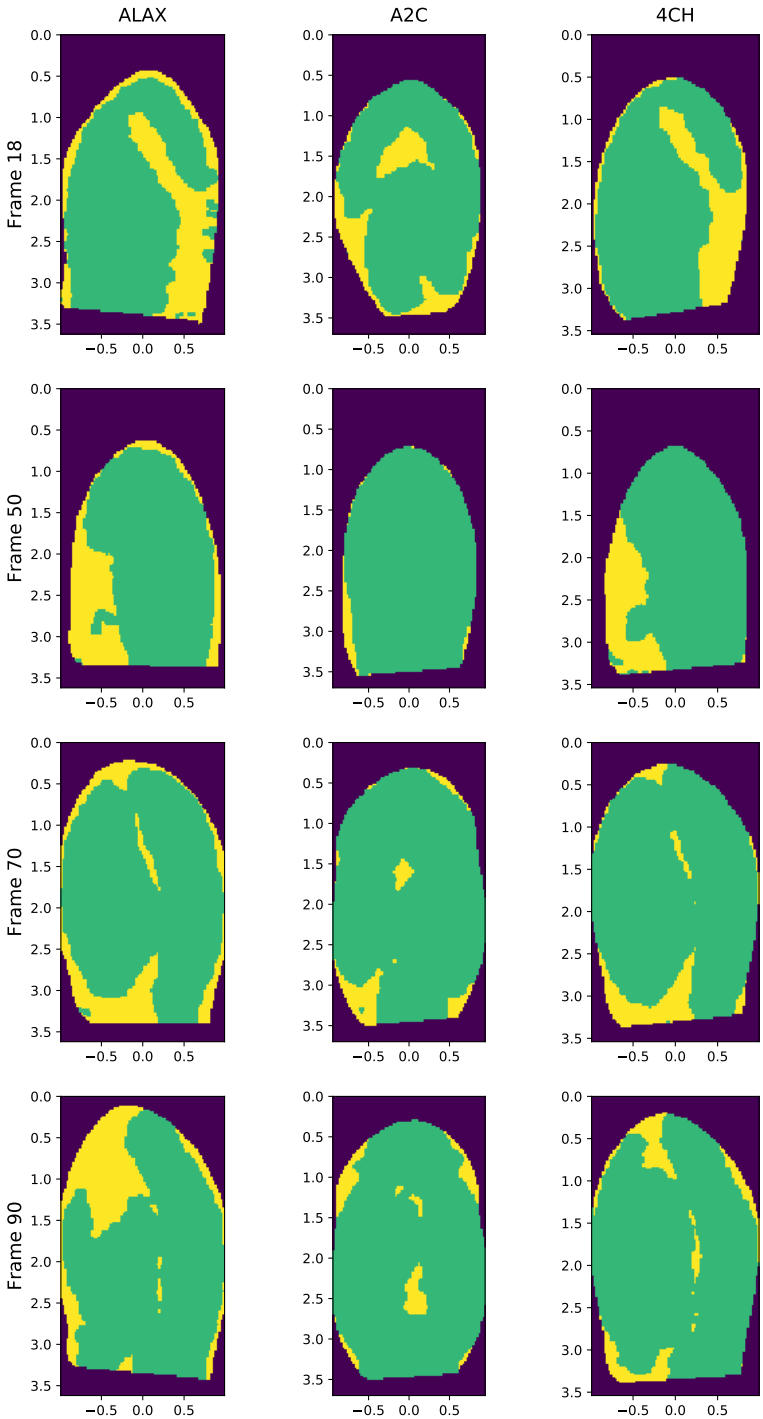

Figure 4.1: Δ masks (green) and Δ_4 masks (yellow) used for the all the FUSK iVFM analysis.

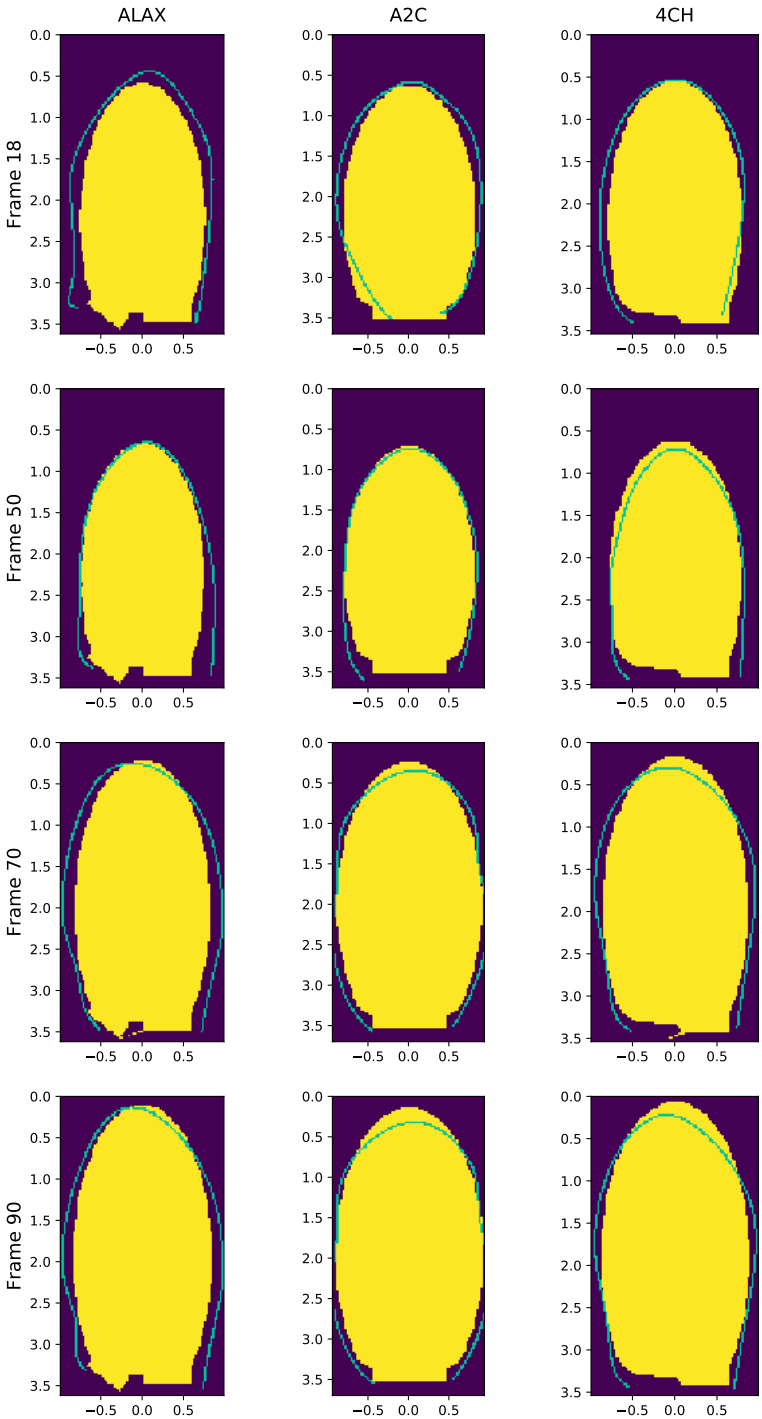

Figure 4.2: True phantom domain (yellow) and manually drawn walls (green) used for the all the FUSK iVFM analysis.

4.1 λ_4 – the radial flow constraint parameter

This section contains the results from the analysis of the introduced radial flow constraint parameter, λ_4. The FUSK results for all 100 frames and all three cardiac views is summed up in figure 4.3, which shows the RMS error for iVFM running with and without λ_4 regularization. The new constraint is observed to have a larger impact on the errors in the radial z-direction than the lateral x-direction. While the V_z errors are considerably or slightly lowered in most of the regions and almost identical in the other regions, the error in V_x ranges from slightly lower to about the same in most regions, but is also slightly higher in a few others. This increased V_x error can be seen for A2C and 4CH from frame 20 to 40. The most prominent finding is however how the λ_4 presence effectively dampens the V_z error spikes in the region from frame 0 to frame 40, and that this error reduction is achieved without the introduction of any other significant errors.

The six figures that follow show how the radial flow constraint affects the flow results for three FUSK views at different points in the heart cycle. Figure 4.4 and 4.5 show the iVFM results for the ALAX view for frame 18, respectively without and with λ_4 regularization. Observe how both the V_z and V_x absolute errors are reduced when $\lambda_4 = 0.5$ in the lower right region of the domain. In figure 4.6 and 4.7, which both show 4CH frame 50, mainly the V_z error is affected by λ_4, while the error in V_x stays more or less unchanged. The third FUSK example is from ALAX frame 90 and is visualized in figure 4.8 and 4.9. This last FUSK example shows a decent error reduction in the apical regions for both V_z and V_x for the λ_4 regularized flow field solution.

An in vivo example showing the iVFM analysis result with and without the radial flow constraint can be viewed in figure 4.10, and figure 4.11 shows the Δ and Δ_4 masks used for the analysis of this particular frame. The effect of λ_4 regularization within the Δ_4 mask is clear: Almost no radial motion is allowed in these regions in the solution flow. Notice however how radial flow is allowed outside the Δ_4 mask in the upper left part of the ventricle. This region lies outside the ROI, and iVFM cannot penalize the radial flow here on the same basis as within the ROI, since this region can contain radial velocities that will never show up as Doppler estimates. The result is an iVFM flow solution without the artificially high radial velocities in dropout regions, but only within the ROI.

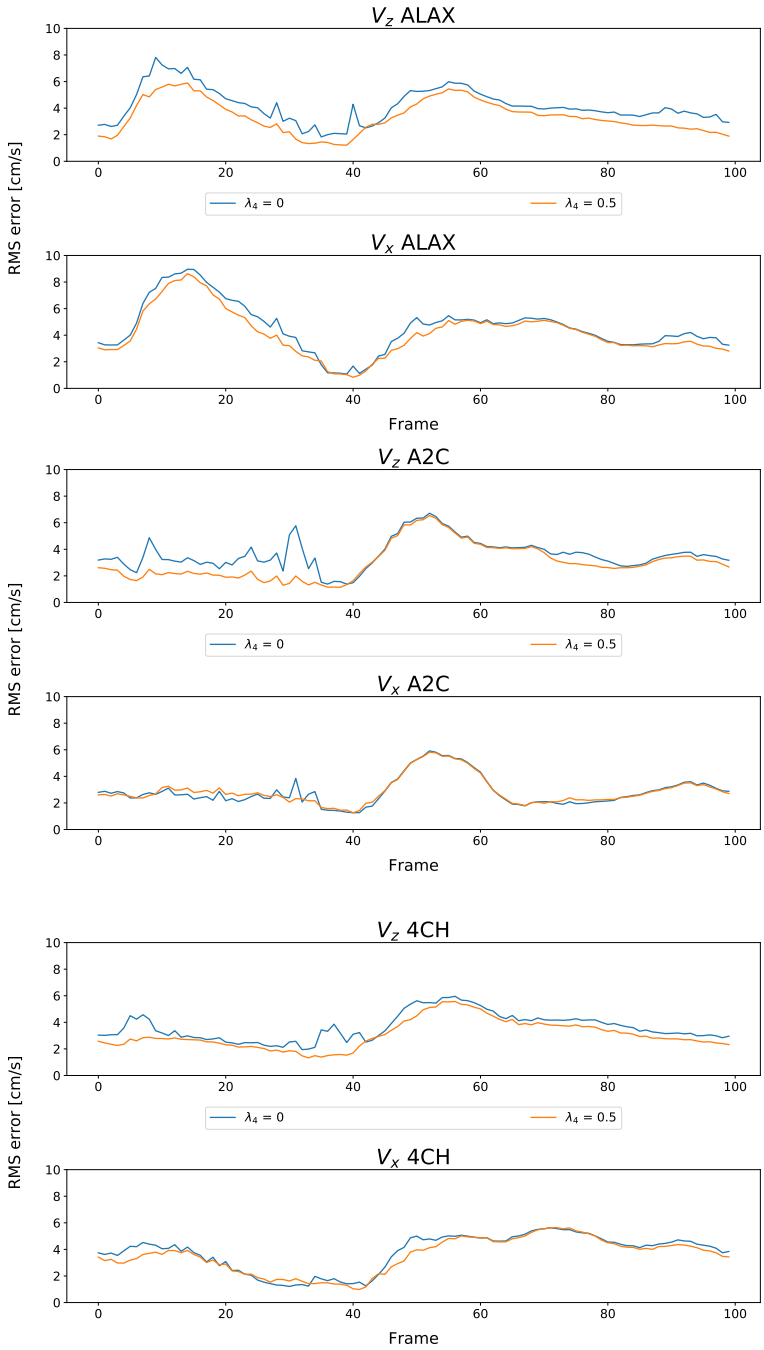

Figure 4.3: iVFM RMS errors in the whole domain for raw FUSK data, with $\lambda_4 = 0$ and $\lambda_4 = 0.5$. Views from the top: Apical long axis (ALAX), apical 2 chamber (A2C) and 4 chamber (4CH).

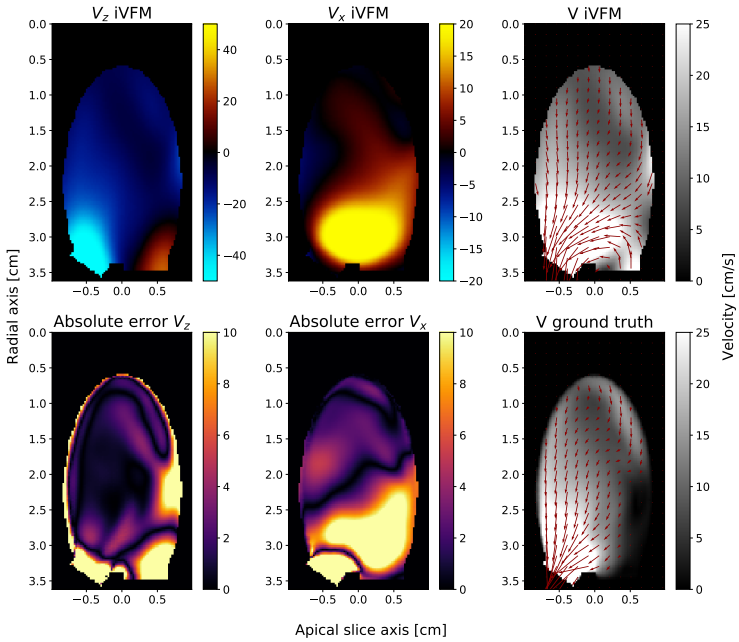

Figure 4.4: iVFM ALAX result for frame 18 from the clutter filtered and aliasing corrected FUSK data, analyzed with $\lambda_4 = 0$. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

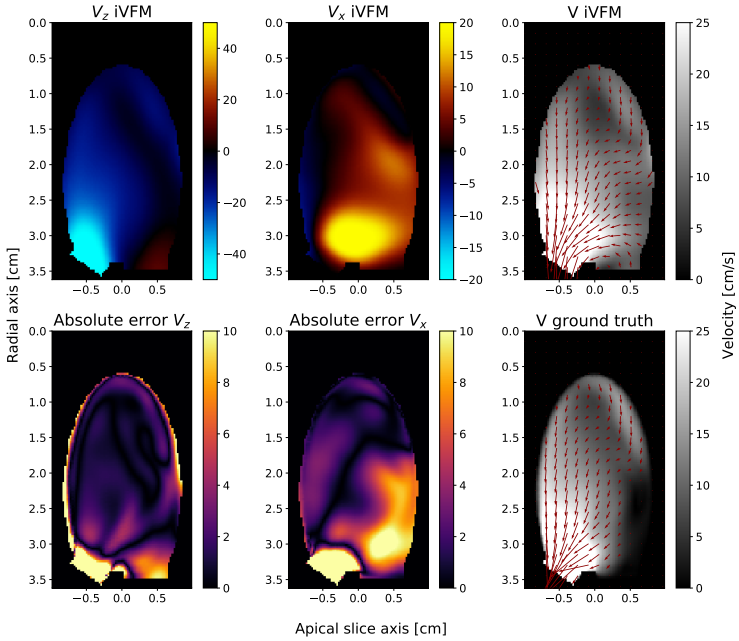

Figure 4.5: iVFM ALAX result for frame 18 from the clutter filtered and aliasing corrected FUSK data, analyzed with $\lambda_4 = 0.5$. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

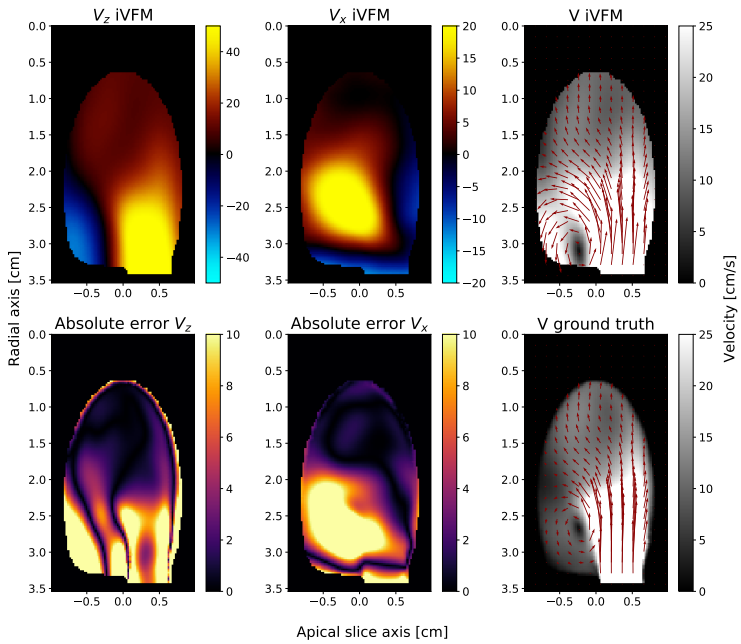

Figure 4.6: iVFM 4CH result for frame 50 from the clutter filtered and aliasing corrected FUSK data, analyzed with $\lambda_4 = 0$. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

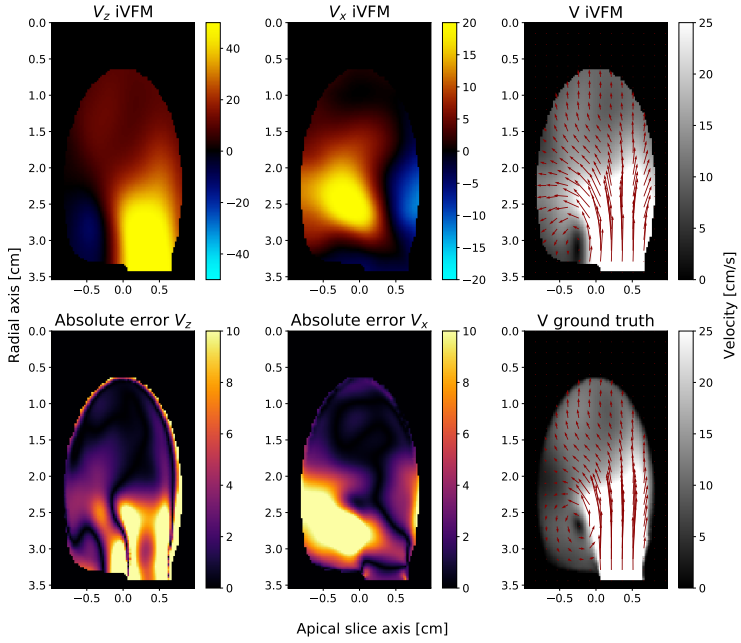

Figure 4.7: iVFM 4CH result for frame 50 from the clutter filtered and aliasing corrected FUSK data, analyzed with $\lambda_4 = 0.5$. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

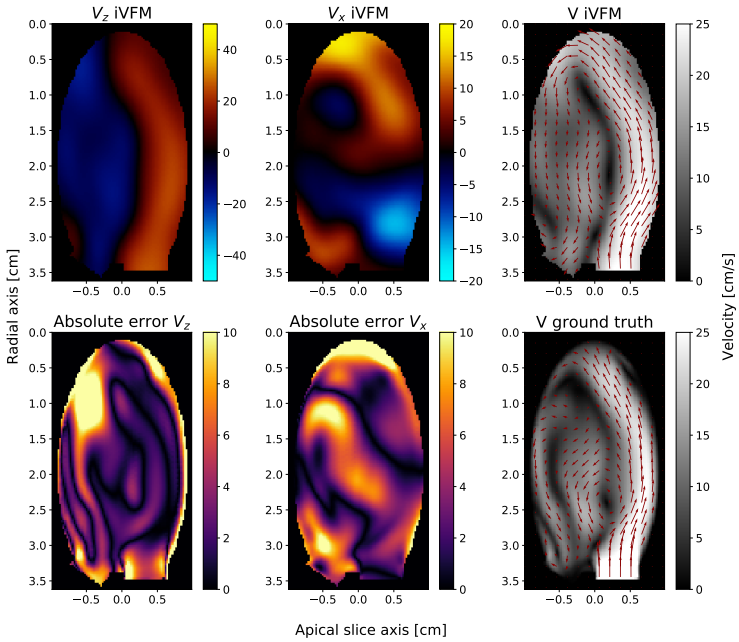

Figure 4.8: iVFM ALAX result for frame 90 from the clutter filtered and aliasing corrected FUSK data, analyzed with $\lambda_4 = 0$. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

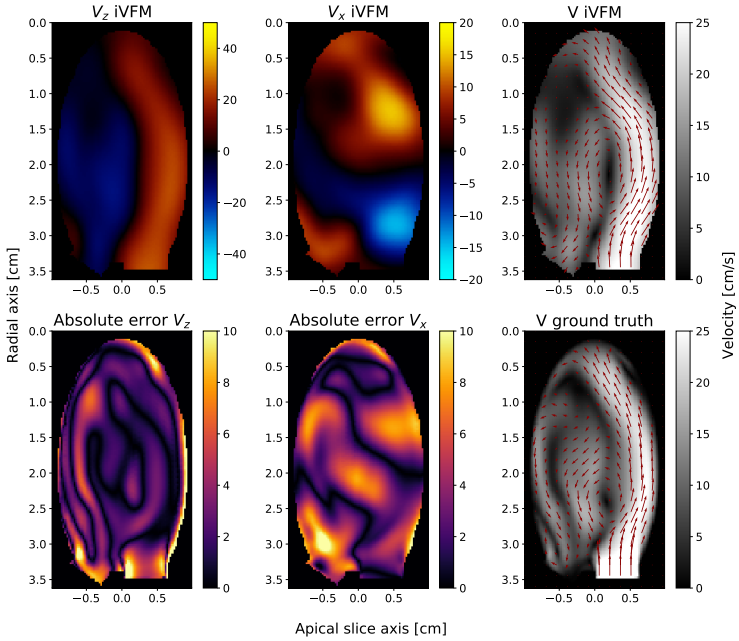

Figure 4.9: iVFM ALAX result for frame 90 from the clutter filtered and aliasing corrected FUSK data, analyzed with $\lambda_4 = 0.5$. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

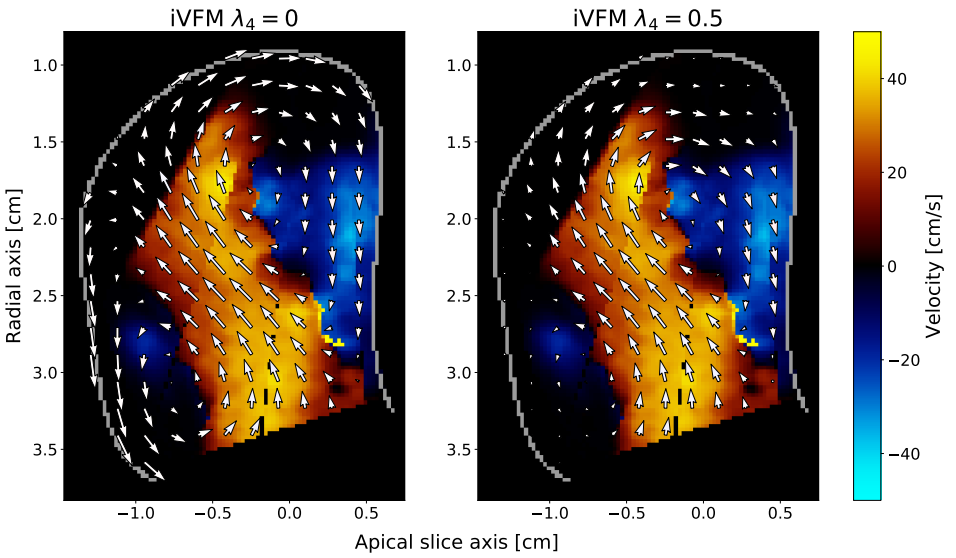

Figure 4.10: In vivo iVFM analysis with and without λ_4 regularization. Frame 48 from patient 4002.

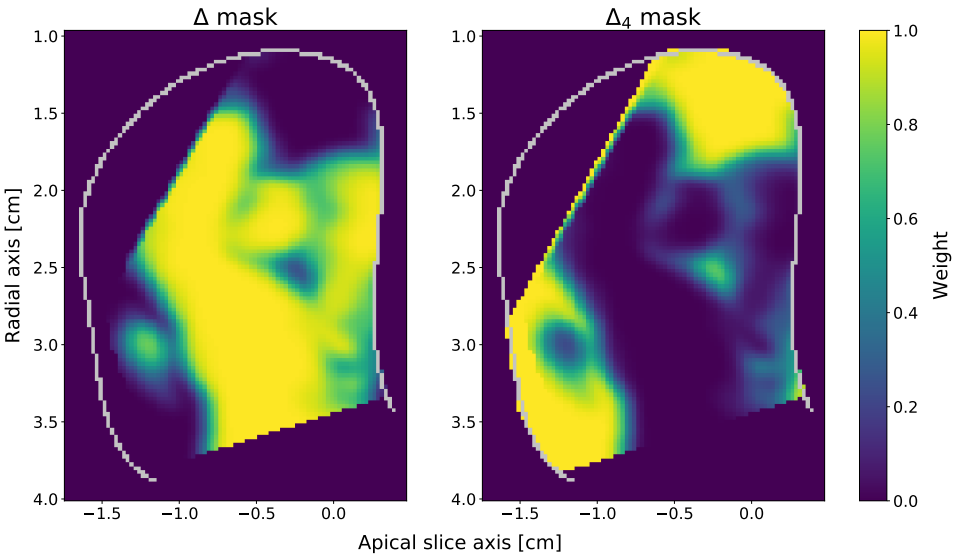

Figure 4.11: The Δ and Δ_4 masks used in the in vivo iVFM analysis of frame 48 from patient 4002.

4.2 Comparison of iVFM and ST

This section contains a comparison between iVFM and ST applied to both FUSK and in vivo data. Figure 4.12 summarizes the iVFM and ST performance for all the views and frames in an RMS error plot. The figure also contains iVFM results with V_x estimates, which will be presented in detail in section 4.3. The RMS error plots show a tendency for iVFM to have a slightly smaller V_z error, and for ST to have a significantly smaller V_x error. This general difference in performance can be seen in all frames, except for the almost flowless time interval around frame 40, and is more prominent for ALAX and 4CH than for A2C.

Figure 4.13 to 4.18 show ST and iVFM results for ALAX frame 18, A2C frame 70 and 4CH frame 90. All three examples illustrate the tendency for ST to underestimate and iVFM to overestimate velocities. This becomes especially evident in figure 4.17 and 4.18, where the details of the flow are lost in both the ST and iVFM solution, due to under- and overestimation respectively. The result is that the field produced by iVFM suffers from V_x errors and the field produced by ST suffers from V_z errors.

Difference plots for iVFM and ST were constructed to investigate the under- and overestimation in ST and iVFM more thoroughly. The plots can be seen in figure 4.19 to 4.24. These are scatter plots with the true velocity plotted along the x-axis and the difference between the iVFM or ST velocity and the true velocity on the y-axis. Underestimated velocities will lead to points below the $y = 0$ line for positive true velocities and above for negative true velocities, and the opposite for overestimated velocities. The difference plots were made for all three views and V_z and V_x separately. ST shows a clear underestimation trend, especially for V_x, but also for V_z at high absolute velocities. For small positive velocities, ST has a tendency to overestimate the velocity. iVFM, on the other side, shows no overall over- or underestimation trend in V_x. For V_z the iVFM trend is an overestimation of small absolute velocities and an underestimation of high absolute velocities.

In vivo comparisons of ST and iVFM are found in figure 4.25, 4.26, 4.27 and 4.28 for patient 4002 and 4.29 for patient 4018. ST and iVFM seem to agree on the direction in most cases, but iVFM appears to overall result in flow fields with higher velocities, as seen for example in figure 4.26 and 4.28. ST sometimes clearly underestimates radial velocities, such as in the central part of the ventricle in figure 4.25 and in frame 49 in figure 4.29. Another important observation is that seemingly small differences in the Doppler input can lead to widely different iVFM results, as seen from frame 46 to 47 in figure 4.29, where the aliasing correction has failed to correct every aliased velocity in frame 47. The corresponding ST solutions are more or less unaffected, while the iVFM field is influenced globally.

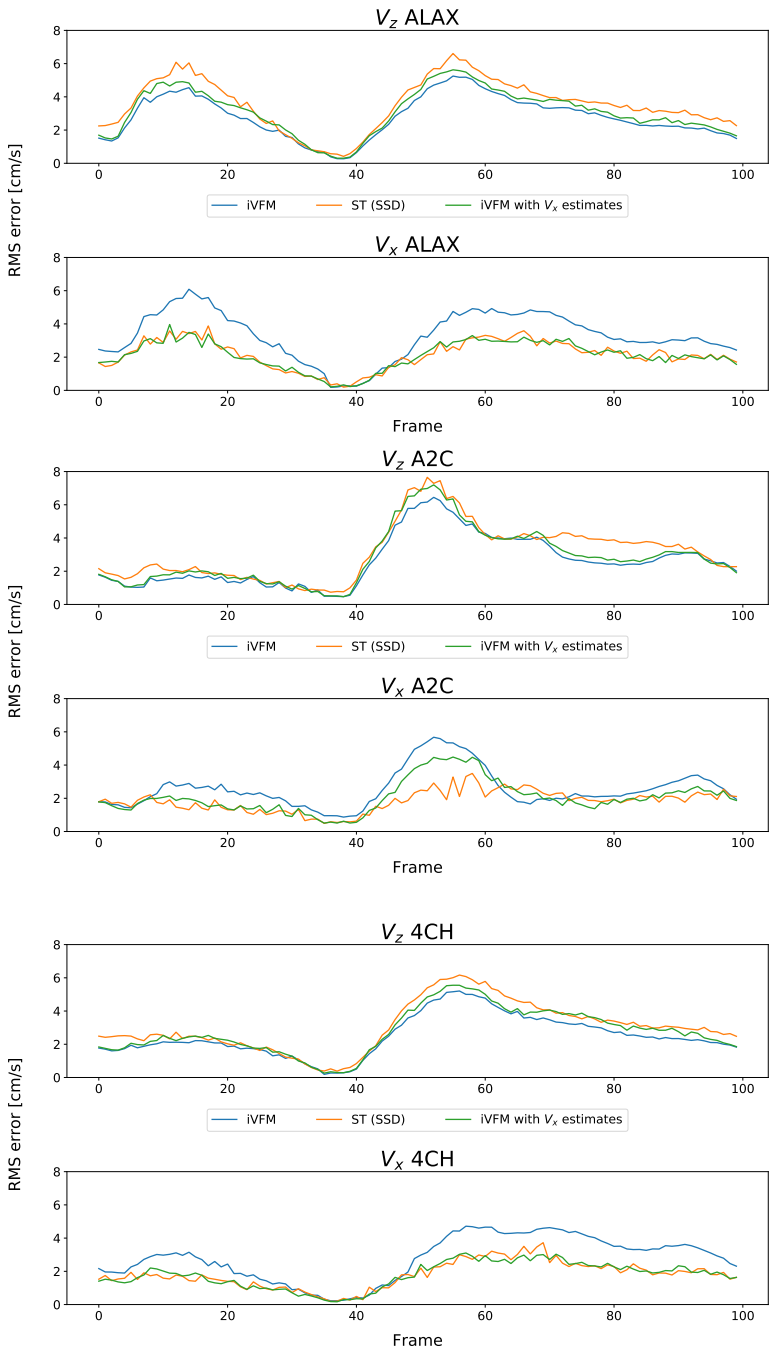

Figure 4.12: iVFM, ST and iVFM with V_x estimates from ST RMS errors in the whole domain for raw FUSK data. Views from the top: Apical long axis (ALAX), apical 2 chamber (A2C) and 4 chamber (4CH).

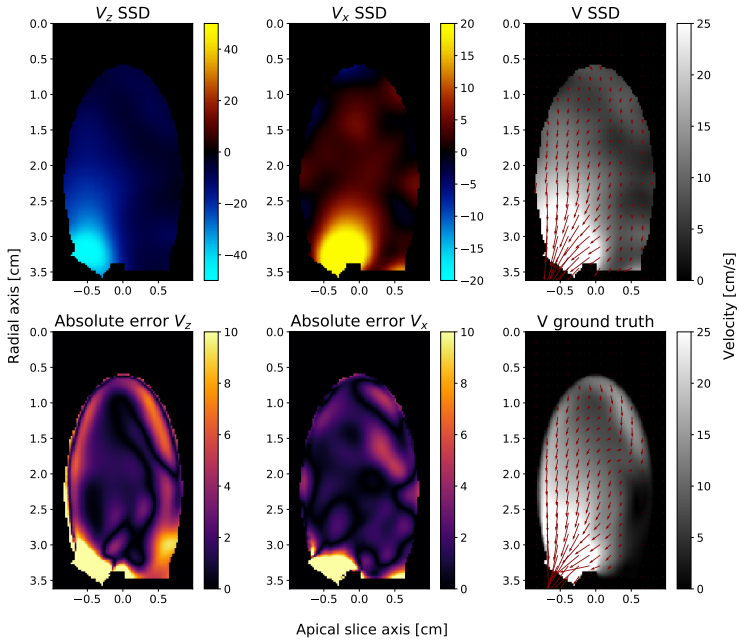

Figure 4.13: ST ALAX result for frame 18 from the clutter filtered and aliasing corrected FUSK data, with a Gaussian smoothing. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

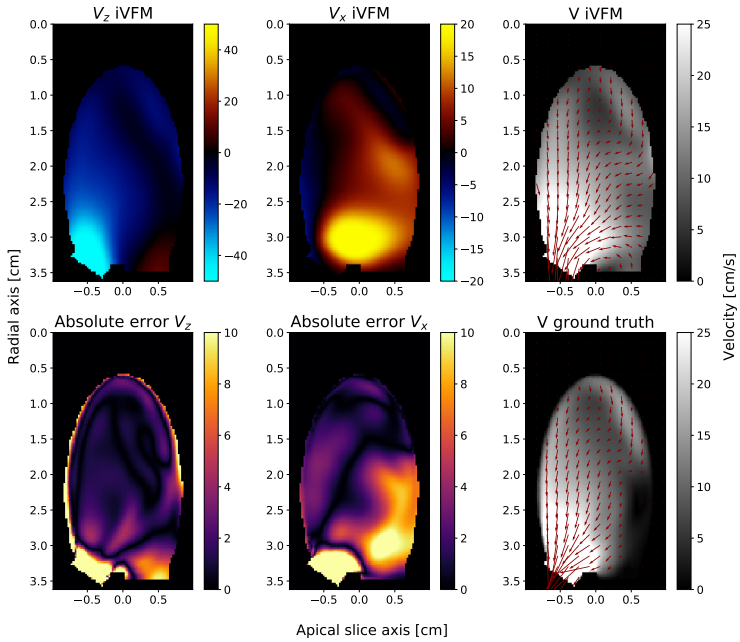

Figure 4.14: iVFM ALAX result for frame 18 from the clutter filtered and aliasing corrected FUSK data. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

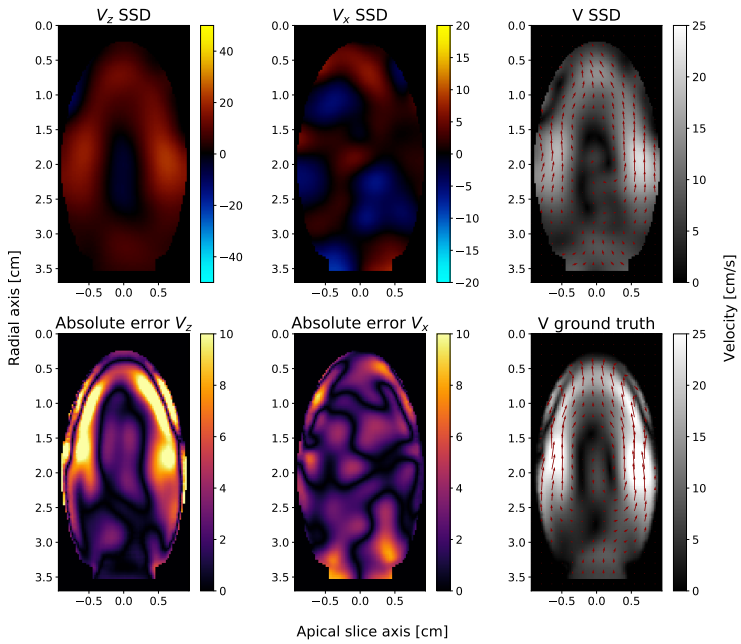

Figure 4.15: ST A2C result for frame 70 from the clutter filtered and aliasing corrected FUSK data, with a Gaussian smoothing. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

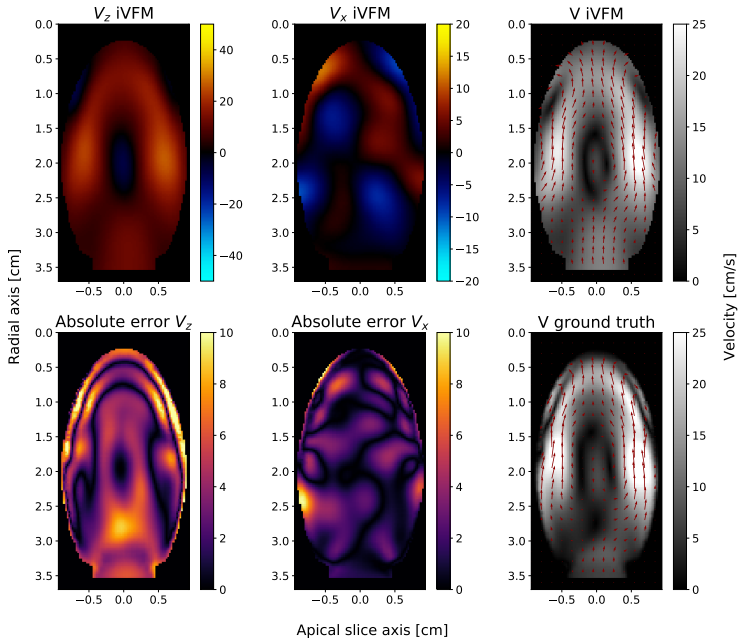

Figure 4.16: iVFM A2C result for frame 70 from the clutter filtered and aliasing corrected FUSK data. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

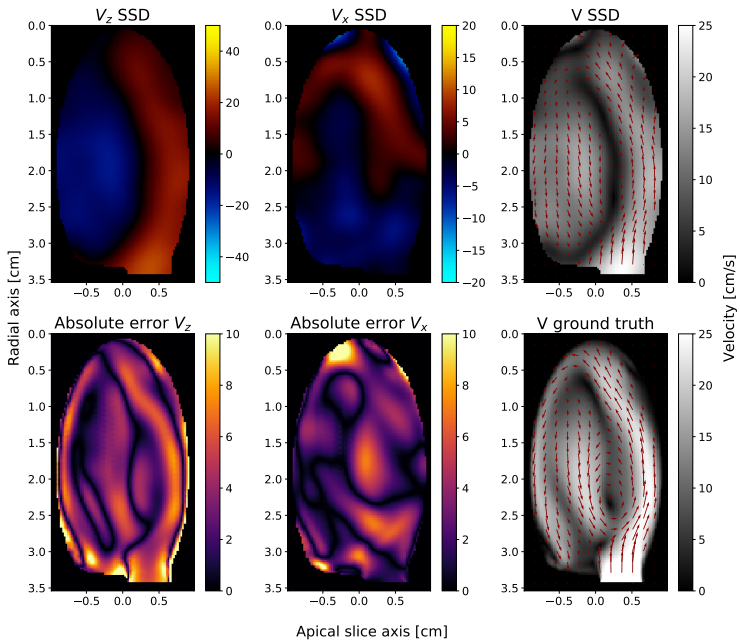

Figure 4.17: ST 4CH result for frame 90 from the clutter filtered and aliasing corrected FUSK data, with a Gaussian smoothing. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

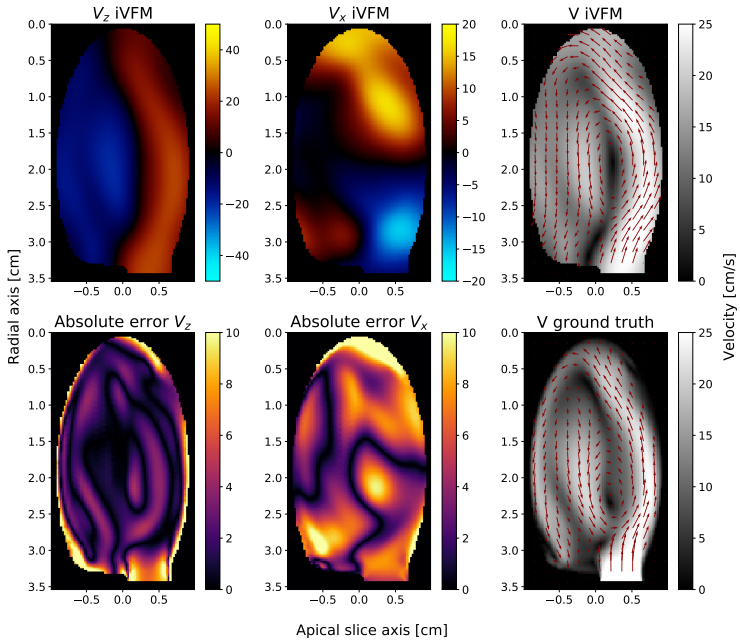

Figure 4.18: iVFM 4CH result for frame 90 from the clutter filtered and aliasing corrected FUSK data. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

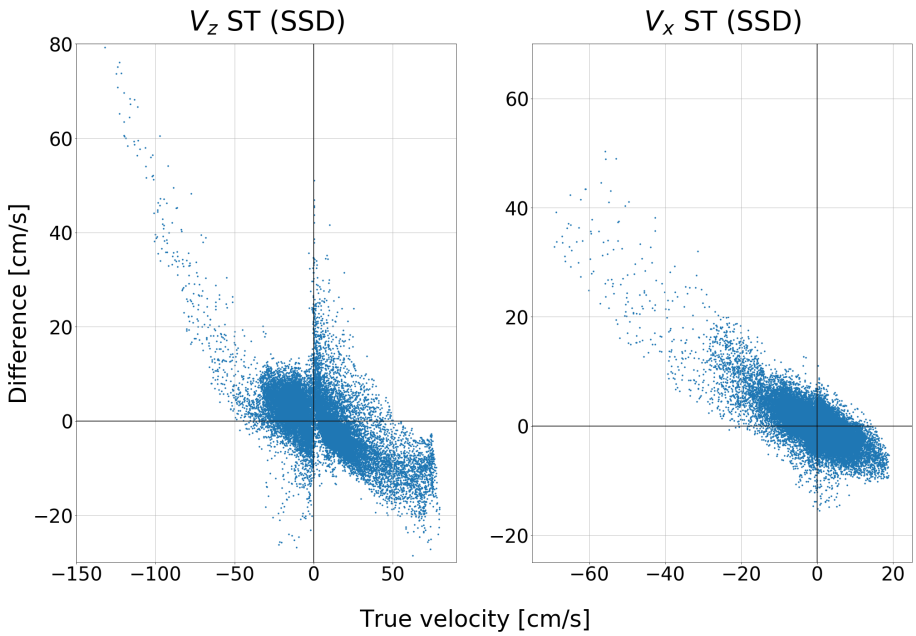

Figure 4.19: ST ALAX difference plot. The difference is defined as the ST velocity minus the true velocity.

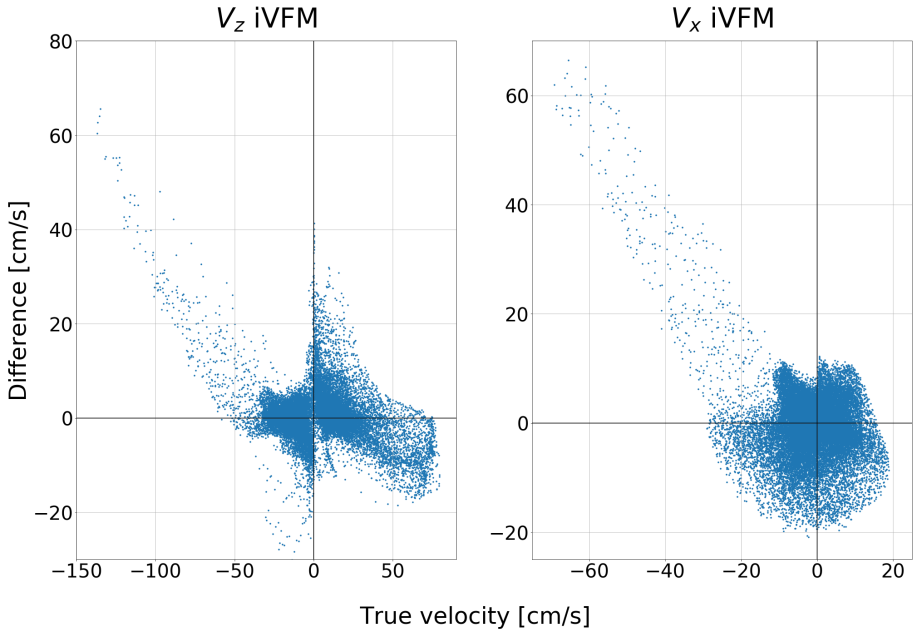

Figure 4.20: iVFM ALAX difference plot. The difference is defined as the iVFM velocity minus the true velocity.

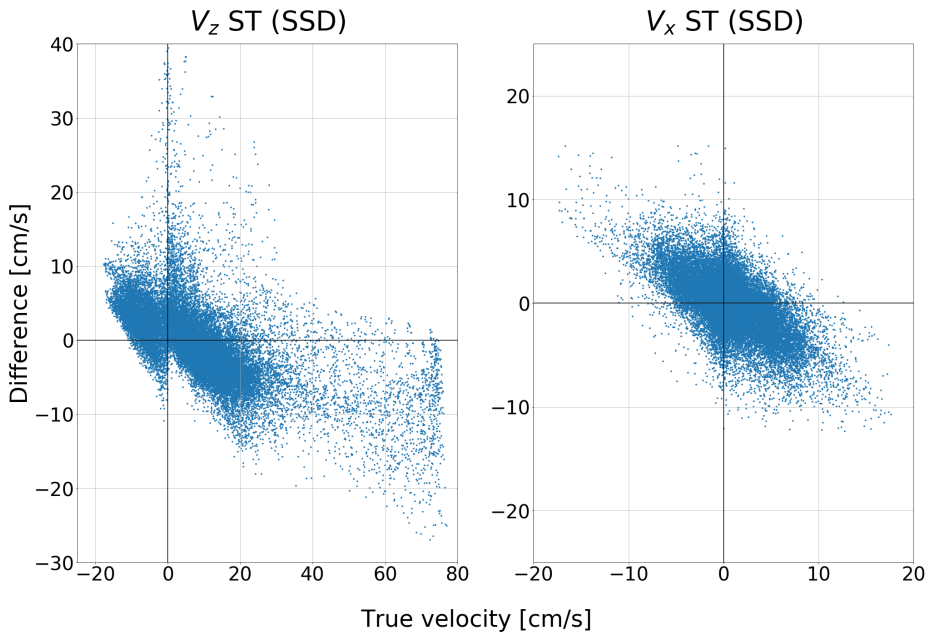

Figure 4.21: ST A2C difference plot. The difference is defined as the ST velocity minus the true velocity.

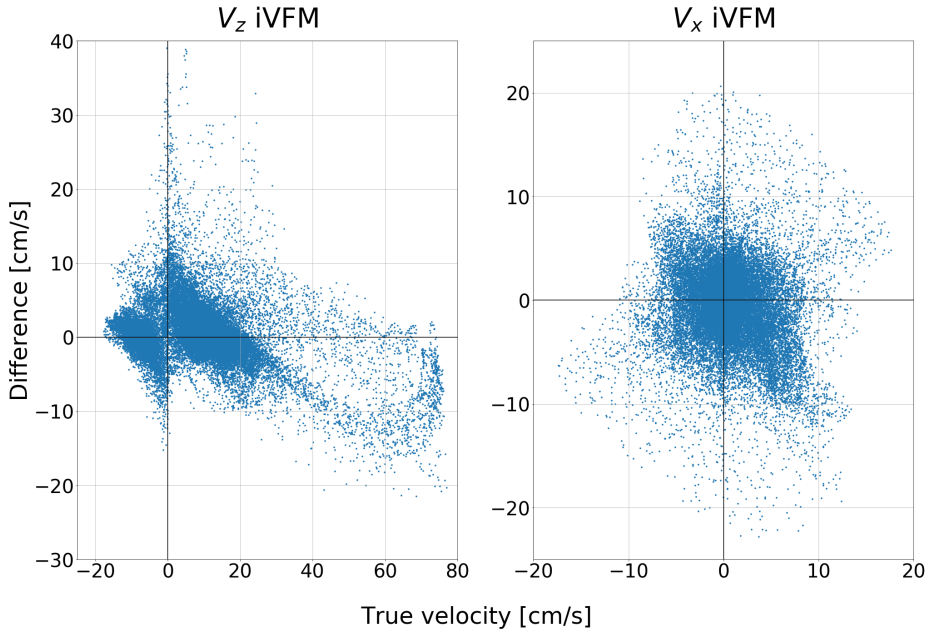

Figure 4.22: iVFM A2C difference plot. The difference is defined as the iVFM velocity minus the true velocity.

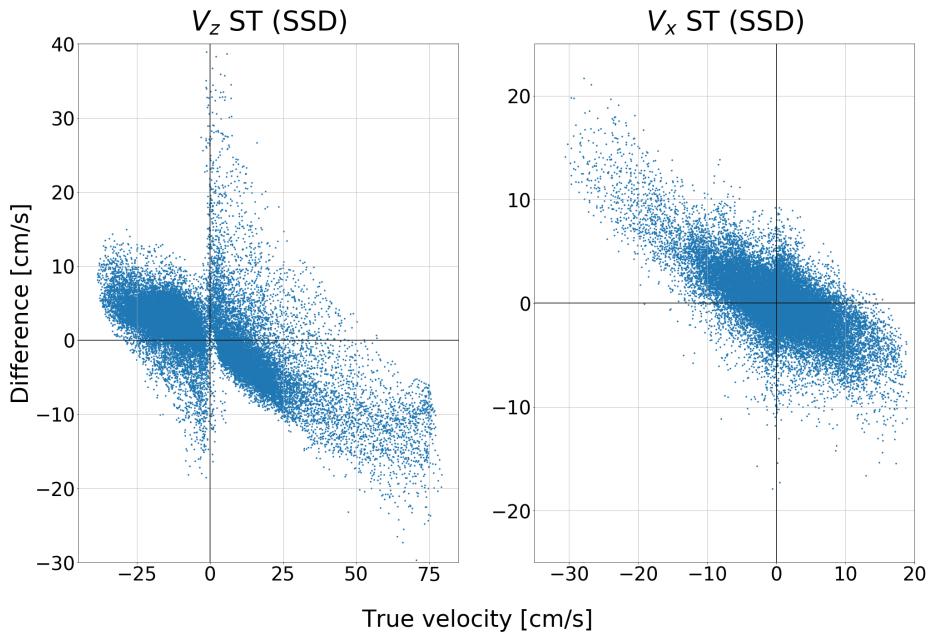

Figure 4.23: ST 4CH difference plot. The difference is defined as the ST velocity minus the true velocity.

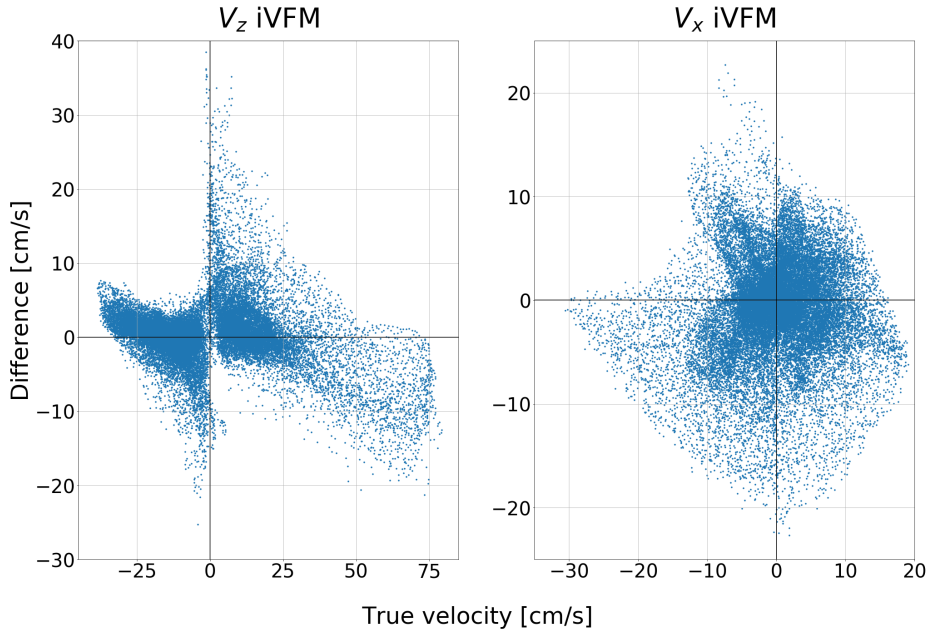

Figure 4.24: iVFM 4CH difference plot. The difference is defined as the iVFM velocity minus the true velocity.

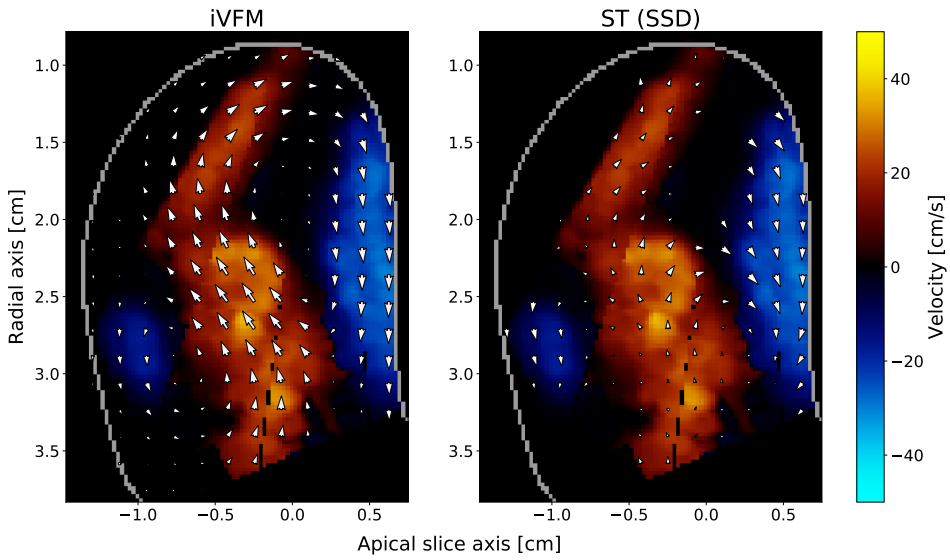

Figure 4.25: In vivo iVFM and ST analysis. Frame 27 from patient 4002.

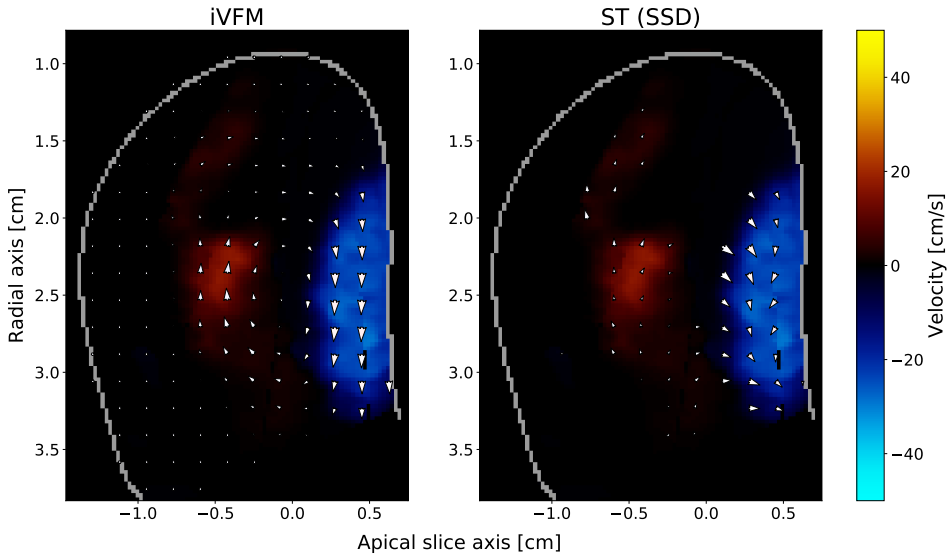

Figure 4.26: In vivo iVFM and ST analysis. Frame 30 from patient 4002.

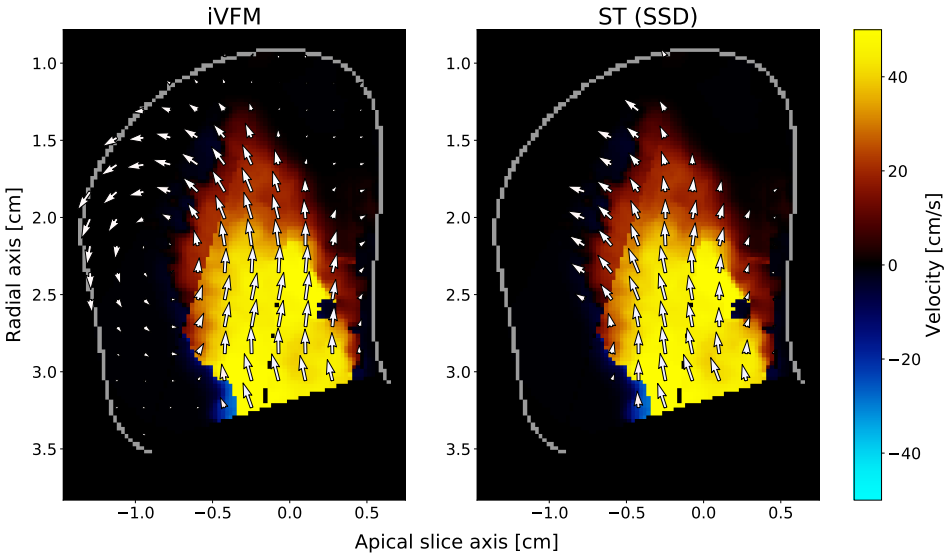

Figure 4.27: In vivo iVFM and ST analysis. Frame 45 from patient 4002.

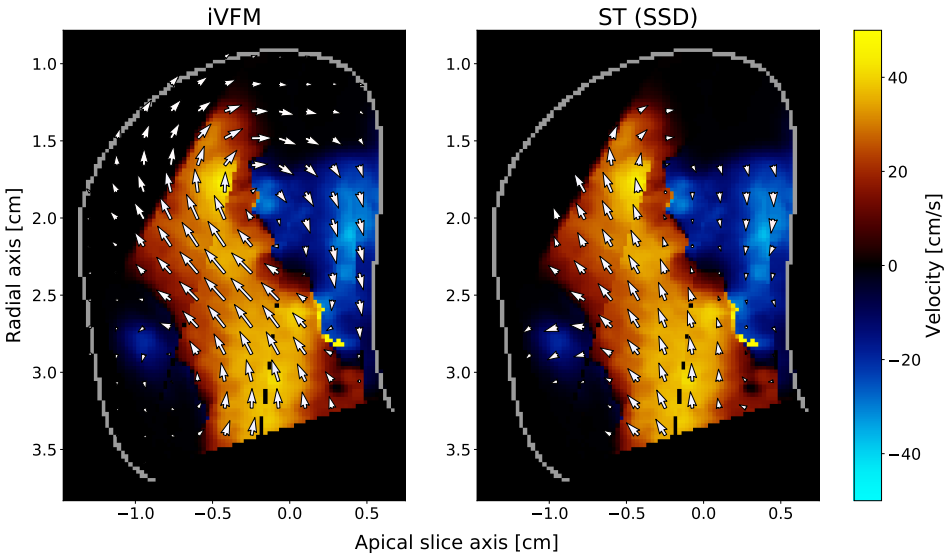

Figure 4.28: In vivo iVFM and ST analysis. Frame 48 from patient 4002.

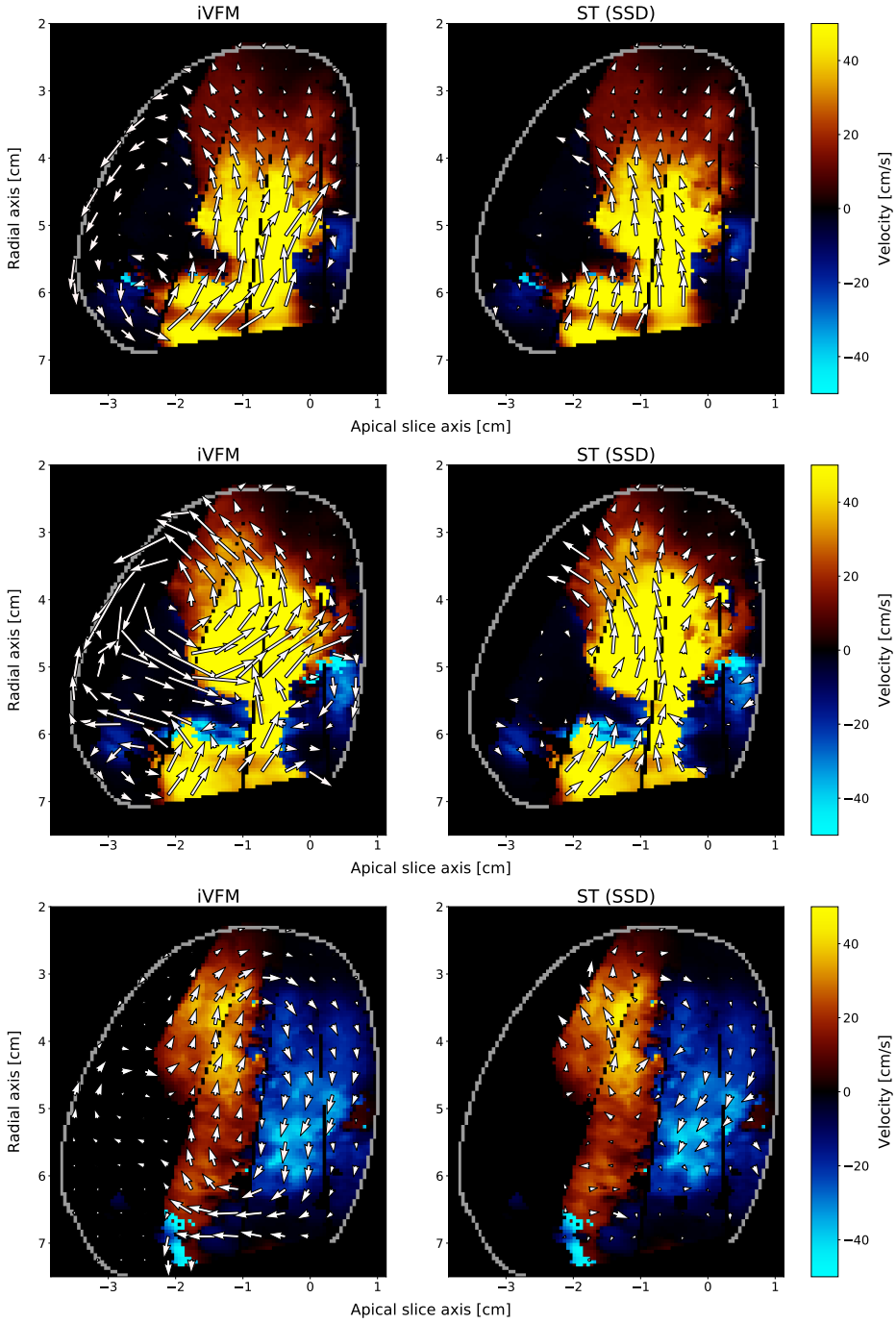

Figure 4.29: In vivo iVFM and ST analysis. Frame 46 (top), 47 (middle) and 49 (bottom) from patient 4018.

4.3 iVFM with V_x input from ST

The figure that shows how iVFM with V_x input from ST performs compared to the pure iVFM and ST, 4.12, can be found in section 4.2. For the most part, this iVFM version has the same level of V_x RMS error as the ST result, and almost the same level of V_z RMS error as the iVFM result. The RMS error is sometimes as high as the RMS error for the standard iVFM, as seen when comparing figure 4.30 to figure 4.31, but the typical behaviour is better described by comparing figure 4.32 to 4.33. As seen here, the V_z errors are almost identical and the V_x errors are heavily reduced in the iVFM versions that takes V_x estimates. By comparing with the V_x error in figure 4.17, we see that iVFM with V_x estimates is more or less on the same V_x error level as ST.

An in vivo analysis of iVFM with V_x input from ST and the conventional Doppler iVFM can be seen in figure 4.34. In frame 46 in this figure, the main differences between using and not using the V_x estimates consist of a few underestimated z-velocities in the central part of the domain and the suppression of a lateral left-to-right flow near the ventricle basis in V_x version. The removal of a strong lateral current can also be seen in frame 49. The usage of V_x estimates has however the biggest impact on frame 47, which is dominated by lateral flows in both directions in the Doppler iVFM. By using the V_x estimates the flow field behaves much more as expected, judging by the color Doppler and domain.

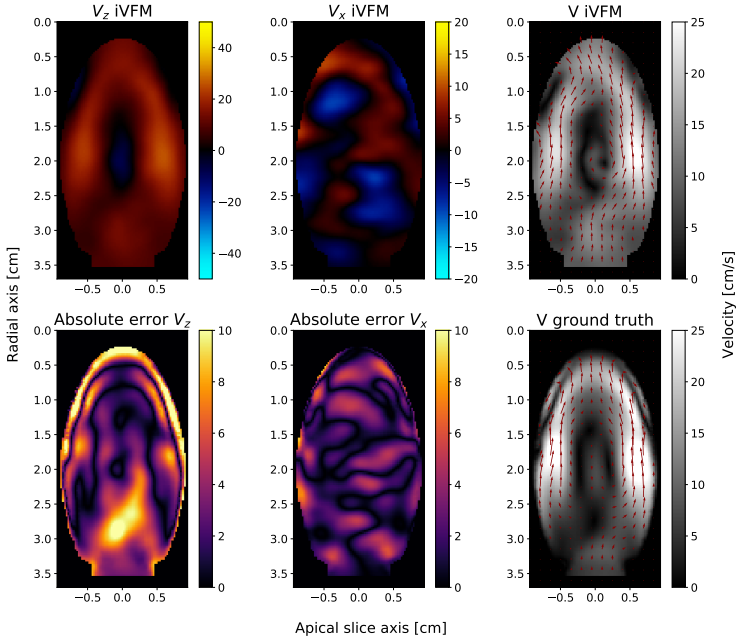

Figure 4.30: iVFM A2C result for frame 70 for iVFM with V_x estimates from ST. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

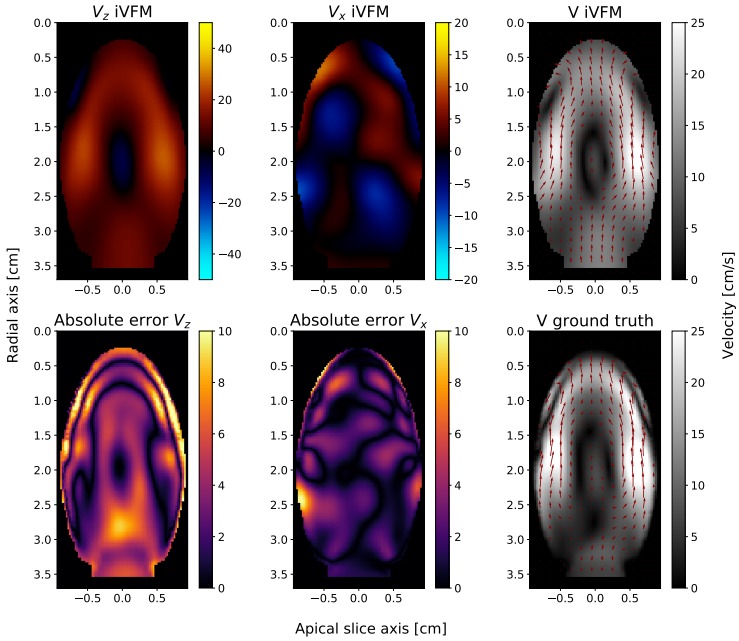

Figure 4.31: iVFM A2C result for frame 70 from the clutter filtered and aliasing corrected FUSK data. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

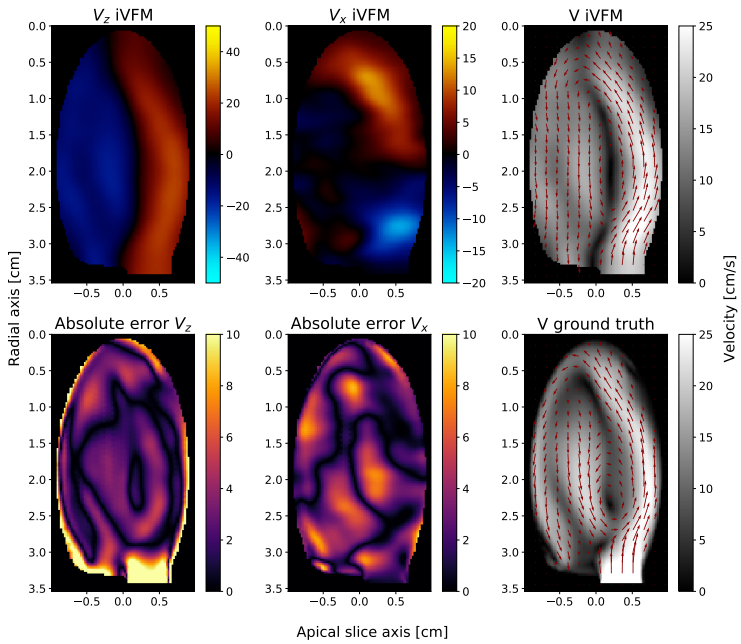

Figure 4.32: iVFM 4CH result for frame 90 for iVFM with V_x estimates from ST. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

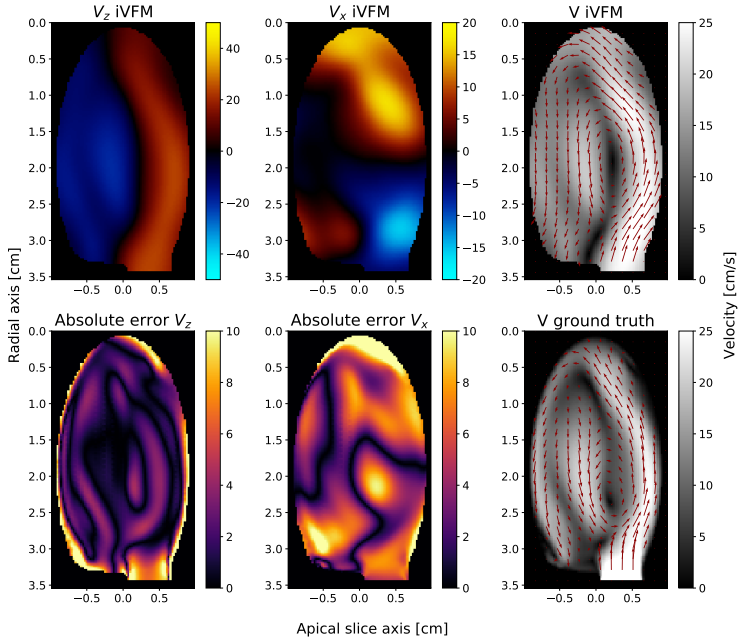

Figure 4.33: iVFM 4CH result for frame 90 from the clutter filtered and aliasing corrected FUSK data. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

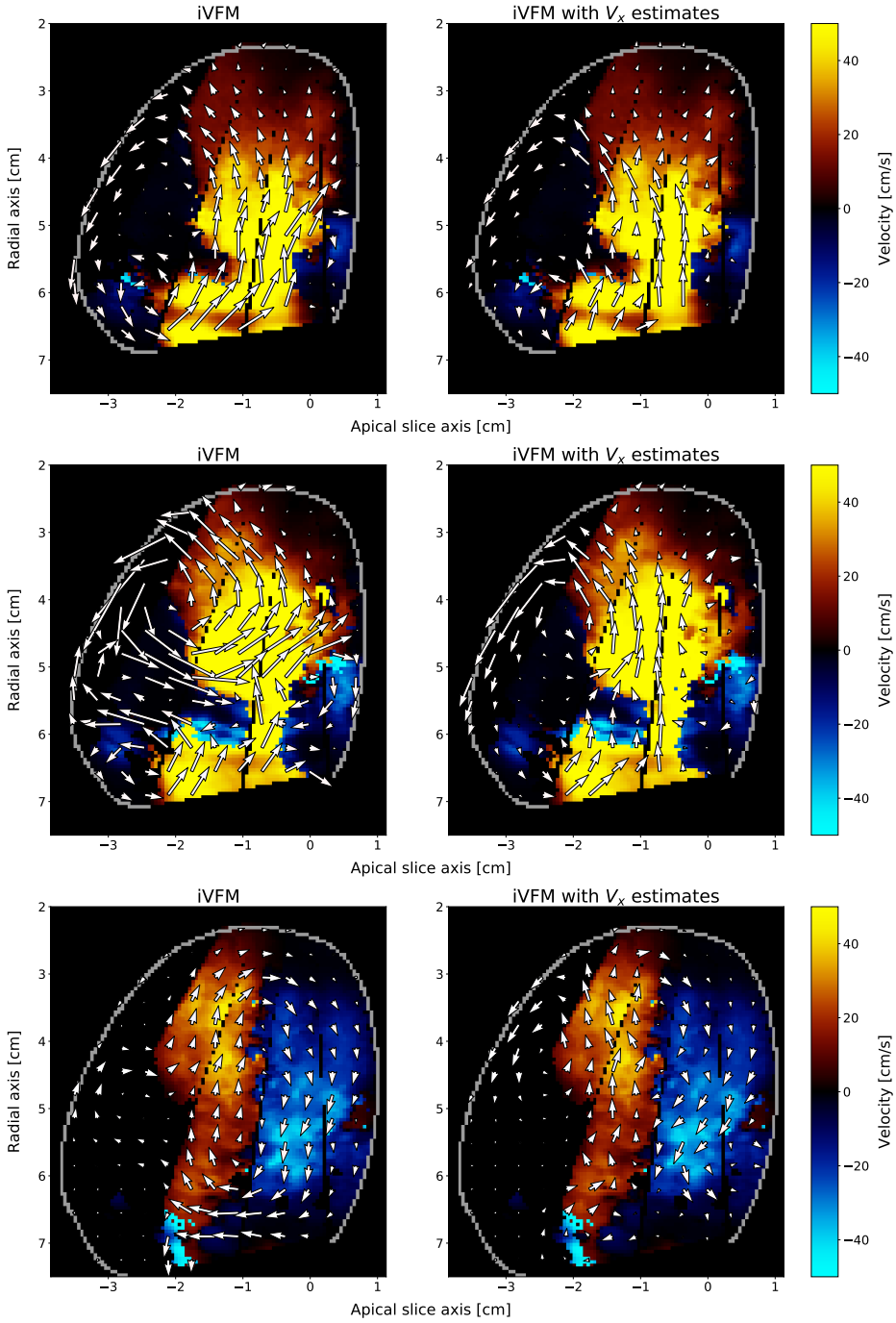

Figure 4.34: In vivo iVFM and iVFM with V_x estimates analysis. Frame 46 (top), 47 (middle) and 49 (bottom) from patient 4018.

4.4 iVFM with V_y gradient on FUSK data

The results from the iVFM with V_y gradient is summarized in figure 4.35. The RMS error plots show modest reductions in V_z error and fairly large reductions in V_x error. The V_x error reductions are present for all views and all frames except a small time interval around frame 35 with very little flow. The FUSK examples in figure 4.36 to 4.41 show the same trend of massive V_x error reduction and almost no changes to the V_z. In figure 4.37 and 4.39 the flow escaping through the lateral walls is reduced, and in 4.41 the vortex center is moved much closer to the true center which makes the field resemble the true field to a much larger extent. The remaining V_x errors are mainly situated along the domain walls, except for 4.39, where all the V_x error that is left is a small central patch.

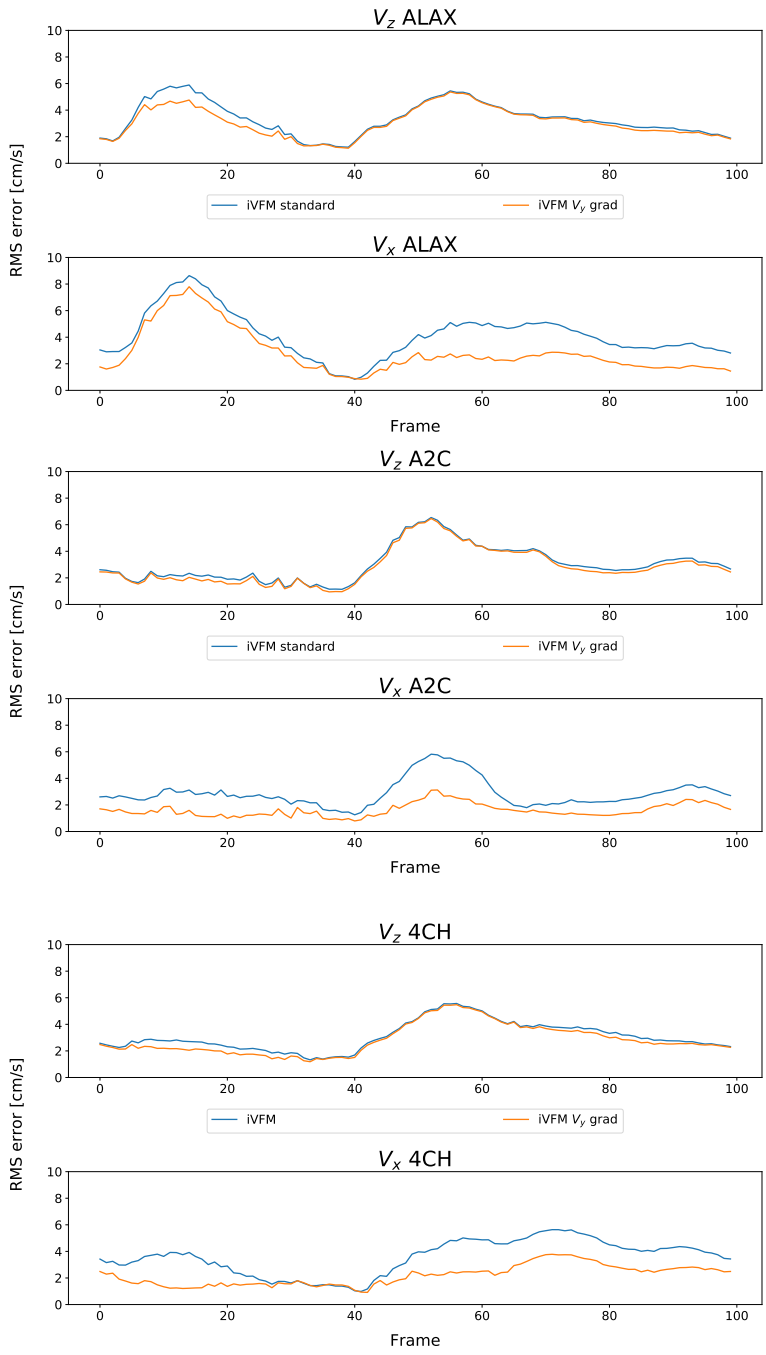

Figure 4.35: iVFM RMS errors in the whole domain for raw FUSK data. iVFM V_y grad is provided with the exact V_y gradient, all other parameters are kept at default values for both iVFM versions. Views from the top: Apical long axis (ALAX), apical 2 chamber (A2C) and 4 chamber (4CH).

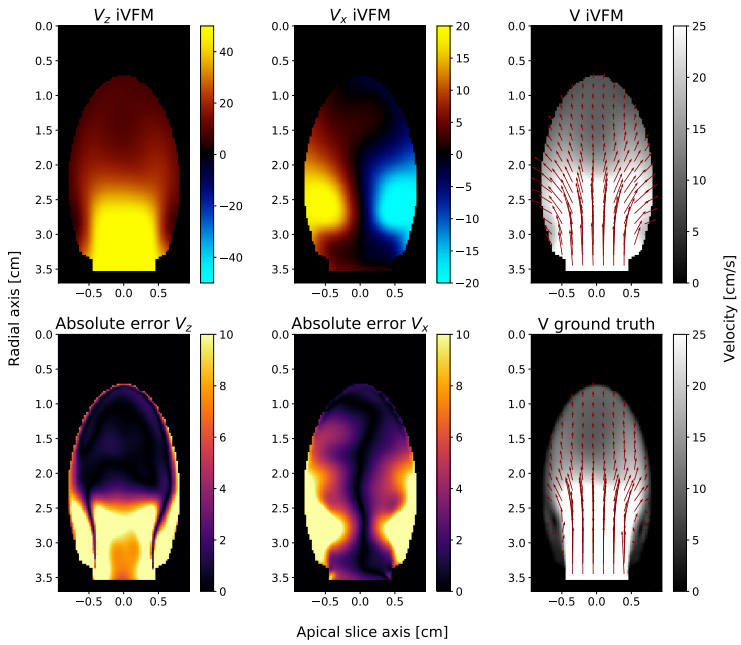

Figure 4.36: iVFM A2C result for frame 50 from the clutter filtered and aliasing corrected FUSK data. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

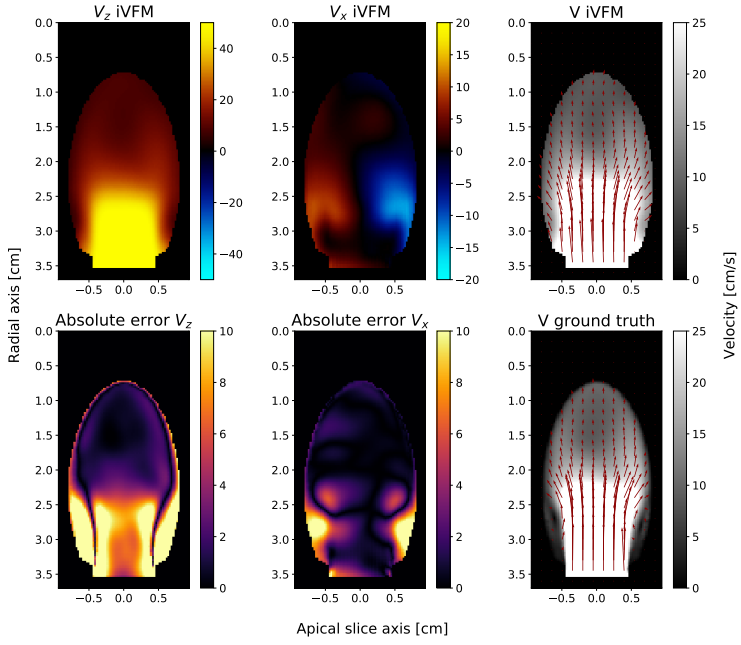

Figure 4.37: iVFM A2C result for frame 50 from the clutter filtered and aliasing corrected FUSK data. iVFM is provided with the exact V_y gradient, all other parameters are kept at default values. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

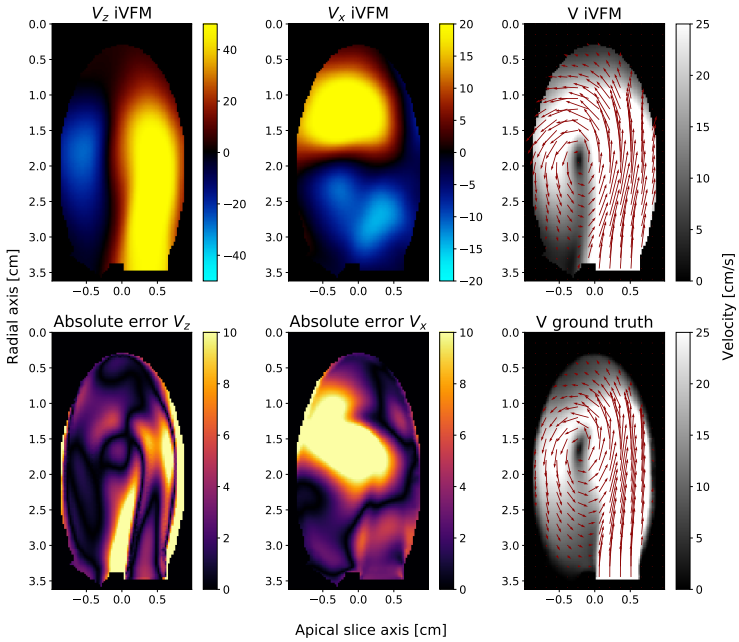

Figure 4.38: iVFM ALAX result for frame 60 from the clutter filtered and aliasing corrected FUSK data. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

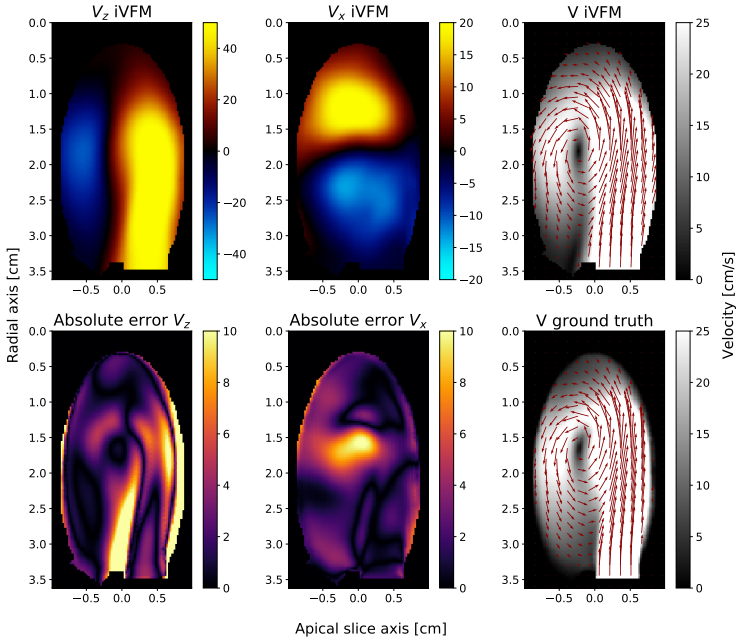

Figure 4.39: iVFM ALAX result for frame 60 from the clutter filtered and aliasing corrected FUSK data. iVFM is provided with the exact V_y gradient, all other parameters are kept at default values. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

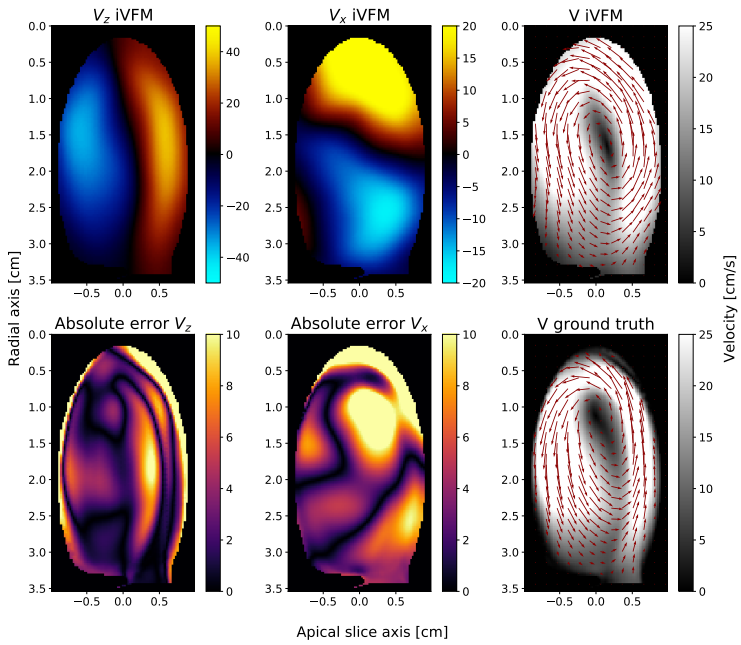

Figure 4.40: iVFM 4CH result for frame 70 from the clutter filtered and aliasing corrected FUSK data. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

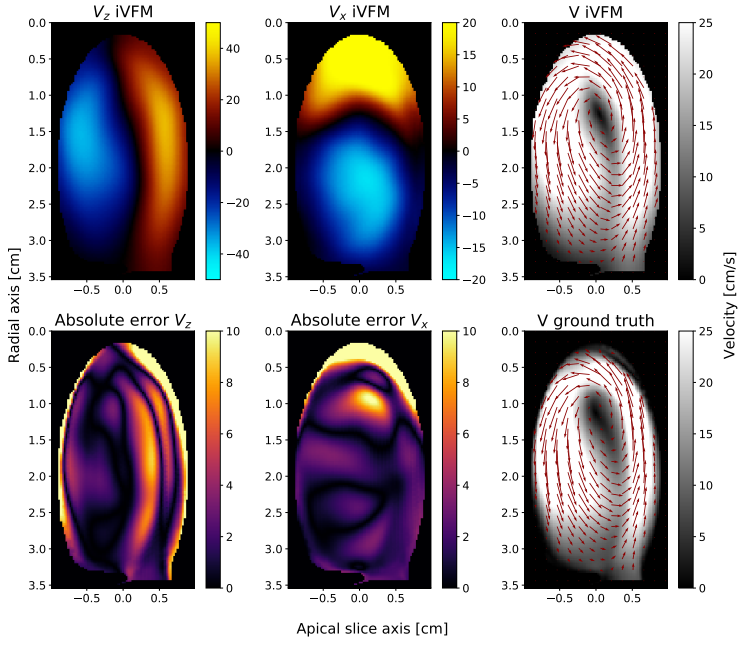

Figure 4.41: iVFM 4CH result for frame 70 from the clutter filtered and aliasing corrected FUSK data. iVFM is provided with the exact V_y gradient, all other parameters are kept at default values. The top row shows the reconstructed velocity field, while the bottom row shows the absolute error and ground truth.

Chapter 5

Discussion

The main goals of this master thesis were to implement and evaluate an additional regularization parameter in iVFM, evaluate the effect of out-of-plane motion and show how new out-of-plane measurements could be included in iVFM. Another goal was to implement iVFM in a practical pipeline for in vivo data analysis before conducting a thorough comparison between ST and iVFM both in silico and in vivo. This chapter discusses the most important findings.

5.1 Aliasing correction

The ST approach to aliasing correction has shown promising results. The time intervals from the FUSK data set that originally contained aliased velocities, and therefore had to be left disregarded in the writer's project thesis, could now be included in the analysis. Furthermore, the establishment of a proper anti-aliasing treatment opened the door to in vivo iVFM analysis. The aliasing correction was used non-critically with default settings throughout this thesis, with the consequence that it sometimes failed to correct all aliased velocities. This can be seen in figure 4.29, where it becomes apparent that iVFM can be very sensitive even to small errors in the aliasing correction procedure. Since iVFM is an algorithm that finds a global minimum, part of the flow going in the wrong direction can have consequences for the whole flow domain. When using iVFM in a clinical context, extra attention should therefore be put into ensuring that the aliasing correction has succeeded, compared to for instance ST with Gaussian smoothing or the iVFM version with V_x estimates from ST.

5.2 λ_4 – the radial flow constraint parameter

The new constraint on radial flow was created to constrain an effect that was discovered when starting to give different weight to different Doppler estimates for in vivo iVFM analysis. iVFM tended to suggest illogically strong flows in areas that were given little or no weight because they were situated outside the color flow mask. The new constraint on high radial velocities outside the color flow mask has shown itself to work as intended. The FUSK results are characterized by fairly small dropout regions, as illustrated by figure 4.1, and therefore gives the λ_4 constraint a limited scope. The effects are however a reduction of the error in most frames, without contribution to more error in the frames where it turns out less effective. The λ_4 constraint has proven even more useful in in vivo analysis, where the dropout regions generally are much bigger than the FUSK ones. Radial flows are effectively restricted and the new global solution adapts accordingly, as shown in the in vivo figure 4.10.

There are two main challenges with the new restriction on radial flow. The first is its penalization of all radial flows outside the color flow mask. Even small flows are drawn towards zero, when all one really knows for certain is that the radial flow is somewhere between zero and the cutoff velocity of the clutter filter in the dropout areas. If the penalization of velocities over the cutoff velocity only could be implemented, the λ_4 regularization could probably prove itself even more effective. One could argue that adjustments of the λ_4 parameter would serve more or less the same function, regulating the amount of radial flow allowed, but that would require different parameter adjustments for different cutoff values. It could be interesting to establish a tuning relation between the cutoff velocity and the λ_4 value, especially if the non-linear penalization of radial flow suggested above proves too difficult. iVFM is after all based on the solving of a linear problem, and it is not obvious how one would implement such a non-linear penalization.

The other issue with the radial flow constraint becomes apparent when parts of the ventricle domain lies outside the ROI, which defines where Doppler estimates are available. This sometimes leads to "highways" of radial flow that are unlikely to be real, like in frame 46 in figure 4.29. However, it is less obvious how to deal with this artifact, given that we cannot really implement on a general basis whether or not to constrain radial velocities in these areas.

5.3 Comparison to ST and iVFM with V_x estimates from ST

Even though a comparison between iVFM and ST was conducted in the writer's project thesis, it was rather brief and only gave clues about the most important differences between the two VFI methods. In this master thesis, a new comparison

was completed with alias corrected FUSK data sliced in three different ways to mimic cardiac views used in the clinic. Furthermore, this time the performance of the two methods could be compared on in vivo data analysis and, as first purposed in the project thesis, a hybrid iVFM method combining Doppler estimates with V_x estimates from ST was analyzed. This was compared to the conventional iVFM and ST on both FUSK and in vivo data.

Not surprisingly, ST shows much less error for lateral velocities while iVFM is better at reconstructing the radial velocities. The general impression of underestimation in ST and overestimation in iVFM was partly confirmed by the difference plots: ST clearly has an higher tendency to underestimate velocities than iVFM. This conclusion is easier to draw from the V_x difference plots then from the V_z difference plots. The V_z differences between ST and iVFM are more vague, a result suspected to come from underestimated high velocities in the Doppler estimates. Nevertheless, the scatter plots show that iVFM produces solutions with both higher V_z and V_x velocities than ST. This general conclusion comes as no surprise, given that the ST fields underwent spatial Gaussian smoothing. The spatial smoothing smears out extreme values, but is an absolute necessity to get a coherent flow field from the raw ST results. Even though the FUSK analysis shows less overall errors for ST than for iVFM, they both seem to reconstruct the velocity field characteristics to more or less the same extent in the FUSK examples. Many smaller flow details are simply lost in both cases, and even though ST has the ability to trade some of the smoothness for more details by shrinking the Gaussian smoothing kernel, it is necessary to apply quite a lot of smoothing to prevent a wrinkled field.

The differences between the ST and iVFM performance seem to become more distinct for the in vivo comparisons. ST struggles a lot more than what the RMS plots from the FUSK analysis would suggest, and sometimes makes large underestimations of radial velocities such as in figure 4.25. This can possibly be corrected with some tuning of the ST settings, but it looks like a reoccurring source of ST error when the a group of radial velocities are placed in a relatively narrow band. iVFM, on the other hand, gets thrown off easily by errors in the anti-aliasing procedure, like in frame 47 in figure 4.29.

The perhaps most interesting finding comes not from the comparison of ST and iVFM, but from the iVFM version that combines Doppler values and V_x estimates from ST. This seemingly combines the best parts of iVFM and ST into an overall better VFI method, but only when the V_x ST estimates are more accurate than the V_x that results from the conventional Doppler iVFM. Since the ST library is already used for the aliasing correction, the practical transition to producing additional ST V_x estimates is minimal.

Even though iVFM with V_x estimates from ST looks like the best performing iVFM version so far, the "best of both worlds"-trend observed in the results of this master thesis, should not be taken for granted. The combined iVFM version might easily inherit the underestimating nature of the ST from heavily underestimated V_x input, or the chaotic nature of iVFM when given aliased Doppler estimates. The reconstruction of velocities outside the ROI will no matter what suffer from the same "highway"-effect mentioned in section 5.2 and seen in frame 47 in figure 4.34.

5.4 iVFM with V_y gradient on FUSK data

The inclusion of the out-of-plane flow gradient in the divergence term of iVFM, has shown itself to be a computationally cost effective way to exploit ultrasonic measurements in three dimensions. It omits the colossal matrices one has to face in a full 3D iVFM implementation, but can nevertheless make use of 3D ultrasound measurements, which will become more available as the world of ultrasound progresses towards 3D. The fact that the V_x error was substantially reduced for all three views points in the direction that the 2D iVFM is more prone to errors originated from the faulty non-divergence assumption than assumed earlier. For views with a small out-of-plane flow, such as ALAX, this assumption has been considered a satisfactory approximation. This master thesis has however not tested a scheme for including in vivo V_y gradient measurements, so bear in mind that these conclusions are drawn from an idealized simulation set up.

The remaining error sources are believed to be the imperfect walls and imperfect nature of the FUSK Doppler estimates. This conclusion is supported by a disappearance of virtually all errors for brief test completed with only CFD data as both Doppler input and V_y gradient, as well as with almost perfect walls. Another possible error source is the smoothing regularization, which may have a lower optimal value when the V_y gradient is provided to iVFM. The V_y gradient gives iVFM more precise information especially about the areas where the 2D flow field has rapid spatial changes as it flows out of the plane, and a high smoothing regularization could potentially smear out correct flow details.

5.5 Further work

A natural next step in the iVFM development outlined in this master thesis, would be to set up an analysis of the V_y gradient on in vivo data. The results from the FUSK V_y gradient test are clear: V_x errors can be heavily reduced with good measurements of the out-of-plane flow. Such measurements could come from full volume ST or from ST in a sector just wide enough to get an a out-of-plane measurement.

One of the strengths of a global minimization approach with regularization terms, such as iVFM, is the possibility to add new ways to penalize unwanted or illogical solutions through the addition of regularization terms. Thus, it is a relatively straight forward process to combine all iVFM improvements to an improved iVFM. The λ_4 regularization and V_y gradient measurements can therefore be included in an iVFM version that also uses V_x estimates from ST. How well this combined iVFM version can perform also depends on the domain definitions, that up till now have been defined manually. Domain has not been the focus of this thesis, but it is clear that a connection between iVFM and the work done on domain segmentation with machine learning could ease the domain definition process itself, but more importantly, lead to more consistent flow results.

One of the weaknesses with the addition of new regularization terms is the growing number of regularization parameters. The tuning of these needs to be handled in a more objective manner than what has been done in this master thesis, should iVFM be enrolled in a clinical context. Future work should definitely look into how parameter tuning for iVFM can be implemented in an effective way.

Chapter 6

Conclusion

This master thesis intended to implement iVFM in a practical in vivo analysis pipeline, upgrade the traditional iVFM with an extra regularization term and show how out-of-plane flow could be included. It also aimed to evaluate these innovations through in silico and in vivo analysis. In addition to the iVFM development, an in silico and in vivo comparison of iVFM and ST was conducted to shed light on individual strengths and weaknesses of the two VFI methods, and to look into the possibilities for unification.

The results of the analysis showed that the regularization term that constrains radial flow, λ_4, contributes to a reduction of radial error due to iVFM overestimation in areas outside the color flow mask. But even with this regularization, iVFM showed a tendency to overestimate small radial velocities in the comparison to ST. ST, on the other hand, has challenges related to underestimation of both radial and lateral velocities. It proved to have a consistently lower V_x RMS error than iVFM, but the underestimation in ST causes iVFM to have a lower V_z RMS error than ST. The iVFM version that combines Doppler values with lateral V_x estimates proved itself as a good compromise, and had the lowest overall RMS error.

The in vivo results backed up the general conclusions from the in silico validation, but also helped to lay bare a few flaws of both iVFM and ST. The λ_4 regularization cannot constrain radial flow outside the ROI, which typically leads to overestimation and even completely artificial flows. iVFM also proved very sensitive to errors in the aliasing correction, which can leave the whole flow field affected. ST sometimes suffers from majorly underestimated velocities, but the iVFM version with V_x estimates from ST seems to hit a good compromise also for the frames where these flaws are present.

The analysis of the inclusion of the spatial gradient of V_y in the out-of-plane direction shows that out-of-plane motion has a lot to say for the result of the two dimensional iVFM. The idealized test shows that a perfect measurement of this flow gradient could reduce the lateral error massively, and it stands out as an alternative to a full 3D implementation of iVFM.

iVFM is now fitted into the visualization pipeline of the ultrasound viewer PyUS-View, developed at ISB, NTNU. It utilizes the ST library that is already incorporated in the ultrasound viewer to aliasing correct Doppler estimates and provide lateral velocity estimates to iVFM, which is now fully capable of velocity reconstruction of two-dimensional in vivo ultrasound data. With the implemented preconditioner for the iterative conjugate gradient solver that is being used, computation times are kept well within reasonable limits.

Future iVFM development should focus on trying to harvest the great error reduction potential that lies in the inclusion of a well measured out-of-plane gradient, as well as effective implementations of parameter tuning and automatic domain definitions.

Bibliography

[1] G. Pedrizzetti and F. Domenichini, "Left ventricular fluid mechanics: the long way from theoretical models to clinical applications," *Annals of biomedical engineering*, vol. 43, no. 1, pp. 26–40, 2015.

[2] J. A. Jensen, S. I. Nikolov, C. Alfred, and D. Garcia, "Ultrasound vector flow imaging—part i: Sequential systems," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 63, no. 11, pp. 1704–1721, 2016.

[3] D. Garcia, J. C. del Álamo, D. Tanné, R. Yotti, C. Cortina, É. Bertrand, J. C. Antoranz, E. Pérez-David, R. Rieu, F. Fernández-Avilés, *et al.*, "Two-dimensional intraventricular flow mapping by digital processing conventional color-doppler echocardiography images," *IEEE transactions on medical imaging*, vol. 29, no. 10, pp. 1701–1713, 2010.

[4] T. Tanaka, R. Asami, K. Kawabata, E. K. Itatani, T. Uejima, T. Nishiyama, and T. Okada, "Intracardiac vfm technique using diagnostic ultrasound system," *Hitachi Rev*, vol. 64, no. 8, pp. 488–492, 2015.

[5] K. C. Assi, E. Gay, C. Chnafa, S. Mendez, F. Nicoud, J. F. Abascal, P. Lantelme, F. Tournoux, and D. Garcia, "Intraventricular vector flow mapping—a doppler-based regularized problem with automatic model selection," *Physics in Medicine & Biology*, vol. 62, no. 17, p. 7131, 2017.

[6] M. Sælensminde, "Evaluation of vector flow mapping for intraventricular flows," *NTNU*, 2017.

[7] D. Kane, W. Grassi, R. Sturrock, and P. Balint, "A brief history of musculoskeletal ultrasound:'from bats and ships to babies and hips'," *Rheumatology*, vol. 43, no. 7, pp. 931–933, 2004.

[8] J. Curie and P. Curie, "Développement, par pression, de l'électricité polaire dans les cristaux hémièdres à faces inclinées," *Comptes rendus*, vol. 91, pp. 294–295, 1880.

[9] K. Dussik, "On the possibility of using ultrasound waves as a diagnostic aid," *Neurol Psychiat*, vol. 174, pp. 153–168, 1942.

[10] J. Kirkhorn, "Introduction to iq-demodulation of rf-data," *IFBT, NTNU*, vol. 15, 1999.

[11] D. Robinson, F. Chen, and L. Wilson, "Measurement of velocity of propagation from ultrasonic pulse-echo data," *Ultrasound in medicine & biology*, vol. 8, no. 4, pp. 413–420, 1982.

[12] L. Bohs, B. Geiman, M. Anderson, S. Gebhart, and G. Trahey, "Speckle tracking for multi-dimensional flow estimation," *Ultrasonics*, vol. 38, no. 1, pp. 369–375, 2000.

[13] T. Grønli, "Applications of penalized b-spline grids in ultrasound blood flow imaging," *NTNU*, 2017.

[14] X. Lai, H. Torp, and K. Kristoffersen, "An extended autocorrelation method for estimation of blood velocity," *ieee transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 44, no. 6, pp. 1332–1342, 1997.

[15] D. Calvetti, S. Morigi, L. Reichel, and F. Sgallari, "Tikhonov regularization and the l-curve for large discrete ill-posed problems," *Journal of computational and applied mathematics*, vol. 123, no. 1, pp. 423–446, 2000.

[16] M. J. Grote and T. Huckle, "Parallel preconditioning with sparse approximate inverses," *SIAM Journal on Scientific Computing*, vol. 18, no. 3, pp. 838–853, 1997.

[17] J. Van Cauwenberge, L. Lovstakken, S. Fadnes, A. Rodriguez-Morales, J. Vierendeels, P. Segers, and A. Swillens, "Assessing the performance of ultrafast vector flow imaging in the neonatal heart via multiphysics modeling and in vitro experiments," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 63, no. 11, pp. 1772–1785, 2016.

[18] T. Varslot and S.-E. Måsøy, "Forward propagation of acoustic pressure pulses in 3d soft biological tissue," *Modeling, Identification and Control*, vol. 27, no. 3, pp. 181–200, 2006.

[19] T. Hergum, S. Langeland, E. W. Remme, and H. Torp, "Fast ultrasound imaging simulation in k-space," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 56, no. 6, pp. 1159–1167, 2009.