A fast 4D B-spline framework for model-based reconstruction and regularization in vector flow imaging

Thomas Grønli*†, Morten Wigen*†, Patrick Segers‡ and Lasse Lovstakken*†
*Department of Circulation and Medical Imaging, Norwegian University of Science and Technology, Trondheim, Norway
†Centre for Innovative Ultrasound Solutions, Norway
‡IBiTech-bioMMeda, Ghent University, Ghent, Belgium

Email: thomas.gronli@ntnu.no

***Abstract*—A generic framework for model-based regularization and reconstruction is described, with applications in a wide range of noisy measurement scenarios. The framework employs automatic differentiation and stochastic gradient optimizers to perform online measurement fitting and regularization, and was implemented as a scalable CPU and GPU library with high-performance operation even in compute- or memory-intensive contexts, such as for 4D cardiac vector flow imaging. The framework was demonstrated by reconstructing 4D vector flow mapping through the incorporation of the incompressible Navier-Stokes equations. Furthermore, the achieved performance was within bedside applicability requirements.**

I. Introduction

Ultrasound vector flow imaging (VFI) has been demonstrated using different approaches such as vector-Doppler, blood speckle tracking (BST), or by imposing tissue boundary conditions and mass conservation to estimate the blood velocity vector from color-Doppler images, a method known as vector flow mapping (VFM) [1]. The main challenge is to estimate the lateral velocity component, which for transthoracic cardiac ultrasound implies a small transducer footprint and a limited lateral resolution. Further, to increase the temporal resolution, we would like to take advantage of broad ultrasound beam emissions to enable a high degree of parallel beamforming. In result, the signal-to-noise suffers due to the loss of transmit focusing, and the measurement variance increases. Finally, to be able to extract the blood signal we perform tissue rejection by high-pass filtering the received temporal data sequences, suppressing the more echoic and stationary tissue signal. This filtering will currently also suppress parts of the blood signal depending on its direction and velocity, which leads to measurement dropouts in the images where the flow is near perpendicular to the beam angle.

Current state-of-the-art 2D methods employing model-based regularization and dropout reconstruction suffer under the assumption of zero through-plane flow gradients and scale poorly when moving to 3D/4D imaging due to memory constraints when solving matrix formulations of the problem. Another limitation with this approach is the difficulty to incorporate nonlinear models. The problem of model-based reconstruction and regularization has roots in data assimilation and nonlinear constrained optimization theory. In a noisy measurement scenario, many sources of conflicting information may be available, ranging from noisy measurements (sensors), idealized system models (PDEs) and smoothness constraints on the solution. In this context, we want a robust method that allows arbitrary information sources to be interpolated onto a solution variable, while minimizing a global energy term defined through data fitting terms with added physical constraints.

With this project we aim to solve 4D measurement regularization and reconstruction challenges by adopting optimization techniques used in machine learning, and by building upon the massively scalable technology that powers this field. The goal is to provide a general framework that allows rapid and symbolic prototyping and deployment of fast model-based regularization even on consumer-grade hardware. By adhering to the energy optimization formulation when designing the framework, the generic nature of the method is retained and applications to various regularization problems can be imagined and explored.

II. Method

A. Components

1) Tensor product splines: The linear combination of 1D B-splines [2] is generalized to a scalar spline field of dimension D by

$$\begin{aligned}\mathcal{I}_{\mathbf{C}}(\mathbf{x}) = &\sum_{i_1}\sum_{i_2}\cdots\sum_{i_D} c_{i_1 i_2 \cdots i_D} \times \\ &\frac{\partial^{n_1}}{\partial x_1^{n_1}} B^{\mathbf{t}_1}_{i_1,p_1}(x^1) \frac{\partial^{n_2}}{\partial x_2^{n_2}} B^{\mathbf{t}_2}_{i_2,p_2}(x^2) \cdots \\ &\cdots \frac{\partial^{n_D}}{\partial x_D^{n_D}} B^{\mathbf{t}_D}_{i_D,p_D}(x^D), \end{aligned} \tag{1}$$

where $\mathbf{C}$ is the coefficient tensor for the grid of B-spline functions positioned at the knot vectors $\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_D)$, $\mathbf{p} = (p_1, p_2, \cdots, p_D)$ are the spline orders in each dimension and $\mathcal{I}$ denotes the combined operator for interpolation and differentiation to orders $\mathbf{n} = (n_1, n_2, ..., n_D)$ on the grid $\mathbf{C}$. From the definition of B-splines, (1) requires the computation of the contribution from the $\prod_{d=1}^{D}(p_d + 1)$ coefficients closest to the query position $\mathbf{x}$.

2) Automatic Differentiation: Reverse mode automatic differentiation (AD) is a method for computing the gradient of a function composition $\mathcal{J} = (f_k \circ f_{k-1} \circ \cdots \circ f_1)(\mathbf{C})$ by traversing the chain rule starting with the innermost partial derivatives,

$$\frac{\partial \mathcal{J}}{\partial \mathbf{C}} = \left(\left(\cdots\left(\left(\frac{\partial f_k}{\partial f_{k-1}}\frac{\partial f_{k-1}}{\partial f_{k-2}}\right)\frac{\partial f_{k-2}}{\partial f_{k-3}}\right)\cdots\right)\frac{\partial f_2}{\partial f_1}\right)\frac{\partial f_1}{\partial \mathbf{C}}. \tag{2}$$

Differentiating (1) with respect to the coefficient tensor $\mathbf{C}$ causes the gradient to be given simply by the elementwise support at the query point provided by each function on the grid. Since the gradient is nonzero only for the limited number of coefficients whose function provides support at $\mathbf{x}$, the gradient is well suited for a sparse representation. Registering the gradient $\frac{\partial \mathcal{I}}{\partial \mathbf{C}}$ in an AD system enables the computation of gradients in nonlinear function compositions that depend on interpolated and differentiated splines in their graph.

3) Stochastic Gradient Descent: Stochastic Gradient Descent (SGD) is an iterative procedure for minimizing an objective function whose gradient has a known analytical form. The term stochastic refers to the use of a limited set of randomly drawn samples to obtain an approximated function gradient at each iteration such that the approximation stochastically drives convergence towards a minimum of the system.

Coupling a SGD optimizer to an AD system allows the optimizer to obtain the analytical gradients of any differentiable cost function through the AD system at each iteration and update the current state according to the optimizer heuristics.

B. Implementation

A few elements had particular focus in the design of the framework

- **Easy**
  The framework should be easy to use even for users unfamiliar with the implementation details
- **Flexible**
  The framework should be flexible enough to handle a range of problems without modification
- **Scalable**
  The framework should not have any limitation on the problem size given enough computation time
- **Fast**
  The framework should be fast enough to use for moderately sized problems on common hardware
- **Deployable**
  The framework should be available for most common platforms

The open source TensorFlow machine learning framework was chosen as suitable a platform [3]. TensorFlow provides a major part of the required infrastructure for the method with an expressive symbolic system for constructing computational graphs with full AD functionality and a set of powerful adaptive optimizers that are designed for interoperation with the graphs.

In addition, an interpolator was required for arbitrary measurement support. A large part of the effort in this project went into the development and testing of a spline interpolation operator along with gradient computations for all inputs. The resulting interpolation operator was made compatible with TensorFlow and its AD system for seamless integration with other elements of the energy term formulation. The developed interpolator provided support for tensor product splines with arbitrary spline order, unlimited directional differentiation and optional periodicity in each grid dimension independently. Cross-platform CPU and GPU codes were written for both forward and backward operations to improve performance and enable use on multiple compute system configurations.

C. VFM application

For the VFM approach, a single vector component and the boundaries of the fluid domain are required. We targeted a duplex mode ultrasound acquisition scheme which simultaneously records color-Doppler and B-mode images. In this setup, radial blood velocities are obtained from the averaged autocorrelation at each beamformed voxel from the color-Doppler after high-pass filtering to remove clutter from the tissue. Boundary conditions are obtained from automatic or manual segmentation of the endocardium layer from the anatomical B-mode sequence.

1) Analysis scheme: Given a spline grid with 4D time-periodic spatiotemporal 3-channel coefficient state $\mathbf{C}$, we define the operator $\mathcal{I}$ in the analysis as the vector function interpolating the channels (V_x, V_y, V_z) at observation locations $\mathbf{x}$, which represent the 4-vector (time-inclusive) position. In the VFM approach only the radial velocity component V_z is measured and inference of the states V_x and V_y is achieved through a coupling of the scalar grids and their derivatives using a set of constraints, specifically free slip conditions at the boundaries, incompressibility and momentum conservation in the fluid.

Descriptions of the energy terms employed in the VFM reconstruction problem are presented followingly.

- **Data fit**
  The data fit requires that the radial component of the interpolation at each measurement location $\mathbf{x}$ agrees with the corresponding Doppler estimate v_D.

  $$\mathcal{I}(\mathbf{x}, \mathbf{C}) \cdot \hat{\mathbf{n}}_D = v_D, \tag{3}$$

  where $\hat{\mathbf{n}}_D$ is the ultrasound beam angle at the measurement location.
- **Boundary condition**
  In the reconstruction problem we include free slip boundary conditions. Given a boundary segment $\partial\Omega$ positioned at $\mathbf{x}_{\partial\Omega}$ and its corresponding normal vector $\hat{\mathbf{n}}_\perp$, we impose a tangential velocity at this surface by requiring

  $$\mathcal{I}(\mathbf{x}_{\partial\Omega}, \mathbf{C}) \cdot \hat{\mathbf{n}}_\perp = v_\perp, \tag{4}$$

  where $v_\perp$ is the normal velocity of the boundary segment. Note that the boundary condition is identical to the Doppler fitting term in that a scalar velocity is imposed along a specified direction in the fluid.

- **Mass continuity**
  To model an incompressible fluid, we impose the incompressible continuity equation at each observation location $\mathbf{x}$ inside the fluid

$$\nabla \cdot \mathcal{I}(\mathbf{x}, \mathbf{C}) = 0. \tag{5}$$

- **Momentum balance**
  We impose momentum balance in the fluid according to the incompressible Navier-Stokes convection-diffusion equation

$$\frac{\partial \mathcal{I}(\mathbf{x}, \mathbf{C})}{\partial t} + \nabla \cdot (\mathcal{I}(\mathbf{x}, \mathbf{C}) \otimes \mathcal{I}(\mathbf{x}, \mathbf{C})) = \nu \nabla^2 \mathcal{I}(\mathbf{x}, \mathbf{C}), \tag{6}$$

  where ν is the dynamical viscosity parameter of the fluid. External forces and internal source terms have been excluded.

2) Optimization procedure: All the necessary operations to calculate the interpolation errors and PDEs described in the analysis scheme were expressed in terms of the interpolation kernel $\mathcal{I}(\mathbf{x}, \mathbf{C})$ in the symbolic framework provided by TensorFlow. The total sum of all deviations was constructed as a computational graph and minimized with the Adam [4] optimizer using minibatches with N = 10000 samples randomly drawn with replacement from the set of measurements until convergence.

3) In silico validation: A computational fluid dynamics (CFD) phantom [5] of a neonatal left ventricle created with the Ansys Fluent CFD package was used as a reference for the flow reconstruction. The phantom captures the main ventricular blood flow events such as filling, vortex characteristics and ejection.

4) In vivo feasibility: In vivo 4D data was acquired using a GE Vivid E95 system with a 4V phased array probe on a healthy adult volunteer, where 3D hybrid BST provided dealiased Doppler velocity radially over the entire temporal data sequence. The left ventricular (LV) domain was extracted automatically using the open source FAST framework [6].

III. Results

A. Kernel performance

The framework performance was evaluated on two heterogeneous compute systems

- Workstation [4x Xeon E5, TITAN V]
- Mobile workstation [1x Core i7, Quadro M2200]

Figure 1 shows the kernel performance on the different compute units. On average, the 4D reconstruction converged after 3 minutes with tuned gradient stepping parameters on the mobile workstation.

B. In silico analysis

Figure 2 shows the distribution in of the reconstructed velocity error along with comparison to the reference velocities. The scatter diagrams point to some underestimation in both angular directions, but stronger in elevation. From the spatial distribution we observe that the majority of the error is located near the base of the ventricle close to the inlets and outlets. Overall, root-mean-square error (RMSE) in the reconstructed velocities was 11.3% normalized at the 99.9th angular magnitude percentile (3.1 cm/s).

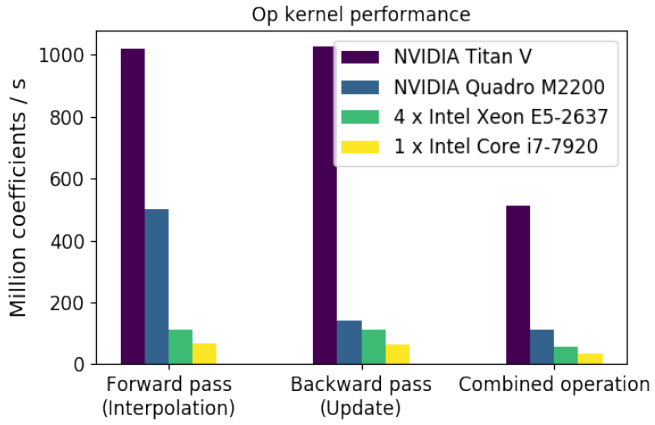

Fig. 1. Performance of the interpolation kernel and gradients across different compute platforms. The number of coefficients involved in each interpolation and update depends on the spline order.

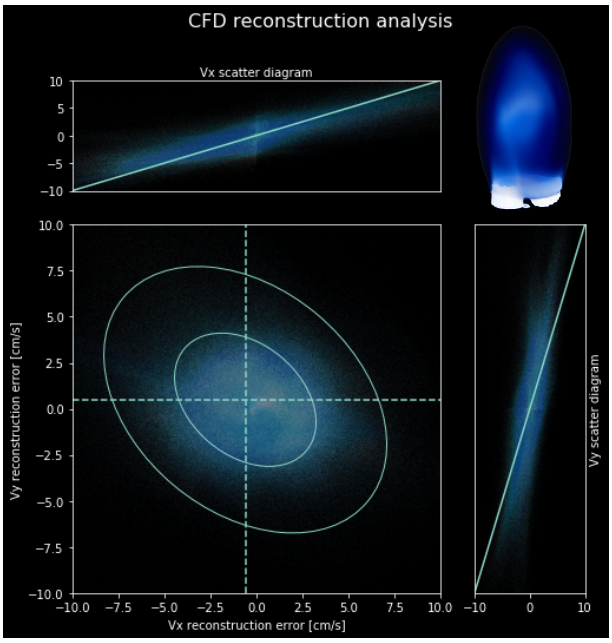

Fig. 2. The figure summarizes the component-wise reconstruction error in the two angular directions viewed in the ALAX-orientation. Scatter diagrams comparing the reconstructed velocities to the phantom are drawn along their respective axes with the (long) abcissæ representing the reference velocity. The relative distribution of the error inside the reconstructed field is indicated in the top right corner of the figure.

1) In vivo demonstration: Figure 3 shows the framework applied to a 4D dataset, resulting in physically convincing reconstructions with the radial velocity magnitude being preserved after reconstruction of the angular velocities. The boundary mesh was sliced at the base to avoid spurious wall bouncing in this area.

IV. Discussion

As observed in figure 2, the errors are within tolerable ranges. Most of the error is located near inlet or outlets, and this may be caused by the lack of internal source terms in the Navier-Stokes regularizer to account for accelerative pressure forces during the valve events in the phantom. Due to the global nature of the regularization procedure, this likely impacts the internal field reconstruction also. The framework relies on a combination of measurements and regularization terms and this data driven approach may allow for simplified models. However, the effect of imprecise boundaries or pressure gradients should be investigated further.

The strengths of the framework lie in its ability to rapidly prototype and iterate on different model constraints used in the regularization by the symbolic representation of the problem. The main benefits over a matrix formulation is the ability to perform online training on memory-constrained systems, in addition to avoiding setup steps for the required sparse matrices for the observation and finite difference operators. The exact energy gradient with respect to the solution is obtained by automatic differentiation of the symbolic formulation of the cost function. The performance of the regularization method is sufficient on common hardware to suggest clinical applicability in the future after further validation. Moving forward, the framework should undergo thorough in vivo vector flow validations, e.g. by comparison to phase contrast MRI.

V. Conclusion

A fast regularization framework with an efficient B-spline interpolator at its core was successfully developed and demonstrated for 4D VFM. As volume flow methods become increasingly available, efficient processing libraries are necessary in order to successfully incorporate physical models and the described framework provides a positive preliminary outlook.

References

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

[2] C. De Boor, C. De Boor, C. De Boor, and C. De Boor, *A practical guide to splines*, vol. 27. Springer-Verlag New York, 1978.

[3] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning.," in *OSDI*, vol. 16, pp. 265–283, 2016.

[4] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[5] 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.

[6] E. Smistad, M. Bozorgi, and F. Lindseth, "Fast: framework for heterogeneous medical image computing and visualization," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 11, pp. 1811–1822, 2015.

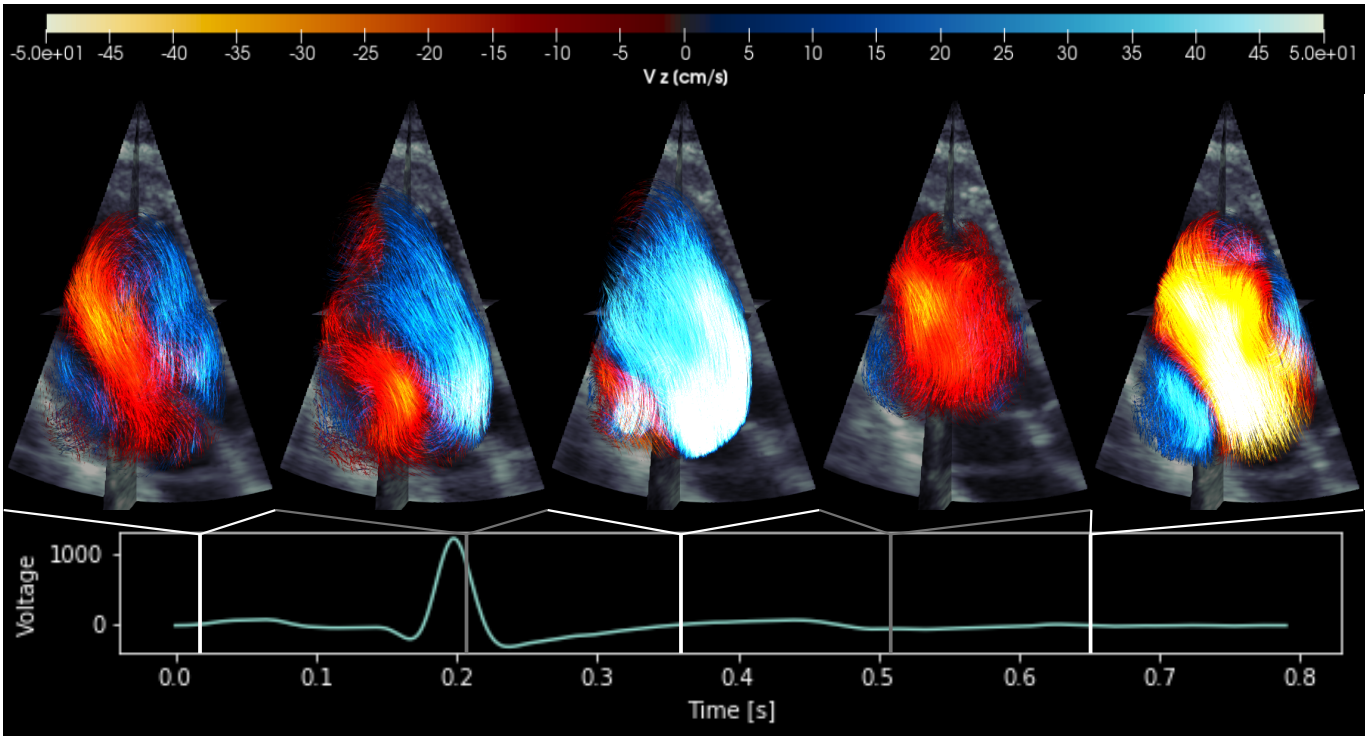

Fig. 3. The figure shows left ventricular vector flow reconstruction in a healthy adult volunteer at five different points during the cardiac cycle. Dealiased Doppler measurements were used in the reconstruction and the streamlines are colored according to the estimated velocity.