Computer Methods and Programs in Biomedicine 238 (2023) 107604

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

FLUST: A fast, open source framework for ultrasound blood flow simulations

Ingvild Kinn Ekroll[a,*], Anne E.C.M. Saris[c], Jørgen Avdal[a,b]

[a] CIUS and the Department of Circulation and Medical Imaging, Norwegian University of Science and Technology, Norway
[b] Department of Health Research, SINTEF Digital, Norway
[c] Medical Ultrasound Imaging Center, Department of Medical Imaging, Radboud University Medical Center, Nijmegen, The Netherlands

ARTICLE INFO

Article history:
Received 27 February 2023
Revised 8 May 2023
Accepted 14 May 2023

Keywords:
Ultrasound imaging
Blood flow
Doppler ultrasound
Velocity estimation
Simulation tools
Open source

ABSTRACT

Background and objective: Ultrasound based blood velocity estimation is a continuously developing frontier, where the vast number of possible acquisition setups and velocity estimators makes it challenging to assess which combination is better suited for a given imaging application. FLUST, the Flow-Line based Ultrasound Simulation Tool, may be used to address this challenge, providing a common platform for evaluation of velocity estimation schemes on *in silico* data. However, the FLUST approach had some limitations in its original form, including reduced robustness for phase sensitive setups and the need for manual selection of integrity parameters. In addition, implementation of the technique and therefore also documentation of signal integrity was left to potential users of the approach.

Methods: In this work, several improvements to the FLUST technique are proposed and investigated, and a robust, open source simulation framework developed. The software supports several transducer types and acquisition setups, in addition to a range of different flow phantoms. The main goal of this work is to offer a robust, computationally cheap and user-friendly framework to simulate ultrasound data from stationary blood velocity fields and thereby facilitate design and evaluation of estimation schemes, including acquisition design, velocity estimation and other post-processing steps.

Results: The technical improvements proposed in this work resulted in reduced interpolation errors, reduced variability in signal power, and also automatic selection of spatial and temporal discretization parameters. Results are presented illustrating the challenges and the effectiveness of the solutions. The integrity of the improved simulation framework is validated in an extensive study, with results indicating that speckle statistics, spatial and temporal correlation and frequency content all correspond well with theoretical predictions. Finally, an illustrative example shows how FLUST may be used throughout the design and optimization process of a velocity estimator.

Conclusions: The FLUST framework is available as a part of the UltraSound ToolBox (USTB), and the results in this paper demonstrate that it can be used as an efficient and reliable tool for the development and validation of ultrasound-based velocity estimation schemes.

1. Introduction

Within the medical ultrasound imaging community, ultrasound simulations are widely used during the development and validation phase of new technology, ranging from innovative ultrasound transmission sequences and new transducer designs to blood flow imaging and velocity measurements. With the use of simulated data, effects of different design choices can easily be studied. In addition to tools for simulation of ultrasound pressure fields [1,2], several ultrasound imaging simulators are available [3–8], although not necessarily open-source, each with their own specifics, underlying theory and assumptions.

Ultrasound-based blood flow imaging is one of the most important and often-used modes on a clinical ultrasound scanner. Due to its importance, the medical ultrasound community is continuously searching for new and improved ways to detect and quantify blood flow, resulting in a vast number of proposed velocity estimators. Many velocity estimation techniques are based either on the autocorrelation [9] or the cross-correlation [10] approach, but tai-

* Corresponding author.
E-mail address: ingvild.k.ekroll@ntnu.no (I.K. Ekroll).

https://doi.org/10.1016/j.cmpb.2023.107604

lored to suit a given application [11–17]. Others could be based on a combination of the two approaches [18–20] or on Fourier filtering techniques [21–23].

The performance of any velocity estimator may depend on many factors, such as estimator design, imaging application, flow complexity, probe design, acquisition setup and beamforming parameters. In addition, estimators are often tuned to a specific application of interest and performance assessment done based on data sets that are not publicly available. Consequently, there are a vast number of publications describing an increasing number of velocity estimators, but no established methodology for comparing estimator performances between publications. This makes it challenging to evaluate which imaging acquisition and velocity estimator to use in a new application, and leads to repeated investigations of similar problems.

The lack of common data sets was one of the driving forces for imaging challenges such as PICMUS [24] in 2016 and SAVFI [25] in 2018, in which simulated, phantom and *in vivo* data sets were made openly available, and standardized performance metrics were used to evaluate the contributions. In these challenges, a fair comparison of the different approaches was possible. However, the results and conclusions were limited to the specific data sets and acquisition setups used, and can therefore not be used as foundation for a comprehensive comparison of all new or existing velocity estimation schemes.

To investigate and quantify the performance of an estimator, often the behaviour over several estimates is investigated. The bias and variance of the estimator are determined, as described in e.g. [25], assuming the estimates are independent. To obtain a conclusive statistical assessment of estimator performance, a sufficient number of estimates should be taken into account. A single estimate is typically based upon a number of emitted pulses, called an ensemble, to reduce the variance caused by the underlying speckle variations. In addition, to get a measure of the true bias and variance of an estimator, many ensembles from a stationary flow field are required. This means that when using *in silico* flow models and ultrasound simulations to assess these properties, several realizations of the same flow field must be generated to yield conclusive results.

With the gradual increase in computational power, there has been a trend towards using increasingly complex flow models when investigating new or existing ultrasound blood velocity estimators. Through coupling with computational fluid dynamics (CFD) models, simulations can be performed yielding realistic ultrasound images and Doppler signals from complex blood flow [26]. An advantage of these sophisticated *in silico* models is that they provide a good representation of realistic *in vivo* flow fields, and can therefore give a good indication of overall performance of an estimator. However, it can be challenging to decompose the results into correct sources of error contributing to the final performance. It is also difficult to predict the effect of slight changes to the flow velocities or patterns, such as different velocity gradients, changes in geometry and out-of-plane motion. Another disadvantage is that running realistic flow simulations can be time-consuming. Consequently, results are often based on single, or a small number of flow field realizations, yielding performance indicators that are strongly affected by variance.

Aiming to reduce runtime while preserving integrity, the flow line-based simulation approach was introduced in a previous work [27] by Avdal *et al*. The FLUST technique enables multiple realizations of signals from both simple and complex flow models at low computational cost, facilitating the calculation of statistical expectation values and the variance of a velocity estimator at any point in the *in silico* flow field.

In this work we describe the advancement of the theoretical foundation and initial implementation of FLUST towards an easy-to-use, robust simulation software package available as open-source code for the wider community. Important implementation details are described to ensure users are well informed about the software design, to allow them to review design choices and facilitate adaptation of the open-source code by advanced users. Furthermore, we illustrate FLUST's application in the development of novel ultrasound blood flow imaging schemes and estimators, where, not different from any other biomedical research, extensive optimization and validation is an essential step prior to clinical translation of developed technologies. The FLUST package is accompanied by examples describing a range of transducer types, acquisitions and flow phantoms. This will support the user in setting up and running ultrasound simulations of both simple and complex flow scenarios, avoiding duplication of effort.

One limitation of the initial approach was the need for careful selection of spatial and temporal time steps in the simulation chain to avoid significant interpolation errors and speckle artefacts in the resulting simulated data. This was particularly challenging for phase sensitive acquisition setups. Another factor limiting the use of the FLUST technique was the lack of features ensuring signal integrity, which in the original implementation depended on manually selected parameters. Without such features, verification of the integrity of the simulated data would be left to each user prior to any further use of FLUST.

In this work, these shortcomings are addressed. First, we propose an additional phase modulation step in the simulation chain to significantly reduce the impact of interpolation errors. This facilitates a significant reduction in FLUST simulation time without compromising the integrity of the results. Second, we propose methods for automatic selection of spatial and temporal discretization steps to assure the integrity of the FLUST simulator and resulting Doppler signals, features that also improve the user-friendliness of the framework. Third, we propose a weighting approach to reduce the variability of signal power between different flow lines. Thereafter, the impact of the proposed improvements on FLUST simulations is assessed, and the integrity of the resulting simulated data is quantified. With the improvements proposed in this work, the FLUST framework may contribute to accelerate development and assessment of novel velocity estimators and facilitate the process of comparing performance of estimators, even from different acquisition schemes.

Section 2 starts with a description of the FLUST framework and a reiteration of the FLUST simulation technique and its motivation, before proposing a series of important features in the FLUST framework that ensures user independent integrity of the simulated IQ Doppler signals. Section 3 motivates and describes the simulation setups and validation techniques used in this work. In Section 4 simulation results are presented that illustrate the underlying issues of the original approach, the effectiveness of the proposed solutions, and an extensive integrity study of the Doppler signals produced by the FLUST framework. Furthermore, this section ends with two illustrative examples of FLUST, showing its potential in the design process of velocity estimators. For the second example, a runtime comparison is made with the often used FIELD II software. Results are discussed in Section 5 and Section 6 concludes the paper.

2. FLUST

2.1. The FLUST simulation framework

The purpose and novelty of the FLUST framework is its ability to produce multiple realizations of high integrity IQ Doppler signals from simple or complex flow fields at low computational cost. It utilizes other sources such as Field II [3,4], SIMUS [5], or water tank measurements for generation of point spread functions (PSFs),

Table 1
The steps of the FLUST method .

Steps	Parameters	Output
1: PSF calculation	Acquisition setup, flow geometry, PSF density	s_F^r : PSFs on regular spatial grid along flow line
2: Interpolation of PSFs	Doppler PRF, flow velocity, oversampling factor	s_F^t : PSFs on regular temporal grid along flow line
3: Temporal convolution	Number of realizations, ensemble size	s_F : Realization for one flow line
4: Repeat for all flow lines and sum	Flow line weighting	s : Realization for all flow lines

which are ultrasound radiofrequency (RF) signals from single scatterers. FLUST is distributed as a part of the UltraSound ToolBox (USTB), which is an open source MATLAB toolbox for processing of ultrasonic signals [28]. Transparency and reproducibility of results are key elements in both the FLUST and USTB projects, and the code (found at http://www.ustb.no/flust) includes customizable acquisition setups and flow phantoms, as well as tools for validation of (Doppler) signal integrity, estimator performance, and visualization. The repository includes examples of simulation setups with a range of *in silico* flow phantoms, various arrays and transmission/acquisition sequences. The setups can be adapted by the user for a given purpose, and may be used both for research and teaching purposes.

2.2. The inner workings of FLUST

FLUST can produce multiple realizations of the received ultrasound signal from a stationary flow field. This is achieved by decomposing the flow field into flow lines instead of individual point scatterers. For each flow line F, scatterers are seeded at equidistant points with distance Δr at which PSFs are calculated (Step 1). Let the IQ demodulated PSFs be denoted by $s_F^r(d_F, \mathbf{r})$, with $\mathbf{r}$ denoting pixel position and d_F denoting the propagated distance of the scatterer along flowline F. The pre-calculated PSFs are then interpolated to a regular temporal grid with sampling interval Δt, resulting in the sampled signal s_F^t from a single scatterer moving along F (Step 2)

$$s_F^t(k, \mathbf{r}) = I\{s_F^r\}(d_F(k\Delta t), \mathbf{r}),\ k \in \mathbb{Z}. \tag{1}$$

Here $d_F(t)$ denotes the position of the scatterer at time t and $I\{s_F^r\}$ denotes the interpolant agreeing with s_F^r at equidistant scatterer positions along F. The index k denotes the temporal sample index within an ensemble, corresponding to successive ultrasound transmissions in the same direction. In step 3, the signal s_F from multiple scatterers moving along F is produced by applying a temporal filter to s_F^t,

$$s_F = n *_t s_F^t, \tag{2}$$

where the convolution is applied along the temporal dimension. Each entry of n is a real valued random variable with Gaussian distribution, representing the scattering strength of single scatterer entering the flow line at that time point.

The signal s_F is then one realization of the spatiotemporal ultrasound signal, also called the Doppler ensemble, from flow line F. In step 4, steps 1–3 are repeated for all flow lines and the resulting signals are summed to produce a full flow field realization s:

$$s = \sum_{F \in \Omega} w_F s_F \tag{3}$$

where w_F is a weighting function used to ensure incompressibility and conservation of mass, as described in Section 2.6. The number of flow lines in Ω is determined by the size of the flow phantom and the spacing between flow lines, set to $\lambda/2$ at the flow phantom inlet. Generating multiple ensemble realizations of the same flow field is computationally cheap as it only requires repeating steps 3 and 4 with different temporal filters n. The FLUST steps are illustrated in Fig. 1 and summarized in Table 1. A more elaborate description of the underlying theory can be found in the work by Avdal *et al.* [27].

2.3. The need for multiple realizations and high integrity signals

The output of FLUST is multiple realizations of the IQ Doppler signal s at all image points, from a known flow field. By applying a velocity estimator to these realizations, it is possible to quantify the performance of the estimator in terms of bias and variance. Increasing the number of realizations improves the integrity of the performance evaluation. More precisely, applying a velocity estimator to a pixel or region in realization i yields a velocity estimate v_i that is a stochastic variable following a probability distribution $\mathcal{P}$

$$v_i \sim \mathcal{P}(\mu, \sigma^2). \tag{4}$$

Here μ and σ^2 denote the mean and variance of $\mathcal{P}$. Note that the estimator v_i may have a non-zero bias, in which case μ would differ from the true velocity. Applying the velocity estimator to all N realizations and averaging the result yields the new estimator

$$v^N = \frac{1}{N}\sum_{i=1}^{N} v_i. \tag{5}$$

As N increases, the probability distribution corresponding to v^N will converge towards a normal distribution with the same mean as v_i, but with variance reduced by a factor of N.

$$v^N \sim \mathcal{N}(\mu, \sigma^2/N). \tag{6}$$

Consequently, the estimate v^N yields a better approximation of the expectation value μ than those from individual realizations v_i, and improves with increasing number of realizations N. A similar analysis can be used to show that estimation of the variance σ^2 also improves with increasing N. Because the FLUST simulator is able to produce multiple realizations of the same flow field efficiently, it becomes easier to obtain estimates of μ and σ that themselves have low bias and variance.

In addition to a large number of realizations, an important prerequisite for high integrity evaluation of estimators is that each individual IQ signal itself has high integrity. The original FLUST method [27] had multiple user-defined parameters influencing the signal integrity. To enable a more widespread use of FLUST, steps have been taken to automate determination of parameters that are essential to the integrity of the simulated data. In the following sections, we describe the most important sources of error and how they are handled by the FLUST framework.

2.4. Interpolation errors and proposed FLUST solutions

As illustrated in Fig. 1, FLUST uses interpolation of PSFs calculated on a regular spatial grid with sampling distance Δr along a flow line F, to produce the signal from a single scatterer moving along F on a regular temporal grid with interval Δt. Significant interpolation errors may arise if the spacing Δr between the precalculated PSFs along flow lines is too large, which again may cause errors in the magnitude and phase information in the resulting simulated IQ signals. Two steps have been taken to limit the effect of interpolation errors.

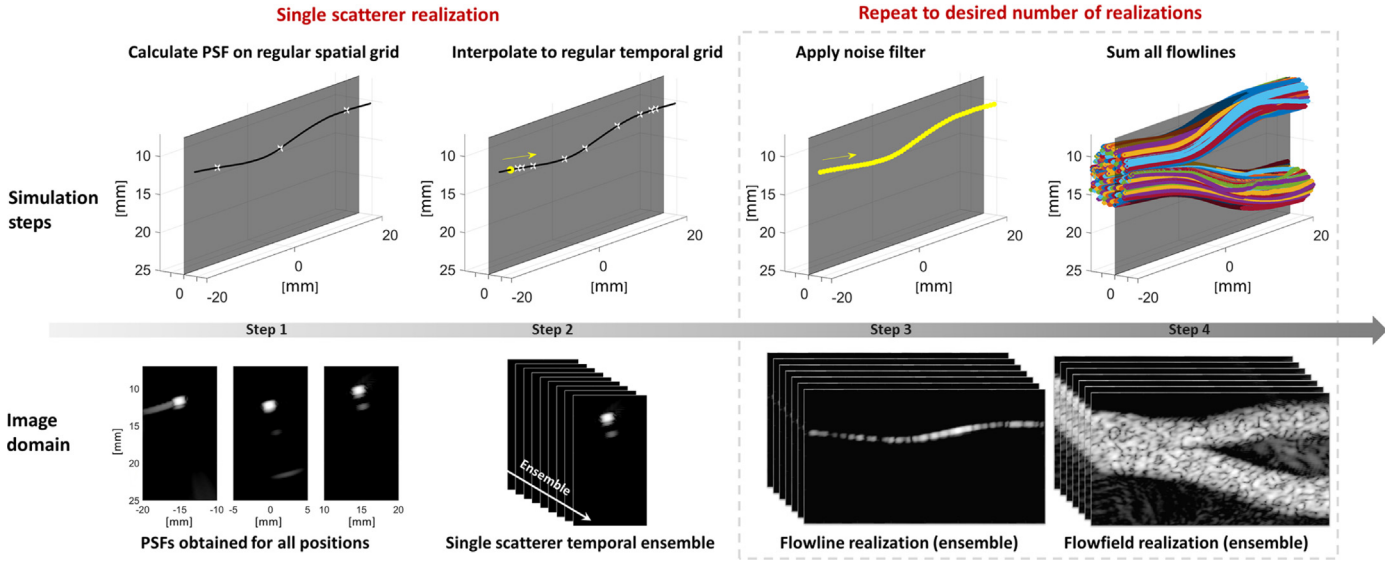

Fig. 1. Step-wise illustration of FLUST: For a given flow line, PSFs are calculated for scatterer positions (using e.g. Field II[4], SIMUS[5], or even water tank measurements) distributed along the flow line with spacing Δr. The signal from a single scatterer moving along the flow line is obtained by interpolation to a temporal grid with interval Δt. In the next step, a temporal filter is applied to obtain the signal from multiple scatterers moving along the flow line, yielding one realization of a single flow line. To produce a full flow field realization, the signal from all flow lines are summed. By repeating the two last steps, the desired number of ensemble realizations can be obtained at a low computational cost. The first two steps are illustrated with low spatial and temporal density for visualization purposes.

2.4.1. Reducing phase interpolation errors

The phase of the main lobe of a PSF is dependent on the transmit and receive directions and the spatial position of the point scatterer. This phase is typically used in Doppler imaging for velocity estimation, and may be written as

$$\phi(\mathbf{r}, \hat{\mathbf{u}}_{\mathbf{tx}}, \hat{\mathbf{u}}_{\mathbf{rx}}) = \frac{f_0}{c}\,\mathbf{r}\cdot(\hat{\mathbf{u}}_{\mathbf{tx}} + \hat{\mathbf{u}}_{\mathbf{rx}}). \tag{7}$$

Here f_0 is the pulse center frequency, c is the speed of sound, $\mathbf{r}$ is the position of the scatterer, and $\mathbf{u}_{tx}$ and $\mathbf{u}_{rx}$ are unit vectors in the transmit and receive direction, respectively.

Interpolation errors due to rapid phase variations may significantly reduce the integrity of the Doppler signals. To minimize this effect, we suggest multiplying the PSFs with a corresponding phase factor $e^{-i\phi}$ before the interpolation in step 2, and reapplying the phase to the PSFs after interpolation. The expression in (1) is thus modified to

$$s_F^t(k, \mathbf{r}) = I\{s_F^r e^{-i\phi}\}(d_F(k\Delta t), \mathbf{r})\, e^{i\phi(\mathbf{r})},\ k \in \mathbb{Z}. \tag{8}$$

This approach will result in a slower phase variation of the PSFs as a function of position along the flow line. As a consequence, for a predefined acceptable interpolation error value, the required PSF spacing Δr may be significantly increased.

2.4.2. Reducing spatial interpolation errors

To reduce potential interpolation errors further, we have introduced calculation of a recommended distance Δr between PSF positions along flow lines prior to step 1. To achieve this, PSFs are calculated for P positions spaced 10 μm apart along straight lines in the axial and lateral directions, both centered around the centroid $\mathbf{c}_0$ of the flow field of interest. The PSFs are evaluated at $\mathbf{c}_0$, producing reference temporal, or *slow-time*, signals s_1 and s_2 corresponding to scatterers moving in the axial and lateral directions, sampled at spatial intervals of 10 μm. A series of interpolation functions are then defined, corresponding to linear interpolation with different sampling rates. Linear interpolation with an undersampling rate of k may be described as filtering a zero-filled undersampled signal using a triangle function h_k as kernel, with

$$h_k(n) = \frac{k - |n|}{k},\ \forall n \in [-k, k]. \tag{9}$$

The interpolation errors for the different undersampling rates may now be estimated as the energy loss associated with the filtering operation

$$E_i(k) = 1 - \frac{\|h_k * s_i\|}{\|s_i\|},\ i = 1, 2, \tag{10}$$

where $i = 1, 2$ corresponds to the axial and lateral dimensions, respectively. The recommended PSF spacing is determined by finding the largest undersampling factor k yielding interpolation errors $E_1(k)$ and $E_2(k)$ that are both smaller than a predefined threshold, for instance 4%.

2.5. Temporal sampling and proposed FLUST solution

After the second step of FLUST method, the resulting signal consists of PSF functions along a flow line on a regular temporal grid. Given that the transit time of the scatterer through a sample volume is large compared to the temporal step size Δt, the signal s_F^t from the scatterer moving along the flow line will be sufficiently sampled. However, if the transit time is low compared to Δt, i.e. when a scatterer passes through the sample volume in a very low number of slow-time samples, the signal will be undersampled. Consequently, the resulting realization s_F after step three will not sufficiently represent the phase variation in signal components resulting from multiple scatterers entering the sample volume at slightly different time points.

As a solution to this problem, we propose that the temporal grid resolution Δt_F for flow line F should be selected such that any scatterer should never be displaced by more than the spatial discretization step Δr between consecutive temporal samples.

$$\Delta t_F < \frac{\Delta r}{v_F^m}, \tag{11}$$

where v_F^m is the highest scatterer velocity along flow line F. This requirement ensures that the phase variation between temporal samples is limited.

As a practical choice, it is proposed that Δt_F should yield a slow-time sampling rate that is the smallest integer multiple of the desired firing rate of the simulation satisfying the above con-

dition. Thereafter, the realizations may be produced by decimating the simulated signal, avoiding further interpolation errors.

2.6. Flow line weighting and proposed FLUST solution

In the third step of the FLUST method, realizations of the signal from a single flow line are generated using convolution with a temporal filter. As discussed in 2.2, each entry of n represents the backscattered signal magnitude from a scatterer entering the flow line at that time point.

When using FLUST, one scatterer enters the flow line every time step Δt, and the time each scatterer spends in a sample volume is inversely proportional to the scatterer velocity v. It follows that the scatterer density ρ from a sample volume satisfies

$$\rho \propto \frac{1}{v\Delta t}. \tag{12}$$

It is assumed that the mean power in an image pixel is proportional to the number of scatterers that are simultaneously within the corresponding sample volume [29,30]. Without any weighting, a low velocity flow line will thus tend to have a higher signal strength than a high velocity flow line. However, blood is assumed incompressible, and therefore the average power of each flow line should be the same. A weighting function for flow line F, compensating for the scatterer density in (12), is

$$w_F = v_F \Delta A_F \Delta t, \tag{13}$$

where v_F is the velocity at the inlet of flow line F and ΔA_F is the area of the corresponding cross-section represented by F at this inlet. If the flow lines are uniformly spaced at the inlet, the factor ΔA_F may be skipped. The weights w_F are applied when summing the signals from individual flow lines, as shown in (3).

3. Methods

A series of simulations were performed to illustrate the issues discussed above, show the effectiveness of the proposed solutions, and assess the integrity of the FLUST approach. The different setups and their motivation is described in the following. Unless explicitly stated, FLUST simulations were performed using PSFs generated from Field II. An overview of the parameters used for the different simulation setups is included in Table 2.

3.1. Interpolation errors

Two setups were used to illustrate the effect of interpolation errors. First, a series of single flow line simulations were run, in which the spatial step size (Δr) for PSF scatterer positions along a flow line were 5, 25, 65 and 105 μm. The flow line had a beam-to-flow angle of 60 degrees, passed through the point (x,z) = (0 m, 0.02 m) and had a constant velocity such that point scatterers moved a distance of 5 μm between consecutive slow-time samples. The effect of interpolation errors was then assessed on the PSF signal s_F^t with regular temporal grid. When using a PSF step size of 5 μm, all time points evaluated will coincide with a pre-calculated PSF scatterer position. The signal will then have no interpolation error and can be considered as a reference signal. For the other three step sizes, however, only a subset of time points will coincide with pre-calculated PSF position, and the rest will be generated using interpolation.

The second setup was used to assess the effect of interpolation errors on the flow field realizations s and corresponding slow-time spectra. A simulation was run using a 2-D gradient tube phantom with diameter 6 mm, beam-to-flow angle 60 degrees and velocities ranging from 0.2 m/s to 0.5 m/s.

3.2. Temporal sampling

To visualize clearly the effect and necessity of temporal, or slow-time, oversampling, a setup was designed in which the displacement of point scatterers was large between consecutive slow-time samples. In order to achieve this, a 2-D plug flow tube phantom with diameter of 1 mm, beam-to-flow angle of 60 degrees and velocity of 3 m/s was used. The corresponding acquisition setup used a single plane wave with no steering, pulse center frequency of 5 MHz and a firing rate of 4 kHz. With this setup, the selected step length between calculated PSF functions was 160 μm, whereas the displacement of scatterers between consecutive slow-time samples was 750 μm. Simulations were performed with and without slow-time oversampling, using temporal discretization steps of 50 μ s (temporal grid size according to (11)) and 250 μ s, respectively. To illustrate the effect, 100 ensemble realizations were produced with and without oversampling. Then, for each pixel, lag 1 auto-correlation phase estimates were calculated

Table 2
All simulation setups .

Setup	Interpolation error	Oversampling	Weighting (tube)	Weighting (disk)	Integrity 1 cross-corr	Integrity 1 auto-corr	Integrity 2 (phase)	Example (design)
Number of realizations	1/100	100	100	100	100	100	100	10, 2000
Ensemble size	40	40	40	40	40	40	40	10
Transmit frequency [MHz]	6	5	5	5	5	5	3, 5, 10	6, 12
Number of pulse cycles	1.5	4.5	1.5	1.5	1.5	5.5	5.5	4.5
Element pitch [μm]	250	200	200	200	200	200	200	100, 200
Element height [mm]	5	5	5	5	5	5	5	5
Number of elements	160	192	192	192	192	192	192	128, 256
Transmit angles [degrees]	−15	0	0	0	0	0	−20, −10, 0, 10, 20	-10
Receive angles [degrees]	0	0	0	0	0	0	−20, 0, 0, 0, 20	−10
Receive F-number	0.5	2	0.5	0.5	1	3	3	1
Doppler PRF [kHz]	6	4	12	12	8	8	2	12
Phantom type	Line/Tube	Tube	Tube	Disk	Tube	Tube	Tube	Tube
Flow angle [degrees]	60	60	90	N/A	0	0	0, 30, 60, 90	70
Phantom depth [mm]	20	20	20	20	20	20	20	20
Phantom diameter [mm]	0.1/6	1	6	6	6	6	0.5	2
Flow line spacing [mm]	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1, 0.05
PSF interpolation error limit [%]	N/A	4	4	4	4	4	4	4
Minimum velocity [cm/s]	3/20	300	10	20	50	50	7.5, 15, 22.5, 30, 37.5, 45	5
Maximum velocity [cm/s]	3/50	300	70	100	50	50	7.5, 15, 22.5, 30, 37.5, 45	60

as

$$\angle\tilde{R}(1) = \angle\left(\sum_{n=1}^{N-1} s_t^*(n)s_t(n+1)\right) \tag{14}$$

and mean power Doppler as

$$\overline{P} = \sum_{n=1}^{N-1} |s_t(n)|^2, \tag{15}$$

where s_t denotes the slow-time signal.

3.3. Flow line weighting

As shown in (13), the parameters affecting the expected power level of a flow line include the flow line length, the propagation time for a single scatterer passing through the flow line and the temporal step size. Two different simulations were run to illustrate the necessity of flow line weighting. The first was a tube phantom with gradient flow in which all flow lines have the same length, but the propagation time varies. The second was a spinning disk phantom in which all flow lines have different length, but the propagation time is the same. The velocities of the phantoms were selected such that no temporal oversampling was necessary for the gradient flow phantom, whereas the outermost line in the spinning disk phantom had a temporal oversampling factor of two.

3.4. Signal integrity study 1

A series of simulations were run to verify the integrity of the resulting IQ signals. First, a tube phantom with constant axial velocity of 0.5 m/s and diameter 6 mm was simulated using two different acquisition setups, both using a single plane wave transmission and a pulse center frequency of 5 MHz. The first setup used 1.5 pulse cycles and an F-number of 1, yielding higher resolution more suitable for a cross-correlation based velocity estimator. The second setup used 5.5 pulse cycles and an F-number of 3, which is more suitable for an auto-correlation based estimator. A total of 100 realizations of the flow field were simulated, each containing 40 slow-time samples. Noise was added to the realizations, producing signals with a signal-to-noise ratio of 10 dB.

To assess the integrity of the signal from both setups for use with a cross-correlation based estimator, an M-mode signal s_M was produced by extracting the signal along a straight line in the flow direction for all slow-time samples. The 2-D spatio-temporal autocorrelation of this signal was then calculated as

$$R_{xx}(m,u) = \langle s_M^*(r,n)\, s_M(r+m, n+u)\rangle \tag{16}$$

where $s_M(r,n)$ is the M-mode signal at spatial position r along the straight line and temporal sample n. The signal R_{XX} should be strong when the ratio between the spatial lag m and the temporal lag u is equal to the flow velocity v. Thus, for an appropriately designed acquisition setup, the energy of R_{XX} should be concentrated around a straight line with slope v.

The integrity of the simulated signal from both setups for use with an auto-correlation based velocity estimator was studied by selecting a rectangular kernel K of 21 x 21 pixels, corresponding to 1.6 mm x 1.6 mm, from the middle of the tube. For all pixels in this kernel, the lag 1 auto-correlation phase angle was estimated using (14), and the spectral power was estimated as

$$S_t(\omega) = \left|\sum_{n=1}^{N} s_t(n)e^{-i\omega n\Delta t}\right|^2, \tag{17}$$

where Δt is the slow-time sampling, N is the ensemble size and s_t is the full flow field realization slow-time signals. The spectral power was averaged over all pixels in K and over all realizations. The reference phase angle corresponding to the acquisition setup and the true velocity vector $\mathbf{v}$ is given by

$$R(1) = \frac{\pi}{2v_N}(\hat{\mathbf{u}}_{\mathbf{tx}} + \hat{\mathbf{u}}_{\mathbf{rx}}) \cdot \mathbf{v}, \tag{18}$$

where v_N is the Nyquist velocity

$$v_N = \frac{c\,\mathrm{PRF}}{4f_0}. \tag{19}$$

Here c is the speed of sound, PRF refers to the Doppler PRF, and f_0 is the pulse center frequency.

3.5. Signal integrity study 2

To further verify that the slow-time phase behavior of the simulated signals is correct for different acquisition setups, a series of small tube (0.5 mm diameter) simulations with varying flow velocities, flow directions and pulse center frequencies were generated. The acquisition was a plane wave setup with five transmission directions ranging from -20 to +20 degrees with steps of 10 degrees. The receive steering angles were [-20, 0, 0, 0, 20] degrees, yielding five combinations of transmit and receive angles. The firing rate of the system was 10 kHz, yielding a Doppler PRF of 2 kHz. The setup, while not necessarily useful in a practical setting, was designed to verify that the slow-time signal has the expected behavior for different combinations of transmit and receive steering angles. The setups included six different velocities, four different flow directions, and three different pulse center frequencies, resulting in a total of 72 different simulation conditions. For each condition, 100 realizations were generated, each consisting of 40 slow-time samples. For each condition and per transmit-receive combination, the lag 1 auto-correlation phase angle was calculated according to (14) within a kernel region of 11 x 11 pixels (0.8 x 0.8 mm) and averaged over 100 realizations. Results were compared to reference phase values obtained using (18). All FLUST simulations were performed twice, first using point spread functions generated from Field II [3], then from the MUST simulator [5].

3.6. Using FLUST: An example of probe and acquisition design

Two alternative imaging setups are considered: Alternative 1 (A1) is a high resolution, low penetration setup with pitch of 100μm, center frequency of 12MHz and 256 elements. Alternative 2 (A2) is a low resolution, high penetration setup with a transducer pitch of 200μm, a center frequency of 6MHz and 128 elements. Flow in a straight tube with parabolic velocity profile is simulated, using angled plane wave transmissions with equal aperture sizes for the two alternative setups, see details in Table 2. The FLUST simulations are used to assess which of the two alternative designs yields axial velocity estimates with highest accuracy and precision. The velocity estimator assessed in this example is the autocorrelation estimator [9], where

$$v_{ax} = \frac{\angle\tilde{R}(1)}{\pi} v_N. \tag{20}$$

3.7. Using FLUST: Runtime comparison

Imaging setup A2 and the flow phantom described in 3.6 are further used to illustrate the difference in runtime between FLUST and Field II at an increasing number of realizations. FLUST was run and timed for 2, 20, 200, 2000 and 20,000 realizations. Using a tube diameter of 2 mm and a flow line spacing of 0.1 mm, each flow field realization is thus based on 21 flow lines. The tolerable threshold for interpolation errors was set to 4%, resulting in precalculation of PSFs with a sampling distance Δr of 60 μm. With the imaging setup yielding a resolution of 0.26$\times$ 0.58 mm, this

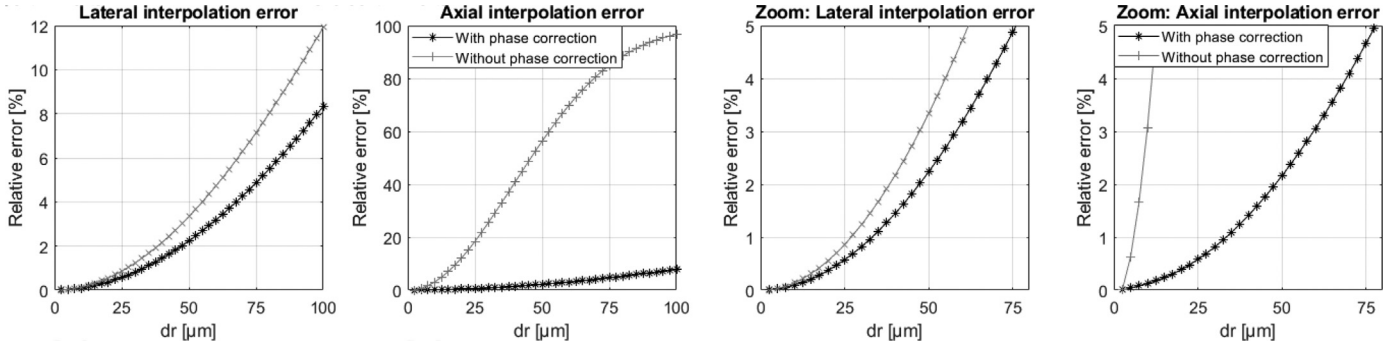

Fig. 2. The two leftmost panels show examples of PSF interpolation errors in step 2 of the FLUST technique as a function of spatial sampling distance (Δr) for scatterers moving in axial and lateral directions. Interpolation errors are shown with phase correction (black with star markers) and without phase correction (gray with crossed markers). Zoomed in versions are shown in the two rightmost panels. Using an upper limit on interpolation error of 4%, the required spacing would be 11 μm without phase correction. With phase correction, the PSF spacing can be increased to 67.5 μm while yielding the same interpolation errors.

resulted in precalculation of approximately 25 PSFs per resolution cell for the FLUST simulations. Field II was run and timed using the same phantom and imaging setup for a single realization, utilizing the standard point scatterer density value of 10 point scatterers per resolution cell. Runtime values for Field II were obtained by multiplying the single-realization runtime with the desired number of realizations (2, 20, 200, 2000 and 20000). Simulations were all run on an Intel(R) Xeon(R) Platinum 8260 CPU with 192 GB RAM available.

4. Results

4.1. Interpolation errors

Fig. 2 shows examples of interpolation errors calculated according to (10), for scatterers along axial and lateral directions, as a function of spatial sampling distance Δr. Interpolation errors with and without phase correction are shown in black and grey respectively. The predefined threshold for acceptable interpolation error could for instance be 4%, and the corresponding required PSF spacing read out from the right panels of Fig. 2. Without phase correction, the figure shows that the limiting factor would be the axial interpolation error, yielding a required spacing would of 11 μm. With phase correction, the limiting factor would be the lateral interpolation error, and the required spacing would be increased to 67.5 μm. Incorporating phase correction would in this example enable reduction of the run time with a factor of approximately 6 without compromising the integrity of the simulated data.

The interpolated signals corresponding to different spatial PSF densities are shown in Fig. 3. When using phase correction, both the magnitude and phase of the IQ signal are largely preserved when using a spatial step size of 65 μm. Using a step size of 105 μm yields a small change in the magnitude, but still largely retains the phase information of the signal. In contrast, when not using phase correction, the magnitude is significantly changed for a step size of 25 μm, significant phase changes are observed for a step size of 65 μm, and the phase of the signal is severely distorted for a step size of 105 μm. Note that when using phase correction (left column), the spatial sampling distances in Fig. 3 correspond to interpolation errors of 0%, 0.6%, 3.6% and 9.5%, respectively.

The impact of interpolation errors on the corresponding slow-time Fourier spectra is shown in the bottom row of Fig. 3. When using phase correction, the slow-time spectra are largely unaffected by the evaluated spatial step sizes. When not using phase correction, broadening of the spectra is observed when using a step size of 65 μm, and the center frequency changes significantly when using a step size of 105 μm.

4.2. Temporal oversampling and flow line weighting

Fig. 4 shows average power and lag 1 autocorrelation phase estimates for a simulated small tube phantom with velocities of 3 m/s. Without temporal oversampling, distinct stripe artifacts may be observed both in power and phase images. When using oversampling, no such stripe artifacts are visible.

Fig. 5 shows power Doppler images averaged over 100 realizations of ensembles of 40, with and without the suggested flow line weighting approach. Images in the upper row are from a gradient flow phantom, and the bottom row is from a spinning disk phantom. Without weighting, it can be seen that the distribution of power is non-uniform over the flow lines. As seen in the right column, the weighting in (13) compensates both for the temporal (over)sampling factor and the mean scatterer velocity in a flow line, resulting in an even distribution of power over the whole phantom region.

4.3. Signal integrity

Fig. 6 shows results of the signal integrity analysis for two different imaging setups. The distribution of the real and imaginary parts of the IQ signals are both normally distributed, indicating that the underlying speckle signal is fully developed [31]. In the 2-D auto-correlation R_{xx} of the M-mode signal, it may be observed that the energy is largely concentrated around the line corresponding to the underlying velocity of the scatterers, indicating that a cross-correlation function along the corresponding spatial line would have a peak corresponding to the correct velocity. The lag 1 phase angle estimates inside the kernel are all close to the reference value predicted by (18). This indicates that an auto-correlation based estimator would attain approximately the correct mean frequency, although a small systematic underestimation is observed. Finally, the averaged slow-time spectrum is centered around the correct angular frequency, indicating the usefulness of the signals for validation of spectral estimators.

The result of the second integrity test is shown in Fig. 7. Each of the figures contain 360 measurement points, corresponding to 72 different simulations, each containing five combinations of transmit and receive angles. The expected lag 1 phase values correspond well with the reference phase predicted using true velocities and steering angles, with root-mean-square deviations of 0.10 radians for the results using Field II and 0.19 radians when using MUST.

4.4. Using FLUST: An example of probe and acquisition design

Fig. 8 shows the PSFs from the two alternative probe and acquisition designs described in 3.6, whereas Fig. 9 provides a

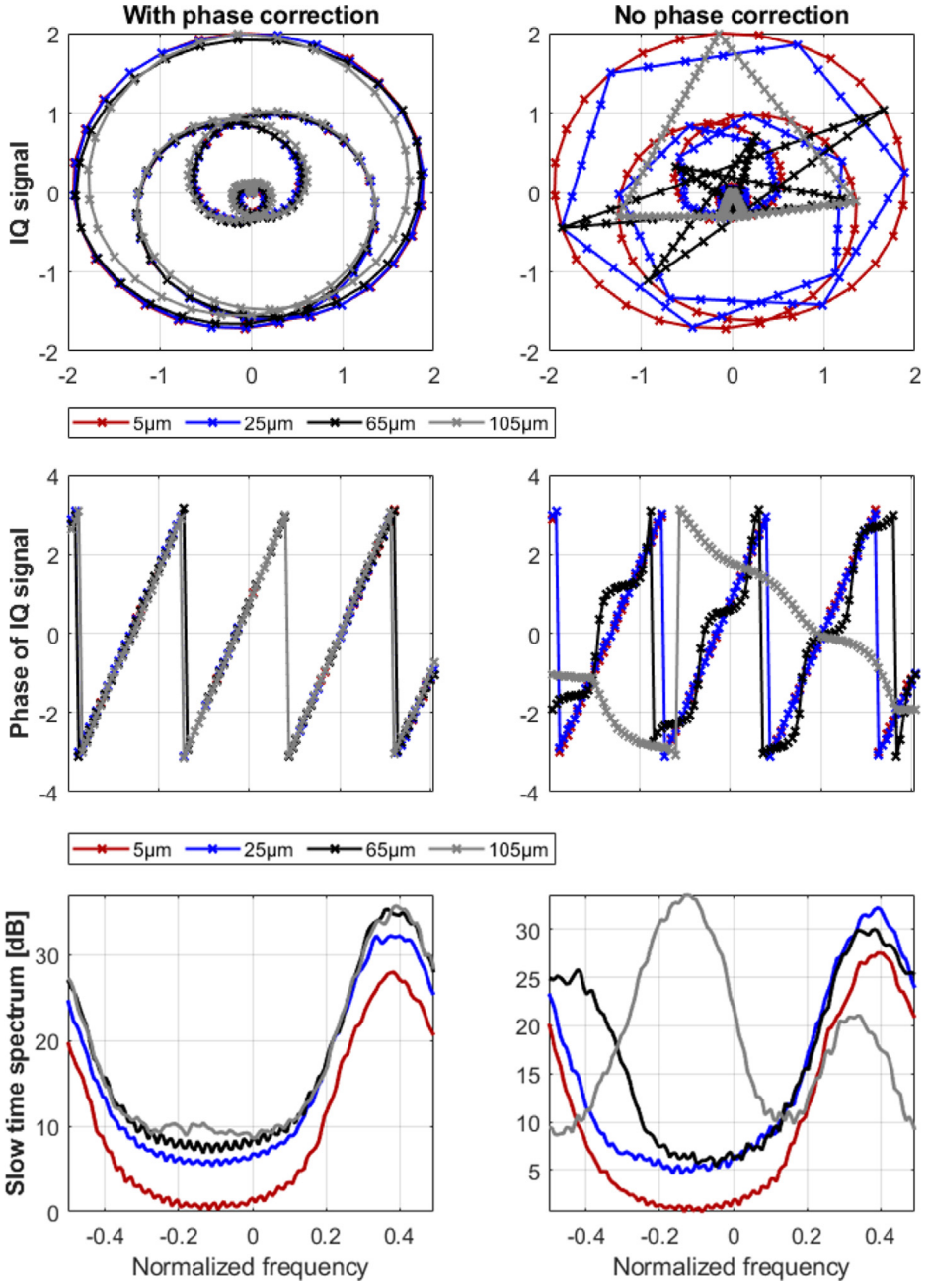

Fig. 3. The top row shows interpolated signals s_F^r, with and without phase correction, for setups in which a point scatterer move 5 μm each slow-time sample, and the corresponding signals s_F^r have spatial sampling distances Δr equal to 5, 25, 65 and 105 μm. For the latter three cases, the signals s_F^r coincide with s_F^r for each 5th, 13th and 21st sample, respectively. The rest of the samples are generated using interpolation. The middle row shows the effect of interpolation on the phase of the IQ signals. The bottom row show slow-time spectra, using the same set of sampling distances, from the central pixel in simulations of a tube phantom with velocities varying from 0.2 to 0.5 m/s.

comparison of the two in terms of mean and standard deviation of axial velocity estimates. The results shown in the left panels are based on 10 realizations of the simulated Doppler signal, whereas the right panels show mean and standard deviation using 2000 realizations. The ensemble length is 10 for both cases, and only velocity estimates along a single cross section of the tube are considered. The left panels indicate that A1 yields a much smaller negative bias for the maximum velocity, a smaller positive bias for the lower velocities near the walls, and an overall lower variance. By increasing the number of realizations to 2000, however, it may instead be observed that the two setups yield a similar and small negative bias for the maximum velocities. The difference in standard deviation also becomes clearer and easier to quantify. With all other parameters being equal, the difference in conclusions may only be attributed to the increased variance of the estimator v^{10} compared to v^{2000} as shown in (6).

4.5. Runtime comparison between FLUST and FIELD II

Fig. 10 shows a runtime comparison between FLUST and Field II using acquisition design 2 (A2). Each bar in the left panel shows runtime in seconds for FLUST (blue) and Field II (red) for an increasing number of realizations. Numbers on top of each bar are

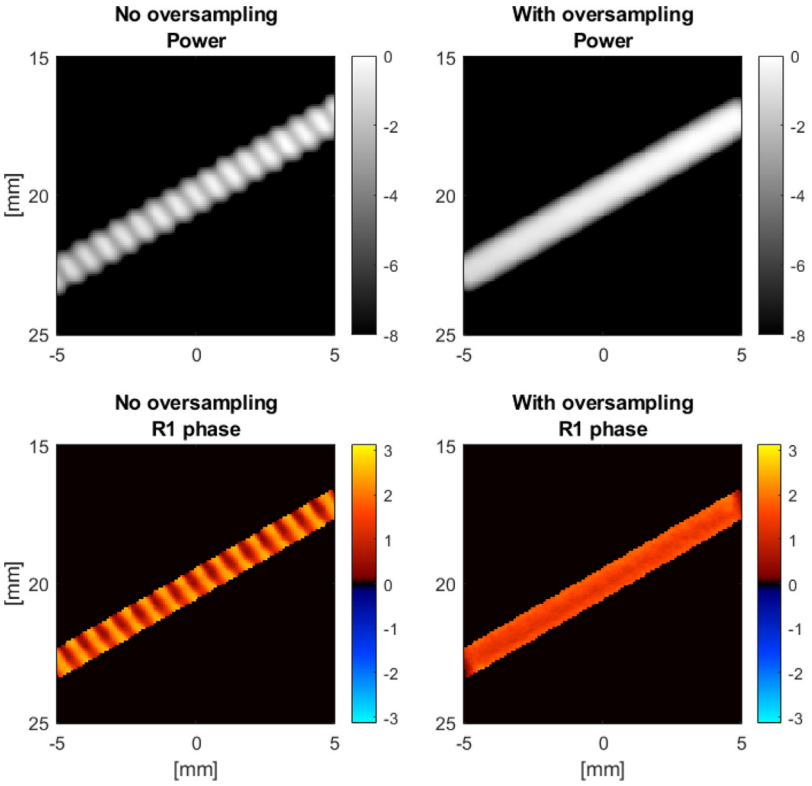

Fig. 4. Results from a simulation, with and without slow-time oversampling, of a 2-D plug flow tube phantom with constant velocity of 3 m/s. One new scatterer is spawned every slow-time sample at one edge of each flow line. Using the current setup without oversampling, these scatterers move 750 μm between each slow-time sample. The top row shows power in dB, the bottom row shows the phase of the autocorrelation function with lag 1. Both are averaged over 100 ensemble realizations.

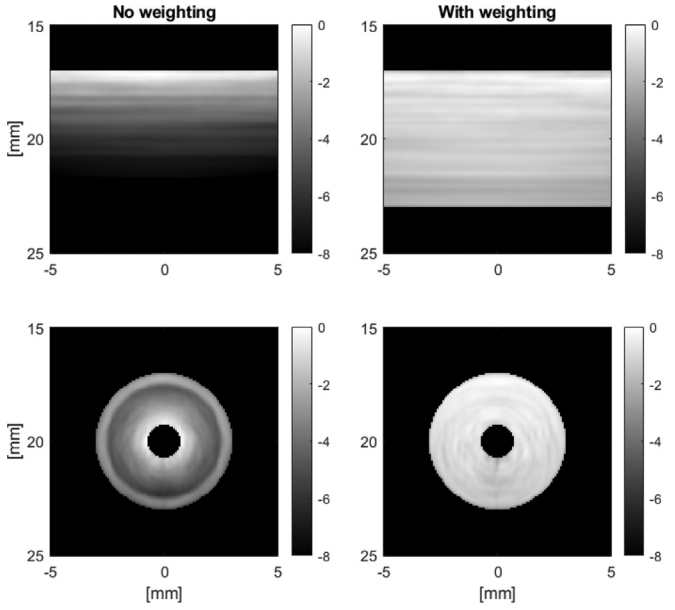

Fig. 5. Average power over 100 realizations with an ensemble size of 40, with and without weighting of flow lines. The top row shows a tube with diameter 6 mm and a velocity gradient from 0.1 m/s to 0.7 m/s, the bottom row shows a spinning disk with diameter 6 mm and velocity range from 0.2 m/s to 1.0 m/s.

runtime in seconds, but to fit the large variations in runtime, the vertical axes is in log-scale. The right panel shows the ratio between runtime in Field II and FLUST. Generating the results seen in Fig. 9 would take 9.3 hours (33546 s) using Field II, whereas FLUST can provide the same results in 2 minutes (120 s).

5. Discussion

The FLUST framework is an open source simulation tool that enables multiple realizations of high integrity IQ Doppler signals from simple or complex flow fields at low computational cost. Based on theoretical concepts introduced in our previous work [27], a framework consisting of a database of *in silico* flow phantoms and functions for acquisition and processing has been built and distributed as a part of the USTB project [28][1]. The current work also proposes and investigates several important contributions to the integrity of the flowline based simulation technique.

First, signal integrity after the spatial interpolation step of FLUST was ensured by controlling interpolation errors and automatically selecting the spatial density of precalculated PSFs. More specifically, a phase correction technique was proposed to reduce interpolation errors, and a method for automatic selection of spatial discretization of point spread functions was presented. It was observed that the use of phase correction allowed for using a significantly smaller number of precalculated PSFs (Fig. 2, Fig. 3), and therefore reduction in simulation time, without compromising the integrity of the simulated data. The spatial discretization of PSFs is determined automatically based on an upper bound for the acceptable energy loss after interpolation. The acceptable energy loss is left as a user-defined parameter determining a trade-off between accuracy and computational cost. Notably, using this parameter allows for an automated selection of spatial discretization without performing calculations for each new selection of probe geometry and/or pulse center frequency. In this work, energy loss due to interpolation errors were required to be smaller than 4%. However, as can be observed from results in Fig. 3, larger interpolation er-

[1] http://www.ustb.no/flust

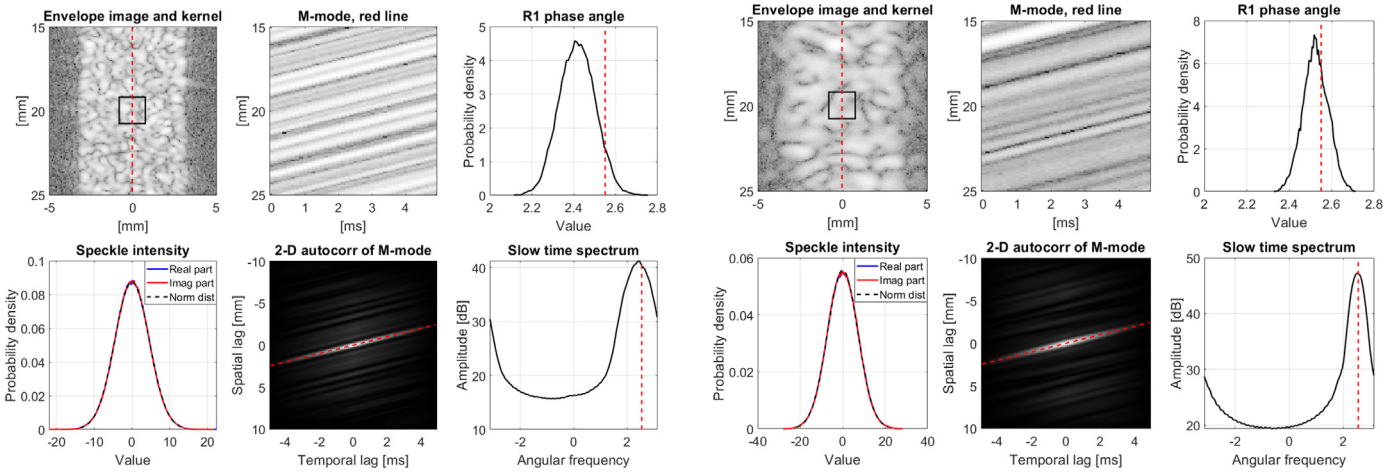

Fig. 6. Integrity analysis for a high resolution setup (left) and a low-resolution setup (right). The upper left panels show single frames from each simulation, with kernel of interest and M-mode line indicated. The lower left panels show signal statistics from the kernel of interest over all ensembles and realizations, with the normal distribution predicted from the signal amplitude as reference. The middle panels show M-mode signals, i.e. signals from the red line in the upper left panel over all slow-time samples in the ensemble, and the 2-D autocorrelation function of the M-mode signal. The right panels show the distribution of the R(1) phase estimate inside the kernel for all ensembles and realizations, and the slow-time spectrum, averaged over the kernel, ensembles and realizations. In these last two panels, the red line indicates the reference values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

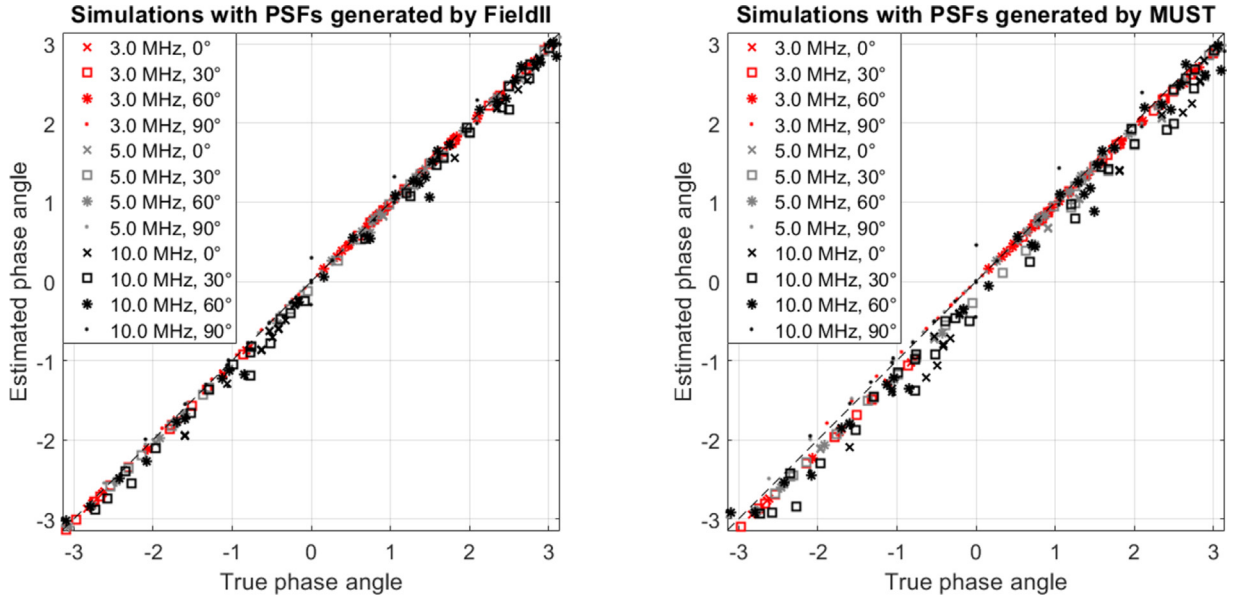

Fig. 7. Validation of the R(1) phase angle for 72 simulation setups, with varying flow velocity (7.5 to 45 cm/s), flow direction (0 to 90 degrees) and pulse center frequency (3 to 10 MHz). Phase estimates are presented in radians. The true velocities are as high as 6 times the Nyquist limit for the highest frequency. Each setup contains five combinations of transmit and receive angles, yielding a total of 360 phase estimates. The results in the left panel are generated using point spread functions (PSFs) obtained with FIELD II, whereas the results on the right are based on PSFs generated using the MUST simulator.

rors may be allowed in some cases without substantially degrading the results.

Second, steps were taken to ensure integrity of the final simulation results. It was shown that the use of temporal oversampling is necessary for high velocity flow scenarios, and a method for automatic determination of the oversampling factor was presented. Further, it was demonstrated that the expected power along different flow lines may vary, and that appropriate weighting resolves this issue. An integrity parameter that is currently not automatically determined is the flow line spacing. Whereas automatic determination of spatio-temporal PSF interpolation parameters ensure high signal integrity *along* flow lines, sufficient flow line density ensures high signal integrity *between* flow lines. In this work, a flow line spacing of $\lambda/2$ at the inlet was used. This spacing may need to be lowered if the cross-sectional area increases downstream, as is the case in bifurcations or aneurysms. Alternatively, phantoms may be designed such that additional flow lines are included in regions with higher cross-sectional area. This, however, will require new solutions to ensure correct weighting of flow lines in such regions.

The integrity of the simulation technique was assessed by confirming that the spatial and temporal correlation, frequency content and statistical properties all corresponded well with theoretical predictions. It may be observed from Fig. 6 that the mean phase shift seems slightly lower than the reference values, especially for the high resolution setup. This does not imply an inaccuracy of the simulator. Instead, it may be explained by the use of a large aperture, as the observed Doppler shift decreases away from the aperture center. Indeed, the observed bias is lower for the smaller aperture used in the low reso-

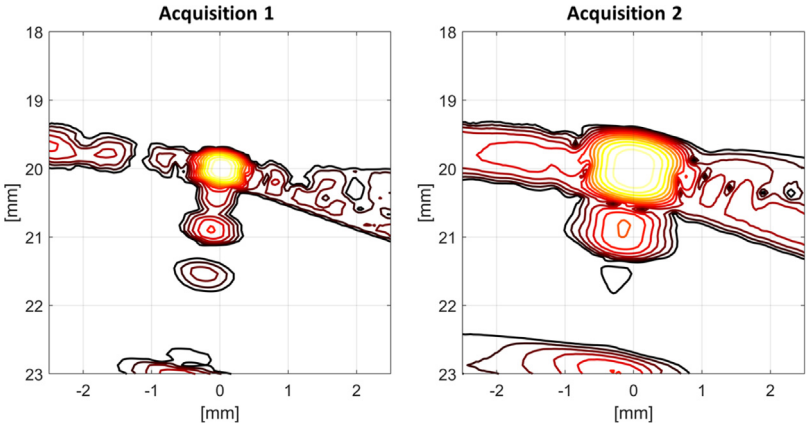

Fig. 8. Point spread functions using plane wave transmission from two different probe designs with equal aperture size. Acquisition 1: 100μm pitch, 4.5 cycles@ 12MHz pulse center frequency and 256 transducer elements. Acquisition 2: 200μm pitch, 4.5 cycles@ 6MHz and 128 transducer elements. Contour lines are drawn every 5 dB, covering a 60 dB range.

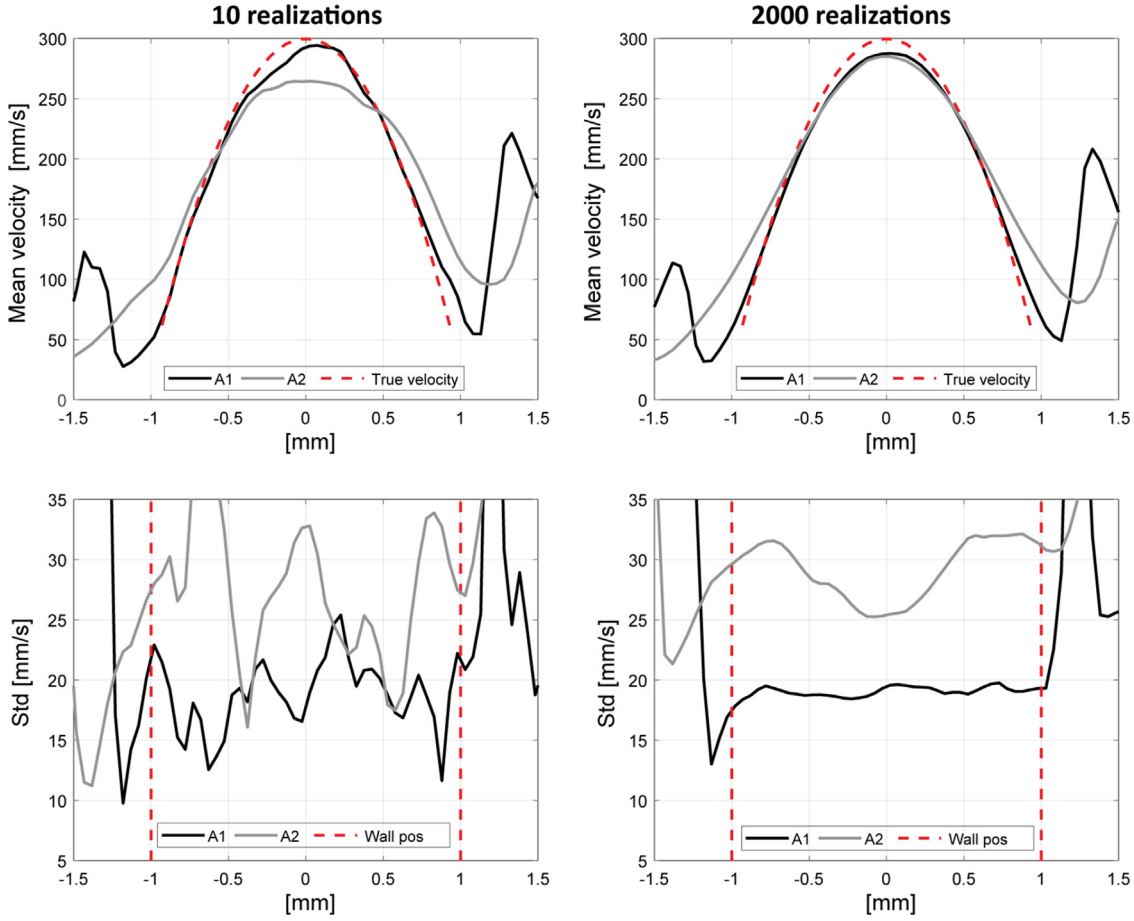

Fig. 9. A comparison of the two different probe designs with PSFs shown in Fig. 8, in terms of their respective autocorrelation based velocity estimates in a tube with a parabolic flow profile. Mean (top panels) and standard deviation (bottom panels) are based on 2000 (right) and 10 (left) realizations of the 10 samples long ensembles. The red dashed line in the upper panel indicates the true flow profile, whereas they indicate positions of the tube walls in the bottom panel. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

lution case. Another observation is that the slow-time spectral bandwidth is lower for the low resolution setup, yielding lower variance of an auto-correlation estimator, whereas the energy of the M-mode auto-correlation function is more concentrated for the high resolution setup, yielding lower variance for a cross-correlation based method. These observations confirm that the two types of estimators each have their own preferred acquisition setup.

Finally, an illustrative example of how FLUST may be used throughout the design and optimization process of a velocity estimator was included. The example also demonstrated the importance of utilizing a sufficient number of realizations of the flow field for performance evaluation of a velocity estimator, and that FLUST may be used to obtain statistically sound conclusions while keeping reasonably low runtimes. In fact, results in Fig. 10 show that compared to Field II, simulation time using FLUST is only to

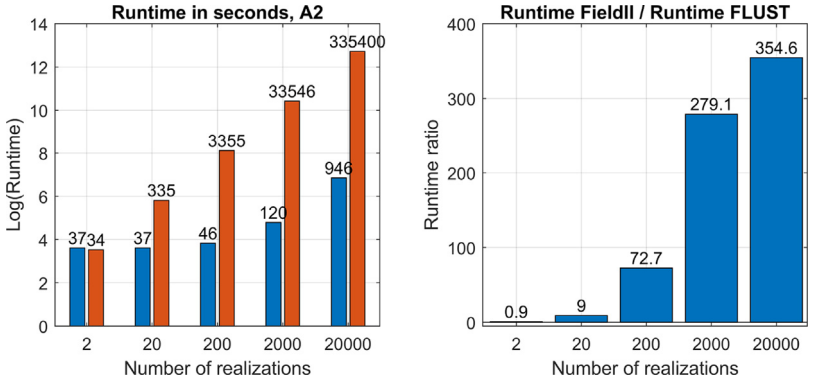

Fig. 10. Runtime comparison between Field II and FLUST for an increasing number of realizations. The left panel shows runtime in seconds for both simulation tools using an Intel(R) Xeon(R) Platinum 8260 CPU with 192 GB RAM. The right panel shows runtime ratios between Field II and FLUST.

a small extent dependent on the number of realizations. These results were based on an effective number of scatterers per resolution cell of 10 for Field II versus 25 for FLUST, where the latter is the result of an automatic parameter selection process. Users may therefore expect high integrity IQ Doppler signals at significantly reduced runtimes when using FLUST.

The way forward

One of the aims of the FLUST project is to provide a database of *in silico* flow phantoms with known velocity fields, available to the ultrasound community as a whole. In addition to single scenario phantoms with user-selected velocities, spatial gradients or degrees of out-of-plane motion, the database will include more complex flow scenarios such as vortices, helices or models derived from patient specific geometries. With FLUST, it should be easy to do simulations with various phantoms and acquisition setups, and to tune these to specific imaging interests. FLUST could be used in conjunction with either in vivo data, flow phantoms or complex simulations, providing challenging flow scenarios that can be separately recreated *in silico*. Complex flow simulations or in vivo data may for instance provide information on what kind of velocity gradients or vortices are likely to arise in a given imaging application. FLUST may then be used to investigate the true performance of the estimator in these conditions, and used to determine for which conditions the estimator will provide reliable estimates. The low computational cost allows for multiple iterations of this step, which may be used to improve the estimator itself, or the applied acquisition setup.

A current limitation with the FLUST approach is that it does not support pulsatile flow simulations. If such a feature should be implemented in practice, spatial interpolation can no longer be performed exclusively along flow lines as they may change over time. Instead, PSFs could be calculated on a regular grid, with sufficient density to reconstruct a PSF at any position without significant interpolation error. The phase correction step proposed in this work facilitates such a development, as it significantly decreases the required density of calculated PSFs.

We hope that a common library of phantoms and a standardized manner of performance assessment will contribute to making comparisons of different velocity estimation approaches easier and more reliable.

6. Conclusion

This work describes the open-source simulator FLUST, a computationally cheap and user-friendly framework to simulate ultrasound data from stationary velocity fields. In addition to the development of software, several improvements to the original FLUST technique have been proposed, resulting in reduced interpolation errors, reduced variability in signal power, and automatic selection of spatial and temporal discretization parameters. The integrity of the resulting simulated signals has been validated in an extensive study, with results indicating that speckle statistics, spatial and temporal correlation and frequency content all correspond well with theoretical predictions. In conclusion, the FLUST simulation framework was shown to be efficient and robust, and will hopefully prove useful as a tool for development, validation and comparison of ultrasound based velocity estimation schemes.

Declaration of Competing Interest

All authors declare that they have no conflicts of interest.

References

[1] D. Chen, J.F. Kelly, R.J. McGough, A fast near-field method for calculations of time-harmonic and transient pressures produced by triangular pistons, J. Acoust. Soc. Am. 120 (5) (2006) 2450–2459.
[2] E. Bossy, M. Talmant, P. Laugier, Three-dimensional simulations of ultrasonic axial transmission velocity measurement on cortical bone models, J. Acoust. Soc. Am. 115 (5) (2004) 2314–2324, doi:10.1121/1.1689960.
[3] J. Jensen, N.B. Svendsen, Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers, Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on 39 (2) (1992) 262–267.
[4] J.A. Jensen, Field: a program for simulating ultrasound systems, in: 10th Nordicbaltic Conference on Biomedical Imaging, vol. 4, Supplement 1, Part 1: 351–353, Citeseer, 1996.
[5] D. Garcia, Simus: an open-source simulator for medical ultrasound imaging. part i: theory & examples, Comput Methods Programs Biomed 218 (2022) 106726, doi:10.1016/j.cmpb.2022.106726.
[6] B. Treeby, B. Cox, K-wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields, J Biomed Opt 15 (2) (2010) 021314, doi:10.1117/1.3360308.
[7] H. Gao, H.F. Choi, P. Claus, S. Boonen, S. Jaecques, G.H. Van Lenthe, G. Van der Perre, W. Lauriks, J. D'hooge, A fast convolution-based methodology to simulate 2-dd/3-d cardiac ultrasound images, IEEE Trans Ultrason Ferroelectr Freq Control 56 (2) (2009) 404–409.
[8] T. Hergum, S. Langeland, E.W. Remme, H. Torp, Fast ultrasound imaging simulation in k-space, IEEE Trans Ultrason Ferroelectr Freq Control 56 (6) (2009) 1159–1167.
[9] C. Kasai, K. Namekawa, A. Koyano, R. Omoto, Real-time two-dimensional blood flow imaging using an autocorrelation technique, IEEE Transactions on sonics and ultrasonics 32 (3) (1985) 458–464.
[10] G.E. Trahey, J.W. Allison, O.T. von Ramm, Angle independent ultrasonic detection of blood flow, IEEE Trans. Biomed. Eng. BME-34 (12) (1987) 965–967, doi:10.1109/TBME.1987.325938.
[11] B.Y.S. Yiu, S.S.M. Lai, C.H. Alfred, Vector projectile imaging: time-resolved dynamic visualization of complex flow patterns, Ultrasound in Medicine and Biology 40 (9) (2014) 2295–2309.
[12] K.L. Hansen, H. Møller-Sørensen, J. Kjaergaard, M.B. Jensen, J.A. Jensen, M.B. Nielsen, Aortic valve stenosis increases helical flow and flow complexity: a study of intra-operative cardiac vector flow imaging, Ultrasound in Medicine & Biology 43 (8) (2017) 1607–1617.
[13] S. Holbek, C. Ewertsen, H. Bouzari, M.J. Pihl, K.L. Hansen, M.B. Stuart, C. Thomsen, M.B. Nielsen, J.A. Jensen, Ultrasonic 3-d vector flow method for quantitative in vivo peak velocity and flow rate estimation, IEEE Trans Ultrason Ferroelectr Freq Control 64 (3) (2016) 544–554.

[14] M. Correia, J. Provost, M. Tanter, M. Pernot, 4D ultrafast ultrasound flow imaging: in vivo quantification of arterial volumetric flow rate in a single heartbeat, Physics in Medicine & Biology 61 (23) (2016) L48.
[15] F. Vixège, A. Berod, P.-Y. Courand, S. Mendez, F. Nicoud, P. Blanc-Benon, D. Vray, D. Garcia, Full-volume three-component intraventricular vector flow mapping by triplane color doppler, Physics in Medicine & Biology 67 (9) (2022) 095004.
[16] A.E. Saris, H.H.G. Hansen, S. Fekkes, M.M. Nillesen, M.C.M. Rutten, C.L. De Korte, A comparison between compounding techniques using large beam-steered plane wave imaging for blood vector velocity imaging in a carotid artery model, IEEE Trans Ultrason Ferroelectr Freq Control 63 (11) (2016) 1758–1771.
[17] S. Fadnes, M.S. Wigen, S.A. Nyrnes, L. Lovstakken, In vivo intracardiac vector flow imaging using phased array transducers for pediatric cardiology, IEEE Trans Ultrason Ferroelectr Freq Control 64 (9) (2017) 1318–1326.
[18] I.K. Ekroll, J. Avdal, A. Swillens, H. Torp, L. Løvstakken, An extended least squares method for aliasing-resistant vector velocity estimation, IEEE Trans Ultrason Ferroelectr Freq Control 63 (11) (2016) 1745–1757.
[19] J. Avdal, L. Løvstakken, H. Torp, I.K. Ekroll, Combined 2-D vector velocity imaging and tracking doppler for improved vascular blood velocity quantification, IEEE Trans Ultrason Ferroelectr Freq Control 64 (12) (2017) 1795–1804.
[20] M.S. Wigen, S. Fadnes, A. Rodriguez-Molares, T. Bjåstad, M. Eriksen, K.H. Stensæth, A. Støylen, L. Lovstakken, 4-D intracardiac ultrasound vector flow imaging–feasibility and comparison to phase-contrast MRI, IEEE Trans Med Imaging 37 (12) (2018) 2619–2629.
[21] V. Perrot, I.K. Ekroll, J. Avdal, L.M. Saxhaug, H. Dalen, D. Vray, L. Løvstakken, H. Liebgott, Translation of simultaneous vessel wall motion and vectorial blood flow imaging in healthy and diseased carotids to the clinic: a pilot study, IEEE Trans Ultrason Ferroelectr Freq Control 68 (3) (2020) 558–569.
[22] S. Rossi, A. Ramalli, F. Fool, P. Tortoli, High-frame-rate 3-d vector flow imaging in the frequency domain, Applied Sciences 10 (15) (2020) 5365.
[23] J. Porée, G. Goudot, O. Pedreira, E. Laborie, L. Khider, T. Mirault, E. Messas, P. Julia, J.-M. Alsac, M. Tanter, et al., Dealiasing high-frame-rate color doppler using dual-wavelength processing, IEEE Trans Ultrason Ferroelectr Freq Control 68 (6) (2021) 2117–2128.
[24] H. Liebgott, A. Rodriguez-Molares, F. Cervenansky, J.A. Jensen, O. Bernard, Plane-wave imaging challenge in medical ultrasound, in: 2016 IEEE International Ultrasonics Symposium (IUS), 2016, pp. 1–4, doi:10.1109/ULTSYM.2016.7728908.
[25] J.A. Jensen, C.A. Villagómez-Hoyos, 2018 IEEE IUS SA-VFI Challenge, Tech. Rep. (2017).
[26] A. Swillens, L. Lovstakken, J. Kips, H. Torp, P. Segers, Ultrasound simulation of complex flow velocity fields based on computational fluid dynamics, IEEE Trans Ultrason Ferroelectr Freq Control 56 (3) (2009) 546–556, doi:10.1109/TUFFC.2009.1071.
[27] J. Avdal, I.K. Ekroll, H. Torp, Fast flow-line-based analysis of ultrasound spectral and vector velocity estimators, IEEE Trans Ultrason Ferroelectr Freq Control 66 (2) (2019) 372–381, doi:10.1109/TUFFC.2018.2887398.
[28] A. Rodriguez-Molares, O.M.H. Rindal, O. Bernard, A. Nair, M.A. Lediju Bell, H. Liebgott, A. Austeng, L. Lovstakken, The ultrasound toolbox, in: 2017 IEEE International Ultrasonics Symposium (IUS), 2017, pp. 1–4, doi:10.1109/ULTSYM.2017.8092389.
[29] L.Y.-L. Mo, R.S.C. Cobbold, A unified approach to modeling the backscattered doppler ultrasound from blood, IEEE Trans. Biomed. Eng. 39 (5) (1992) 450–461.
[30] J.-F. Chen, J.B. Fowlkes, P.L. Carson, J.M. Rubin, R.S. Adler, Autocorrelation of integrated power doppler signals and its application, Ultrasound in medicine & biology 22 (8) (1996) 1053–1057.
[31] T.A. Tuthill, R.H. Sperry, K.J. Parker, Deviations from rayleigh statistics in ultrasonic speckle, Ultrason Imaging 10 (2) (1988) 81–89.