Make the most of MUST, an open-source Matlab UltraSound Toolbox

Damien Garcia

Make the most of MUST, an open-source MATLAB UltraSound Toolbox

Damien Garcia

***Abstract*—Medical ultrasound imaging requires recurring use of elementary principles, namely, the choice of the transducer, the design of the emitted waves and their wavefronts (e.g. focused or wide), the demodulation of the received signals, their beamforming, and their post-processing to generate B-mode or flow images. The ultrasound signals used for research purposes can be synthetic or acquired. To make the whole pipeline easily accessible to many researchers and students, the objective was to provide an open-access toolbox, widely documented, and adapted to ultrasound imaging research involving experimental or simulation methods.**

The MUST *Matlab UltraSound Toolbox* contains algorithms that focus on the development, simulation, and analysis of ultrasound signals for medical imaging. The user can design various ultrasound-imaging scenarios and analyze their performance using simulated or acquired data. The MUST functions allow studying the characteristics of transducers and waveforms, analyze signals, and construct ultrasound images. The many examples provide a starting point for students and researchers to quickly gain an understanding of the essentials of ultrasound imaging. The simulators integrated into MUST provide very realistic acoustic pressure fields and ultrasound images. The Matlab MUST toolbox is freely available at https://www.biomecardio.com/MUST.

Before engaging in advanced ultrasound techniques and comparing them with so-called standard methods, it is necessary to have a good understanding of the latter and their advantages and limitations. With this in mind, the MUST toolbox includes everything needed for comprehensive ultrasound imaging: simulators of acoustic pressure fields and backscattered RF signals, delay-and-sum beamforming, B-mode imaging, wall filtering, color or vector Doppler, speckle tracking... The documentation and open access facilitate easy and intuitive use. If the MUST toolbox proves to be interesting, the author plans to integrate advanced features depending on the demands of the ultrasound community.

***Index Terms*—Ultrasonic imaging, Medical simulation, Signal processing, Image processing, Open access**

I. Introduction

SINCE the commercialization of research ultrasound scanners, researchers and engineers working in ultrasound imaging have access to digitized radio frequency (RF) signals. Software analysis has overtaken hardware, which has strongly contributed to the development of innovative ultrasound methods. These relate to the transmission of the waves, as well as the processing of the recorded echoes. Ultrasound research systems, such as Verasonics scanners, offer great flexibility for biomedical research and non-destructive testing. Figure 1 shows the number of articles referenced in Google Scholar that contain the word "*Verasonics*". It is apparent that testing and experimentation with such a system have quickly gained momentum, and this trend continues to grow. Terabytes of RF signals had to be acquired and digitally analyzed to produce the thousands of ultrasound images published in the literature.

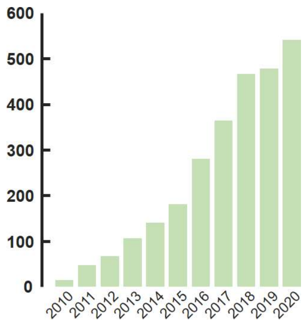

Fig. 1 – Number of yearly papers in which "*Verasonics*" appears, according to Google Scholar. The curve shows the growing interest of researchers in accessing raw ultrasound data.

Before conceiving *in vitro* or *in vivo* experiments, it may be advisable to first use computational ultrasound imaging. This methodological approach enables the exploration of multiple configurations before experimental evaluation and validation. For example, it helps in implementing optimized ultrasound sequences or transducer arrays, and developing beamforming and post-processing algorithms. Since shortly, ultrasound simulations are also used to train neural networks when hundreds or thousands of *in vitro* or *in vivo* acquisitions cannot be considered. The first wide-scope simulator (Field II) was introduced by J.A. Jensen [1]. Its free access and ease of use democratized the practice of ultrasound simulations and accelerated the development of new processing methods. Subsequently, B.E. Treeby and B.T. Cox introduced the software k-Wave [2], which is also widely used by the ultrasound community. Field II, k-Wave, and the Verasonics scanners work in a MATLAB environment because of its rich repertoire of built-in functions for data analysis and processing, and image display.

This work was supported in part by the LABEX CeLyA (ANR-10-LABX-0060) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-16-IDEX-0005) operated by the French National Research Agency.
D. G. (e-mail: damien.garcia@inserm.fr; garcia.damien@gmail.com) is with INSERM at CREATIS (Centre de Recherche en Acquisition et Traitement de l'Image pour la Santé), CNRS UMR 5220 – INSERM U1294 – Université Lyon 1 – INSA Lyon. D. G. began this work when he was head of the RUBIC (Research Unit of Biomechanics and Imaging in Cardiology) at the LBUM, CRCHUM (Centre de recherche du centre hospitalier de l'Université de Montréal), Montreal, QC, Canada.

978-0-7381-1209-1/21/$31.00 ©2021 IEEE

Once RF signals are acquired or simulated, they need to be analyzed to reconstruct interpretable images of the insonified medium, whether stationary or in motion. The processing steps typically include demodulation, beamforming, filtering, and motion detection techniques such as Doppler and speckle tracking. Although essential, these analysis techniques are sometimes not fully understood. It is uncommon for the codes to be provided by research teams after publication, which can be a barrier for students and researchers entering the field of ultrasound imaging. An attempt to address this gap was the USTB toolbox [3], which was created following the PICMUS challenge [4]. The claimed objective of the USTB was to facilitate the comparison of imaging techniques and the dissemination of research results. It mostly offers the experimental data that were downloadable during the PICMUS challenge and includes several beamforming codes. This toolbox is primarily intended for an experienced user with specific needs for comparisons of beamforming methods.

For educational purposes, and to facilitate the handling of ultrasound signals, I have developed a Matlab toolbox for ultrasound imaging called MUST. To make it easy to use, MUST is extensively documented by a website with many examples (www.biomecardio.com/MUST). The MUST toolbox contains open functions for common tasks in ultrasound imaging: emission design, demodulation, beamforming, imaging, wall filter, Doppler, speckle tracking... MUST also includes an acoustic pressure field simulator (PFIELD) and an RF echo simulator (SIMUS) to generate realistic ultrasound images. In contrast to Field II, which works in the time domain, the MUST simulators PFIELD and SIMUS operate in the Fourier domain.

II. Description of MUST

The MUST toolbox enables the design of transmit sequences, RF signal simulation, post-processing and demodulation, beamforming and image generation, motion estimation, and the creation of educational figures and animations. All the functions, their description, and the examples illustrating their application are available on the MUST website (Figure 2). The flowchart in Figure 3 shows how the MUST functions can be used in experimental or computational ultrasound imaging.

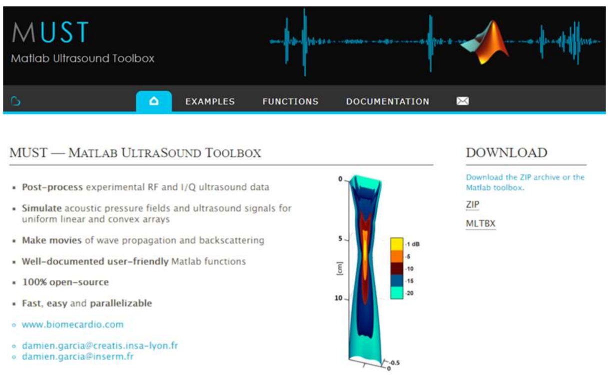

Fig. 2 – MUST website, www.biomecardio.com/MUST (2021). The MUST Matlab UltraSound Toolbox is freely available. The front-page image represents a simulated acoustic field of a focused wave with a cardiac phased array.

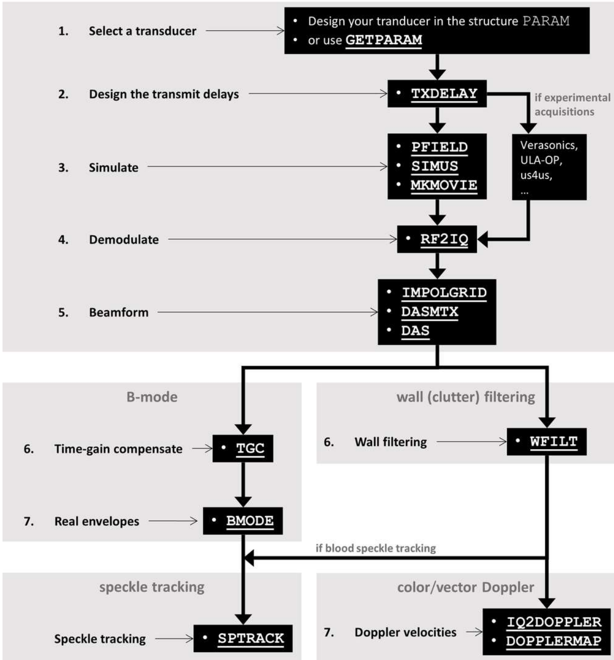

Fig. 3 – This flowchart shows how some of the MUST functions can typically be used for a comprehensive study in ultrasound imaging. A brief description of the MUST functions is given in TABLE I.

TABLE I briefly describes the open functions that are included in MUST (Release 2021). In particular, it includes functions that have been the subject of publications: DASMTX [5] to create a delay-and-sum matrix for beamforming with sparse matrix-vector multiplication, SPTRACK for speckle tracking [6], [7], PFIELD and SIMUS for simulations of acoustic pressure fields and ultrasound RF signals [8]–[11].

TABLE I – List of the MUST functions (Nov. 2021)

Acquire	
txdelay	Generate transmit delays
rf2iq	I/Q demodulation of RF data
tgc	Time-gain compensation
Beamform	
das	Delay-and-sum of RF and I/Q signals
dasmtx	Delay-and-sum matrix for beamforming
Motion	
wfilt	Wall (clutter) filtering
iq2doppler	Convert I/Q signals to Doppler velocities
sptrack	Motion estimation by speckle tracking
Simulate	
getparam	Get parameters of a linear or convex array
getpulse	Get the one-way or two-way transmit pulse
pfield	Simulation of RMS acoustic pressure fields
simus	Simulation of RF signals
Display	
impolgrid	Polar-type grid for ultrasound images
bmode	Generate 8-bit B-mode image
dopplermap	A color map for color Doppler
mkmovie	Make movies and animated GIF of wave propagation

The functions DASMTX, SPTRACK, PFIELD, and SIMUS, are briefly introduced in the next paragraphs to illustrate the functionality of MUST.

A. DASMTX – Delay-and-Sum Matrix

The function DASMTX generates a sparse matrix for delay-and-sum beamforming of RF or I/Q (in-phase/quadrature) signals. Let s_k be the RF or I/Q signal recorded by the k^{th} transducer of an N-element array. If the time series are stacked in a column vector noted $s = [s_1, \ldots, s_k, \ldots, s_N]^{\mathrm{T}}$, the beamformed signals s_{bf} are given by the following sparse matrix-vector multiplication:

$$s_{\mathrm{bf}} = M_{\mathrm{DAS}}\, s, \text{ with } M_{\mathrm{DAS}} \text{ being the DAS matrix} \quad (1)$$

The DAS matrix can be real (when beamforming RF signals) or complex (when beamforming I/Q signals) and contains the interpolation coefficients that are used to beamform the temporal signals at specific locations. The sparsity of the M_{DAS} matrix depends on both the number of samples in the s_k and the interpolation scheme (nearest vs. linear vs. quadratic, etc. See equation (19) in [5]). If the ultrasound sequence (array, transmit, receive) and the beamforming point locations are kept unchanged, the M_{DAS} matrix (Figure 4) needs to be calculated once.

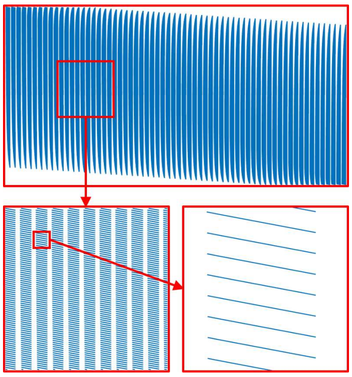

Fig. 4 – An example of sparse DAS matrix for diverging-wave imaging (size = 65536×126464, sparsity = 99.92%) as created by DASMTX.

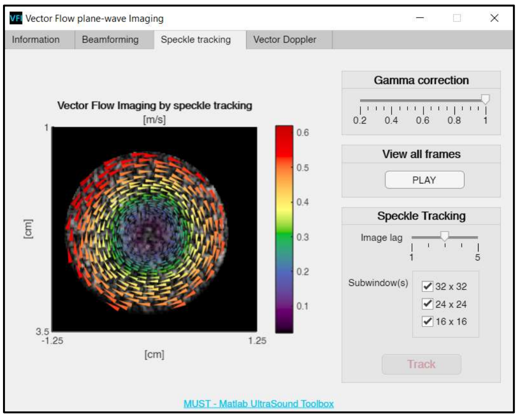

Fig. 5 – Speckle tracking with SPTRACK in a rotating disk insonified by unsteered plane waves. This image is from the application VFI-by-PWI (vector flow imaging by plane wave imaging) used during the *IUS* 2021 short course on motion estimation (www.biomecardio.com/files/VFI_by_PWI.zip).

B. SPTRACK – Motion Estimation by Speckle Tracking

Speckles, resulting from interferences between backscattered echoes, are acoustic markers of the insonified tissues. In a time series of ultrasound images, these markers are sufficiently preserved from one image to the next if the frame rate is set high enough. It is then possible to follow the speckle patterns and derive the local displacements of the tissues with a frame-by-frame method. A widely used approach is local block matching based on cross-correlation measures [12]. The MUST function SPTRACK computes FFT-based normalized cross-correlations [13]. To detect both large and small displacements, SPTRACK includes a coarse-to-fine multiscale approach, i.e. the displacement estimates are iteratively refined by decreasing the size of the blocks. Subpixel displacements are determined by a parabolic fitting around the cross-correlation peaks. The estimated motion field is finally smoothed with a robust unsupervised spline smoother [7], [13]. Figure 5 displays an *in vitro* example of a velocity field estimated by the MUST function SPTRACK with ultrafast plane-wave imaging.

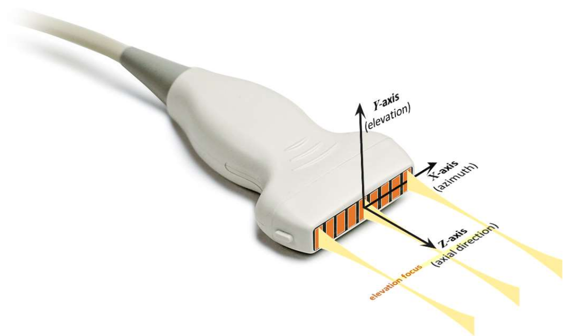

Fig. 6 – Coordinate system for a rectilinear array.

C. PFIELD – Acoustic Pressure Fields

The function PFIELD (in Release 2021) simulates acoustic pressure fields radiated by a uniform linear or convex array. PFIELD works in the "conventional" coordinate system illustrated in Figure 6. PFIELD works in the frequency domain. It is based on far-field (Fraunhofer) and paraxial (Fresnel) approximations. The transducer elements are split into sub-elements along the X direction to make the far-field hypothesis valid. The paraxial approximation is also applicable if there is no significant deviation from the X-Z azimuth plane. In PFIELD, the harmonic pressure at time t, for an angular frequency ω (wavenumber k), and at position $\boldsymbol{X} = (X, Y, Z)$, is $P(\boldsymbol{X}, \omega, t) =$

$$P_{\mathrm{Tx}}(\omega) e^{-i\omega t} \sum_{n=1}^{N} W_n \frac{e^{ikr_n}}{r_n} D(\theta_n, k) \delta(Y, r_n, k)\, e^{i\omega\Delta\tau_n}. \quad (2)$$

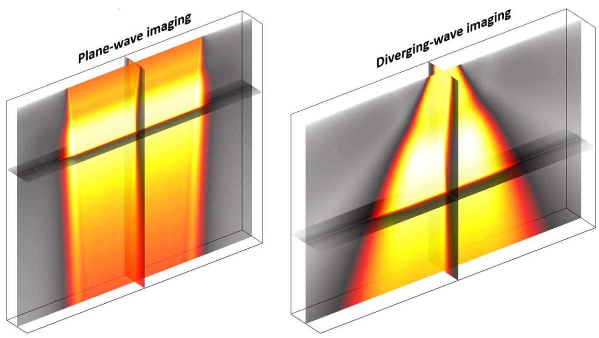

Fig. 7 – Root-mean-square pressure fields by PFIELD for a 128-element linear array and 64-element cardiac phased array. Adapted from [11].

$\mathcal{N}$ stands for the total number of sub-elements, r_n is the distance that separates the position $\boldsymbol{X}$ from the n^{th} sub-element, and θ_n is the azimuth angle that defines the obliquity of position $\boldsymbol{X}$ with respect to the n^{th} sub-element. $P_{\text{Tx}}(\omega)$ represents the spectrum of the transmit pressure pulse. The D and δ parameters relate to the directivity of a sub-element and the elevation focus, respectively. The $\Delta\tau_n$ are the transmit delays, and the W_n are the transmit apodizations. Figure 7 shows simulated acoustic pressure fields for ultrafast ultrasound imaging.

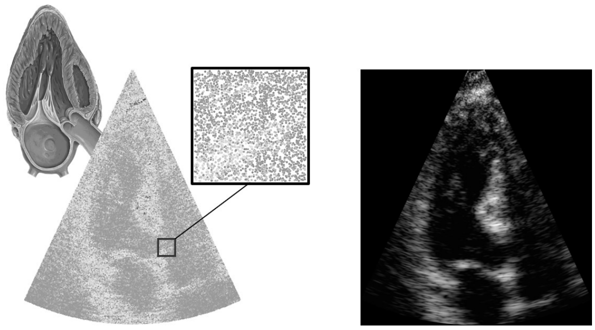

Fig. 8 – A simulated three-chamber-view echocardiographic image. The RF signals were generated by SIMUS using ~40,000 scatterers (left).

D. SIMUS – Simulation of Ultrasound RF Signals

The function SIMUS (Release 2021) simulates ultrasound RF radio-frequency signals generated by a uniform linear or convex array insonifying a medium of scatterers. SIMUS uses PFIELD and point scatterers to generate echoes. The scatterers become individual monopole sources when an incident wave reaches them. They do not interact acoustically (weak scattering assumption). Each scatterer is assigned a reflection coefficient that describes the amplitude of the backscattered wave. The pressures recorded by the transducer elements are derived from Equation (2) by using the principle of acoustic reciprocity. The mathematics inside PFIELD and SIMUS are described in detail in [10]. Figure 8 shows a cardiac three-chamber view obtained with the MUST toolbox: RF signals were simulated with SIMUS, demodulated with RF2IQ, and beamformed with DASMTX. The B-mode image was then obtained with the function BMODE (see TABLE I).

III. IS MUST A MUST-HAVE?

In this *IUS* proceeding was presented the Matlab UltraSound Toolbox MUST. It allows students, researchers, and other ultrasound imaging enthusiasts to post-process experimental RF and I/Q ultrasound data, and to simulate acoustic pressure fields and ultrasound signals emitted and received by ultrasound transducers. The MUST toolbox contains user-friendly and well-documented Matlab functions. It is 100% open-source, fast, and easy to use. Examples of MUST applications include experimental design, demodulation and beamforming, B-mode imaging, color and vector Doppler, speckle tracking, realistic simulations, etc. MUST also offers the possibility to make wave propagation and backscattering movies. The reader is invited to visit the website (www.biomecardio.com/MUST) and run the examples that illustrate the diversity of MUST. The functions of MUST have all been the subject of published scientific articles, for example with ultrafast Doppler echocardiography in volunteers with a Verasonics scanner [6], [9], [14], or for the study of vector Doppler *in silico*, *in vitro* and *in vivo* [8], [15]. MUST has its own simulator SIMUS, which is very easy to use. For users with the MATLAB *Parallel Computing Toolbox*, SIMUS can be run on a parallel pool of workers. Unlike Field II, its code is open, allowing the advanced user to make modifications or implement it with a different programming language.

The MUST toolbox will evolve according to the users' requests. It will probably include functionalities for volume ultrasound imaging with matrix ultrasound probes. The current version of MUST contains only popular techniques (e.g. DAS beamforming, autocorrelation-based Doppler, block matching by cross-correlation, wall filter by polynomial regression or SVD, etc.) If the MUST toolbox shows some success, it is likely that less classical approaches, which have proven to be successful, will be integrated into future versions. Standalone executables will also be offered to extend the outreach of MUST.

Its format and simplicity have been designed to allow the users to make the most of MUST. The usage will tell us if the MUST toolbox is a must-have, or at least a must-try, for research in ultrasound imaging.

IV. REFERENCES

[1] 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*, 1996, pp. 351–353.

[2] B. E. Treeby and B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," *JBO, JBOPFO*, vol. 15, no. 2, p. 021314, 2010.

[3] A. Rodriguez-Molares *et al.*, "The UltraSound ToolBox," in *2017 IEEE International Ultrasonics Symposium (IUS)*, 2017.

[4] H. Liebgott *et al.*, "Plane-wave imaging challenge in medical ultrasound," *IEEE International Ultrasonics Symposium (IUS)*, 2016.

[5] V. Perrot *et al.*, "So you think you can DAS? A viewpoint on delay-and-sum beamforming," *Ultrasonics*, vol. 111, p. 106309, 2021.

[6] P. Joos *et al.*, "High-frame-rate speckle-tracking echocardiography," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 65, no. 5, pp. 720–728, 2018.

[7] V. Perrot and D. Garcia, "Back to basics in ultrasound velocimetry: tracking speckles by using a standard PIV algorithm," *IEEE International Ultrasonics Symposium (IUS)*, 2018.

[8] S. Shahriari and D. Garcia, "Meshfree simulations of ultrasound vector flow imaging using smoothed particle hydrodynamics," *Phys Med Biol*, vol. 63, pp. 1–12, 2018.

[9] J. Porée, D *et al.*, "High-frame-rate echocardiography using coherent compounding with Doppler-based motion-compensation," *IEEE Transactions on Medical Imaging*, vol. 35, no. 7, pp. 1647–1657, 2016.

[10] D. Garcia, "SIMUS: an open-source simulator for ultrasound imaging. Part I: theory & examples," *arXiv:2102.02738*, 2021.

[11] A. Cigier *et al.*, "SIMUS: an open-source simulator for ultrasound imaging. Part II: comparison with three popular simulators," *arXiv:2103.05521*, 2021.

[12] D. Garcia *et al.*, "Introduction to speckle tracking in cardiac ultrasound imaging," in *Handbook of speckle filtering and tracking in cardiovascular ultrasound imaging and video*, IET, 2017.

[13] D. Garcia, "A fast all-in-one method for automated post-processing of PIV data," *Exp Fluids*, vol. 50, pp. 1247–1259, 2011.

[14] J. Faurie *et al.*, "Intracardiac vortex dynamics by high-frame-rate Doppler vortography – in vivo comparison with vector flow mapping and 4-D flow MRI.," *IEEE Trans Ultrason Ferroelectr Freq Control*, vol. 64, no. 2, pp. 424–432, 2017.

[15] C. Madiena *et al.*, "Color and vector flow imaging in parallel ultrasound with sub-Nyquist sampling," *IEEE Trans Ultrason Ferroelectr Freq Control*, vol. 65, no. 5, pp. 795–802, 2018.