The Journal of Open Source Software

QUPS: A MATLAB Toolbox for Rapid Prototyping of Ultrasound Beamforming and Imaging Techniques

Thurston Brevett [1]

1 Stanford University

DOI: 10.21105/joss.06772

Software

- Review
- Repository
- Archive

Editor: Adam Tyson

Reviewers:

- @rafaelorozco
- @tomelse

Submitted: 18 April 2024
Published: 29 August 2024

Summary:

Medical ultrasound encompasses a rapidly developing field of research with many potential applications from diagnostics to therapeutics. Public data initiatives and recent industry trends have reduced the cost and complexity to engage in research via publicly available data, on-board data acquisition functions, and commercial off the shelf (COTS) imaging systems. However, terminology, data formatting standards, and computing practices tend to be highly fragmented throughout the ultrasound research field, leading to a high barrier of entry that hinders communication, code reuse, and shareability.

`QUPS` aims to facilitate ultrasound research by offering a flexible, performant, modular, and intuitive toolbox that can be used by experienced researchers as well as curious students for data acquisition, generation, representation, and processing. It pursues this goal by offering tools designed to reduce the burden of common computational tasks involved in the scientific process by standardizing data formats and offering hardware accelerated functionality.

Statement of need:

The lack of data and terminology standardization in ultrasound technology research is an easily overlooked impediment to developing research, as it encourages reinventing the wheel and can easily lead to misinterpretation. At its lowest level, `QUPS` provides memory efficient definition classes capable of representing a wide variety of common imaging systems and enforces standardized units, coordinate systems, and nomenclature. All classes have data visualization support, making it far easier to share and validate data amongst collaborators and with the public.

`QUPS` provides a variety of compute and memory optimized implementations of common signal processing methods (e.g. beamforming, resampling, or convolution), which take advantage of native parallelization tools within MATLAB or hardware acceleration via CUDA or OpenCL when feasible by leveraging `MatCL` (Heinisch & Ostaszewski, 2018). `QUPS` exploits data parallelism primarily using a single program multiple data (SPMD) approach, and optimistically dispatches to the most efficient implementation available on a user's computer. For access to cutting edge research, it can also export or import to `USTB` (Rodriguez-Molares et al., 2017), an ultrasound toolbox which contains a library of advanced algorithms from the literature.

To facilitate data acquisition, `QUPS` provides tools to import data from Verasonics Vantage ultrasound research scanners, which uses a flexible yet highly complex data standard. Similarly, to facilitate data generation, `QUPS` also interfaces with multiple popular free ultrasound field simulators including `FieldII` (Jensen, 1996), `MUST` (Garcia, 2021), and `k-Wave` (Treeby & Cox, 2010). Simulations can be batched on a compute cluster via MATLAB's parallel server toolbox.

Alternatively, when a coarse but fast approximation is preferable, `QUPS` offers a hardware

accelerated simulator with minimal complexity. This method uses a green's function to simulate ideal isotropic point sources (transmitters), point receivers, and point scatterers with radial propagation loss using the equation

$$x_{mn}(t) = \sum_s \frac{v(t - c_0^{-1}\left(||\mathbf{p}_s - \mathbf{q}_m|| + ||\mathbf{p}_s - \mathbf{r}_n||\right))}{||\mathbf{p}_s - \mathbf{q}_m|| \cdot ||\mathbf{p}_s - \mathbf{r}_n||}$$

where t is time, c_0 is the speed of sound, $v(\cdot)$ is the transmit waveform, m, n, and s are the transmit, receive, and scatterer indices, $\mathbf{q}_m$ and $\mathbf{r}_n$ and $\mathbf{p}_s$ are the transmit elements, receive elements, and scatterer locations in 3D Cartesian coordinates, and $x_{mn}(\cdot)$ is the received echo waveform as a function of time. This expression is embarrassingly parallel over m and n, and can be further accelerated with a branch-and-bound approach used to minimize unnecessary computation. Both of these aspects are exploited in native MATLAB, as well as in CUDA and OpenCL kernels.

Broadcasting conventions are employed throughout to reduce the memory footprint while offering significant flexibility. For example, to implement delay-and-sum beamforming, one of the most common operations in ultrasound imaging, `QUPS` provides a GPU-enabled method employing the equation

$$b_{ijkf} = \sum_m \sum_n \alpha_{ijknm} \cdot x_{mnf}(c_0^{-1}\left(||\mathbf{p}_{ijk} - \mathbf{r}_n|| + (-1)^{s(\mathbf{p}_{ijk}, \mathbf{q}_m)} \cdot ||\mathbf{p}_{ijk} - \mathbf{q}_m||\right) - t_m)$$

where m, n, and f are the transmit, receive and (image) frame indices, i, j, and k are the voxel (or pixel) indices, $x_{mnf}(\cdot)$ is the voltage signal (trace) as a function of time, t_m is the start time for transmit m, $\mathbf{p}_{ijk}$ are the voxel positions in 3D cartesian coordinates, $\mathbf{q}_m$ and $\mathbf{r}_n$ are the virtual transmit and receiver positions, and $s(\cdot,\cdot)$ is a threshold function based on the relation between the virtual transmit position and the imaging voxel position. Rather than require the user to specify the full five dimensional tensor α_{ijknm}, any subset of these dimensions may be singular to efficiently form e.g. a transmit by lateral weighting (apodization) scheme or a receiver by depth weighting scheme. Permutations of the input data e.g. x_{nfm} or $\mathbf{p}_{kji}$ are handled via properties specifying the indices rather than by enforcing a convention, which may lead to costly memory operations.

Engaging in research often involves a plethora of small choices in order to configure software to suit the needs of a particular scientific inquiry. While exposing the user to these choices is necessary for some fields of research, mandating a suitable input forces all users to consider each aspect of the methodology. In most cases, the vast majority of these choices are either ultimately inconsequential or there exists a clear optimal or, at a minimum, a clear *acceptable* choice. For example, for finite-difference time-domain (FDTD) simulators, knowledge of a transducer's frequency bandwidth is sufficient to determine a reasonably sized temporal sampling interval that satisfies the Courant-Friedrichs-Lewy (CFL) stability criterion. For research regarding the stability and accuracy of such simulators, the sampling interval is a critical choice. In many other cases however, this aspect is irrelevant as long as the sampling interval is *sufficiently* small. Requiring a user to be sufficiently well-versed with such minutia creates a barrier to entry with little benefit.

Throughout the toolbox, `QUPS` provides reasonable defaults that abstract and simplify the number and complexity of such choices so that researchers can focus on the big picture questions. When these choices do become critical as is the case for researchers and experts, such users can engage with the toolbox at a lower level, or simply use the available functionality as a template to be modified to suit their needs.

Acknowledgements

This material is based upon work supported by the National Institutes of Health under award number R01EB027100 and by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1656518. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Garcia, D. (2021). Make the most of MUST, an open-source matlab UltraSound toolbox. *2021 IEEE International Ultrasonics Symposium (IUS)*, 1–4. https://doi.org/10.1109/IUS52206.2021.9593605

Heinisch, P., & Ostaszewski, K. (2018). MatCL: A new easy-to use OpenCL toolbox for MathWorks Matlab. *Proceedings of the International Workshop on OpenCL*, 8:1–8:1. https://doi.org/10.1145/3204919.3204927

Jensen, J. (1996). FIELD: A program for simulating ultrasound systems. *Medical and Biological Engineering and Computing*, *34*, 351–352.

Rodriguez-Molares, A., Rindal, O. M. H., Bernard, O., Nair, A., Lediju Bell, M. A., Liebgott, H., Austeng, A., & L vstakken, L. (2017). The UltraSound ToolBox. *2017 IEEE International Ultrasonics Symposium (IUS)*, 1–4. https://doi.org/10.1109/ULTSYM.2017.8092389

Treeby, B. E., & Cox, B. T. (2010). K-wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. *Journal of Biomedical Optics*, *15*(2), 021314. https://doi.org/10.1117/1.3360308