KSIM

Acoustoelectric Simulator

*EUNIL*

Software written and maintained by Chet Preston

Email questions to [createfacts@gmail.com](mailto:createfacts@gmail.com)

**Table of Contents**

[KSIM Functionalities 4](#_heading=h.gjdgxs)

[GUI Overview 5](#_heading=h.30j0zll)

[How to run a full 2D acoustoelectric simulation 6](#_heading=h.1fob9te)

[Creating a simulation grid (kgrid) 7](#_heading=h.3znysh7)

[Creating a medium 8](#_heading=h.2et92p0)

[Creating sensors 9](#_heading=h.tyjcwt)

[Creating an acoustic transducer 10](#_heading=h.3dy6vkm)

[Running and acoustic simulation 13](#_heading=h.1t3h5sf)

[Multiple Simulations 14](#_heading=h.4d34og8)

[Viewing Pressure output in KSIM 15](#_heading=h.2s8eyo1)

[Viewing a sensor in KSIM 15](#_heading=h.17dp8vu)

[Creating currents sources 18](#_heading=h.3rdcrjn)

[Running an acoustoelectric simulation 19](#_heading=h.26in1rg)

[Analyzing the results 20](#_heading=h.lnxbz9)

[Running a full 3D acoustoelectric Simulation 22](#_heading=h.35nkun2)

[Load CT 23](#_heading=h.1ksv4uv)

[Creating a 2D Medium 23](#_heading=h.44sinio)

[Creating a 3D Medium 25](#_heading=h.2jxsxqh)

[Electroids 26](#_heading=h.z337ya)

[Automated functions 28](#_heading=h.3j2qqm3)

[Aberration Correction 29](#_heading=h.1y810tw)

[Analytical corrections 30](#_heading=h.4i7ojhp)

[Time Reversal 32](#_heading=h.2xcytpi)

[Collecting time-reversal delays 32](#_heading=h.1ci93xb)

[Converting collected time delays into corrected values 33](#_heading=h.3whwml4)

[Adding Noise 34](#_heading=h.2bn6wsx)

[Beautify 35](#_heading=h.qsh70q)

# KSIM Functionalities

* Conduct 2D and 3D acoustic simulations
* Use heterogenous media
* Record 2D (plane) or 3D (volume) acoustic data
* Utilize independent acoustic sources
* Construct 2D and 3D acoustic arrays (simulated transducers)
* Perform focused, plane wave, or Hadamard encoded transmits
* Perform multiple transmits in a single simulation
* Display and analyze metrics from recorded images/movies
* Save simulations
* Create and save external figures
* Perform AE time-reversal simulations
* Create rudimentary current sources
* Perform acoustoelectric scans and reconstructions
* Implement lead fields for AE scans
* Export to beautify for full analysis

# GUI Overview

Graphical user interface

Description automatically generated

The GUI is broken up into different sections that all contribute to a single simulation. Each section is highly modifiable. Built in displays and analysis tools are provided, but exportation to separate software is also viable (and usually preferable).

# How to run a full 2D acoustoelectric simulation

The following steps involved are typically as follows:

1. Create kgrid
2. Create medium
3. Create sensors
4. Create transducer
5. Run acoustic simulation
6. Create current sources
7. Run AE simulation
8. View in ksim or beautify

## Creating a simulation grid (kgrid)

The *kgrid* structure in matlab contains all of the variables associated with the spatiotemporal domain of the simulation such has number of dimensions, pixels and time points along with their sizes. Note that this does not include anything to do with the medium properties of each pixel (that is set in the [**medium**](#_heading=h.2et92p0)section).

The eight boxes of this panel design the simulation space. They are**: Nx, Ny, Nz, Nt, dx, dy, dz,** and **dt.** The boxes labeled with “N” dictate the number of pixels (or timepoint) of their dimensions. The boxes labeled with “d” dictate the size of each unit for their associated dimensions. Note that spatial dimensions are in millimeters and the time dimension is in microseconds. For example, an Nx of 100 and dx of 0.1 will give an X dimension of 10mm.

By definition in kwave (and KSIM but association), a 2D simulation will use the X and Y dimensions only. In order to do this, set Nz to 0 when making the grid. For a 2D simulation, X will determine the azimuthal (lateral) dimension, whereas Y represents depth. If Z is included, the simulation becomes 3D where Z is elevational and where Y becomes azimuthal and X becomes depth.

After setting parameters, press **Make Grid** to create or update the *kgrid* variable in the workspace.

The **Viewer** button in this section is not used for making a kgrid. Instead it is a link to open beautify.

When creating a kgrid, make sure to input sane parameters. For example, don’t large dx or dt if you are doing a high frequency simulation, or don’t input a high Nx or Nt if you only need to run a small simulation. Figuring out good values is informed by theory but optimized through practice.

Graphical user interface, application

Description automatically generated

## Creating a medium

Creating a medium consists of 4 parameters: **Density, Sound Speed, Alpha Coefficient, and Alpha Power.** The former two can be applied individually to each pixel/voxel. Moreover, note that density and sound speed relate to Young’s modulus (K) by The latter two alpha parameters must be uniform throughout the entire medium. This is not realistic, as these parameters do vary in actuality, and thus makes up a limitation of the program. Alpha coefficient effects how the pressure is attenuated over propagation. Alpha power effects the dispersive properties of the medium (how the sound speed varies with frequency). Higher alpha coef results in more attenuation, and higher alpha power results in faster dispersion into the different frequencies composing a broadband wave.

A medium can be created in two ways in KSIM. The first way is to use the four boxes provided in the many GUI itself and then pressing **Make Medium.** Doing it this way is simple and fast, but limited the simulation to a uniform medium. To create layered or heterogenous media, used the **Load CT** button. This brings up the load\_ct gui. For more information on using this gui for making 2D or 3D media, check out the [Load CT](#_heading=h.1ksv4uv) section below.

Graphical user interface, diagram, application

Description automatically generated

## Creating sensors

Sensors in ksim represent the pixels/voxels of your grid at which output values will be calculated at during each time point of the simulation. Obviously, the more sensors used and more calculations made, the larger the output variables will be. Thus, careful calculation of sensor location is sometimes necessary to optimize simulation performance on slower computers or in special cases. However, for the most part, 2D simulations are small enough such that using all pixels as sensors is satisfactory and the simplest to run.

To use all pixels as sensors in a 2D simulation, simply check the **all** box and click **Make Sensor.** Also, make sure the **3D Sim** box remains unchecked. A simulation will always provide the pressures at each sensor. Additional parameters can be recorded by checking their specific boxes. However, it should be noted that no unique parameters can be calculated during the sim, i.e. all parameters can be calculated after simulation from the pressure. Therefore, in most cases, these additional parameters will be left unchecked.

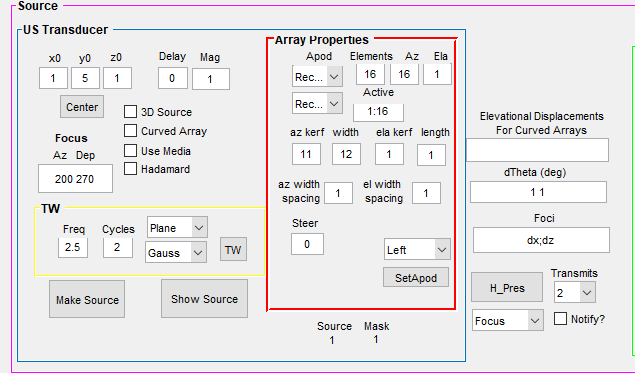
For information on the other boxes, check the [3D simulation](#_heading=h.35nkun2) section.

Graphical user interface, application

Description automatically generated

## Creating an acoustic transducer

Creating an acoustic transducer is the heart of the simulation environment. Generally, k-wave allows for two ways of creating acoustic sources. One is by turning pixel or voxel locations into acoustic emitters. In this case, every chosen point to be an acoustic source is its own independent emitter and is controlled uniquely. Because of this, grouping sources together to form elements that all share the same properties can be problematic, especially in the case of curved 2D arrays. However, despite the coding difficulties that arise, this method allows for a complete control of the acoustic source environment. The other method for creating transducers is using the built in trans function in kwave. This method works well for linear arrays but breaks for curved 2D arrays in the current kwave version. As a result, everything is coded using the former method to keep things consistent between 2D and 3D sims.



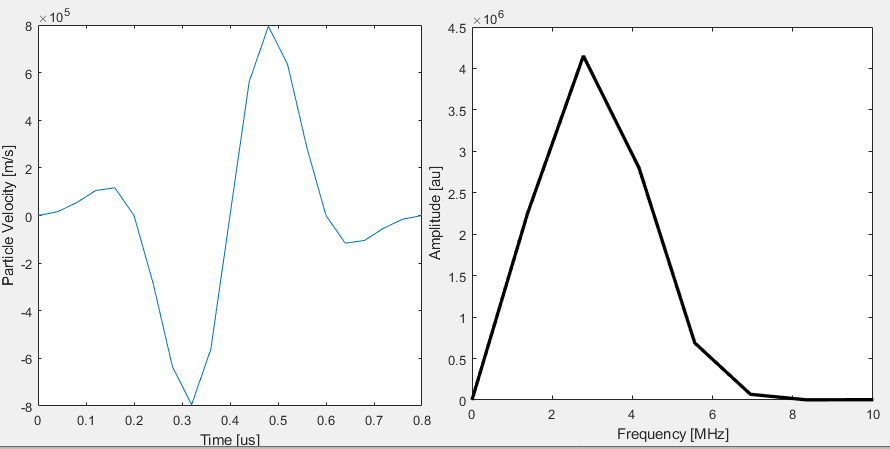
This section is a bit more complicated than the previous sections, but lets dive in.

First, at the top left you can choose where to center the US array. Here, **x0** represents the center of the array in the lateral plane, and **y0** is the center of the depth axis. Its good practice to set y0 off of 1 since the simulation is symmetrical, (using 5 or 10 is good). These get flipped for 3D sims, but for 2D sims that is the case. The **Delay** and **Mag** options give a global delay and a scalar amplification factor that might be useful in some cases when using comparing different settings, but in most cases they can remain at 0 and 1 (there default settings), respectively.

The next four boxes in the US Transducer panel are generally not used in regular 2D sims. The **3D source**  and **Curved Array** boxes are specific to 3D sims, the **Hadamard** box is for 2D Hadamard scans (but we’ll see it will be automatically toggled in cases where we use Hadamard encoding), and the **Use Media** button will consider the individual pixel properties of the media when doing focused beamforming. This button is useful for optimizing focus through aberrating layers, but in general cases we are using general beamforming assuming uniform media. If the media is not uniform, then when this button is unchecked, it considers the top left pixel of the medium to be the homogenous throughout the entire environment.

The **Focus** box is used for choosing the focal point (in pixels) of the beam. This box is only considered when the transmission type is set to *Focus*. The entries for a 2D scan are the lateral point and the depth point. Note that center of the Lateral plane is NOT 0 and that it goes from 1 to Nx.

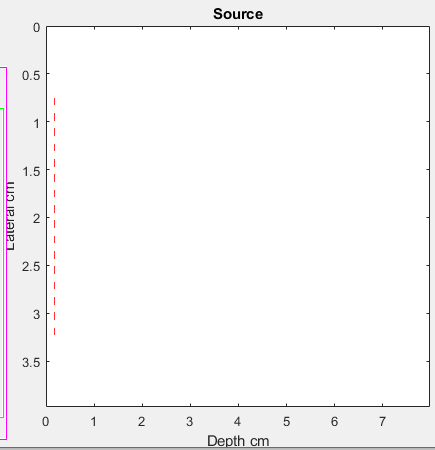
The **TW** panel will design the simple output waveform. Note that a kgrid and a medium most be set before creating a waveform. The **Freq** box indicates the frequency of your waveform, while the **Cycles** box gives how many cycles will occur. The top drop-down menu lets you choose between **Plane Wave** or **Focused Beam** excitation. The differences between these will not appear on the output waveform shape, but instead how the delays for each element are computed. Lastly, the bottom box will allow you to either fit a **rectangular** window or **Gaussian** window to the output waveform. The below picture shows a 2.5MHz 2 cycle wave with a Gaussian window that is presented when pressing the **TW** button. The left window is the waveform in time domain, and the right is its frequency domain representation. Note that **bandwidth** is determined by cycles and window shape and cannot be set directly.



Next is the **Array Properties Panel** where the array is actually designed. For now, we will ignore the apodization features. Starting at the top, we choose the number of **Elements** we want in our array. The two boxes to the right let you choose how that number of elements is split into the **Azimmuthal** and **Elevational** axis. The product of these two must equal the number of elements chosen, so if you have a 96 element array, this can be split into 96x1, 48x2, 16x6, 3x32, etc. In the case of a 2D simulation, however, the Elevational elements will always be 1, therefore, the Azimthal elements will equal the total elements. Next, we can choose which of these deigned elements will be **Active**, i.e. will transmit a wave. In most cases we will want all elements to fire, but in niche cases only a subset is desired. This box input is written in typical MatLab code, so to fire all elements from 1 to 96, enter 1:96. Smaller subsets can be entered such as 33:45, or even sparse groups such as 12:40 60:82 or 2 10 50 90 95.

After designing the number of elements, we can choose how these elements are split or expanded using the **kerf**, **width,** and **width spacing** boxes. In a 2D sim, only the **Az Kerf,** **Az Width** and **Az Width Spacing** boxes are used. All boxes are set to pixel number. The **Kerf** box sets how many pixels there will be between neighboring elements, the **Width** box sets how many pixels will be used per element, and the **Width Spacing** box sets how many pixels between each emitting pixel in a single element. Generally, the **Width Spacing** box is unused in 2D sims but comes in handy with curved 2D arrays in 3D sims. Because the **Width** box will multiply the number of active emitting pixels but its quantity, the number of emitting pixels can vary from the number of elements chosen. Also, care must be taken not to overlap the width of adjacent elements. To prevent this, make sure that the kerf between two elements is large enough to accommodate the chosen pixel width. An example of a 16 element transducer with 7 pixel kerf and 4 pixel width is shown below.

To make the transducer, press the **Make Source** button. To view the transducer, press the **Show Source** button. Note that a **Kgrid** and a **Medium** must be made before making a **Source**.



Because errors can occur due to overlapping elements or by having elements be placed outside of the chosen grid (by having kerf times elements too large or by being off-center), there is a quick feedback mechanism when **Make Source** is pressed. Below the **Array Properties** panel there are **Source** and **Mask** indicators. The **Source** number tells you how many pixels you designed to be emitters, whereas the **Mask** number informs you how many pixels were able to become emitters. If these are not equal the **Mask** number will be printed in red and the simulation will not run.

The **Steer** box in this section is used for plane waves and is the angle in degrees you want to fire the plane wave at.

To the right of the **Array Properties panel** are additional settings for running multiple sims in succession. The top box is for curved 2D arrays only and can be ignored for now. At the bottom is a drop down menu labeled **Transmits** with numbers in it which will set the number of simulations run. This was originally designed for Hadamard scans, so all options right now are even numbers. To the left is another drop down box that will let you choose whether you want to do **Focused, Plane Wave,** or **Hadamard encoded** simulations.

For **Focused** sims, you need to enter the dx and dz for the set of simulations in the format of dx; dz in the box labeled **Foci**. In most cases we change the dx and leave the dz constant. This can be done by setting dz to 0 by omitting it entirely. Additionally, the Focus box in the **US Transducer** panel will indicate the original focal position. The simulations will then vary from -dx\*(number of sims)/2+focus to dx\*(number of sims)/2 + focus.

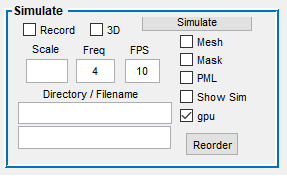
For **Plane Wave** sims, the dTheta in degrees need to be entered in the **dTheta (deg)** box. Also, the Steer box must be set to indicate the central simulation angle. Therefore a Steer of 0, a dTheta of 5 and a Transmit number of 8 will run simulations from -20:5:20 degrees, i.e. -20, -15, -10, -5, 0, 5, 10, 15, 20.

Note that for both Focused and Plane Wave sims, the total number of simulations run will be the number chosen + 1.

For **Hadamard** sims, no additional boxes are necessary, but the number of **Elements** must equal the number of **Transmits**. By altering **Kerf, Width,** and **Width Spacing,** a transducer of smaller element number can be modelled as having more elements, i.e. a 16 element design can function as if it had 96 elements with proper geometry.

## Running and acoustic simulation

To run a simulation, a **Kgrid, Medium, Sensor,** and **Source** must all be defined.



Pressing the **Simulate** button will run the simulation.

The **Mask** and **PML** buttons can be ignored. The **Show Sim** button will display the simulation as it is conducted. The **Mesh** button only works when **Show Sim** is active and it will convert the display into a mesh plot. The **Record** button will save the movie and also requires Show Sim to be active. The Directory of the file to be saved can be set below, along with the filename beneath it. The scale and fps of the recorded video can be set above that. If scale is left empty, it will automatically scale to the relative min and max per frame.

Lastly, the **GPU** box will allow the simulation to be run on the GPU. This can save a tremendous amount of time since graphics processors are designed to handle 2D and 3D matrices. However, running sims on a GPU obviously requires a good graphics card, but also can limit the size of a single simulation due limited GPU RAM. Therefore, GPU simulations are typically smaller, but sometimes simplifications of larger simulations can be made to run them on the GPU as well. A good amount of RAM for a GPU is 6+ B for 2D sims and 12+ for 3D sims.

In the case of a single simulation, the output matrix is represented in the workspace variable ***pressure***. This matrix is 3D for a 2D sim (lateral, depth, time) and 4D for a 3D simulation (depth, lateral, elevational, time).

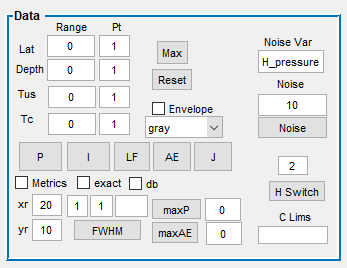
### Multiple Simulations

As alluded to before, a sequence of simulations can be run rather than a single. This is accomplished by using the **H\_Pres** button in the **US Transducer** panel. It is recommended not to use **Record** or **Show Sim** when using the **H\_Pres** button. When multiple sims are run this way, each simulation is run separately and then stored in a variable in the workspace named **H\_pres**. For 2D sims, **H\_pres** will be 4D (lateral, depth, time, sim). Each individual simulation can be run on the GPU (and should be whenever possible). However, **H\_pres** is stored in computer RAM. Therefore, the size of each individual sim is contingent on GPU RAM, but the entire set is contingent on computer RAM. In order to use this function, it is recommended to have at least 50GB of RAM allocated to MatLab for every 16 simulations run. Up to 32 can be run for smaller simulations. A PC with 128GB of Ram would be recommended for sets of up to 32 sims, however.

In order to design the set of simulations to be conducted, follow the instructions in the relevant part of the [*Creating an acoustic transducer*](#_heading=h.3dy6vkm)section.

## Viewing Pressure output in KSIM

KSIM comes with an in-app feature allowing for the visualization of the 2D output matrices. These would be the **pressure, lead fields, current,** and **AE** images which are denoted **P, L, I,** and **AE** in the GUI within the **Data** panel.



The **Max** button sets the fields to the maximum values for the **pressure** variable.

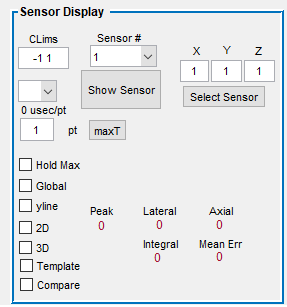
The **Reset** button resets the in-app figures.

The **Envelope** box will force an envelope on the data. Note that this is performed over the Depth axis.

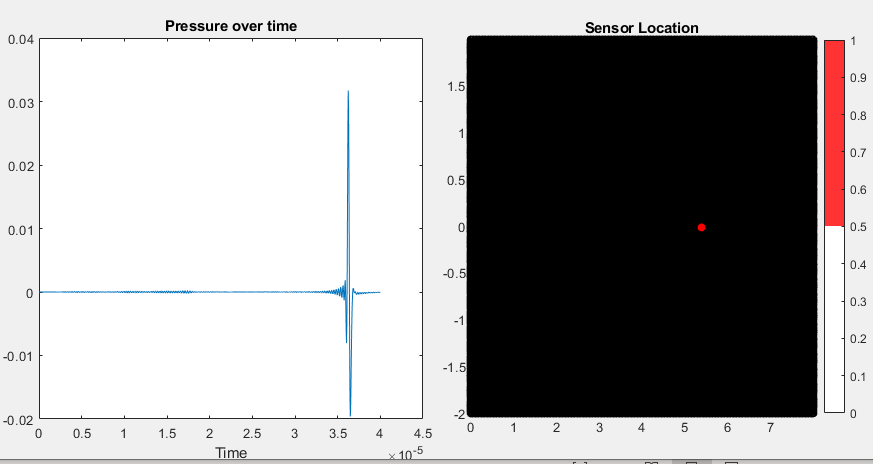
For viewing **pressure,** press the **P** button after setting the axes; this will display the pressure from the lateral and depth ranges set and at the time noted in **Tus**. The variable **Tc** is for current time, and is not available for pressure analysis and is also currently not programmed into the electrical analysis either as only static currents are implemented. All figures in KSIM are displayed with their converted units (such as or mm).

### Viewing a sensor in KSIM

Aside from data images, the sensors themselves can be viewed in KSIM. This is done in the **Sensor Display** panel.



When **Show Sensor** is pressed, the select sensor will be displayed on the bottom right and the pressure over time for that sensor will be plotted to the left of it.



The sensor number drop down menu contains all of the sensors in the grid from 1 (bottom left) to the max which is at the top right. This can be cryptic, so two options were implemented to simplify things. First, if a focused simulation was run, the sensor number will automatically be updated to the center of focus. Second, to choose any sensor by its x, y coordinates, just enter them into the boxes above **Select Sensor** and then press the button to choose that sensor number.

There is also a **MaxT** button that will select the time point at which the select sensor receives its largest value. If **envelope** is checked, then the max value selected will be from the enveloped signal.

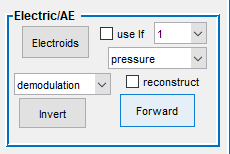
## Creating currents sources

Although KSIM is designed to be the centerpiece of complete acoustoelectric simulations and inverse reconstructions, the current version of KSIM utilizes a simplified version of electrical systems that a built within a separate GUI: [Electroids](#_heading=h.z337ya). A button to open Electroids is in the **Electric/AE** panel.

The output of Electroids will give the current field in variable **I** and any lead fields in variable **L.** See the linked section on Electroids learn how to create these current sources.

## Running an acoustoelectric simulation

After a pressure matrix and current have been created, an acoustoelectric simulation can be conducted. This is performed in the **Electric/AE** panel.



If a lead field is desired to be used in the forward simulation, make sure to check the **Use LF** box. To the right of that box is a drop down menu populated with the number of lead fields that were created. Select the lead field you want to use and press the **Forward** button to run the forward AE simulation. The drop down menu below the lead field menu is allows you to choose either the variable **pressure** or the variable **H\_pres** to run the simulation with. Remember that **H\_pres** is just a set of **pressures**. The resulting variable containing the AE data is named **M.** Check the **reconstruct** box to include an accurate construction of the AE image named **V** in the workspace from the set of **M** AE signals. Note that reconstruct only with **H\_pres** and that it is the convolution of the pressure variables with the current (and LF if selected). The results of this kind of reconstruction are very accurate, but unrealistic of what can be actually performed in practice since in the simulation we have true values of pressure and current for every pixel, in reality we do not. Therefore, real beamformed AE images are generally filled with more artifacts than what is obtained using this simulated reconstruction.

## Analyzing the results

Viewing and analyzing the results of either an acoustic or an acoustoelectric simulation in KSIM can be performed rudimentarily in KSIM. More complete analyses can be performed with other programs such as [Beautify](#_heading=h.qsh70q).

To perform analysis in KSIM, both the **Data** and **Sensor Display** panels can be used. Some analyses that can be performed in KSIM are:

* Line plots through images and FWHM
* Peak locations, amplitudes, and times

Also, figures can be easily exported using the built-in features in the **Save and Auto** panel. Originally, all figures are displayed inside the GUI, so to enable external figures the **Ext Fig** box must be checked. Then enter the figure number of the external figure in the **Fig #** box, the directory and file name in the **Folder** and **File** boxes, and press **Save** to save the figure using the Export Figure function. This function is currently set to save the figure transparently.

To create line plots for FWHM or peak analysis as shown below, just click on a displayed image where you want to generate the line plots. This will create line plots through both axes as shown below. Additionally, the peak, lateral fwhm, and axial fwhm values will be plotted in red in the **Sensor Display** panel. How many points away from the clicked point that are plotted in each line plot can be controlled using the **xr** and **yr** boxes in the **Data** panel. These units are in mm.

Graphical user interface

Description automatically generated with low confidence

In order to choose the exact point for the center of the two line plots rather than estimating with the mouse cursor, the exact coordinates (as displayed in the image (0mm center for lateral)) can be placed in the boxes next to **xr** and **yr** in the **Data** panel (lateral pt, depth pt). Then check the **Exact** box above and click anywhere on the image to plot through the chosen coordinates. This can be very useful when comparing the exact same point between multiple simulations.

Any image can also be plot in decibel scale but checking the **dB** box in the **Data** panel. Make sure that the clims are set in order to view the image (something like -18 0 is a good start).

# Running a full 3D acoustoelectric Simulation

# Load CT

Load\_CT is a small GUI designed to enable simple implementation of 2D and 3D images for the generation of heterogenous or layered media in KSIM. The **Load CT** button can be found in the **Medium** panel.

Graphical user interface

Description automatically generated

In general, Load\_CT allows you to load an image or DICOM file, crop it, resize it, and assign properties to 3 different brightness values of the images.

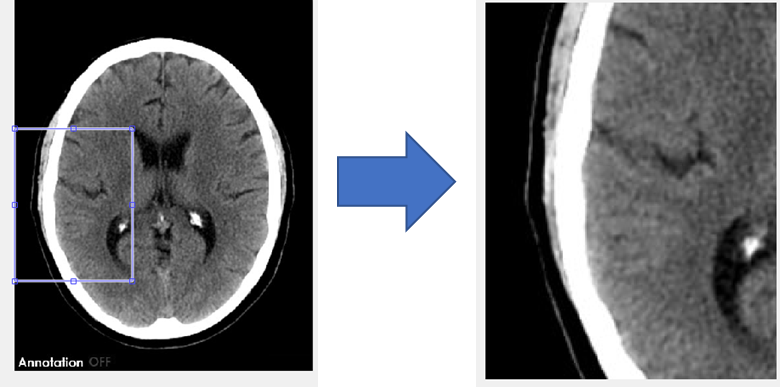
### Creating a 2D Medium

Creating 2D media begins by pressing **Load** at the top left. This will bring up the option of loading an image file from your computer. It will be converted to black and white.

Next, press the **Show Original** button to display the loaded image inside the GUI. Below is the image of a loaded CT scan of the brain and skull.



Generally, for an US simulation, you only want part of the image, and certainly without the words at the bottom, so next you can crop the image by pressing **Crop.** This will bring up an expandable box on the image that you can use to select your ROI. After choosing the ROI, right click on the box and select Crop Image. Then press the **Show Cropped** button to display the new cropped image.



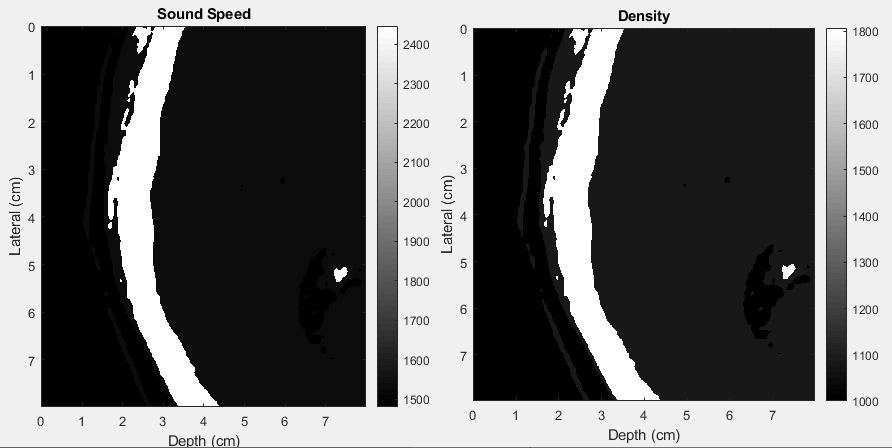
On the right of the display buttons are options for rotating or flipping the image. The **Fliplr** button will flip the image in along the Y axis and **Flipud** will flip it along the x axis. The image can be rotated by entering a number in degrees into the box next to rotate and pressing the **Rotate** button. Importantly, these modifications will be applied either to the original or the cropped image depending on whether the **Cropped** box is checked.

Values in Load\_CT are restricted to three layers and four properties. The layers by default are labeled **Water, Brain,** and **Skull,** although the content of these layers is arbitrary. They allow for separate property values based on pixel brightness which is applied in the **Thresholds column**. Two values for each property needs to be entered into the **Thresholds** box. These values define which pixels are assigned which layer. For entered thresholds *a b,* where *a* and *b* are >0 and <1 and where *b > a,* the property value for **Water** will be assigned to pixels between 0 and *a,* **Brain** will bebetween *a* and *b*, **Skull** will be between *b* and 1.

While there are four properties that can be set in Load\_CT, only **Sound Speed, Density,** and **Alpha Coefficient** can be set on a pixel to pixel basis. **Alpha Power,** on the other hand, is set uniformly throughout the medium. For reference, **Alpha** determines the amount of absorption attenuation within that layer, where lower values give lower attenuation, and **Power** effects the amount of dispersion in the simulation, i.e. how the sound speed varies with frequency in a broadband acoustic transmission.

After assigning values to each layer, you need to make the **medium** size the same as the **kgrid** size. This is accomplished in Load\_CT by entering the number of pixels you want in each dimension into the **X** and **Y** boxes at the top of the GUI and then pressing the **Interpolate** button. This will try to interpolate the image to the selected number of pixels. However, this can sometimes be off by a pixel or two, so feedback as to the sizes of the dimensions is displayed at the bottom of the GUI. For example, sometimes when trying to make a 400x400 cropped image, 401 may need to be entered into one of the dimension boxes. Always trust the feedback at the bottom of the GUI rather than the number entered at the top for interpolation.

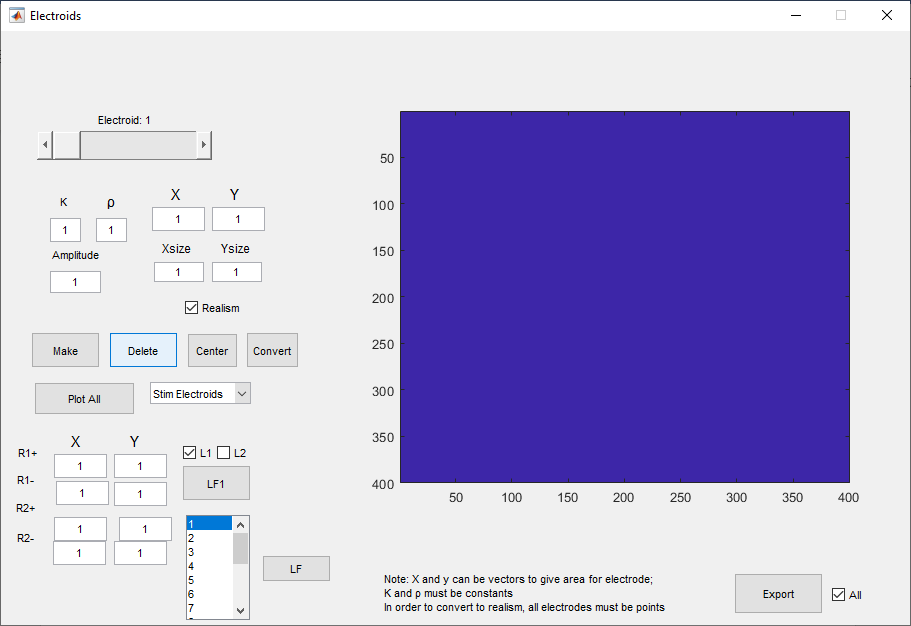
Lastly, in order to apply these values to the **Medium** variable in the workspace, press the **Generate Medium** button in **Load\_CT.** Images of the exported medium’s layer properties can then be viewed in **KSIM** by pressing the **Plot** button in the **Medium** panel.



### Creating a 3D Medium

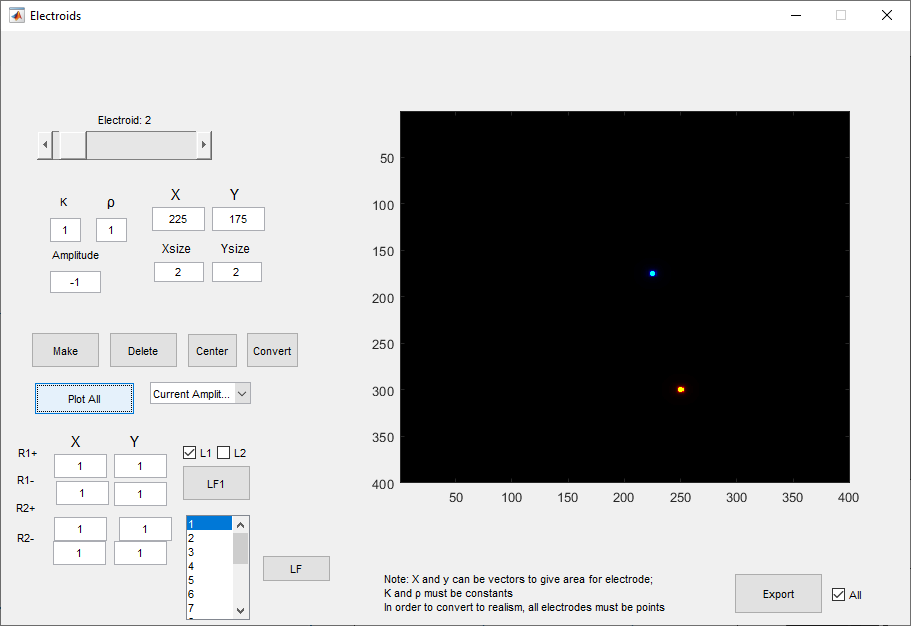
# Electroids

Electroids is a GUI for basic electric field (or current source) generation to be used with KSIM for ascoustoelectric simulations. Many assumptions are made in the calculations in this GUI, thereby limiting its effectiveness, accuracy, and applicability. One such assumption is that electric field calculations can adequately represent current fields, at least in close vicinity to the charge source itself. Nevertheless, the current field outputs can be used accurately in many basic acoustoelectric simulations.

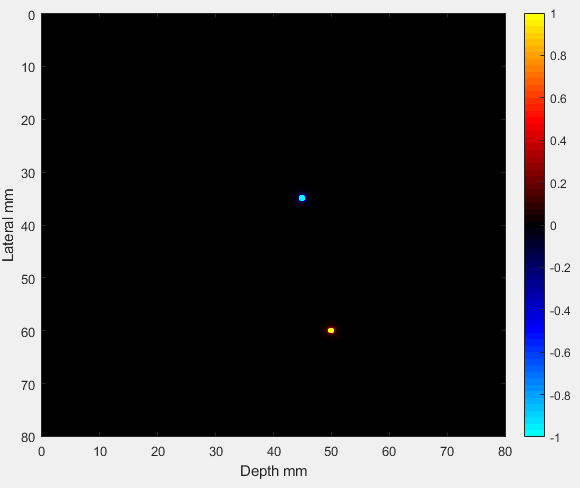


A major limitation of Electroids is that it is limited to 2D static currents. However, creating the current sources is simple and straightforward. Although there are many buttons on the GUI, only a few are needs to create a current field.

The bar at the top is the current source you are going to place. This can be slid between 1 and 20. Start with electrode 1. The **X** and **Y** boxes indicate the electrode location. The **Amplitude** indicates the relative amplitude of the current source; this can be positive or negative. The **Xsize** and **Ysize** boxes indicates how many pixels in each direction the current sources will be. After settings these boxes to your desired value press the **Make** button. You can then press the **Plot All** button to visualize your electrode. Other options for viewing your current fields are present in the drop down box to the right of the Plot All button. To make a second electrode, use the slider at the top to go to Electroid 2 and then repeat the above steps for creating the electrode. This can be repeated up to 20 times.



Even if you do not intend to use a lead field, one still needs to be created. To do this, select the lead field number you want in the menu at the bottom (start with 1) and then press the **LF** button to the right of it. This will bring up two sets of crosshairs. Click on the image in the GUI to place the lead field pair where you want. After creating the lead fields press the **Export** button to create the **I** variable in the workspace which is used as the current field in the forward AE simulation. The **I** variable can be viewed anytime by pressing the **I** button in the **Data** panel in **Ksim.**



# Automated functions

Running repeated simulations or generating a particular setup you want to use frequently can be tedious. Therefore, customizable automatic functions can be built by each user. Custom auto functions are not saved on official KSIM releases, so its important to save them to your own local machines. In order to use this functionality, search for “autofill\_Callback” in ksim.m. This will bring you to where you can code your custom automatic function. Look at the examples that are already there to get an idea of how to add your own. Additionally, you will need to add your function into the drop menu in the KSIM GUI **Save and Auto** panel. To do this:

* Type “guide ksim” into matlab
* Double click on dropdown menu left of the **Auto** button which will bring up the widget’s settings
* Navigate to the **String** option and add the name of your function somewhere in the list
* Close the widget and save the GUI

# Aberration Correction

A lot of the time we want to fire US through media that is not homogenous. Instead, the media can consist of scattered heterogeneities or be composed of various layers of homogenous mediums. See the [Load\_CT](#_heading=h.1ksv4uv) section to see how to easily load these media. Sometimes it is useful to see what these heterogeneities will do to propagating US waves; other times we want to determine ways to correct for these heterogeneities to return structure to the US wave as if it were traveling through a homogenous medium when the waves reach a particular depth.

## Analytical corrections

The easiest way to correct for aberrations in the media when performing focused ultrasound is through analytical corrections. This method takes into account the sound speed values of the pixels/voxels in the actual simulation to perform the corrections. While this method is very powerful, it is also very limited. Due to its reliance on property values, said values must be accurate for accurate corrections. In a controlled simulation this is trivial to accomplish since the values are chosen, but in experimental or practical applications these values must be estimated from images and nominal values.

To perform this form of correction in KSIM, check the **Use Media** box in the **Source** panel when making a focused US source.

Graphical user interface, application

Description automatically generated

The image on the left shows centered uncorrected delays; the right shows the analytically corrected delays for the below image. Below each set of delays is the envelope pressure field produced at the desired focal point through the media shown.

Graphical user interface

Description automatically generated with low confidence

It can be seen that aside from the shape difference in the delay profiles, the max delay is also higher for the corrected set. This is because the increased sound speed makes it the sound waves focus further away from the origin, thus requiring sharper focusing to accommodate. This is typical of transcranial US corrections.

## Time Reversal

The other approach to correcting for varied media is by leveraging data from specific transmits. One simple way to accomplish this, and the way implemented in KSIM, is through *time reversal.* This technique gets its name from pulse echo US due to the fact that a wave traveling one way through a medium would take just as long as if it were going the other way. Therefore, if a wave bounces off of a point target at some distance from the transducer and reflects back to the transducer, the variable delays that each element on the transducer receives that echo can then be used, in reverse order, to focus to the point that caused the reflection.

For AEI, or in KSIM using an acoustic detector, this process is even further simplified. Rather than having an acoustic wave reflect back to the US transducer, these methods use the unidirectional time-of-flight from each element firing a diverging wave one at a time from that element to the detector. The upside to this is the simplicity, the downside is the duration and pressure. Since this method uses sequential single element firings, the pressure is limited and the time to acquire all elements corrected delays is dependent on the number of elements in the transducer. This is opposed to the pulse-echo method where all elements can be fire simultaneously to increase output pressure and that all delays are calculated from a single echo.

Performing this in KSIM uses the **Time Reversal** panel located inside the **Source** panel.

Graphical user interface, application, Word

Description automatically generated

### Collecting time-reversal delays

In order to run a TR scan to collect the delays to perform the corrections the following steps need to be performed.

1. Choose a save name for the variable to be saved as
2. Choose whether u want to use a current target (AE), a pressure target (P) or both by highlighting both.
   1. P is selected, make sure to enter the sensor number below.
   2. If AE is selected, make sure a current field **I** variable is already created.
3. Enter the elements of the transducer you want to run TR for.
4. Press **Run TR**
   1. This will run a series of simulations before producing the M3 and the T\_AE or T\_P variables saving the data
   2. **M3** contains the raw AE or pressure data from each element.
   3. **T\_P** or **T\_AE** contains the calculated delays for each element

### Converting collected time delays into corrected values

After collecting M3 and T\_AE or T\_P:

1. Make sure that the transmit type is set to **Plane.**
   1. This might seem counter-intuitive, but it ensures that the TR delays are applied on an unfocused source.
2. Press the **Make Source** button.
3. Check the **All** box in the **Time Reversal** panel.
4. Highlight T\_P or T\_AE, for whichever variable you are using.
5. Press the **TR2** button to create a variable TR2 in the workspace
   1. This variable can be viewed by pressing the **Plot TR** button
6. Press the **Time Reversal** button to apply the new delays onto your existing delays
7. Press **Simulate** to run sim with corrected delays

A picture containing chart

Description automatically generated

# Adding Noise

Noise can be added to any variable in KSIM, however, it is typically added to the output pressure or AE variables. To accomplish this, use the noise section of the **Data** panel.

Graphical user interface

Description automatically generated

Although this is one box, it takes 3 inputs: The number rows, columns and SNR of your noise. Match this to whatever variable you want to add noise to, in this case we’ll add noise to a single AE signal of size 1x1401.

1. Input values of rows, columns, and SNR into the Noise box.
2. Press **Noise** button.
   1. This will create n variable of the chosen size.
   2. This variable is normalize between -1, 1
3. Multiply the **n** variable by either the peak, RMS, or some other variable of your desired variable to add noise to depending on how you want to calculate SNR.
4. Add **n** onto your chosen variable

Below is an example of this process with the noise being generated, modified to the AE signal peak amplitude, and then added to the signal.

Graphical user interface

Description automatically generated

# Beautify