PICKGUI manual

Joseph MacGregor, UTIG (joemac@ig.utexas.edu)
Last updated: 20 June 2014

This manual explains the operation of the MATLAB functions developed to pick layers (i.e., contiguous reflections) in data collected by various airborne ice-penetrating radar systems. There are four primary functions:

1. `radblockproc` pre-processes the radar data into a format that is both consistent and of a manageable size for later use by `pickgui`.
2. `pickgui` is an interactive GUI for picking layers.
3. `mergegui` is an interactive GUI for merging picked layers produced by `pickgui` using elevation-corrected radargrams.
4. `fencegui` is a pair of interactive 3-D and 2-D GUIs for matching picked layers using fence diagrams composed of elevation-corrected radargrams.

There are additional sub-functions (`intersecti`, `ll2ps`, `smooth_lowess`, `topocorr`) that must be available within the user's path to ensure correct operation of the above functions. Their brief capabilities are either called repeatedly by multiple functions or, more likely, they were developed for other projects and I wish to maintain cross-functionality. Several wrappers exist that are convenient scripts for calling pre- or post-processing functions like `radblockproc` across a directory structure.

All functions have help documentation associated with them that can be accessed either by opening the file in Editor or using `help` in the command line. This inline documentation is presented in the standard MATLAB format and is meant to be exhaustive for all functions except the GUIs, whose operation is documented here.

These functions were developed primarily using MATLAB R2012a–R2014a, but are functional in R2011b and should also be compatible with later versions of MATLAB. The Mapping Toolbox is recommended for `radblockproc`. Multi-core workstations and a license for the Distributed/Parallel Computing Toolbox are also strongly recommended once you reach the production stage.

The latest versions of all of these files and this document are available in a shared Dropbox folder called "pickgui". Email me for access to it. Although several functions and scripts other than the above-mentioned currently reside in the Dropbox folder, they are mostly designed for tasks specific to The University of Kansas data and are likely of limited interest to most other potential users. A similarly up-to-date version of what's in the Dropbox folder is available through Github, which enables version control:

https://github.com/joemacgregor/pickgui

1. radblockproc

Prior to layer picking, the radar must be pre-processed into a format easily interpreted by `pickgui`. This task is accomplished by `radblockproc`, which converts the radar data into fields within a uniform structure named `block`. The horizontal phase gradient from single-channel unfocused data is added to `block` separately by `phaseinterp`. Layer slope may also be added using `aresp`, which in practice functions similarly to `phaseinterp` but can be applied to any radar data, not just those for which complex waveforms are available. See `help aresp` for more details regarding that function and related sub-functions.

At UTIG, the CReSIS radar data are stored in `melt.ig.utexas.edu:/disk/big2/icebridge/`, and that is where calls to pre-processing functions `radblockproc` are best performed. The outputted data blocks can then be transferred to Windows workstations in 2.116. To run MATLAB on melt, ssh into melt with your UTIG credentials (e.g., `ssh joemac@melt.ig.utexas.edu`), change to the directory containing the MATLAB scripts/functions, e.g., `cd /disk/big2/icebridge/data/`. If necessary (and it generally will be), update the MATLAB files from the Dropbox folder. Enter `matlab` in the command line to run the latest version of MATLAB without a graphical interface. Commands that do not produce a graphical output can then be run, e.g., `radblockproc`.

The full command-line function call for `radblockproc` is:

```
radblockproc(dir_in, file_in, file_block, num_file_block, num_overlap,
lat_std, do_norm, dir_out)
```

`dir_in` is the directory string where the data are stored relative to the current working directory. If radblockproc is being called on a Unix or Mac, it must end in `/`, because it is a directory. Similarly, on Windows, it must end in `\`.

`file_in` is the name string of the files to be converted into a data block. `file_in` should generally contain the wildcard `*`, likely as a suffix, because the data should include more than one file per block.

`file_block` is the name string for the outputted data blocks. A recommended suffix to this name is `block#`, where `#` is the value of `num_overlap`. A wildcard is not necessary as the filename will have the added suffix `_#`, where `#` is the number of the block. It is further recommended that `file_block` be similar to radar data file names (not necessarily the same `file_in` because of the wild card) so that the blocks formed from original data files can be tracked easily.

`num_file_block` is the number of data files per block. This positive integer should be experimented with so that the resulting blocks are not too large, so as to speed up layer picking in `pickgui`. For depth sounder data, we have found that aiming for a block length of ~50 km strikes a good balance.

`num_overlap` is the number of files that will overlap with the previous block. This positive integer should be less than `num_file_block`, but can be the same in the case of a single block.

`lat_std` is the standard parallel (latitude) from which the polar stereographic coordinates of the transect will be calculated. Values in the southern hemisphere should be negative. For Greenland, `lat_std` should likely be 70, and for Antarctica it should be –71. In Greenland, if available, the EPSG:3413 projection will be used instead.

`do_norm` is a logical scalar (`true/false` or `1/0`) that determines whether the radar data are normalized by their maximum amplitude. For images, rather than natively collected data, this switch should be set to false.

`dir_out` is the directory string for the outputted data blocks.

An example call to `radblockproc` is:

```
radblockproc('data/', '*', '20110502_block15', 15, 2, 70, true,
'block/')
```

This call takes all the `.mat` files (presumably radar data) in the directory `data/`, makes blocks that are 15 files long, with 2 files worth of overlap, produces polar stereographic x/y coordinates that assume a standard parallel of 70º (i.e., Greenland), normalizes the radar data by its maximum amplitude and then saves the results with a base name of `20110502_block15` in `block/`.

`radblockproc` first checks to make sure that each argument is inputted correctly. The function call is preserved in the saved block for future reference. The number of blocks is determined, and then each block is looped through. As files are loaded, their names are outputted in the command window. Once all the files for a given block are loaded, they are merged, filtered, and then saved. These steps are displayed in the command line and repeated for each block. Once the function processing is done, `DONE with this call to radblockproc.` is displayed.

2. **`pickgui`**

To start the GUI, either type `pickgui` in the command line or run it from the Editor after having opened it. If you start it from the command line, it must either be in your path or you must be in the directory where `pickgui.m` is located, which is standard MATLAB behavior.

The GUI was designed on Mac OSX 10.7–10.9 using a 24–27" monitor. The appearance of the GUI differs slightly between Mac and Windows, but the behavior is the same. Depending on the system, `pickgui` will change the default display of figures to ensure that its size is maximized within the Figure Toolbar, using `set(0, 'DefaultFigureWindowStyle', 'docked')`. However, it cannot adjust the location of the Figure Toolbar, which may appear in a docked form within a single MATLAB window. Undock the Figure Toolbar to maximize the GUI's size. To restore normal figure display behavior in MATLAB, enter `set(0,` `'DefaultFigureWindowStyle', 'default')` in the command line.

If the Parallel/Distributed Computing Toolbox is available, then before the GUI is drawn, the workstation's available cores will be located and initialized. This process takes a few seconds.

2.1. GUI layout

The radargram consumes most of the GUI, because this panel is the primary window for viewing the radar data. Dimensional traveltimes and along-transect distances are displayed, but note that most calculations are performed in "index" space, i.e., they are ignore the dimensionality of the data matrix. This dichotomy presents special challenges within the sub-functions themselves but should be transparent to the user.

The slider bars on all sides of the radargram except the top control the maximum and minimum values of the transect distance *x*-axis in kilometers (km, bottom), the traveltime *y*-axis in microseconds (μs, left), and the color scale in decibels (dB, right). If the phase difference is being displayed, the color decibel scale switches to degrees (rad). Similar behavior occurs for ARESP-inferred layer slopes. If the minimum traveltime slider bar is set to a value larger than that of the current maximum traveltime, the slider bar will return itself to the previous value, and vice versa.

Each slider bar can reset be to its default value by pressing the nearest *reset* button, but note that these values are defined in the preamble for `pickgui.m` and are not editable once the GUI has been started except as is done automatically when the radar data are loaded. There is also a *reset x/y* button in the top right corner to reset both *x*- and *y*-axis limits simultaneously. Each axis range can also be fixed using the appropriate checkbox labeled *fix*. This behavior is useful for panning along an axis using the slider bar while keeping its range fixed. Note that the zoom and pan buttons in the Figure toolbar and the arrow keys can also be used to navigate. Finally, the horizontal axis is also divided into "chunks" that can be cycled through using the dropdown menu in the top left corner. The length of each chunk is set by L_{chunk}, whose units are in kilometers (km).

For the color scale, there are two *fix* buttons in the lower right-hand corner, *1* and *2*. The left checkbox (*1*) behaves in the same way as the other *fix* checkboxes. The right checkbox (*2*) adjusts the color scale relative to only the data within the current display window. Specifically, it adjusts the color scale automatically so that it is always centered on the mean value of the currently displayed data and its total range is four standard deviations of those data.

The color scale can also be switched between `bone` (blue-tinged gray) and `jet` (rainbow) using an unlabeled dropdown menu in the top right corner. Grid lines can be displayed using the *Grid* check box in the top right corner.

To switch between data-viewing modes, use the radio buttons in the top left corner. The default is "twtt" (undadjusted two-way traveltime), which is the standard mode for viewing radargrams. Upon loading `pickgui`, only the "twtt" viewing mode is available. Once data are loaded, other viewing modes will be made available depending on the data that are loaded. If the surface is available, then the "~depth" viewing will show the data in terms of traveltime but referenced so that the surface is flat. This view is useful for shallow radar data. "phase" shows the filtered horizontal phase difference that, if available, can be used to predict the internal stratigraphy; the same goes for "ARESP". "flat" shows the data amplitude in the flattened projection, and is only available once the data been flattened.

Above the data-viewing modes, a text box records the filename of the current (loaded) data block. A longer text box in the top right corner displays the current status of a given operation. In general, if the status ends in "…", the current operation is not yet complete.

All operations that manipulate the data are organized above the radargram. Blue signifies loading data operations; magenta signifies layer operations; red signifies deletion or testing; green signifies saving.

There are several user-editable variables displayed in the GUI. To adjust them, enter a new value in the appropriate units and then press "return". The value may then be rounded or adjusted as necessary internally. For example, variables whose units are indices must be positive integers, and they will be edited to be so and corrected in the displayed box.

Check boxes to the right of most operation buttons turn on/off display of that operation's output. If the necessary operation has not been performed, its display cannot be turned on.

When displaying the GUI, nearly all top-level menu items (the pull-down menus across the top of the screen) normally associated with a MATLAB figure should disappear. This behavior is intentional, because of an unresolved issue whereas continued use of the GUI results in display of repeated top-level menus and destabilizes MATLAB.

Below the primary functions of `pickgui` are explained. For each sub-section header, a simple description of the operation is given, followed by the button name in the GUI and the key sub-functions in `pickgui.m` that it calls.

For most functions, a hotkey is available (Table 1). To ensure that hotkeys remain accessible after any GUI interaction, it is sometimes necessary to "refresh" the GUI's focus, which means that the buttons will briefly flash as grayed out.

Table 1. `pickgui` hotkeys

Key	**Action**
1	Toggle display of reference layers
2	Toggle display of phase-predicted layers
3	Toggle display of ARESP-predicted layers
4	Toggle display of manually predicted layers
5	Toggle display of picks
6	Toggle display of smoothed picks
7	Toggle display of surface/bed picks
a	Adjust picked layer

b	Toggle automatic color scale adjustment
c	Predict layers using horizontal phase difference (if available)
d	Delete layer
e	Reset *x*-/*y*-axes
f	Flatten data using predicted layers
g	Toggle grid lines
h	Match current block's picks with overlapping block's reference picks
i	Load picks for this block
j	Predict layers using ARESP (if available)
k	Choose phase- or ARESP-predicted layers to keep for flattening
l	Load radar data block
m	Merge two layers
n	Select next layer
o	Focus on the current layer
p	Semi-automatically pick layers
q	Display separate static figure replicating current display
r	Load reference picks from overlapping block
s	Manually pick layer
t	Trim *y*-axis of data
u	Manually pick predicted layers for flattening
v	Save picks
w	Toggle display's color map
x	Toggle fixing the distance range
y	Toggle fixing the traveltime range
z	Assign a layer to the surface or the bed
←,↑→,↓	Pan display window 25% of current range in arrow direction
spacebar	Toggle display mode
/	Toggle flattening polynomial order (2nd or 3rd)
\	Toggle layers used for flattening (predicted or picked)
Left-click	Select nearest layer
Mouse wheel	Zoom in or out

2.2. Load a radar data block (*Load data*; `load_data`)

The first step is obviously to load some radar data that has been pre-processed by `radblockproc` by pressing *Load data*. An error will be returned if the selected .mat file does not contain a structure named `block`. For reference, the data block's filename, minus the .mat extension, is displayed in the top left text box and is preserved until a new data file is loaded. The data are horizontally averaged and displayed at intervals equal to $N_{decimate}$ indices. CReSIS data in their original format can also be loaded and picked. If a radar block with the same name as the current block but whose block "number" (the last two digits in the filename) is incremented by one is present in the same directory as the current block, then the user will be prompted as to whether they simply want to load the next block in the transect.

2.3. Trim off excess data (*Trim y*; `trim_y`)

The traveltime range of the data will typically exceed that which is of interest to the user, i.e., data earlier than the earliest surface reflection and/or later than the latest bed reflection. Narrow the *y*-axis appropriately and then press *Trim y* to remove these data, which will trim the appropriate matrices and vectors. This operation will speed up subsequent operations (e.g., *Flatten*) that perform calculations upon the entire data matrix. *Trim y* can be performed multiple times, but existing picks cannot be loaded (*Load picks*) if *Trim y* has already been performed.

2.4. Load existing picks for the current block (*Load picks*; `load_pk`)

If the current data block has already been picked, those picks can be reloaded by pressing *Load picks* and then selecting the appropriate file. If the filename of the data block has changed since it was picked (which is strongly contraindicated), then the picks file cannot be loaded. Because picks can only be saved if all requisite operations have been performed (flattening, picking smoothing, matching), reloading the picks is not instantaneous because the data have to be flattened again. Do not attempt to load picks if any operations have already been performed on the data block, e.g., *Trim y*. In other words, run *Load picks* after having just run *Load data* only. If the picks file is found in the anticipated file and path name within the `melt/icebridge/data/` relative directory structure, then they will be loaded without user input.

2.5. Load reference picks from an overlapping block (*Load ref.*; `load_ref`)

If the data block being picked is not the first block in a given transect, then the picks from an overlapping block must be loaded prior to matching (and hence saving). These are called the reference picks, and their filename is the related to the filename of the overlapping block, not the block being displayed currently. The default directory that this function will search within is the same as that for *Load picks* and *Save*. If a filename following the expected pattern is found in that directory, then user will be prompted as to whether they simply want to load that file. The reference picks are shown in yellow, and they can be from the left or right side of the overlap (earlier or later in the transect/sortie, respectively).

2.7. Predict internal stratigraphy using phase (*Phase*; `track_phase`, `prop_phase`)

This operation predicts the internal stratigraphy using the filtered horizontal gradient of the phase only. This quantity represents the rate of change of the range to a reflector from the aircraft. By integrating that change along-track, predicted layers can be traced. This tool only works for data that have had their phase filtered and processed by `phaseinterp`. Hence, it is presently only available for newer CReSIS data (2006–present).

All that is required from the user is a trace from which to start the phase propagation, which is chosen manually after *Track phase* is pressed. Choose a trace with a high signal-to-noise ratio, but more importantly tend towards picking the thickest ice in the block, and pick a point right above the bed reflection from that thickest ice. Precision is not critical. Once chosen, the phase will be propagated at a fixed vertical interval (`int_track`, set to 10 samples) between the user's pick and the surface reflection of that trace. This calculation takes a few seconds.

The resulting predicted stratigraphy is displayed a set of blue lines, with magenta dots at the originating points along the user-chosen vertical trace. Deep in the ice, where the phase tends to be noisier and less reliable, the predicted layers will sometimes intersect and then follow the bed.

The center frequency of the radar system is the fundamental control on the shape of the phase-predicted internal stratigraphy. It is a user-editable variable (f_{center}), not because the radar's center frequency varies significantly within a given transect (it should not, although it does vary between field seasons) but because our interest is in predicting the stratigraphy using the phase gradient only, which is not necessarily an accurate physical representation of the radar system's behavior and processing. A larger center frequency will result in a smaller predicted range rate of change. In other words, the larger the value of f_{center}, the flatter the predicted internal stratigraphy will be. The units of f_{center} as displayed in the GUI are megahertz (MHz). The slope of the phase-predicted layers can be adjusted by changing f_{center} after having already run *Track phase*.

2.8. Pick phase-predicted layers to keep (*keep*; `pk_keep_phase`)

Having predicted the layers using *Phase*, it is now possible to choose the layers that best represent the observed internal stratigraphy, for use in a subsequent data-flattening operation. After pressing the *keep* button to the right of *Phase*, a cursor appears. If the phase-predicted layers are available but not displayed, they will be re-displayed automatically. Left-click anywhere on the radargram to initialize the choosing of layers to keep. This behavior is similar to all other operations where something must be clicked within the data window.

Click on a predicted layer that represents the internal stratigraphy in its vicinity, even if only approximately or not across the whole of the transect. It is important to understand that this predicted layer is not in any way a final pick of that layer, which will come later. The closest layer to the user's pick will change color to white. If this layer is indeed representative of the internal stratigraphy, then hit "return" to keep it, and it will be bolded. If not, then hit any other key and it will be returned to its previous color and not kept. If a layer that was previously kept is no longer favored, then it can be deleted by entering "d" when the cursor is closest to that layer. Once done picking layers to keep, enter "q". This operation can be revisited by again pressing *keep*. It is typically necessary to toggle the phase-predicted layers on and off to see which best match the observed internal stratigraphy and use *keep* iteratively.

Because the data-flattening operation uses a 2nd or 3rd-order polynomial to fit the predicted or observed layers to the layers at the reference position, it is strongly desirable to pick at least three layers to keep, so that this polynomial can be constrained satisfactorily. The surface reflection is also used, as it is an isochrone with zero age. If three good phase-predicted layers are not available, then it is necessary to produce some ARESP-predicted or manually picked to constrain the flattening procedure.

2.9 Predict internal stratigraphy using ARESP (*ARESP*, *keep*; `track_aresp`, `prop_aresp, pk_keep_aresp`)

Predicting the stratigraphy using ARESP (automated radio-echo sounding processing) layer slopes functions in the exact same way as for the horizontal phase gradient, except that the ARESP layer slopes do not depend on the center frequency used. Use the *ARESP* button and the *keep* button next to it in the same manner as described above.

2.10. Track layers manually (*Manual.*; `track_man`)

If the horizontal phase gradient or ARESP are either unavailable or perform poorly in a portion of the radargram, it is valuable to track one or more layers manually prior to data flattening. These manually traced layers need only to capture the general structure of the observed stratigraphy at a given depth, and they do not need to be detailed tracings of the observed stratigraphy.

After pressing *Manual*, a cursor will be displayed and the user should now trace a single layer using the mouse, clicking as appropriate and preferably moving consistently from one end of the block to the other (direction is unimportant but left to right is simplest). Once done picking the manual layer, press "return" and the spline interpolation of the user-traced picks will be displayed. Additional manual layers can then be picked. To delete a manual layer, instead of clicking the left mouse button, highlight the manual layer to be deleted and press "d". The nearest manual layer to the cursor will then be deleted. To undo the previous click, press "u". To cancel or end manual picking, press "q".

2.11. Flatten data using layers (*Flatten*; `flatten`)

"Flattening" the amplitudes of the radar data relative to predicted and traced layers is a powerful tool for better tracing of the observed stratigraphy. This operation is complex and can take a while. It is parallelized, so if multiple cores are available on the workstation, then these cores will be used. Flattening is a two-step procedure. First, polynomial fits are calculated for the relationship between the vertical indices of the predicted layers at each trace and their vertical indices at a reference trace. For phase- or ARESP-tracked layers, this trace will be that from which the layer slopes were originally propagated. Second, the data amplitudes will be interpolated using those polynomials, effectively "flattening" them relative to the reference trace. In other words, if we assume that all visible layers are isochronal, then the data amplitudes are projected onto a space where the vertical scale is still non-linear in age but is now implicitly horizontally uniform across the data block.

The number of full-transect layers used to flatten the data, whether phase- or ARESP-tracked, manually picked or actual traced layers, must be at least three. *Flatten* uses these layers to flatten the radargram. Note that once layers have been traced in an earlier flattened projection, those layers will be used instead by default. These layers are generally more reliable than those predicted by the earlier methods, which initialize the process.

Press *Flatten* to begin this operation, which can take several seconds or more. After the flattening calculations are completed, the radargram view is switched to the flattened projection, decimated by $N_{flat\,mean}$ indices.

2.12. Semi-automatically pick layers (*semi-automatically*, *Delete*, *Choose*, *Focus*, *Adjust*, *Merge*, *last*, *next*; `pk_auto`, `prop_pk`, `pk_select`, `pk_select_gui`, `pk_del`, `pk_focus`, `pk_adj`, `pk_fix`, `pk_merge`, `pk_last`, `pk_next`, `pk_smooth`)

This operation semi-automatically picks the layers that will ultimately be saved. They can then be edited in-line or using various functions. After pressing *semi-automatically*, a cursor will appear. Due to an unresolved issue, click the left mouse button once to initialize the options shown in the status box.

Left-click to pick a layer that is an amplitude high (i.e., white in the bone color map). Only highs should be picked in data that are represented using a decibel scale. The layer will then be propagated horizontally and displayed in pink from the user's pick by searching progressively for the high/low in each adjacent trace. The size of the vertical search window will be $\pm N_{win}$ indices. This peak-following approach is easier in the flattened projection, because the size of the search window can be reduced significantly (typically N_{win} can be set to 1 or 2 indices). However, it can also be performed in the twtt and ~depth views. While picking, you can still pan using the arrow keys (Table 1), increase (decrease) N_{win} using "w" ("s"), and zoom in (out) using "z" ("o").

If the picked layer is poor, it can be undone immediately by then pressing "u". If an earlier picked layer is no longer considered accurate, it can be deleted by moving the cursor close to it and pressing "d". Once done picking layers, press "q". These layers are then sorted, displayed in red, and projected back onto the original data, but the radargram view will remain in the current projection. New (pink) layers can be immediately and merged edited during semi-automatic picking. See functionality for *Adjust* and *Merge* described below. However, layers picked during a previous semi-automatic picking sessions (displayed in red) cannot be adjusted and merged while picking new layers.

These layers are sometimes rough, due to the simplicity of the peak-following method, so a smoothing operation is necessary. All layers are automatically smoothed after picking. This smoothing operation does not overwrite the original layers, but is simply another version of the layers available for later analysis. They can be toggled with the checkbox next to *Smoothed layers*. The smoothed layers are displayed in green. The length scale over which this smoothing is performed is L_{smooth}, which is in units of kilometers (km). The longer the length scale, the longer this operation will take. Layers that do not span the entire block are trimmed at their edges due to the smoothing algorithm used. If a smoothed layer has zero length, then it is deleted.

Only the brightest and flattest layers can be traced accurately across the entire data block a single click. In general, it is necessary to edit several layers and sometimes merge them after they have been picked. Below is the operation of the functions that adjust layers is described.

The dropdown layer menu is used to highlight/select a layer. All versions of this layer will be highlighted, i.e., its representations in both the original and flattened projection, and the smoothed versions if available. This menu applies to picked layers only. It is necessary to highlight a layer prior to *Delete*, *Focus*, *Adjust*, or *Merge*. Layers are ordered from shallowest (first) to deepest (last). This ordering is based on their average *y* indices values, and may produce an unanticipated order if layers with inaccurate segments cross or overlap prior to being edited. Left-click a layer to select it when not picking. To highlight the next layer in the layer list, press *next*, and to highlight the last/previous layer, press *last*. To delete the selected layer, press *Delete*, then press "y" to confirm deletion, otherwise it will be cancelled.

For faster location of layers vertically, press *Focus* to narrow the traveltime window to ±100% of the traveltime range of the selected layer. For well-flattened layers, this range will be quite narrow.

To adjust the selected layer iteratively, press *Adjust*, which will yield a cursor. Again, click the left mouse button once to initialize the available options. To remove the portion of the selected layer to the left of the cursor, press "l". This operation applies to the selected layer only, and will occur regardless of vertical position of the cursor. This behavior is consistent for all possible adjustments. To remove the portion to the right of the cursor, press "r". To remove a finite middle portion, press "c" at the left end of the portion to be cut, move the cursor to the right end of the portion to be cut, and then press any key or click the mouse. To undo the last adjustment, press "u". Once layer adjustments are complete, press "q". Layer adjustments cannot be undone once "q" has been pressed.

To merge two layers, selected one of the layers to be merged and then press *Merge*. Next, move the cursor to the second of the two layers to be merged and press any key or click the mouse. The layer closest vertically to the cursor at the picked trace (horizontal position) will be merged with the first layer.

2.13. Pick layers manually (*manually.*; `pk_man`)

Some layers are difficult to trace using semi-automatic methods, so they are best traced manually instead. Press *manually* to initiate this picking. This picking style can only be done in twtt or ~depth. Red "x" symbols mark picks. Press "return" when done, and the manually picked will be sorted and smoothed in the same way as for semi-automatic layers. A window of N_{win} indices about the manually picked spline will be tested to find each trace's local maxima, so pinpoint accuracy is not necessary, but these points should be relatively dense horizontally.

2.14 Assign layer to surface or bed (*Surf./bed*; `pk_surfbed`)

The surface and bed picks can also be edited as described above. However, if no such picks exist, it is necessary to assign a picked layer to become either the surface or the bed. Pressing *Surf./bed* will prompt the user to select whether the current layer should be assigned to the surface ("s") or bed ("b") and then the assignment will be performed.

2.15. Match picked layers with the reference layers from an overlapping block (*Match*; `pk_match`)

Because data volumes are generally large enough and transect lengths are generally hundreds of kilometers, all layers in a transect cannot be picked from one single data block. Separate data blocks formed by `radblockproc` that have a degree of overlap determined by that function call. To ensure layer continuity across distances larger than the length of each data block, the layers must be matched across this overlap.

After loading the reference picks from an overlapping block, picking all desired layers in the data, and smoothing those layers, the smoothed layers can now be matched across the overlapping portion. Press *Match* to automatically match layers picked from the current block and the reference layers. If the absolute value of the mean difference between two layers is less than Δt, whose units are microseconds (μs), then the layers are considered matched and the matching layer from the current block is assigned the same layer identification number (`pk.ind_match`) as the matching layer from the overlapping block. The matching is only considered successful if a single reference layer matches just one of the current block's layers.

If there are no reference layers loaded, then it is assumed that this block is the first of a transect. If it is indeed the first of a transect, press "y" and then press the left mouse button. If it is not, then the matching will not proceed and reference layers must be loaded first. This approach ensures that layers from different blocks do not get assigned the same layer identification number.

Prior to matching, the colors of all layers in the current block are faded. If a match occurs, that layer is restored to its original color. This color adjustment provides quick identification of picked layers that did not match with any reference layer. In most cases, the lack of a match will be expected and due to the lack of a nearby layer. In other cases, it may be due to a poorly chosen value for Δt.

2.16. Save picks (*Save*; `pk_save`)

Having picked the data and matched the layers, the layer picks can now be saved as a .mat file. Because saving can only be done once all of these procedures are complete in at least some form, budget enough time to get through them prior to picking any data block. Once the data have been saved, a simple figure is displayed and also saved in PNG format with the same filename (before the extension) that shows both the radargram and the smoothed, picked layers. If the surface and/or bed were adjusted, then the block will also be saved in the same location from which it was loaded.

2.17. Pop figure (*Pop figure*; `pop_fig`)

At any point after data are loaded, a separate figure can be made by pressing *Pop figure* that represents the current display. This is a simple way of outputting an image of a feature without having to take screenshot that includes the GUI elements.

3. **mergegui**

Once all the blocks in a transect have been picked using `pickgui` (or at least all of those with distinct layers), the picks from each block must next be merged using `mergegui`. The format of the GUI is similar to that of `pickgui`, and the description of identical or very similar features is not repeated.

The primary purpose of `mergegui` is to verify that the picked layers are unique, and that no layer matches have been neglected that should have otherwise been made during either initial picking or quality control (QC). Most time spent exploring the merged radargram within `mergegui` should be spent with this goal in mind. This goal is best achieved within `mergegui` because layering patterns across continuous blocks become clearer than they may have been within individual blocks.

The *y*-axis of the radargram display panel is now elevation (units of meters), rather than traveltime, although this axis becomes that of depth (same units) once the data are flattened. Merged radargrams will be corrected for the surface elevation. Layers are ordered by their mean surface elevation.

All common commands in `mergegui` have keyboard shortcuts (hotkeys), as for `pickgui`. Table 2 lists them.

Table 2. `mergegui` hotkeys.

Key	**Action**
1	Toggle display of merged picks
2	Toggle display of merged radargram
3	Toggle display of block divisions and labels
4	Toggle display of core intersections
5	Switch standard layer coloring and age coloring (if available)
6	Toggle display of surface and bed picks
a	Display separate static map showing location of transect
b	Toggle automatic color scale adjustment
c	Load core intersection data
d	Delete selected layer
e	Reset *x*-/*z*-axes
f	Flatten radargram with respect to layers
g	Toggle grid lines
h	Toggle check for correct date format
k	Split selected layer
l	Load radar data blocks to merge
m	Merge two layers
n	Select next layer
o	Focus on the selected layer
p	Load pick files to be merged
q	Display separate static figure replicating current display

s	Select previous layer
v	Save merged picks
w	Toggle display's color map
x	Toggle fixing the distance range
y	Toggle fixing the elevation/depth range
z	Interactively select layer to highlight
←,↑→,↓	Pan windows 25% of current range in arrow direction
spacebar	Toggle viewing mode (elevation/amplitude or depth/flattened)
/	Toggle flattening polynomial order (2nd or 3rd)
Left-click	Select nearest layer
Mouse wheel	Zoom in or out

3.1. Load picks from multiple blocks (*Load picks*; `load_pk`)

The first step is to load existing picks for the blocks to be merged, which is accomplished by pressing *Load picks*. These picks must be loaded prior to loading the radar data blocks, because information from the merging of the picks is used to merge the data blocks efficiently. Multiple pick files can be selected. If any selected file does not contain a `pk` structure, an error is returned and the loading is cancelled. Once the pick files are loaded, they are then merged.

After pressing *Load picks*, press either "f" or "d" to selected either a group of files in a directory or all .mat files in a selected directory, respectively. After loading and merging the pick files, a list next to the viewing mode radio list is populated; this pull-down menu lists the radar data blocks that match the pick files that were loaded.

The transect name is displayed in a text box below the *Load picks* button. The delineation of individual blocks is turned off by default but can be turned on by selecting the checkbox next to *block divisions*. The last two characters of the block names, which should be the block number, are displayed at each division.

3.2. Load data from multiple blocks (*Load radar data*; `load_data`, `load_data_breakout`)

Once the picks have been loaded and merged, the radar data blocks can then be loaded by pressing *Load radar data*. The data list that was populated during the *Load picks* operation should be used as a reference for which radar data blocks to load. If the data follow the expect `melt/icebridge/data/` directory structure relative to the path of the current picks files, then they are loaded automatically. Otherwise, they must be selected manually.

Prior to pressing *Load radar data*, check that the value of $N_{decimate}$ is appropriate for the vintage and amount of data to be displayed. For the displayed picks, this value is generally not a concern, but it is for the radar data themselves. Because loading all the relevant radar data blocks into memory is unwieldy, the radar data are decimated as they are loaded. If $N_{decimate}$ is adjusted after the radar data are loaded, then the same radar data blocks are reloaded and decimated using the adjusted value.

3.3. Load core intersection data (*Load core intersections*; `load_core`, `load_core_breakout`)

Once the picks have been loaded, a specific MATLAB file (`core_int.mat`) can be loaded by pressing *Load core intersections*. Using this file, the program then determines whether the current transect passes nearby any of seven deep Greenland ice-core sites. If so, then the locations of the nearest approaches to the ice-core site(s) and the names of the sites. `core_int.mat` was populated by `core_int.m`. The structure of that script can be easily adapted to other regions.

3.4. Load layer age data (*Load ages*; `load_age`, `load_age_breakout`)

Similar to the above, once the picks have been loaded, a specific MATLAB file (`date_all.mat`) can be loaded by pressing *Load ages*. Using this file, the program then color-codes each layer by its age and switches the color bar to display an age range (in thousands of years, ka) that can be adjusted. Undated layers are shown in magenta. The decibel scale of the radar data can be readjusted by switching back to "std" view.

3.5. Merge layers (*Merge*; `pk_merge`)

Layers are merged by first selected the layer to be merged with another the current layer. This can be achieved by either selected it from the layer list or choosing it interactively using *Choose layer*. A cursor will appear and the layer to be merged can be selected. This operation can be cancelled by pressing "q". Once chosen, the layer to be merged will change color to that of the current layer, but its thickness will not change. The merging must then be confirmed by pressing "y", otherwise any other button will cancel the merging. If there is any overlap between the two layers, which there should not be at this stage, the values for the current layer will be used.

3.6. Split layer (*Split layer*; `pk_split`)

Layers can be split using *Split layer*. A cursor will appear and the *x*-axis position where the split is to be made must be selected. This operation can be cancelled by pressing "q". If the layer would be empty to the left or right or the split, the splitting operation is also cancelled.

3.7 SNR analysis (*SNR*, `do_snr`)

Signal-to-noise ratio (SNR) analysis involves mapping the SNR of each core-intersecting layer, for later use in age determination. It loads a series of additional figures (one for each core intersection) from which each layer at that intersection can be highlighted and its local noise floor can be determined.

3.8. Display current map (*Map*; `pop_map`)

Pressing *Map* once picks have been loaded displays a separate static figure showing the location of the transect within Greenland. This map can be helpful for quickly understanding the plane view flight path, any turns that are present, and any intersections of the transect with itself. The beginning and end of the transect are displayed as green and red circles, respectively, and blue circles are markers representing every 100 km in the linear distance vector. Note that these markers are determined by the distance vector for the entire transect and not simply the picks currently being displayed.

3.9. Flatten radar data (*Flatten data*; `flatten`)

Once the picks and data have been both loaded and merged, the data can again be flattened by pressing *Flatten data*. As compared to `pickgui`, the flattened space in `mergegui` is more physically meaningful: it is depth at the reference trace. This viewing mode can be especially helpful for matching layers across long distances, because well-preserved layers traced across long distances are a powerful constraint on the layering pattern elsewhere in the ice column.

Flattening in `mergegui` is simpler than for `pickgui`. In `mergegui`, the flattening polynomials are first calculated using all the layers present at the trace with the largest number of layers, as for `pickgui`. Once this first procedure is done, all other layers that overlap with those present at the reference trace are then iteratively incorporated into the flattening polynomials, starting from the layer with the most overlap and ending with that with the least amount of overlap. This permits the inclusion of well-mapped layers that were not traced at the reference trace directly. A disadvantage of this approach is that flattening can take much longer, especially if parallelization is unavailable.

Because merging of well-separated layers can result in large changes in the flattening, any deletion or merging of layers in `mergegui` after having already flattened means that the merged picks cannot be saved until the flattening is re-done. This restriction ensures that the best possible flattening polynomials for the whole of the merged transect are saved.

3.10. Save merged picks (*Save*; `pk_save`)

Once the merged picks have been QC'd to the user's satisfaction, the merged picks should be saved by pressing *Save*. This process is much simpler than for `pickgui`, although the file size is larger, so it often takes longer. Once saving is complete, the complete merged radargram and picks are displayed. If a layer has been deleted, split, or merged since flattening was last performed, then flattening must be redone prior to saving.

4. `fencegui`

The purpose of `fencegui` is to match picked layers between intersecting transects. It is simplest to select a master transect (preferably one that intersects or crosses near one or more dated ice cores) and match layers from intersecting transects to this master transect. This approach simplifies layer identification. Note that matching layers is distinct from merging them. Merging them reduces the number of layers, whereas matching them in `fencegui` simply records that they are the same layer.

Because of the inherent challenge of viewing and matching layers in 3-D, `fencegui` is divided into two distinct but interconnected GUIs. The first is 3-D, and can display both picks and radargrams in terms of elevation only. The second is 2-D, and it displays the radargrams of the master and intersecting transecting transects in a manner similar to `mergegui`, except that the radargram panel is split in two to accommodate each transect. The primary purpose of the 3-D GUI is to orient the user in terms of the physical location of the various intersections and develop a holistic sense of the transect-wide stratigraphy. It is not recommended to explore the data with the radargram set to viewable in the 3-D GUI because changing the view angle/elevation will be very slow. The purpose of the 2-D GUI is to directly compare radar data and overlain picks in the immediate vicinity of each intersection. Only in the 2-D GUI can layers be matched.

Table 3. 3-D `fencegui` hotkeys

Key	**Action**
1	Toggle display of master transect picks
2	Toggle display of interesting transect picks
3	Toggle display of current transect pair’s intersections
4	Toggle display of core intersections
5	Toggle display of master radargram
6	Toggle display of intersecting radargram
a	Select previous layer in the active transect’s layer list
c	Load core intersection data
e	Reset *x*-/*y*-/*z*-axes
g	Toggle grid lines
i	Load reference intersection data
m	Locate master layer ID list
n	Select next layer in the active transect’s layer list
p	Load master merged picks file
t	Test function
v	Save intersecting transect’s picks
w	Toggle radargram color map
x	Toggle fixing the *x*-axis range
y	Toggle fixing the *y*-axis range
z	Toggle fixing the *z*-axis range (elevation)
spacebar	Switch between 2-D (map) and 3-D view

Table 4. 2-D `fencegui` hotkeys

Key	**Action**
1	Make master transect the active transect
2	Make intersecting transect the active transect
3	Toggle display of master transect picks
4	Toggle display of interesting transect picks
5	Toggle display of master transect data
6	Toggle display of intersecting transect data
7	Toggle display of current intersections in the master transect’s display
8	Toggle display of current intersections in the intersecting transect’s display

9	Toggle display of core intersections in the active transect's display
a	Select the previous layer in the active transect
b	Toggle automatic color scale adjustment for active transect's display
d	Toggle fixing the distance range of the active transect's display
e	Reset *x*-/*y*-axes of active transect's display
g	Toggle grid lines in the active transect's display
h	Toggle highlighting of matched layers
l	Load radar data for the active transect
m	Match current layers in the master and intersecting transects
n	Select the next layer in the active transect's layer list
o	Focus on current layer
t	Select previous intersection
u	Unmatch current in the master and intersecting transects
v	Toggle selection of nearest intersecting layer in inactive transect
w	Toggle radargram color map
x	Interactively select current layer in the intersecting transect
y	Select next intersection
z	Interactively select current layer in the master transect
←,↑→,↓	Pan active transect's display 25% of current range in arrow direction

4.1. Load intersecting transect data (*transects*; `load_int`)

The process of analyzing data in `fencegui` is begun by pressing *transects* in the 3-D GUI to load `merge_xy_int.mat`. The locations of the intersections recorded in this file are determined using the `radarint` script beforehand. These data are used to populate the intersecting transect list once a master transect is chosen. If the `melt/icebridge/data/` directory structure is present, `merge_xy_int.mat` will be loaded automatically, without user input to determine the file location.

4.2. Load core intersections (*cores*; `load_core`, `load_core_breakout`)

Once the transect intersection data have been loaded, `core_int.mat` should be loaded. This file is the same as for `mergegui`, and it behaves similarly. Generally, this file should be in the same directory as `merge_xy_int.mat` and will be loaded automatically because `load_core` will look there first.

4.3. Locate master layer ID list (*master*; `locate_master`)

Once the transect intersection and core intersection data have been loaded, `id_layer_master.mat` should be located. This file represents the master layer ID list that is slowly populated through the use of `fencegui`. This file is not directly loaded into memory until saving, so that multiple individuals can work within `fencegui` at the same time. This file should also be in the same directory as `xy_int.mat` and `core_int.mat`.

4.4. Load transect picks (*Load master picks*; `load_pk`)

Once the intersection data are loaded, the user can press *Load master picks* to load a master transect's merged picks file. The simplified name of the master transect is then displayed, and the list of intersecting transects is populated based on `xy_int.mat`. In parentheses next to each transect name in the list is the number of intersections of that transect with the master transect. In general, this number will be an overestimate of the number of useful/valid intersections because of the way in which the intersections are calculated, and because the portion of the intersecting transect that was actually picked is often shorter than the total length of the transect.

The user can now select an intersecting transect from the list to load. Once a transect is selected from this list, the user will be prompted to locate the appropriate merged picks file. Because a merged picks file might not necessarily cover an entire transect, it is necessary to recalculate the intersections between the two transects once both picks files are loaded, and it may turn out that the selected intersecting transect does not intersect the master transect. The intersections are shown on both GUIs as vertical magenta lines spanning the elevation range. In the 2-D GUI, at each intersection markers are placed at the elevations/depths of the intersecting layers in the other transect. The color of the marker is the same as that of the layer in the other transect, and its marker shape depends on whether the layer has already been matched between transects. Circles are unmatched intersecting layers, squares are intersecting layers that have been matched to another transect but not the current master/intersecting pair, and triangles are intersecting layers that have already been matched between the current pair. If matched between the current pair, the color of the intersecting layers will be that of the master transect's layer.

4.5. Load radar data (*Load master data*, *Load intersecting data*; `load_data`, `load_data_breakout`)

In the 2-D GUI, lists directly above the radargram panels are populated with the names of the data blocks used in their respective merged picks files. Refer to these lists and then press either *Load master data* or *Load intersecting data* to load the appropriate files. As in previous GUIs, if the `melt/icebridge/data/` relative directory structure is present, then by following the path to the merged picks files, the radar data will be loaded automatically. Otherwise, user input is required in locating the appropriate files. If the flattening polynomials are available for the merged picks file, then the data will be flattened and the flattened view will be available.

4.6. Match intersecting layers (*Match*, *Unmatch*; `pk_match`, `pk_unmatch`)

Layers in intersecting transects are matched in a similar way to `mergegui`. This matching is the primary purpose of `fencegui`. After pressing *Match*, the color of the selected layer in the intersecting transect is changed to the color of the selected master layer, the user is prompted to confirm the match by pressing "y", and then the markers of the matched layers are changed to triangles.

More than one layer in the intersecting transect can be matched to a layer in the master transect. This functionality is necessary because of possible omissions in `mergegui`. Similarly, unmatching a layer pair is accomplished by pressing *Unmatch*, but only if the layer pair has been matched during the same `fencegui` session.

If a layer pair has already been matched, and there is yet another match that includes one of those layers, then it may be necessary to uncheck *Match*, which otherwise highlights the original layer pair if one of them is selected. When *Nearest* is checked, the nearest layer at the current intersection in the inactive transect will be highlighted, but this behavior is ignored if a match for the selected layer in the active transect is found.

4.7. Save intersecting picks (*Save*; `pk_save`)

Once all layers have been compared between the master and intersecting transects at all intersections, the intersecting layer's picks should be saved by pressing *Save*. The only variable that is added or edited in the intersecting transect's merged picks file is `pk.ind_layer`. Similarly, the `id_layer_master.mat` will be concatenated with the new matches that were made. `fencegui` assumes that the name of the intersecting transect's merged picks file will be used, but it is not overwritten automatically. All files for saving are selected prior to their actual saving, to ensure that matches are not adjusted without saving ready to go.