Supplementary material for ImpDAR: An open-source impulse radar processor

David A. Lilien, Benjamin H. Hills, Joshua Driscol, Robert Jacobel, and Knut Christianson

# Installation

A full set of instructions, intended for users with little or no experience with Python or git, is presented in Section S1.1. Users with existing Anaconda installations can follow this first set of instructions, skipping the first step. For users who wish to use some other Python installation, a list of the packages to install follows in Section S1.2. For users who wish to get the latest development updates to ImpDAR, rather than the most stable version, instructions are in Section S1.3.

## Installation with Anaconda

For users who are new to Python, do not have Python installed, or do not have experience with git, this method is most straightforward and has no prerequisites.

1. A current (2.7 or 3.5+) Python installation is the first requirement. We recommend getting Python 3, since Python 2 is no longer supported. Straightforward installation is available from Anaconda (<http://anaconda.com>). Anaconda can be installed on Windows, MacOS, or Linux, and it can set up path variables to make programs executable from any directory. On Windows, Anaconda does not recommend modifying path variables—if users opt to leave the path unchanged, they should perform subsequent steps from the “Anaconda prompt” rather than from the command prompt. Follow the instructions on the Anaconda website for your particular operating system. In order to avoid complications, it is easiest just to restart the terminal (MacOS/Linux) or open the Anaconda prompt after Anaconda installation to ensure that the new path specifications are loaded. For the most basic installation, with ability to pick layers and process data from a limited set of inputs, once can skip directly to step 4, but the next two steps will add support for geographic projections and additional data formats.
2. After installing Anaconda, there are several recommended dependencies for ImpDAR. Because the dependencies are hosted at different places, installation takes two steps. First, install GDAL (<https://gdal.org>), which provides the tools to reproject the geographic coordinates associated with radar data and segyio, (<https://segyio.readthedocs.io/en/latest/>), which is needed to export to the SEGY format. These can be installed using:

conda install -c conda-forge gdal segyio

1. One more package, h5py (<https://www.h5py.org>), needed for data from Blue Systems radars, can be installed with:

conda install h5py

1. Your computer should now be set up to install ImpDAR. The latest stable version of ImpDAR can be obtained with

pip install impdar

If no newer features are needed, the version corresponding to the time of this publication can be installed with

pip install impdar==1.0

ImpDAR should now be set up and available for use. All functionality will be included, except for certain migration routines which rely on the external SeisUnix library (<https://github.com/JohnWStockwellJr/SeisUnix/wiki>). Installation of those tools is more complex and is operating system dependent, so the user should follow the instructions from SeisUnix if those routines are desired. However, ImpDAR provides several migration algorithms, which make for a fully functional installation without installing SeisUnix. SeisUnix (as well as GDAL, segyio, and h5py) are searched for at the time that ImpDAR needs them, so they can always be added later with no reinstallation of ImpDAR if need arises the additional functionality.

## Dependencies when using other Python versions

Users with an existing Python installation may wish to use that rather than Anaconda. All required dependencies (numpy, scipy, and matplotlib) and some optional dependencies (h5py, and segyio) can be installed with pip. GDAL, required for reprojection, and PyQt5, required for the picking GUI, depend on external libraries that cannot simply be installed with pip. Rather, these external libraries must be compiled for pip to access. Since this installation will vary across operating systems, we cannot provide a full set of instructions here. ImpDAR will work without these optional packages (and without SeisUnix as discussed above), and they can be added later.

## Development version of ImpDAR

New development of the ImpDAR package is not immediately made available through pip so as to permit testing without interrupting most users. However, this also means that access to readers for additional data types and other improvements are not immediately accessible through the installation method in step 4 of Section S1.1. Instead, users needing such developments will need to get the latest version of ImpDAR from github.com then install it. Here, we assume that required dependencies (numpy, scipy, matplotlib) have already been installed.

1. There are two options to get the code:

* Users without a git installation can download the development source code as a zip file from <https://github.com/dlilien/ImpDAR/tree/devel>.
* Users with Git installations can use

git clone -b devel https://github.com/dlilien/ImpDAR.git

1. Unzip the file (platform dependent) and enter the impdar folder. This step will vary based on where the files were placed, but on MacOS or Linux, this will require a command similar to

cd impdar

1. Install ImpDAR for all users with

python setup.py install

or for the current user only with

python setup.py install –user

# Processing steps corresponding to manuscript figures

Here we give scripts to perform the full processing for Figures 2 and 3 in the main text. Comments give an indication of what is done each step, so these can be used as examples that users can modify as needed for their particular use.

## Northeast Greenland

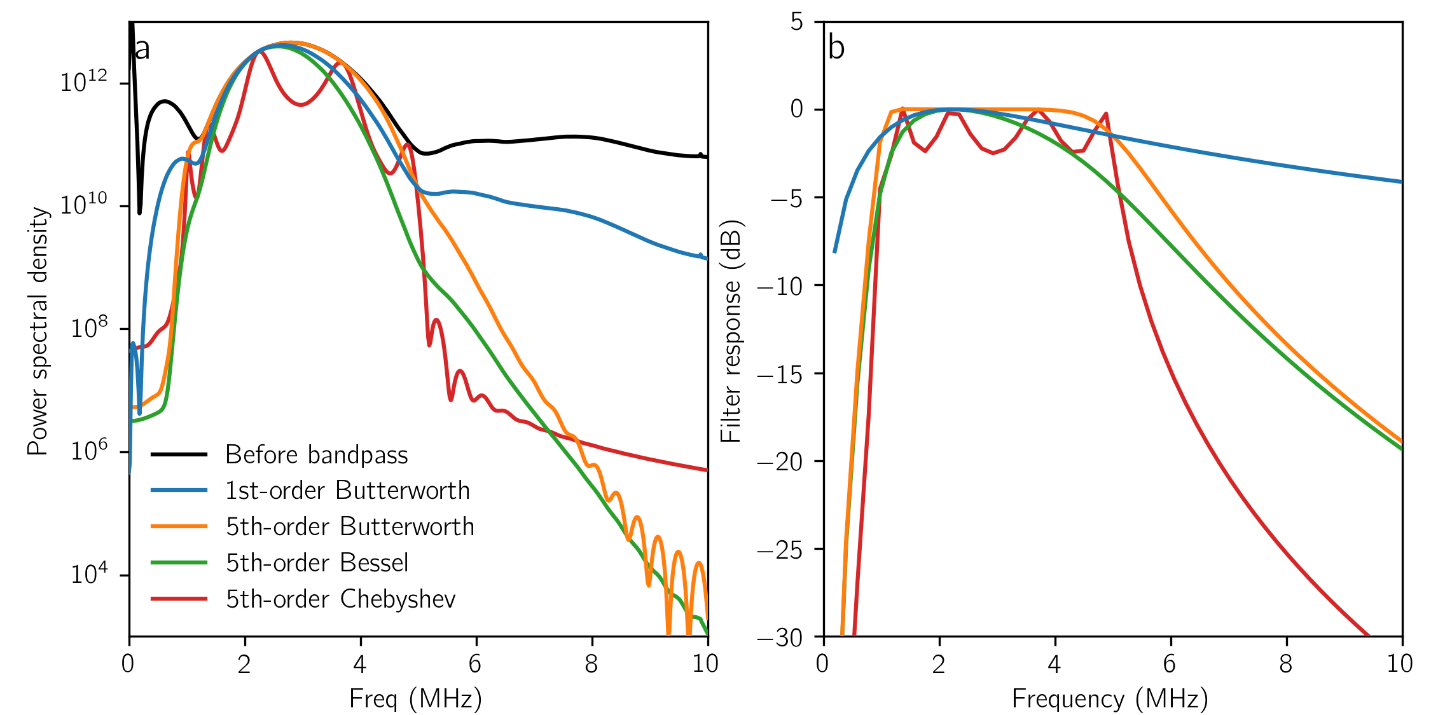
The full processing used to get to the radargram displayed in Figure 3b can be performed with:

1. #! /bin/bash
3. # Begin by simply loading data into ImpDAR
4. # in this case, all the files we will use are listed in proc\_list.txt
5. impdar load gecko $(cat proc\_list.txt)
7. # Reduce noise by vertically bandpassing
8. impproc vbp 1 5 nline5\_raw.mat
10. # Adaptive horizontal filter
11. # Uses average of 100 surrounding traces, tapering off to preserve deep reflections
12. impproc ahfilt nline5\_bandpassed.mat
14. # Next, crop the data to reduce the file size
15. # First take off the pretrigger
16. impproc crop top snum 512 nline5\_bandpassed\_ahfilt.mat
18. # Crop off the bottom using a value from visual inspection
19. impproc crop bottom twtt 38 nline5\_bandpassed\_ahfilt\_cropped.mat
21. # Rename to limit length
22. mv nline5\_bandpassed\_ahfilt\_cropped\_cropped.mat nline5\_bandpassed\_ahfilt\_cropped.mat
24. # Interpolate the data to constant horizontal spacing
25. impproc interp --gps\_fn kinematic\_gps.mat 8 nline5\_bandpassed\_ahfilt\_cropped.mat
27. # Migrate using SeisUnix T-K migration as in the figure
28. impproc migrate --mtype sumigtk --htaper 20 nline5\_bandpassed\_ahfilt\_cropped\_interp.mat -o nline5\_bandpassed\_ahfilt\_cropped\_interp\_sumigtk.mat
30. # Convert to depth while accounting for the spacing between antennas
31. impproc nmo 169.4 nline5\_bandpassed\_ahfilt\_cropped\_interp\_sumigtk.mat
33. # Export a shapefile to plot the maps for the paper.
34. impdar convert -t\_srs 3413 nline5\_bandpassed\_ahfilt\_cropped\_interp.mat shp

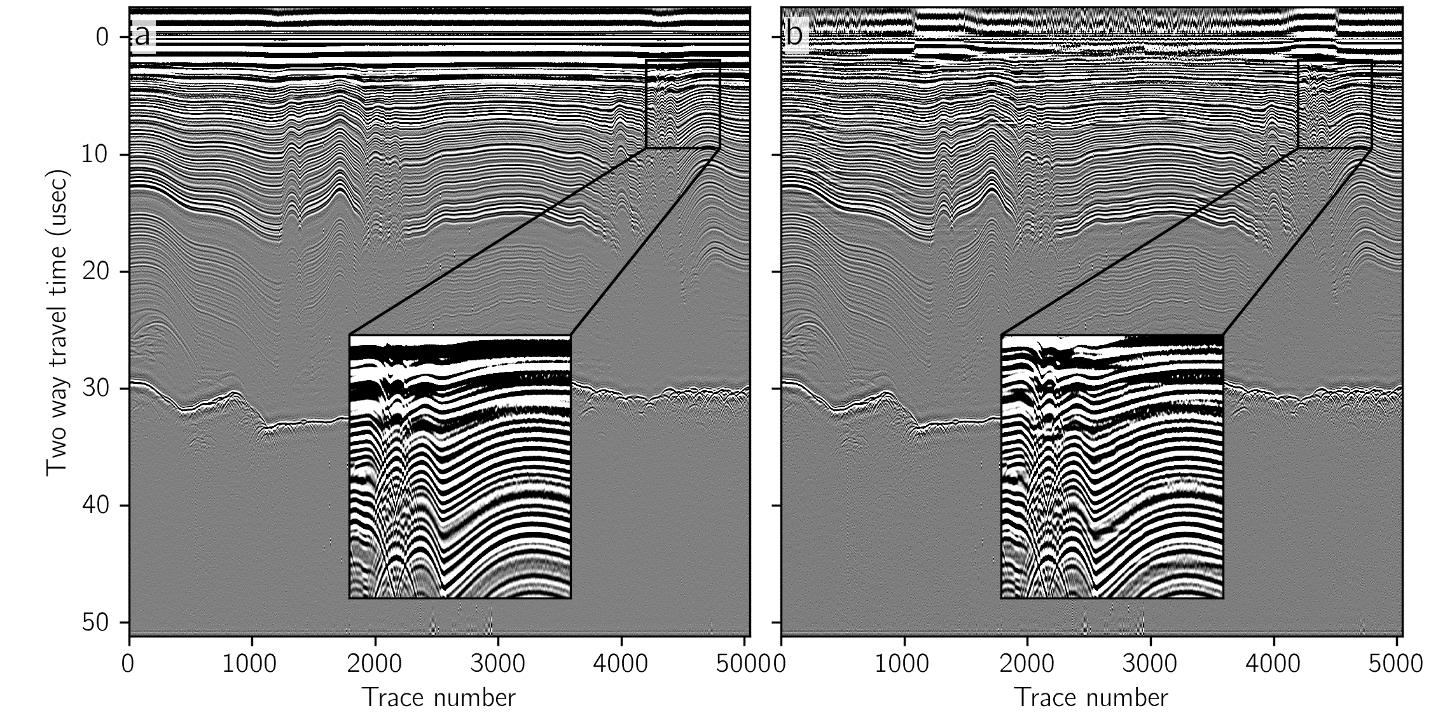
## South Cascade Glacier

The processing for Figure 4b and c is:

1. #! /bin/bash
3. # Begin by simply loading data into ImpDAR
4. impdar load pe LINE03.DT1 LINE04.DT1
6. # Attach the precision GPS data, fine tuning the location using cross-correlation with NMEA strings
7. impproc geolocate --guess digestible\_gps.mat LINE\*\_raw.mat
9. # Concatenate the different profiles
10. impproc cat LINE0[3-4]\_geolocate.mat
12. # Clean up the file name
13. mv LINE0?\_geolocate\_cat.mat LINES.mat
15. # Bandpass filter
16. impproc vbp 300 700 LINES.mat
18. # Pick a reasonable set of traces to get an average trace to subtract
19. impproc hfilt 10000 12000 LINES\_bandpassed.mat
21. # Crop to the returns
22. impproc crop top twtt 0.02 LINES\_bandpassed\_hfilted.mat
24. # Convert travel time to depth using wavespeeds inferred from variable densities
25. impproc nmo 0 --rho\_profile densities.txt LINES\_bandpassed\_hfilted\_cropped.mat
27. # Move to constant trace spacing, removing static traces
28. impproc interp --minmove 0.02 0.1 LINES\_bandpassed\_hfilted\_cropped\_nmo.mat
30. # Produce the variable-surface version for Figure 4c
31. impproc elev LINES\_bandpassed\_hfilted\_cropped\_nmo\_interp.mat
33. # Convert to shapefile for plotting on map in Figure 1
34. impdar convert -t\_srs 32610 LINES\_bandpassed\_hfilted\_cropped\_nmo\_interp\_elev.mat shp



**Figure S1.** Comparison of several finite impulse response filters applied to data from Northeast Greenland. **a**. Power spectral density, calculated per-trace then averaged, before processing (black) and after bandpass filtering between 1 and 5 MHz using a 1st-order Butterworth (blue), 5th-order Butterworth (orange), 5th-order Bessel (green), or 5-th order Chebyshev type-I (red) filter. The goal in filter selection is to preserve the shape of the original data in the pass band but cause the amplitude to decay smoothly outside it. **b.** Filter response as a function of frequency for the three filters in a.



**Figure S2.** Effect of adaptive horizontal filter. **a.** Echogram before horizontal filter. **b.** Echogram after filter. Note that the filter tapers with depth, so deep layers are unaffected, but the filter brings layers out through the noise in some shallow areas (inset in both panels).