C:\Users\burkdani\Downloads\MSU-Wordmark-Green-CMYK.epsCTBTO-logo-Short-Engl-White.eps

PNE2SAC.py users guide

1Daniel Burk

1Department of Earth and Environmental Sciences, Michigan State University

**Abstract.** PNE2SAC is a python program for the conversion of text files representing digitized seismograms into SAC and Miniseed data files. This document describes how to run PNE2SAC.

# Introduction

From the 1950’s through the 1990’s, there were many quality seismic stations scattered throughout the USSR. These networks generated millions of paper seismograms from their short period, long period, and strong motion stations. The eighty years of seismograms represent unique event-to-station vectors and datasets for earthquakes of importance, as well as calibration explosions, both chemical and nuclear. The quality digitization of these paper seismograms means that this dataset can be utilized in new ways with the modern software. PNE2SAC inputs the digitization text files from Wavetrack, and conditions the output to better represent the original analog seismogram. Afterwards, the file is output into SAC and Miniseed files

This document describes how to find the software on GITHUB, how to set up the software on your work station, how to prepare the input file for a typical three-channel station, and how to run the software in order to get a graph and a pole-zero output.

Contents

[Introduction 1](#_Toc32563512)

[PNE2SAC setup 3](#_Toc32563513)

[Anaconda Installation 3](#_Toc32563514)

[Installing Obspy using Anaconda: 3](#_Toc32563515)

[Installing PNE2SAC.py 4](#_Toc32563516)

[Preparing the Wavetrack output file for PNE2SAC 6](#_Toc32563517)

[PNE2SAC operation 10](#_Toc32563518)

[Opening the Anaconda command line window 10](#_Toc32563519)

[Launching PNE2SAC with the Wavetrack output file 12](#_Toc32563520)

[PNE2SAC output files 13](#_Toc32563521)

[Navigating the PNE2SAC graphical output 18](#_Toc32563522)

[The final product: PNE2SAC SAC/Miniseed output file 20](#_Toc32563523)

## PNE2SAC setup

PNE2SAC is a command line utility that uses Python 3.7 as its interpretation language, so in order to run the software, you must first make sure the workstation is set up with Anaconda2 . Anaconda is an entire suite of open source scientific software such as signal analysis modules, along with Python. Once installed, a development environment is created and activated. Then, the Obspy seismic analysis package is installed. Once these steps are complete, PNE2SAC is installed in a root directory called “C:/Pyscripts”. After this is done, the system paths are updated so that the command line interface will look for batch files within this folder. After this step is completed, the workstation will be ready to process text files that we create via PNE2SAC to generate poles/zeros files, and to generate dataless SEED files that will represent our analog station.

### Anaconda Installation

For a step-by-step guide, please refer to Chapter one of the document ***“Seismic Lab Operating Handbook”***, that is available from the MSU Geotectonics laboratory. You can also refer to the following links for how to install the software onto your computer. Please be aware that MSU Python scripts are adapting to Python 3.7

<https://www.anaconda.com/distribution/>

Please remember that all installations must be accomplished by running your window with administrator access. Otherwise, it will experience a write failure at the end of the installation process.

If you already have Anaconda installed, but do not have Obspy, you MUST UPDATE your Anaconda distribution! Obspy is an actively developed application, and as such, it utilizes the latest modules. Old Anaconda distributions often fail installation because it contains outdated modules. You may update from the command line: C:> conda update –all

<https://www.anaconda.com/keeping-anaconda-date/>

### Installing Obspy using Anaconda:

Click on the following link for tips on how to install Obspy, once you have successfully installed and updated Anaconda.

<https://github.com/obspy/obspy/wiki/Installation-via-Anaconda>

Again, make sure that you install the modules from a command line window that has been elevated to administrator privileges. Please refer to the lab handbook for additional details. Once you get the Obspy operating environment set up, you will be ready to install PZcal.py, which is as easy as placing the files into a folder.

### Installing PNE2SAC.py

#### Retrieving PNE2SAC from GITHUB

The latest published version of PNE2SAC is found on Github at: <https://github.com/tychoaussie/converters>

It consists of:

**PNE2SAC.bat** – A batch file that calls the PNE2SAC python script and inputs it into the python interpreter

**PNE2SAC.py** – The python script for PNE2SAC containing all our calibration functions

These two files must be placed into a folder located at the root, called C:\Pyscripts (Figure 1).

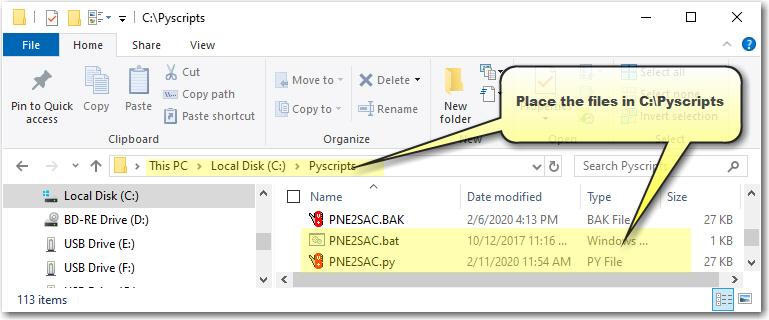


Figure 1: Creation of the Pyscripts folder and placement of the required files within that folder for operation of PNE2SAC

Once the folder c:/Pyscripts is created, you should place it within your path in the system variables. This procedure is covered within the ***“Seismic Lab Operating Handbook”.***

At this point, Anaconda is installed with Python 3.7 , you have created an Obspy environment, and installed Obspy. You have placed the three files within a folder called C:\Pyscripts. One last thing is to set up your user environment variables to include C:\Pyscripts in your user path. This will enable your computer to find PNE2SAC no matter which folder is your current working directory.

#### Placing Pyscripts into your system path:

The previous steps apply to all MSU-derived python applications, and this step is no exception. What is demonstrated applies to Windows 10 machines. Windows 7 and earlier operating systems will vary slightly in updating the system path.

1. A screenshot of a cell phone

   Description automatically generatedFrom the start button, search for “environment variables” and open, from the control panel, the “edit the system environment variables” window (Figure 2).

Figure 2: Searching for Edit the system environment variables from the run tab

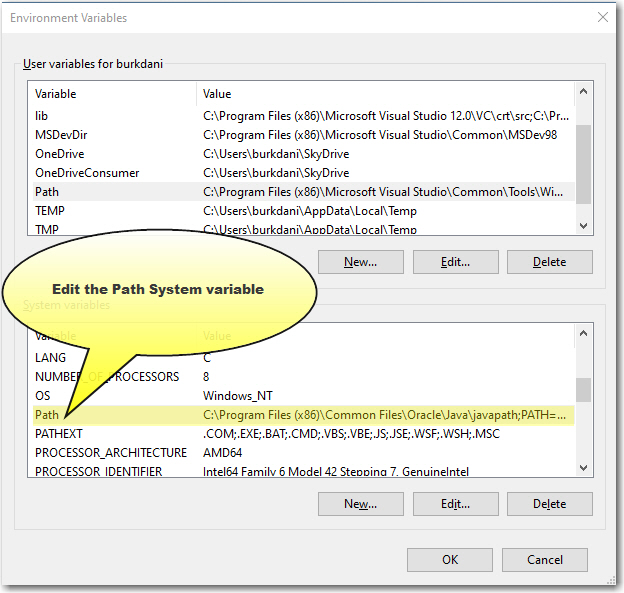
1. Choose Environment Variables at the bottom of the system properties>Advanced tab.
2. From the System Variables list (located BELOW the user variables), scroll down and choose the Path variable, and press the edit button (Figure 3).

Figure 3: Choosing the Path system variable from the Environment Variables list

1. A screenshot of a cell phone

   Description automatically generatedAdd C:\Pyscripts to your list of path variables. The computer will now search this folder for any batch files that you attempt to invoke when typing at the command prompt (Figure 4).

Figure 4: Adding c:\Pyscripts as a new path within the path environment variable

1. Press “OK” and close the system variable list. Now, when you launch the Anaconda command prompt, Pyscripts will be searched whenever you type in commands, such as the batch file “PNE2SAC.bat” that launches PNE2SAC.

### Preparing the Wavetrack output file for PNE2SAC

Wavetrack digitized output files consist of a two-line header, followed by a list of interpolated samples that represent the digitized signal. The header provides insufficient metadata for accurately converting the signal into a useful seismogram. Therefore, the text file must be hand edited and additional information added to the header. In addition, the text file must be checked to make sure the sample rate of the file was appropriate during export.

#### Exporting the waveform from Wavetrack

Before beginning your digitization in Wavetrack, check the export options to be sure that they are properly selected. The sample interval for the interpolated export must be high enough to accurately resolve the amplitudes and frequency response of the waveform. For this reason, (and because disk space is cheap) choose a sample interval of 0.01 seconds, that corresponds to a sample rate of 100 samples per second. This yields a 10:1 oversampling ratio for a 10 Hz signal and ensures for stable conversion of the waveform by PNE2SAC. Additionally, you can insert the channel name and ISC station code into the menu. The comment section is not exported but is retained within the .wsp Wavetrack record. The export will now contain the appropriate station code and channel name, making it easier to edit the text file output to prepare it for PNE2SAC (Figure 5).

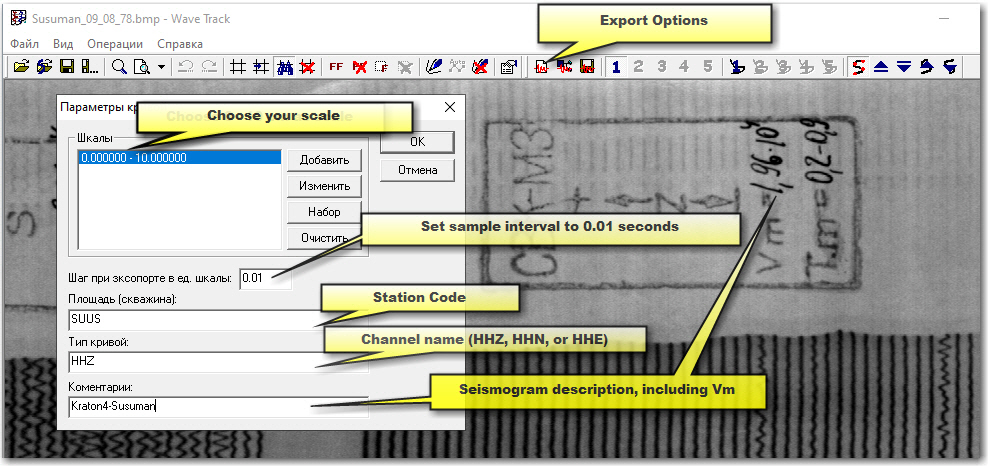


Figure 5: Customizing Wavetrack text output for maximum compatibility to PNE2SAC

Once the digitization is completed, record the apparent start time of the digitization’s first sample, along with the magnification(also known as crest factor) that is listed as Vm on the paper record. Record the time correction (TC) from the paper record. Record the desired start time for the PNE2SAC that corresponds to the nearest minute mark before the arrival of the signal of interest. These four bits of information (Apparent start time, requested start time, TC, and component magnification) must be added to the Wavetrack export file, and are read from the original seismogram (Figure 6). PNE2SAC will use this start time to ‘trim’ the beginning of the waveform. The actual specifics of how to operate Wavetrack are found within the Wavetrack digitization operations guide.

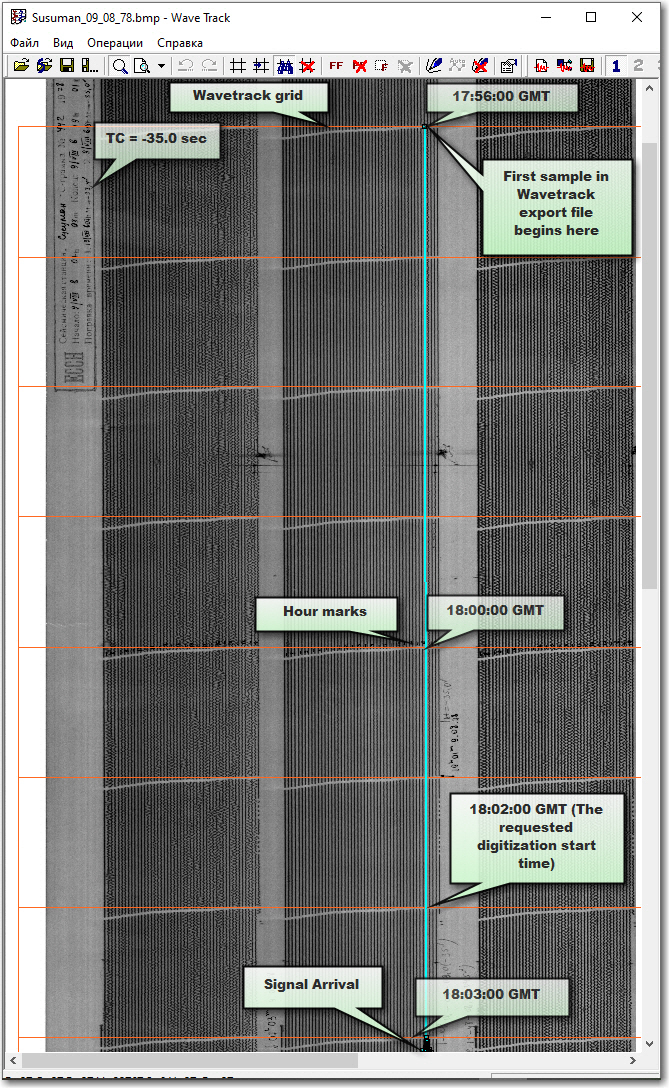


Figure 6: An example of a Wavetrack digitization showing the time correction (TC), time of first sample (RefTime=17:56:00 GMT), Time marker for top of the hour, and minute marker before first arrival. Start time should be chosen to yield about 60 seconds of time-history before first arrival. In this case, 18:02:00 GMT.

An example is this Kraton4 digitization from Station Susuman (Figure 6). The arrival occurs at an apparent 18:03:01.5 GMT. This is too close to the start of the minute, so a good start time for the waveform should be one minute earlier, at 18:02:00.

STARTTIME should therefore be: 09\_08\_1978\_18:02:00.000 “apparent” time.

The start of the digitization occurs at 17:56:00 “apparent” time and would thus be reported as 09\_08\_1978\_17:56:00.000

Time Correction (TC) on this seismogram is reported as -35.0 seconds.

Using these parameters, PNE2SAC will create the seismogram output beginning at 18:02:00 ‘apparent’ time, then apply the -35.0 second adjustment to timing and produce a 18:01:25.000 true start time when generating the resulting SAC & Miniseed files. The Miniseed waveform will then show the corrected phase arrival time of 18:02:26.50 GMT.

#### Editing the text file to prepare it for PNE2SAC

Once the Wavetrack digitization has been exported, the text file must be hand-edited in order to create the header. The header consists of eight sets of fields. The fields are separated by spaces:

**COMMENT** *Information\_about\_this\_waveform*

**NETWORK** *Two\_letter\_Network\_code*

**STATION** *ISC\_Station\_code*

**COMPONENT** *Channel\_name(HHZ, HHN,or HHE)*

**REFTIME** *Time of first sample in the form of: DD\_MMM\_YYYY\_HH:MM:SS.000*

**STARTTIME** *The beginning of waveform export: DD\_MMM\_YYYY\_HH:MM:SS.000*

**CF** The amplification (*Crest factor) for this component in the seismogram*

**TC** *Time correction(in seconds) from the seismogram describing the clock deviation from GMT*

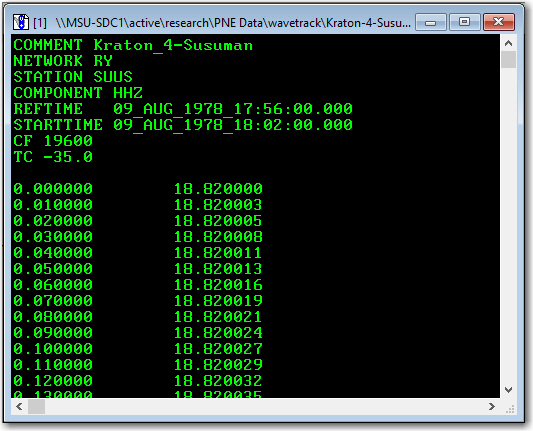
Once the header has been added to the Wavetrack export file (Figure 7), PNE2SAC be used for the creation of the first iteration of the Miniseed file.

Figure 7: Adding the PNE2SAC text header to the Wavetrack export file.

## PNE2SAC operation

PNE2SAC is a python program that runs the script PNE2SAC.py by way of a batch file. The batch file is called PNE2SAC.bat, and it feeds the PNE2SAC script into the Python language interpreter. It requires that you have successfully installed Python 3.7, installed Obspy, opened an Anaconda command line window, and have activated the Obspy environment.

### Opening the Anaconda command line window

A screenshot of a computer screen

Description automatically generatedOpen the Anaconda prompt. You may find it from the start menu button. I personally like to right-click on this item and fix it to my tool bar for easy location. This will open a command line window in which the Anaconda “base” environment is then launched. You will then need to activate your Obspy environment.

Figure 8: Opening the Anaconda Prompt from the start menu

In order to activate your Obspy environment, you need to know the name of your Obspy environment. Yours will be named whatever you called it when installing Obspy. In my case, I have multiple environments for research and development, and for working with different versions of Python. (I have several, but you likely will have only two: base, and your Obspy.)

**Conda info –envs**

will give you a list of your environments. Pick your Obspy environment (mine is called Obspy37) and activate it:

**Activate obspy37** (or whatever your environment is called)

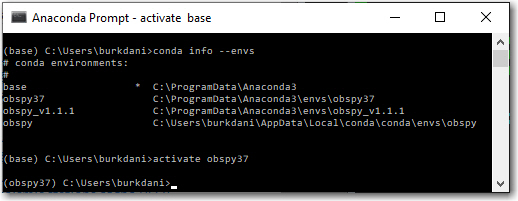
This will activate the environment containing the Obspy package necessary for running PNE2SAC (Figure 9).

Figure 9: Example of how to find your active python environments, and how to activate one.

### Launching PNE2SAC with the Wavetrack output file

Now that the environment is configured and activated, it is time to point PNE2SAC into the folder where your output file is located. There are two methods to do this:

1. Change your current working directory to the one containing the Wavetrack output.
   1. A screenshot of a cell phone

      Description automatically generatedThe command for this is “cd” to change directories(Figure 10).

Figure 10: Example on how to change directories from the Anaconda prompt

* 1. A screenshot of a cell phone

     Description automatically generatedType PNE2SAC, followed by the name of the Wavetrack output file. This will launch the program and input the file for conversion. The file name is now a relative file name, so all output files will be deposited into the current working directory (Figure 11).

Figure 11: Example of launching PNE2SAC with a local copy of the Wavetrack export file

1. A screenshot of a social media post

   Description automatically generatedLeft click, hold, and drag the name of the file from Microsoft Explorer into the command line window (Figure 12). You must type in PNE2SAC first, then add a space, and then you can click and drag the file name into the window. It will fill in the file name with the whole path, and the output of PNE2SAC will likewise use the whole path to output the Miniseed and SAC files into the source directory. Thus, it does not matter what your working directory is, the output will be placed in the same folder as your Wavetrack output file.

Figure 12: Example of clicking on a file name in explorer and dragging it into the Anaconda prompt to start its conversion

### PNE2SAC output files

If your Wavetrack output file has a good header, PNE2SAC should then begin the conversion. PNE2SAC will first open the file and load the header information to retrieve the station name, network name, channel name, reference time, requested start time, and time correction constant. It will then begin processing the Wavetrack output in order to recover the original operator “click points”. Once recovered, PNE2SAC will then use a PChip interpolation algorithm to synthesize the seismogram waveform with a sample rate that matches the original Wavetrack export sample rate. PNE2SAC will then save the file to the original folder in both SAC and Miniseed format.

#### What are Click Points?

A screenshot of a social media post

Description automatically generatedClick points are where the operator “clicked” when digitizing the seismogram inside of Wavetrack (Figure 13). Unfortunately, Wavetrack does not have the option of sending an output consisting of only these points. Instead, the Wavetrack output file consists of a series of linearly interpolated line segments that are then ‘digitized’ to the selected sample rate. This means that sometimes the click point falls in-between two of the linearly interpolated sample points. It also means that large changes in slope can occur at the click point, yielding a very spiky output. It gets worse when you attempt to signal process the waveform for derivative and creates huge signal distortions when viewing the seismogram in the frequency domain. Therefore, PNE2SAC first analyzes the waveform, looks for slope changes, and calculates the intersection of the slopes in order to recover the original click point. The resulting list of click points are then used for generating a more realistic waveform using a PCHIP interpolation algorithm.

Figure 13: Plot of Wavetrack linearly interpolated points (in blue) versus the recovered click points (in green)

#### The PCHIP interpolation algorithm

PCHIP stands for Piecewise Cubic Polynomial Interpolation. It is a form of monotonic cubic spline interpolation in which the derivative is “smoothed” at each click point. A wave form is then “constructed” to connect a curve in-between each click point, such that it intersects each click point. The result is a more natural waveform that mimics the movement of the original galvanometer. Once the line segments are determined, the PCHIP interpolation re-constructs a digitized series of points that mimic the original sample rate of the Wavetrack output file. (This is usually 100 samples/second).

#### Data Discontinuities in the Wavetrack Output:

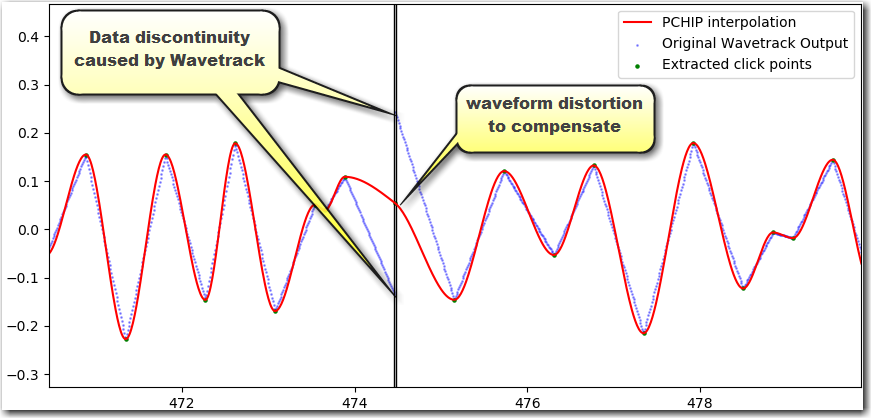
One caveat of the PCHIP algorithm is that the click points must be continuous and increasing in time. Therefore, the data that generates the click points must be free of discontinuities(Figure 14). Unfortunately, Wavetrack sometimes includes discontinuities that are caused by having two click points that occupy the same space in time. PNE2SAC deals with this by flagging the point and generating a synthetic “interpolated point”. This synthetic point will result in a distortion in the waveform, but it will at least enable PCHIP to complete its processing. When this occurs, the proper procedure is to re-edit the Wavetrack output, and correct the problem points. After PNE2SAC generates the SAC/Miniseed files, it will display a graphic of the waveform, including any data discontinuities in need of repair. Once these points are repaired, it is appropriate to re-run the process on the newly generated Wavetrack output file until there are no longer any data discontinuities.

Figure 14: Discontinuity in Wavetrack output (blue dots). PNE2SAC will "distort" the output waveform to avoid crashing (red line), but these errors should be fixed in the Wavetrack .wsp file and re-exported.

#### PNE2SAC diagnostics output

When running PNE2SAC, diagnostics output text is generated. Any data discontinuities will be reported, and the discontinuity will report the exact sample at which the error occurs. It is good practice to trap these errors by directing the text output to a log file using the pipe command (>>):

(obspy37) c:\Seismo\PNE> **PNE2SAC Kraton4\_RY\_SUUS.txt >> Kraton4\_SUUS.log**

If any discontinuities exist within the Wavetrack file, PNE2SAC will generate an entry for that discontinuity and store it in the file (in this example, Kraton4\_SUUS.log)

**Error in calculation of clickpoint time: 474.460 < 477.058 < 474.470 sec.**

**Time discrepancy of 2.598 seconds.**

**Use Linear interpolation set sample time to 474.465 seconds.**

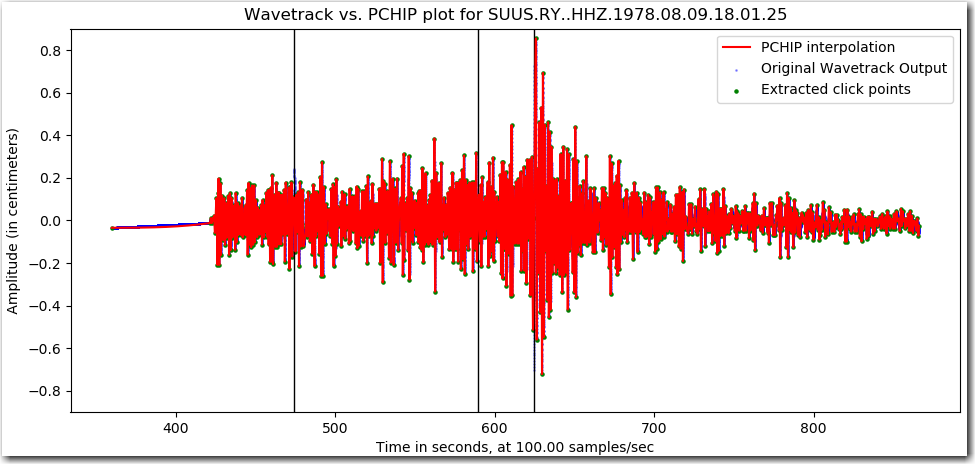
When the discontinuity occurs, a black vertical line will appear on PNE2SAC’s output graph(Figure 15):

Figure 15: PNE2SAC output waveform, with black lines indicating data discontinuities in need of repair.

#### Removing the discontinuity from Wavetrack

A close up of a device

Description automatically generatedThe PNE2SAC text output that identifies a waveform discontinuity lists two different times. These times represent the end point and starting point of two different line segments that bound the calculated click point. If they are continuous, these line segments should intersect at some time in-between the end point and starting point representing the click point. If they do not, it indicates a discontinuity in the Wavetrack output caused by having a click point that is out of sequence, such as being co-located on the same time segment as another click point, or by being mistakenly dragged to an earlier time segment. On Wavetrack, this appears as either a horizontal line, or a ‘rising’ line that moves from one click point to an earlier time (Figure 16). These non-sensical click points might be possible to create within Wavetrack, but are not possible in the real world, where time never stands still, and always moves forward. When navigating in Wavetrack, time, in seconds, is displayed at the bottom of the view. To correct the discontinuity , scroll through the digitized waveform until the cursor is located at the time reported by PNE2SAC as being the source of the discontinuity. It should show up as either a horizontal line, or as a line segment that moves up the graph, (meaning it goes backwards in time). Fix these problem points, re-export the Wavetrack output, rebuild the header, and re-run the waveform with PNE2SAC.

Figure 16: Example of two instances of how a discontinuity occurs in Wavetrack export files.

### Navigating the PNE2SAC graphical output

A screenshot of a cell phone

Description automatically generatedOnce all discontinuities are removed from the Wavetrack output file, PNE2SAC will display a composite image that is free of black vertical lines. What is shown, is a diagnostics plot of three variables(Figure 17):

Figure 17: PNE2SAC final output, showing original Wavetrack output, with click points and PCHIP interpolated waveform for comparison and quality assessment.

(Blue dots) Original Wavetrack Output. These are the linearly interpolated points from the output file.

(Green dots) Recovered Click points. These represent a slope change for a line segment.

(Red lines) PChip-applied interpolated seismogram. This waveform is generated from the click points.

You can zoom in on a specific part of the seismogram by clicking on the magnifying glass in the figure, then drawing a box around the waveform segment that you wish to inspect.

A screenshot of a cell phone

Description automatically generatedNote how the PCHIP interpolation attempts to smooth the transition between the line segments bounded by the click points (Figure 18). It provides a smoother derivative (velocity change) for the wave form that is more realistic to the motion of a seismometer and galvanometer, which because of mass and inertia, must obey the physical laws of motion. After inspection, you may save the graph by pressing on the disk icon if you desire, then close the graph in order to finish the conversion.

Figure 18: Zoomed-in view, showing the linear interpolation of original Wavetrack output versus PCHIP interpolated curve. Recovered click points are displayed as green dots.

### The final product: PNE2SAC SAC/Miniseed output file

Once the graph is displayed, PNE2SAC will have already generated both a SAC file and a Miniseed file representing the digitized waveform. The PCHIP interpolated waveform will generally bear a good likeness of the original seismogram, especially in terms of amplitude. The use of a manual digitization process enables the operator to use their seismic analysis experience to find the inflections within the blurry scan that would otherwise be missed by an automated curve fitting algorithm (Figure 19). PNE2SAC will store the output files in the same folder as the source file. The naming convention for these files was chosen to make them compatible with cataloging software for easy indexing when integrating the file into your seismic database. The naming format is:

Stationcode.Networkcode.channel\_location.channel\_name.YYYY.MM.DD.HH.MM.SS

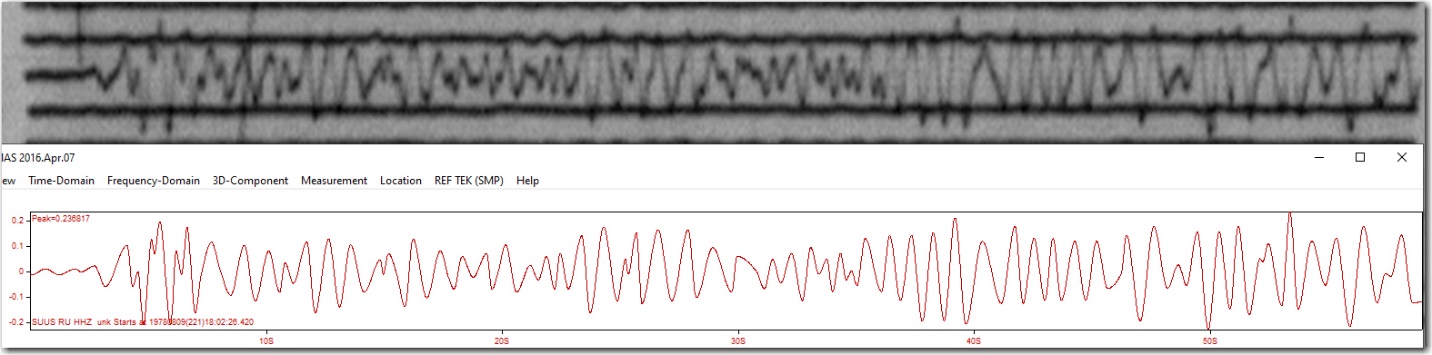
PNE2SAC leaves the channel location (typically blank, but can contain a number such as 00,01,02,etc.) blank by default.

Figure : From scanned seismogram to digitized seismogram: Kraton4, 09AUG1978, Station SUUS (Susuman), using DIMAS.