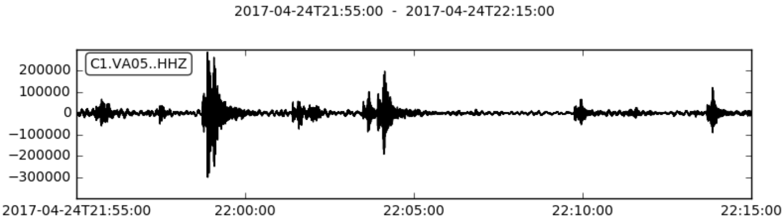
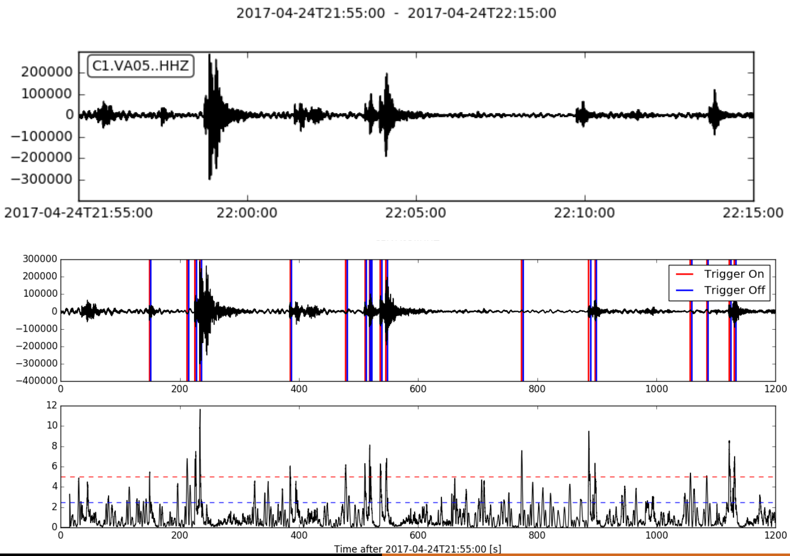
Earthquake detection and location: towards automatization Day2

Christian Sippl

Geofyzikální ústav Akademie věd ČR, v.v.i.

Universidad de Concepcion, January 13-17, 2020


2017-04-24T21:55:00 - 2017-04-24T22:15:00
C1.VA05..HHZ
200000
100000
0
−100000
−200000
−300000
2017-04-24T21:55:00
22:00:00
22:05:00
22:10:00
22:15:00


2017-04-24T21:55:00 - 2017-04-24T22:15:00
C1.VA05..HHZ
2017-04-24T21:55:00
22:00:00
22:05:00
22:10:00
22:15:00
Trigger On
Trigger Off
Time after 2017-04-24T21:55:00 [s]

- **Monday**: Intro and earthquake detection basics (triggers)
- **Tuesday**: Phase pickers and the earthquake association problem
- **Wednesday**: Basic earthquake location techniques, uncertainty estimation
- **Thursday**: Multi-event (re)location techniques, use of cross-correlations
- **Friday**: Putting it all together: how to design an automated approach

- **Monday**: Intro and earthquake detection basics (triggers)
- **Tuesday**: Phase pickers and the earthquake association problem
- **Wednesday**: Basic earthquake location techniques, uncertainty estimation
- **Thursday**: Multi-event (re)location techniques, use of cross-correlations
- **Friday**: Putting it all together: how to design an automated approach

Tuesday: The earthquake association problem, and use of targeted phase pickers

- *9:00 - 10:30*: Arrival association and multistation triggering
- *11:00 - 12:30*: Exercises: Coincidence trigger(s)
- *13:30 - 15:15*: Targeted phase pickers, S-wave picking
- *15:30 - 17:00*: Exercises: Building a polarization picker

Day2: Goals

1. You know different approaches of associating station picks to events
2. You know the basics of targetted P- and S-phase pickers
3. You have practical experience in using coincidence triggers and S-phase pick estimation

Central question: how to define events from lists of trigger alerts?

Basic problems:

- False alerts and missed onsets
- Phase misidentifications
- Overlapping events, teleseisms etc.

Partial solution: looking not at single station but at larger network

Earthquake signal should be coherent over multiple stations, most noise sources are local

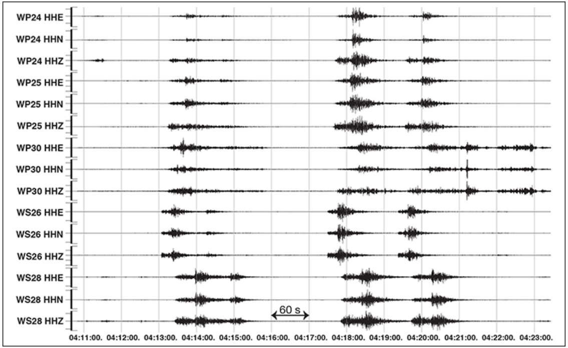
General approaches from simple to complicated:

Arrival time association

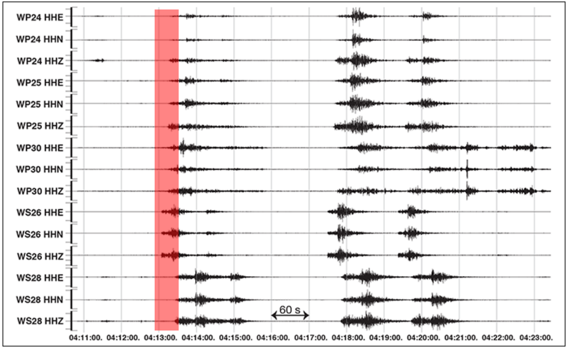
1. Coincidences (= time window)
2. Traveltime grid (= raytracing)
3. Wavefront methods

1-3: also increasing area (local to global) possible

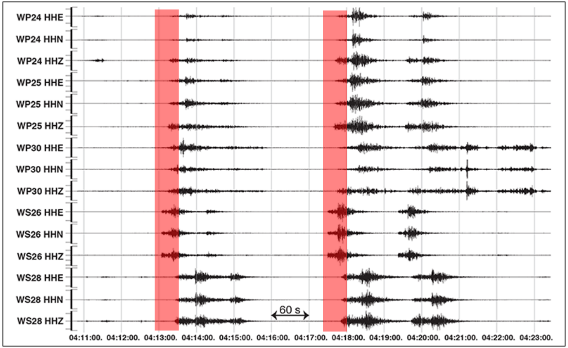
- Define time window length and minimum number of stations
- If at least this amount of stations has trigger alert in such a time window, event is defined
- Problems: Are phases consistent? Only works for small arrays



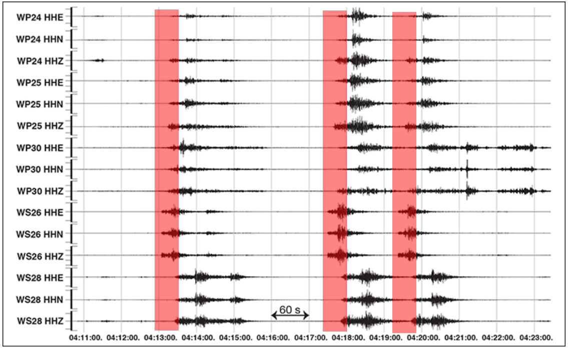
- Define time window length and minimum number of stations
- If at least this amount of stations has trigger alert in such a time window, event is defined
- Problems: Are phases consistent? Only works for small arrays

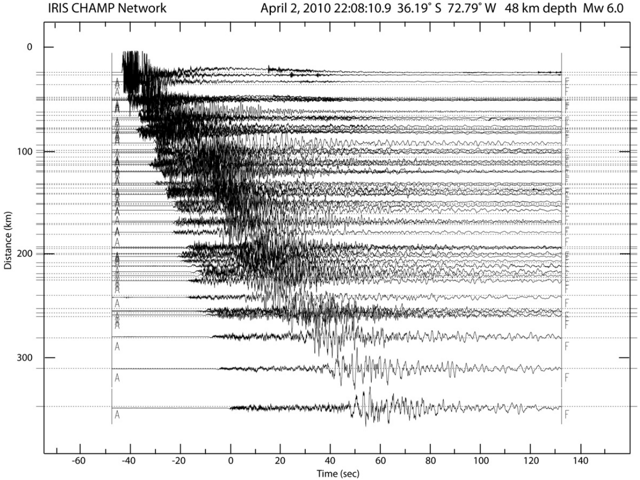


- Define time window length and minimum number of stations
- If at least this amount of stations has trigger alert in such a time window, event is defined
- Problems: Are phases consistent? Only works for small arrays



- Define time window length and minimum number of stations
- If at least this amount of stations has trigger alert in such a time window, event is defined
- Problems: Are phases consistent? Only works for small arrays





Static time window is not appropriate if stations are far apart (or: time window gets larger and larger, may catch other signals)

+ Simple and (for local networks) quite effective
+ Can be directly coupled with the different trigger algorithms
+ Computationally cheap
- Becomes problematic for larger (regional-scale) regions
- Location and origin time estimates are rather rough (closest station)
- Requires "clean" alert input

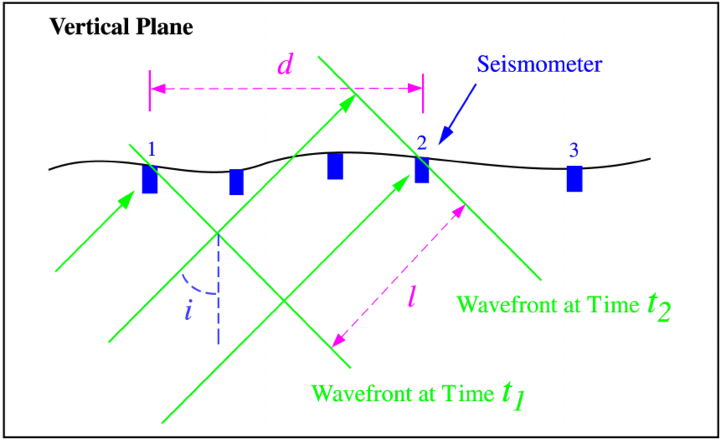
Question

What is necessary to overcome the main deficiency of the coincidence method?

Question

What is necessary to overcome the main deficiency of the coincidence method?

Instead of assuming arrivals at roughly the same time, we should look for arrivals with a consistent velocity



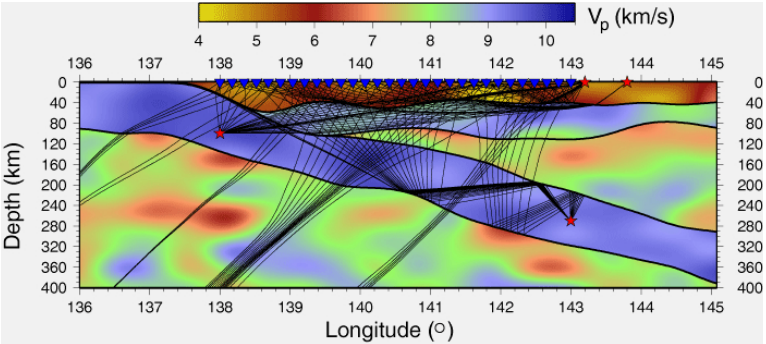
If absolute velocity is V, apparent velocity is $V/\cos\theta$ (always faster, infinity for teleseismic event from directly below)

Instead of static time window, one could write an algorithm that:

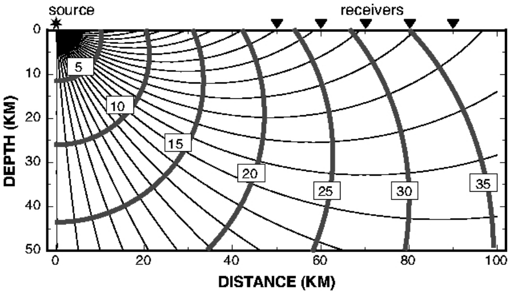
1. identifies potential “groups” of alerts
2. Chooses the closest station (=station with earliest alert)
3. Only retains alerts in a certain corridor of “allowed” apparent velocities

But: apparent velocity depends highly on depth $\rightarrow$ it is hard to determine a meaningful velocity corridor (or at least upper boundary)
more sophisticated method: traveltime grids

Forward calculation, i.e. get theoretical traveltimes for given location and velocity model
Gives ONLY first arrivals (=search for quickest, not shortest, path)!



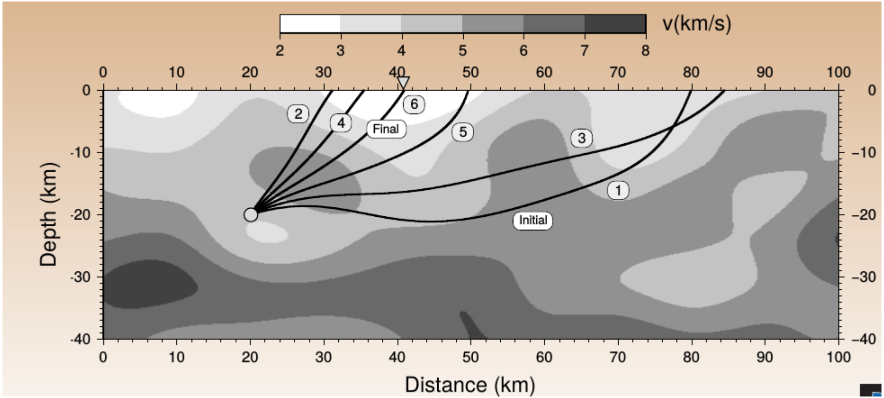
- Assumption of point source and (usually layered) velocity model
- Seismic waves propagation occurs on a spherical surface (distorted if velocity model is not homogeneous)
- First arrivals can be described as “rays” that are normal to this spherical surface



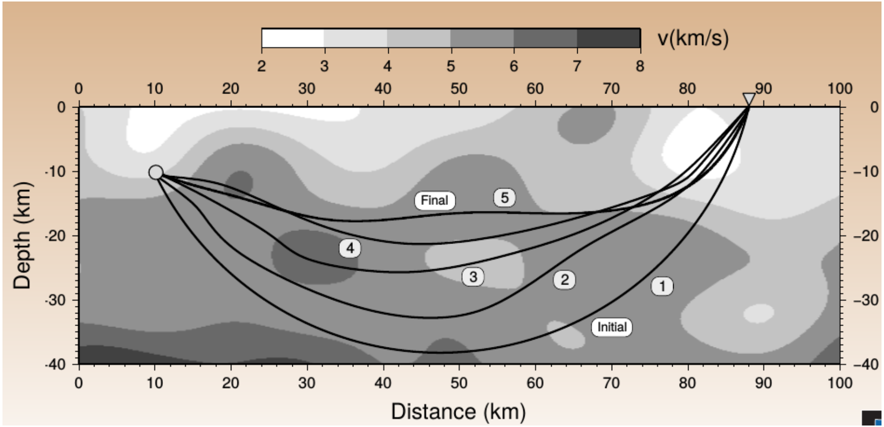
Attention: ray tracing is high-frequency approximation!!

Shooting Methods:

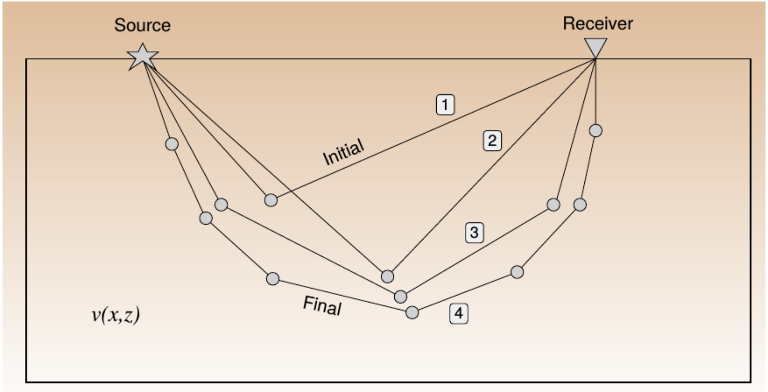
Shoot rays in the general direction and compare travel times



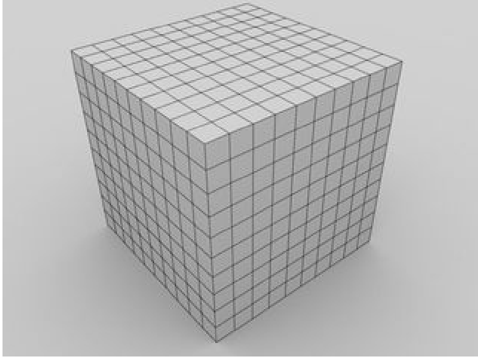
Bending methods:
Assume a rough "first guess" raypath that connects source and receiver; keep ends fixed and perturb ("bend") raypath, minimizing traveltime



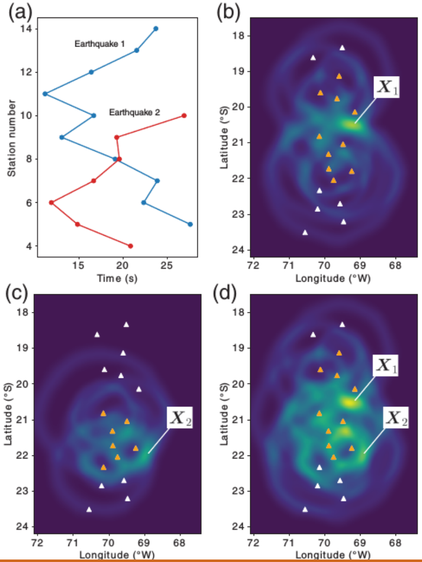
In practice, most methods today use schemes that combine the two concepts in different ways; one example: pseudo-bending

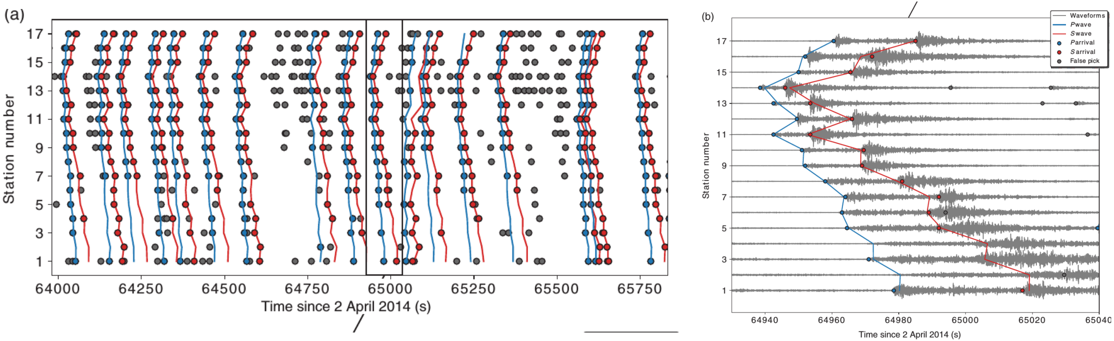


- Subsurface is discretized into a grid of nodes
- These nodes are potential sources, traveltimes from each node to all stations are computed with ray tracing
- Then: input of trigger list; algorithm compares traveltime patterns to pre-computed ones
- Subsets of trigger alerts that fit to traveltime patterns from specific nodes are kept as "events", rest is discarded



- Takes list of picks as input
- Entire waveform is backpropagated, and this is repeated for all stations
- Regional grid: all nodes are checked for coherency, maximum should be approximate earthquake location
- Claim: can also handle simultaneous events (see figure)





Each grey dot is a pick; blue: associated as P; red: associated as S ($\rightarrow$ pair blue+red is event)

```
from obspy.signal import trigger
output = trigger.coincidence_trigger('recstalta',5.,
1,in_stream,5,details=True,sta=0.8,lta=12.)
#Input: trigger type,trigger threshold,detrigger threshold,input
    Stream,coincidence sum)
```

How can we construct an input Stream containing all necessary Traces?

```
from obspy.core import Stream, UTCDateTime
from obspy.clients.fdsn import Client

client = Client('GFZ')
t1 = UTCDateTime(2015,1,1,0,0,0)
st = Stream()
stations = ['PATCX','HMBCX','PB01','PB02','PB03','PB04','PB07','PB09']
for j in stations:
    tr = client1.get_waveforms('CX',j,'--','HHZ',t1,t1+86400.)
    st.append(tr[0])
print(st)

#8 Trace(s) in Stream:

#CX.HMBCX..HHZ | 2014-12-31T23:59:59.680000Z - 2015-01-02T00:00:03.720000Z | 100.0
    Hz, 8640405 samples
```

```
print(type(output))
#list
print(len(output))
5
#Output is a list, each element is one ‘‘event’’
print(type((output[0]))
#dict
```

Accessing parts of this dictionary

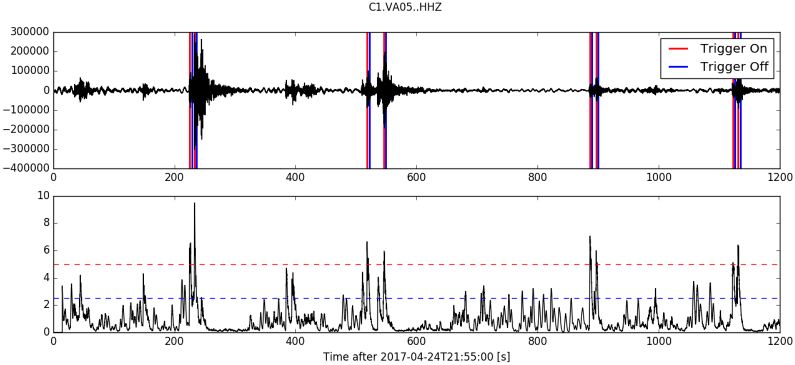
```
print output[0].keys()
#[u'coincidence_sum', u'cft_std_wmean', u'similarity',
# u'stations', u'trace_ids', u'cft_stds', u'time',
# u'duration', u'cft_peak_wmean', u'cft_peaks']
print output[0]['cft_peaks']
#[13.662237430293125, 12.937480467513232, 8.697322694813776, 11.148089376098383,
   6.396752460694139, 3.527443856751875, 3.9680128002287374]
```

Exercises

5. Coincidence trigger

- Download vertical component data for January 1, 2015 from stations PATCX, HMBCX, PB01, PB02, PB03, PB04, PB07 and PB09 from the GEOFON data server (network code CX).
- Put all data traces into one Stream object and apply the coincidence trigger. Adjust the value for coincidence_sum so that you get a good number of events (maybe between 15 and 30).
- Extract approximate origin time and the stations with alerts
- Now we want determine exact arrival times for each station. For this, we can apply a simple STA/LTA on a small time window (discuss strategy). Try this out using plot_trigger() to check if picks are OK.
- Finally, write these picks to a file. Modify your script so it can run in a loop (over all events found by the coincidence trigger).

We have discussed and tried out triggers to find earthquakes; but: these are usually not phase-specific, plus arrival times are imprecise

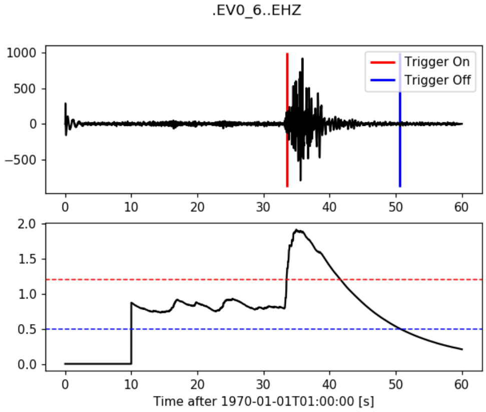


→ Targeted phase pickers: Derive precise phase arrival plus quality rating often not operating on raw data but on pre-defined time windows

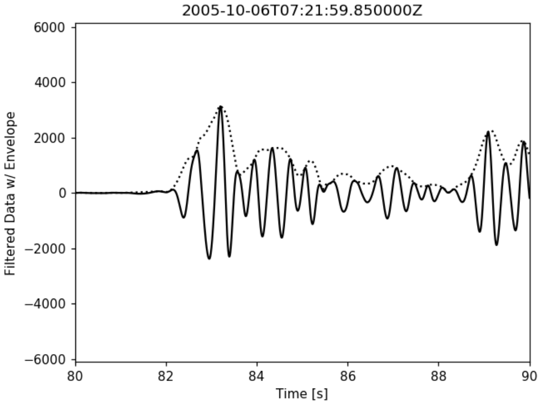
STA/LTA alerts are not very reliable (false alerts), and nearly always late

→They should only be used as "first guesses", and be refined and checked afterwards.

Targeted phase pickers are more precise and usually supply quality weighting (=uncertainty estimate)

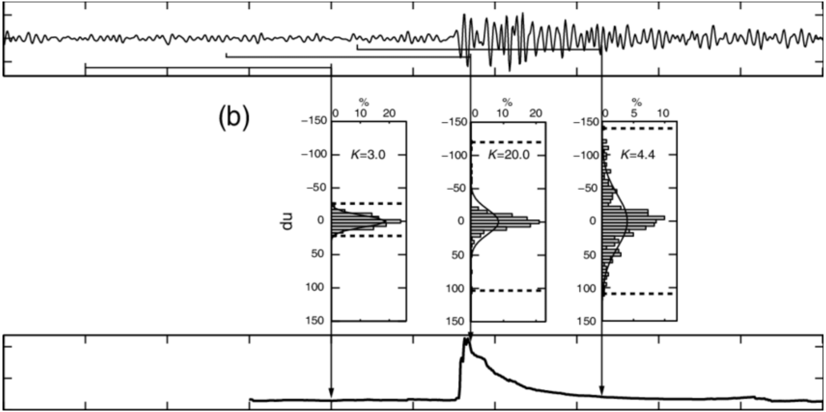


comparable to STA/LTA on envelope function (show what that is)



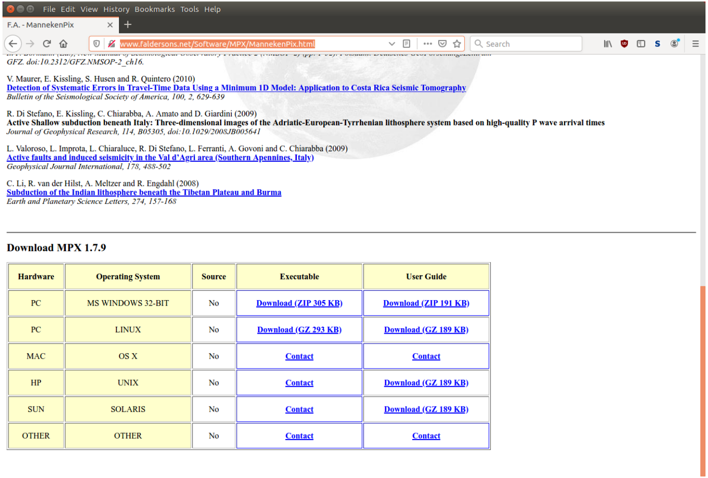
takes typical STA/LTA-delay into account; uses mostly spectral parameters to define weighting

Wrapper is available in obspy.signal.trigger



Kurtosis is a statistical measure that can be roughly described as “tailedness” of a distribution (i.e. presence of outliers → higher kurtosis)

- Searches for P arrival in pre-defined window
- Uses adaptive Wiener pre-filtering, then a modified version of the Baer picker
- Weighting engine based on Fisher statistics, can be calibrated with handpicked dataset
- Problem: only executable is available, no source code



GFZ. doi:10.2312/GFZ.NMSOP-2_ch16.

V. Maurer, E. Kissling, S. Husen and R. Quintero (2010)
Detection of Systematic Errors in Travel-Time Data Using a Minimum 1D Model: Application to Costa Rica Seismic Tomography
Bulletin of the Seismological Society of America, 100, 2, 629-639

R. Di Stefano, E. Kissling, C. Chiarabba, A. Amato and D. Giardini (2009)
Active Shallow subduction beneath Italy: Three-dimensional images of the Adriatic-European-Tyrrhenian lithosphere system based on high-quality P wave arrival times
Journal of Geophysical Research, 114, B05305, doi:10.1029/2008JB005641

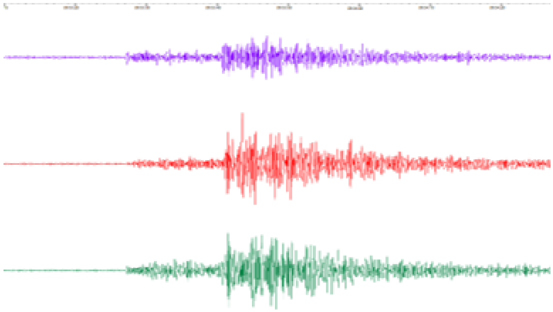
L. Valoroso, L. Improta, L. Chiaraluce, R. Di Stefano, L. Ferranti, A. Govoni and C. Chiarabba (2009)
Active faults and induced seismicity in the Val d'Agri area (Southern Apennines, Italy)
Geophysical Journal International, 178, 488-502

C. Li, R. van der Hilst, A. Meltzer and R. Engdahl (2008)
Subduction of the Indian lithosphere beneath the Tibetan Plateau and Burma
Earth and Planetary Science Letters, 274, 157-168

Download MPX 1.7.9

Hardware	Operating System	Source	Executable	User Guide
PC	MS WINDOWS 32-BIT	No	Download (ZIP 305 KB)	Download (ZIP 191 KB)
PC	LINUX	No	Download (GZ 293 KB)	Download (GZ 189 KB)
MAC	OS X	No	Contact	Contact
HP	UNIX	No	Contact	Download (GZ 189 KB)
SUN	SOLARIS	No	Contact	Download (GZ 189 KB)
OTHER	OTHER	No	Contact	Contact

http://www.faldersons.net/Software/MPX/MannekenPix.html



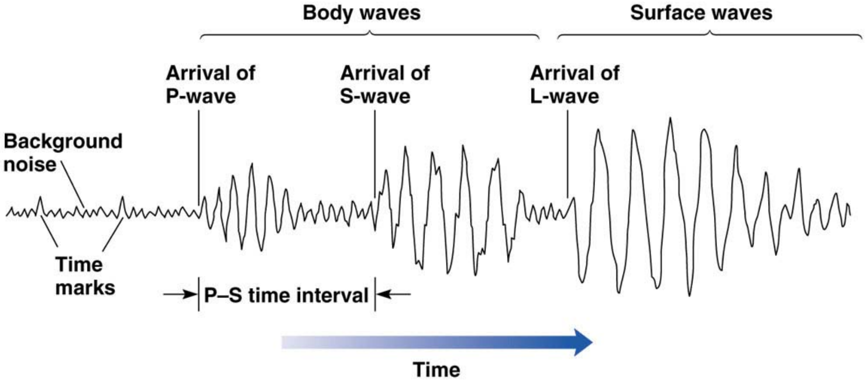
S-picking is nearly always done relative to P (if P exists, look for S in time window after P)
Problem: S-arrival is inside P-wave coda

Question

What changes from P coda to S onset?

Amplitude

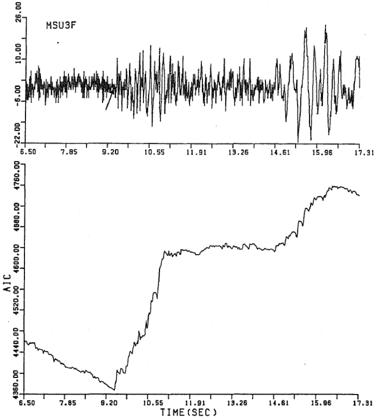
Simple but often functional approach: STA/LTA on horizontals in fixed time window after P



(a)

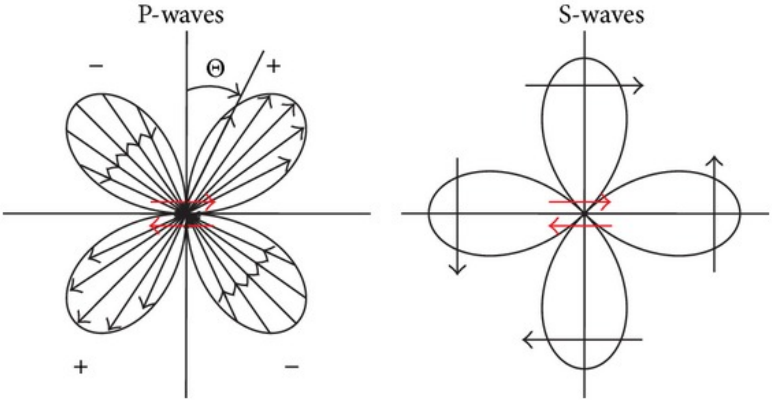
Spectral content

S-waves are usually lower-frequent than P-wave energy



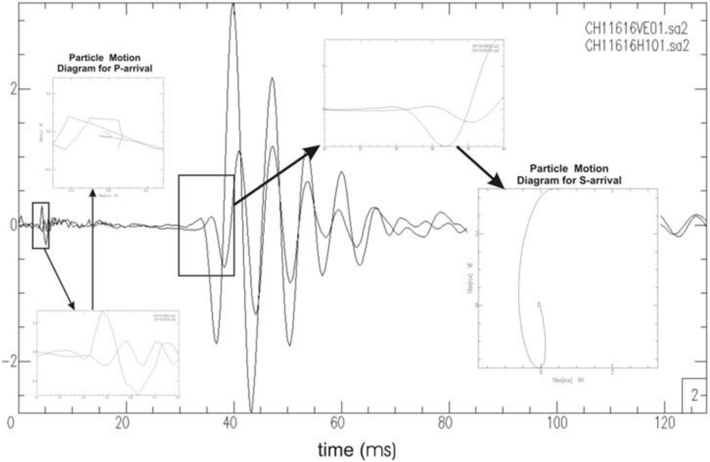
As for P-phases, there are autoregressive S-wave pickers

Polarization

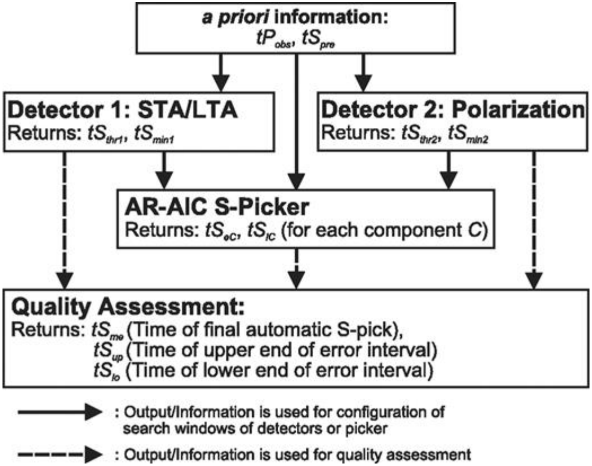


Double-couple radiation patterns, amplitudes are opposite

Linear polarization (in propagation direction) for P, perpendicular to propagation direction for S



Task: continuously measure this polarization direction along the waveform, pick changepoint (=S onset)



- Probably the most sophisticated algorithm for S picking at the moment
- Combines three different approaches (STA/LTA, polarization, AIC)
- Quality weighting by evaluating consistency between the different methods

Exercises

6. S-picker

The next step in automatically detecting and location earthquakes is finding S arrivals.

Take a single test event from the event catalog you built in the morning, and take a time window from the P-pick to 90 seconds after it. On this time window, try out applying an STA/LTA on one of the horizontal components, and also try to evaluate the amplitude ratio between horizontal and vertical trace against time. Does one of these (or both) give a clear indication for S? Extend the analysis to more event (from the catalog), and debate what could best be used as an S-wave detector. Write such a detector that adds an S-pick to your events.