

Entrée [1]:

```

1 %load_ext autoreload
2 %autoreload 2
3 import numpy as np
4 from axitra import *
```

Entrée [2]:

```

1 # 2 sources
2 # index, lat, lon, depth
3 sources=np.array([[1, 45.100, 2.000, 5000.000],
4                   [2, 45.000, 2.000, 5000.000],
```

Entrée [3]:

```

1 # 5 receivers with geographical coordinates
2 # index, lat, lon, depth
3 stations=np.array(
4     [[1, 45.000, 2.000, 0.000],
5      [2, 46.000, 1.000, 0.000],
6      [3, 46.000, 3.000, 0.000],
7      [4, 44.000, 1.000, 0.000],
```

Entrée [4]:

```

1 # 2 layers
2 # thickness (or top), Vp, Vs, rho, Qp, Qs
3 model = np.array([[1000., 5000., 2886., 2700., 1000., 500.],
4                   [2000., 5000., 2886., 2700., 1000., 500.]])
```

Entrée [6]:

```

1 # Compute green's function
2 # fmax = 20Hz
3 # duration = 50 sec
4 # create class for parameters
5 ap = Axitra(model,stations,sources,duration=50.,fmax=20.,latlon=True,axpath=
6
7 #run the Green's function calculation
```

../src/axitra ran sucessfully

Entrée [7]:

```

1 # history of source
2 # index, fx_amp, fy_amp, fz_amp, total_amplitude, time_delay
3 hist = np.array([[1,1.,0.,1.,10.,10.0],
4                  [2,1.,0.,1.,10.,10.0],
```

Entrée [8]:

```

x1= 213730.00000000003  duration= 50.0  nfreq= 1000  fmax= 20.0  path_to_binary=
../src
nsource= 2  nstat= 5  id= 180
```

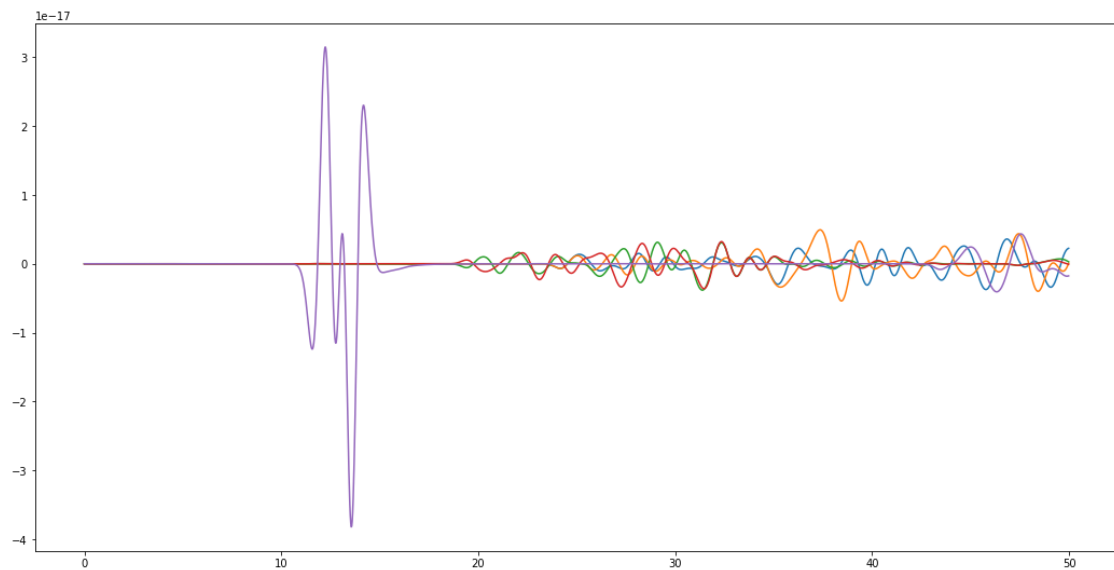
Entrée [9]:

```

1 # first convolution example
2 # source= ricker
3 # source time width = 3 sec
4 # output unit = displacement
```

Entrée [10]:

```
1 pt.figure(figsize=(18, 9))
```

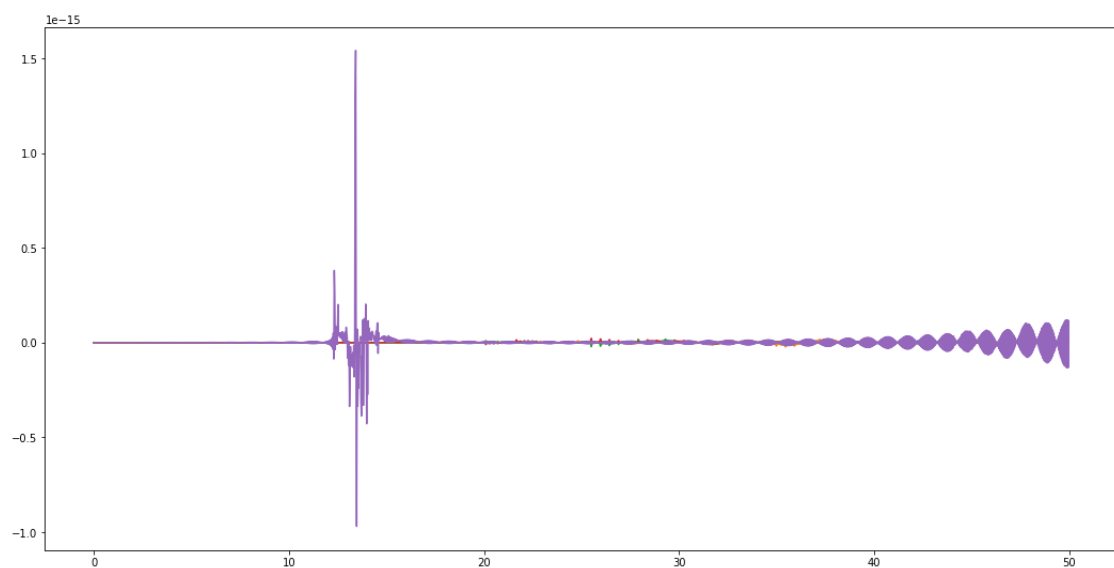


Entrée [11]:

```
1 # convolution with a user provided source function
2 # A dirac of unit 1 at 10th sample
3 sfunc=np.zeros((ap.npt,),dtype='float64')
4 sfunc[10]=1.
```

Entrée [12]:

```
1 pt.figure(figsize=(18, 9))
```



Entrée [13]:

```
1 # Run a new instance by reading existing axitra input files
2 ap2=Axitra.read(str(ap.id),axpath='../src')
3 # clean files associated with previous instance
```

Entrée [14]:

```
1 # Run the second instance that should give identical results
2 force.green(ap2);
```

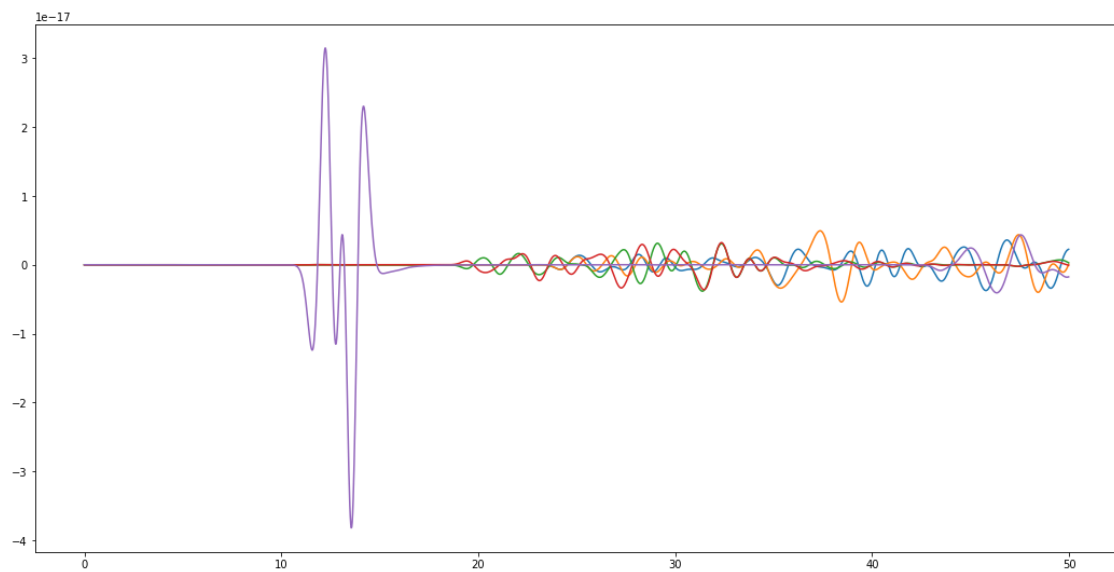
../src/axitra ran successfully

Entrée [15]:

```

1 #clean all files o disk relative to this example
2
3 pt.figure(figsize=(18, 9))

```



Entrée [16]:

Help on method green in module axitra:

green(axitra\_param) method of builtins.type instance

Compute the Green's functions for the set of parameters supplied and for for ce sources

The (3 x nfreq x nsources x nstations) Green's functions are stored in a fil e on disk, the next step is to

convolve them with the source function(s) using moment\_conv() in order to ob tain the

seismograms.

This function write several input files on disk and call the fortran program "axitra" according

to the path 'axpath'. The input/output files are of the form "axi\_???.suffi x" where axi\_??? can be obtain

from the returned Axitra class instance by class.sid

Input parameters:

- axitra\_param = an instance of Axitra class

Return:

- axitra\_param = a copy of the instance of Axitra class

Entrée [17]:

Help on method conv in module axitra:

conv(ap, hist, source\_type, t0, t1=0.0, unit=1, sfunc=None) method of builtins.type instance

Compute the convolution of Green's function obtained by a previous call to moment\_green() by source time functions.

Input:

ap = Axitra class instance from moment\_green

hist = source history array (nsource x 6)

index, moment(Nm), strike, dip, rake, 0., 0., delay

index, slip, strike, dip, rake, width, height, delay

source\_type =

0 : Dirac

1 : Ricker

2 : step

- - - - -