

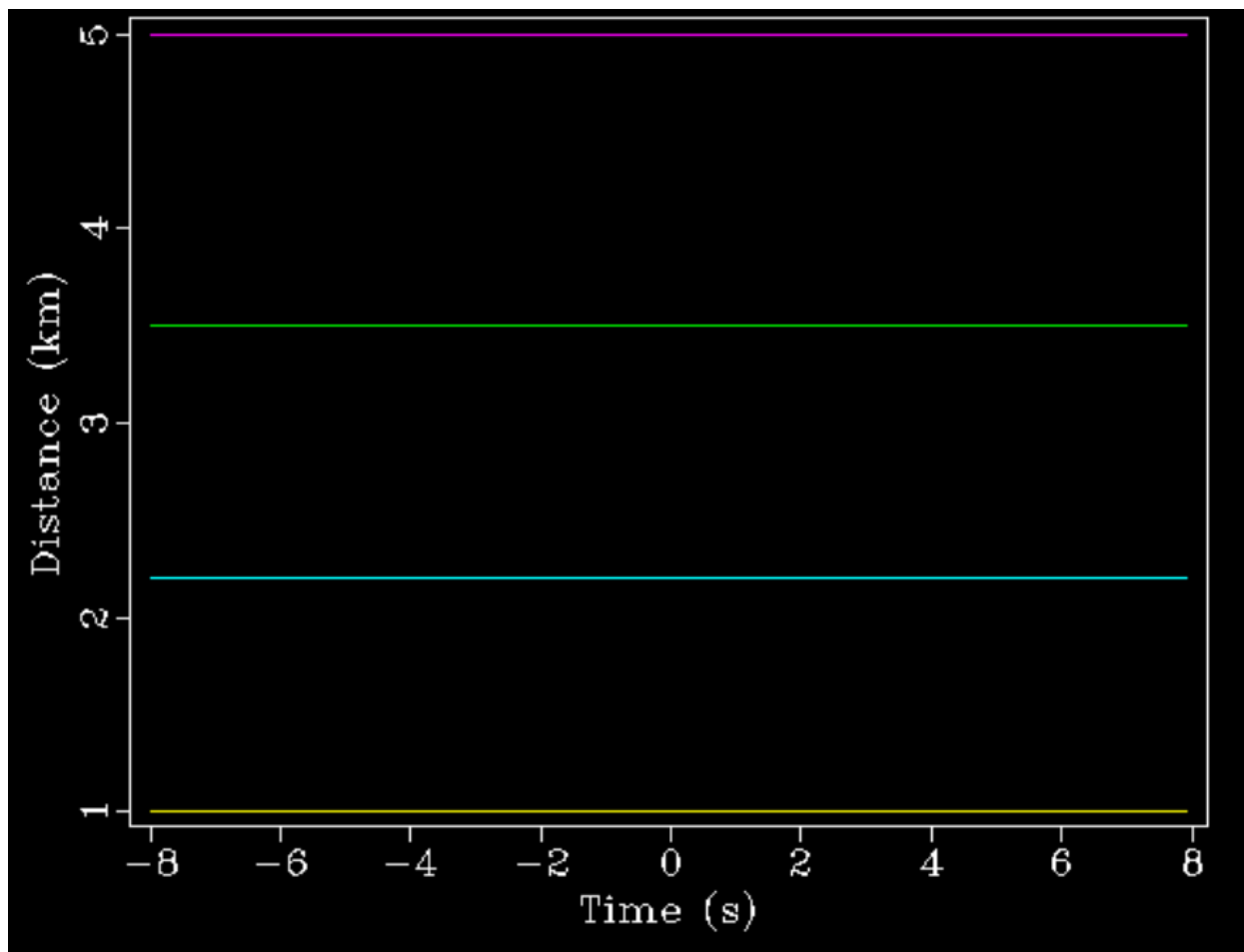
一、spike:生成脉冲序列

1. `n1=200` :生成的脉冲序列的长度
2. `o1=-8` :脉冲序列的起始值
3. `d1=0.08` :脉冲序列的采样间隔
4. `n2=4` :生成几个脉冲序列
5. `nsp=4` :指定脉冲的数量
6. `k2=1,2,3,4` :指定每个脉冲序列的第几个脉冲被赋予不同的振幅
7. `mag=1,2.2,3.5,5` :指定每个脉冲的振幅值

```
# 菲涅耳-基尔霍夫衍射模型
Flow('modl',None,
    ...
    spike n1=200 o1=-8 d1=0.08 n2=4
    nsp=4 k2=1,2,3,4 mag=1,2.2,3.5,5
    '')
```

```
llp@VM-4-3-ubuntu:~/Paper/AVO/stack_of_seismic_AVO_data_using_hybrid_AB_semlance$ sfin modl.rs
modl.rs:
  in="/var/tmp/Paper/AVO/stack_of_seismic_AVO_data_using_hybrid_AB_semlance/modl.rs"
  esize=4 type=float form=native
  n1=200      d1=0.08      o1=-8      label1="Time" unit1="s"
  n2=4        d2=0.1       o2=0       label2="Distance" unit2="km"
  800 elements 3200 bytes
llp@VM-4-3-ubuntu:~/Paper/AVO/stack_of_seismic_AVO_data_using_hybrid_AB_semlance$
```

modl.rs



modl.vpl

- FROM: <https://www.reproducibility.org/RSF/book/uh/avostack/synh.html>

二、window:截取数据

1. `n1=1` :将n1维度上的数据点设置为1
2. `n2`
3. `n3`

```

Flow('data','mod1 refl',
    ...

    kirmod nt=501 dt=0.004 freq=25 refl=${SOURCES[1]}
    ns=12 s0=1.4 ds=-0.1
    nh=64 h0=0.1 dh=0.1
    vel=5 gradz=2 type=v
    '')

Flow('off','data','window n1=1 | math output="0.5*x1" ')

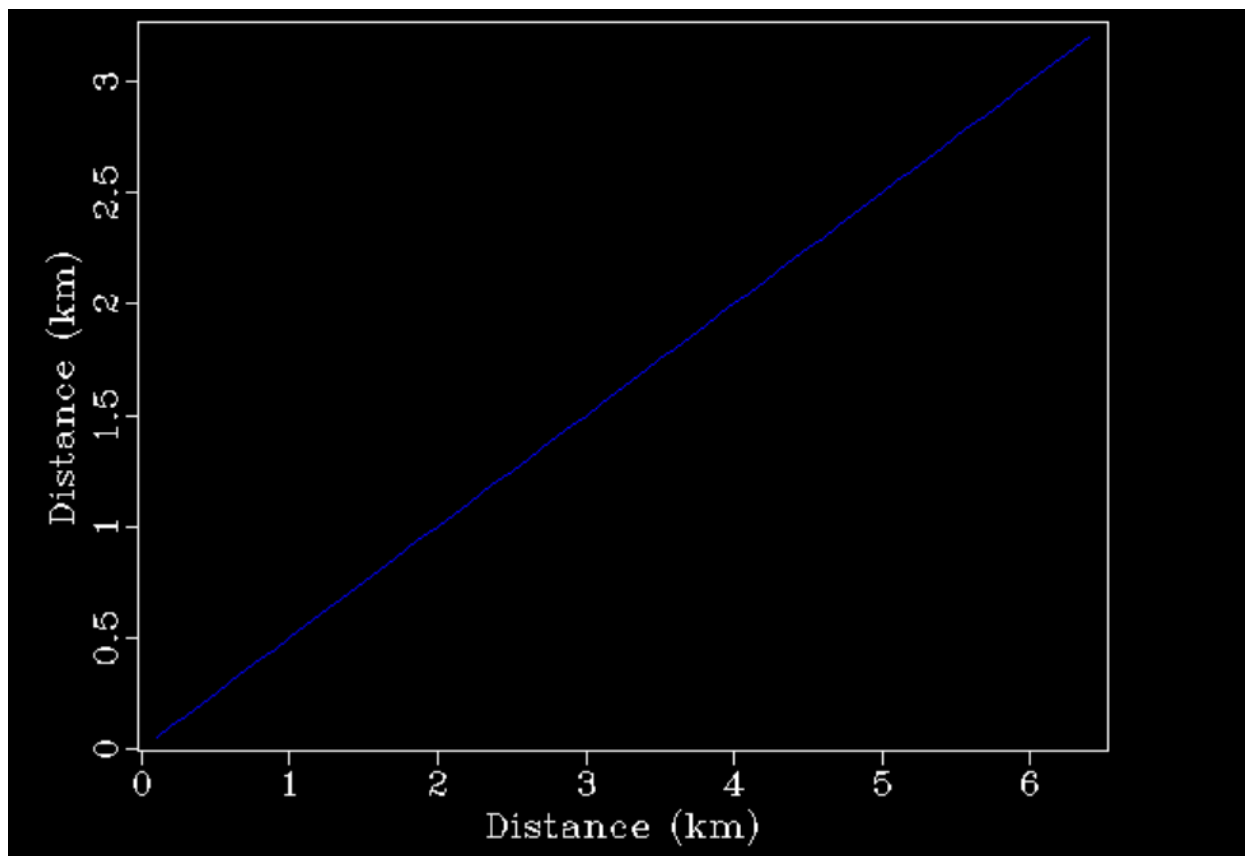
```

```

llp@VM-4-3-ubuntu:~/Paper/AVO/stack_of_seismic_AVO_data_using_hybrid_AB_semnlance$ sfin data.rs
data.rs:
  in="/var/tmp/Paper/AVO/stack_of_seismic_AVO_data_using_hybrid_AB_semnlance/data.rs@"
  esize=4 type=float form=native
  n1=501      d1=0.004      o1=0      label1="Time" unit1="s"
  n2=64       d2=0.1        o2=0.1     label2="Distance" unit2="km"
  n3=12       d3=-0.1       o3=1.4
  384768 elements 1539072 bytes
llp@VM-4-3-ubuntu:~/Paper/AVO/stack_of_seismic_AVO_data_using_hybrid_AB_semnlance$ sfin off.rs
off.rs:
  in="/var/tmp/Paper/AVO/stack_of_seismic_AVO_data_using_hybrid_AB_semnlance/off.rs@"
  esize=4 type=float form=native
  n1=64       d1=0.1        o1=0.1     label1="Distance" unit1="km"
  n2=12       d2=-0.1       o2=1.4     label2="Distance" unit2="km"
  n3=1        d3=0.004      o3=0      label3="Time" unit3="s"
  768 elements 3072 bytes
llp@VM-4-3-ubuntu:~/Paper/AVO/stack_of_seismic_AVO_data_using_hybrid_AB_semnlance$ █

```

data.rs off.rs



off.vpl

- FROM: <https://www.reproducibility.org/RSF/book/uh/avostack/synh.html>

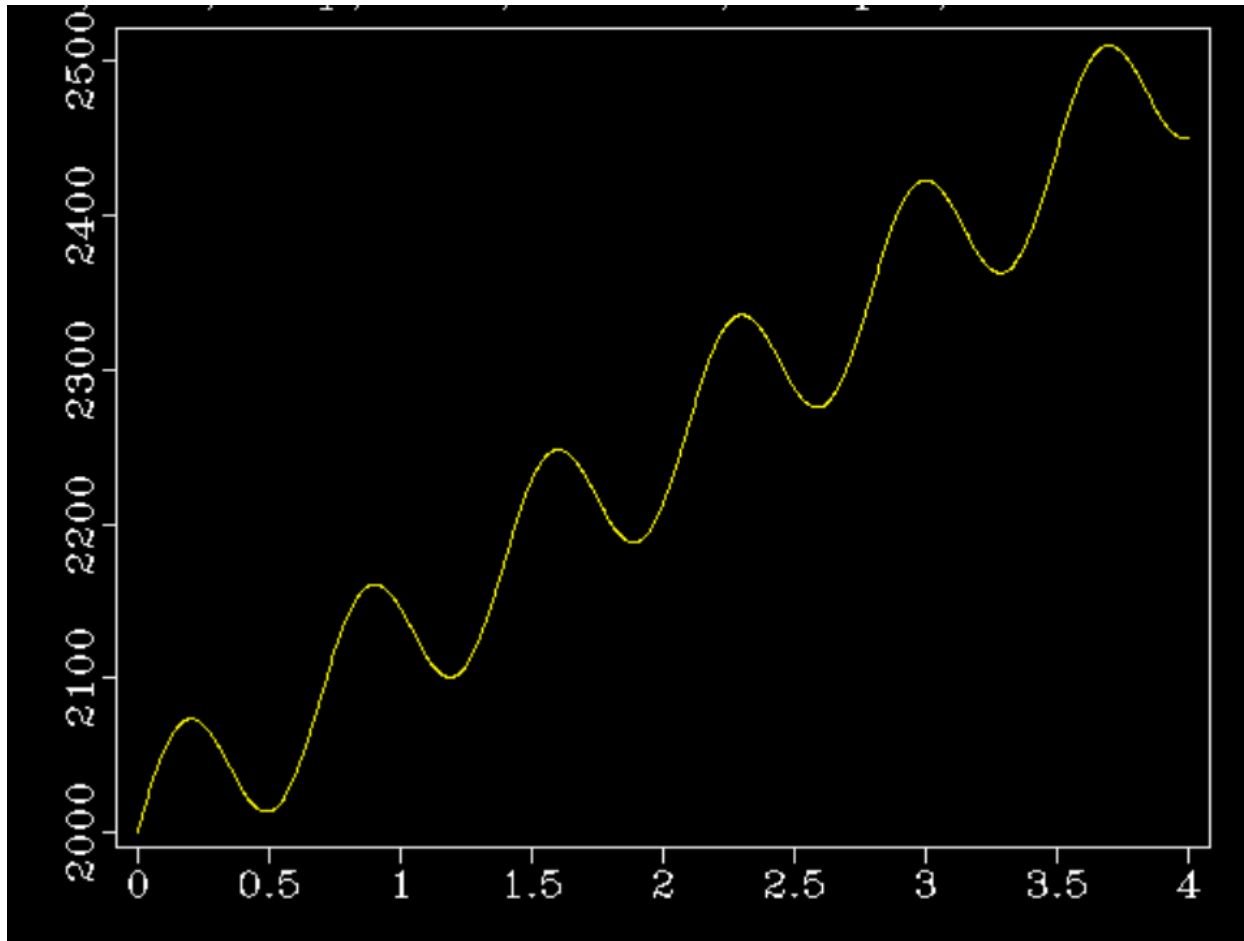
三、math数学运算

1. `d1=0.004` :采样间隔
2. `n1=1001` :生成的数据点数
3. `o1=0` :数据的起始点
4. `output="x1*125+2000+50*sin(9*x1)"` :数学表达式

```
Flow('vrms',None,  
     'math d1=0.004 n1=1001 o1=0 output="x1*125+2000+50*sin(9*x1)" ')
```

```
llp@VM-4-3-ubuntu:~/RSFSRC/book/tccs/svscan/simple$ sfin vrms.rs
vrms.rs:
  in="/var/tmp/tccs/svscan/simple/vrms.rs@"
  esize=4 type=float form=native
  n1=1001      d1=0.004      o1=0
    1001 elements 4004 bytes
llp@VM-4-3-ubuntu:~/RSFSRC/book/tccs/svscan/simple$
```

vrms.rs



vrms.vpl

- FROM: <https://www.reproducibility.org/RSF/book/tccs/svscan/simple.html>

四、noise:生成噪声

1. `rep=y` :允许重复使用相同的噪声

2. `seed=2006` :随机数生成的种子

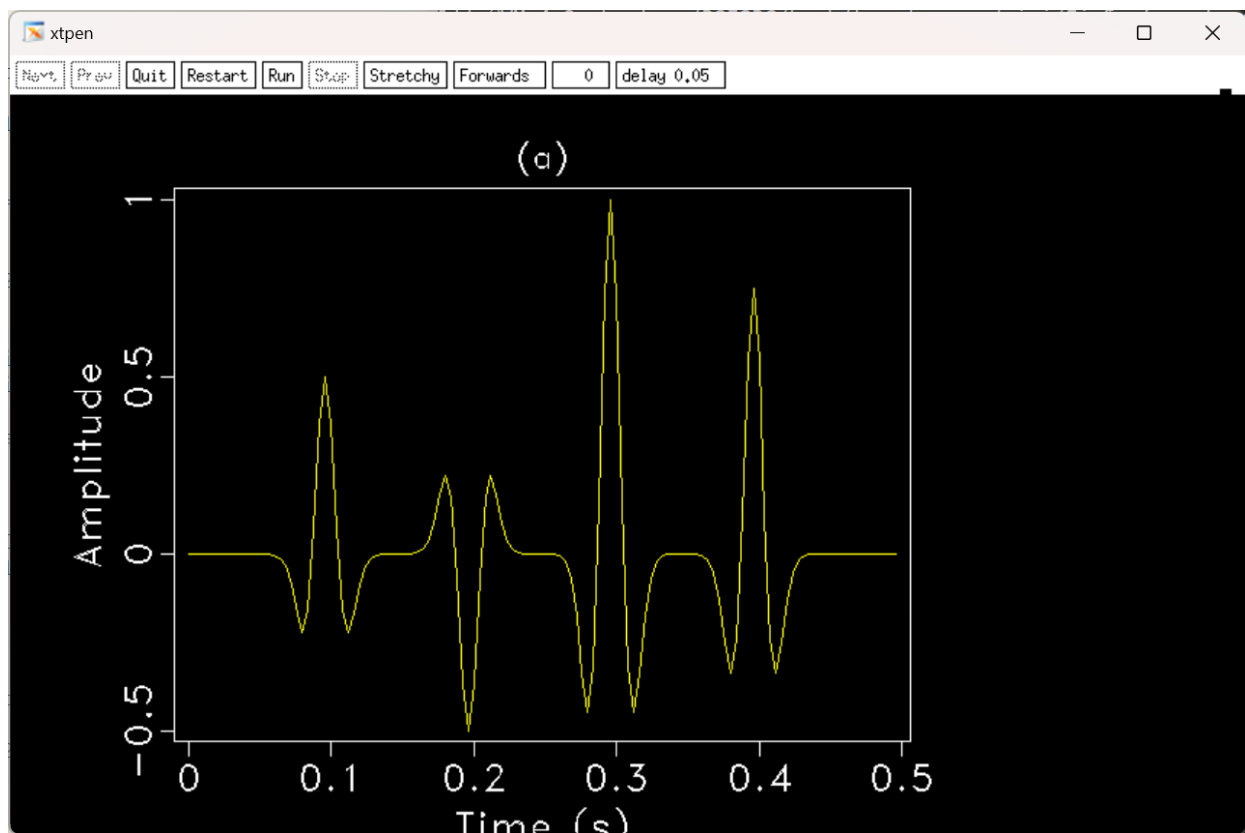
3. `var=1e-20` :方差

```
Flow('trace',None,'spike n1=125 nsp=4 mag=1,-1,2,1.5 k1=25,50,75,100 |  
ricker1 frequency=25 | scale axis=1')
```

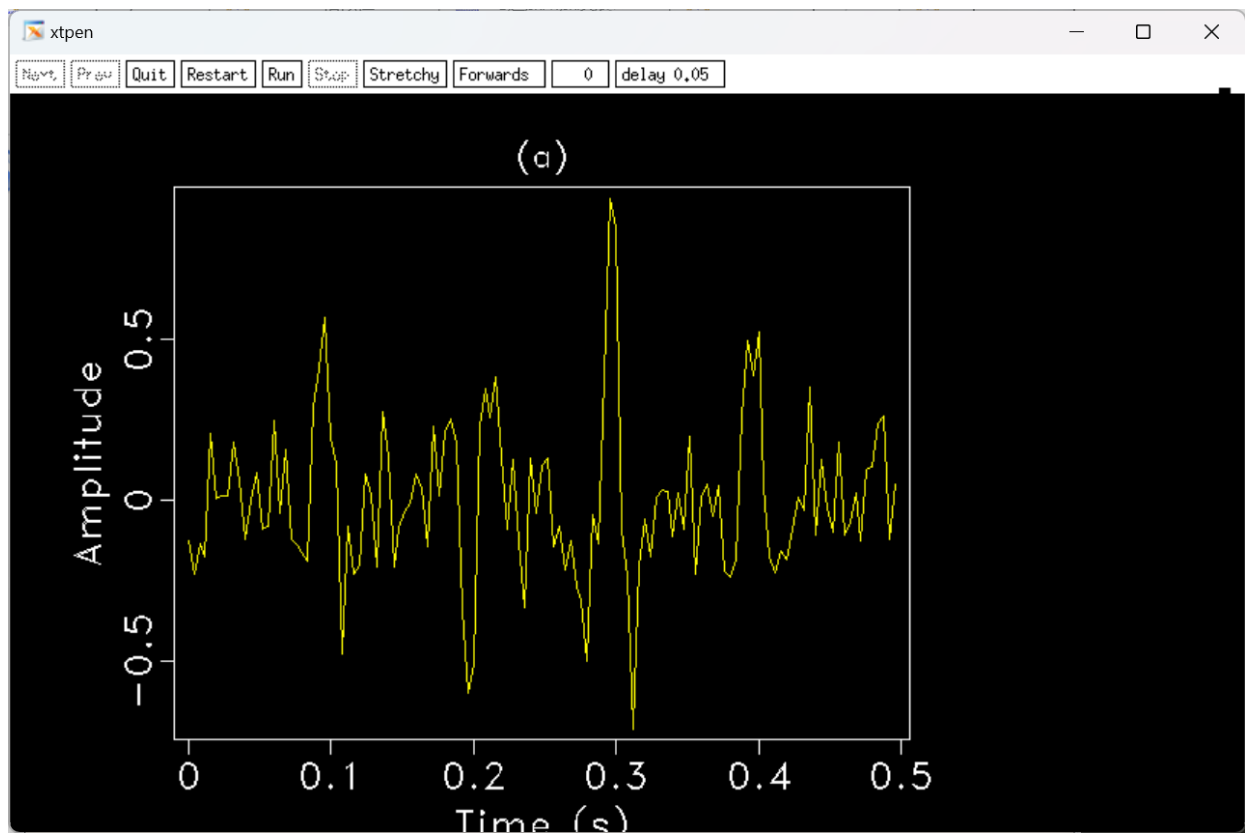
```
Graph('trace','title="(a)"')
```

```
Flow('trace1','trace','noise var=0.02 seed=2014')
```

```
Graph('trace1','title="(a)"')
```



未加噪音的



添加了噪声的`trace1`

- FROM: <https://reproducibility.org/RSF/book/tccs/svscan/simi.html>

五、`bandpass`:带通滤波

1. `fhi=4` :表示低频截止频率为4Hz
 2. `flo=20` :表示高频截止频率为20Hz
- FROM: <https://reproducibility.org/RSF/book/tccs/svscan/class.html>

六、`cut`:将数据集的一部分归零

1. `n1=100` :指定n1维度的0~100的值为零

与`window`函数相反

- FROM: <https://reproducibility.org/RSF/book/tccs/svscan/class.html>

七、**spray**:通过在指定的轴维度扩展数据集

- **axis=2** :表示在数据的第二个维度上添加偏移信息。
- **n=256** :指定生成的偏移数量为256。
- **d=25** :指定偏移之间的间隔为25。
- **o=-3200** :指定第一个偏移的起始值为-3200。
- **label=Offset** :设置偏移的标签为 'Offset'。
- **unit=m** :设置偏移的单位为米。

```
Flow('synt',None,
    ...

    spike d1=0.004 n1=1001 |
    noise rep=y seed=2006 |
    cut n1=100 |
    bandpass flo=4 fhi=20 |
    spray axis=2 n=256 d=25 o=-3200 label=Offset unit=m
    ...)
```

```
llp@VM-4-3-ubuntu:~/RSFSRC/book/tccs/svscan/simple$ sfin synt.rsf
synt.rsf:
  in="/var/tmp/tccs/svscan/simple/synt.rsf@"
  esize=4 type=float form=native
  n1=1001      d1=0.004      o1=0      label1="Time" unit1="s"
  n2=256      d2=25        o2=-3200    label2="Offset" unit2="m"
  n3=1        d3=?        o3=?
      256256 elements 1025024 bytes
llp@VM-4-3-ubuntu:~/RSFSRC/book/tccs/svscan/simple$
```

synt.rsf

- FROM: <https://reproducibility.org/RSF/book/tccs/svscan/simple.html>

八、kirmod:使用解析格林函数进行 Kirchhoff 2-D/2.5-D 建模

nt=501：生成的地震数据的时间点数为501。

dt=0.004：时间采样间隔为0.004秒。

freq=25：地震数据的主频率为25 Hz。

refl=\${SOURCES[1]}：反射数据来源于第二个输入（**refl**）。

ns=12：地震数据中的源数量为12。

s0=1.4：第一个源的位置为1.4。

ds=-0.1：源之间的距离为-0.1。

nh=64：地震数据中的检波器（接收器）数量为64。

h0=0.1：第一个检波器的位置为0.1。

dh=0.1：检波器之间的距离为0.1。

vel=5：地下介质的波速为5。

gradz=2：地下介质的梯度为2。

type=v：地震数据的类型为速度。

- FROM: <https://www.reproducibility.org/RSF/book/uh/avostack/synh.html>

九、ricker1:与Ricker的小波卷积

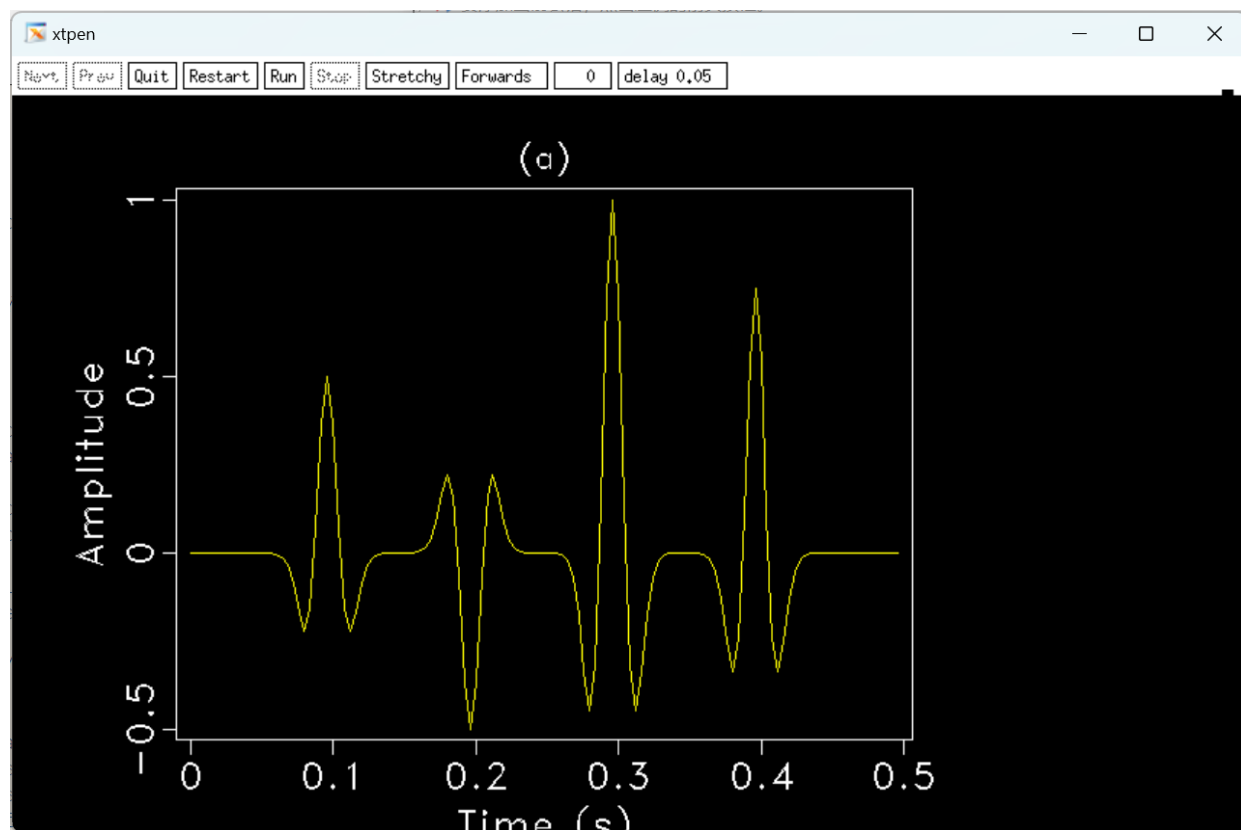
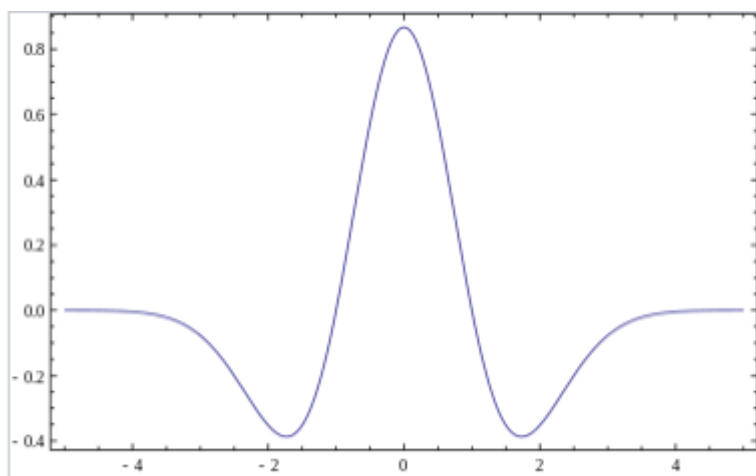
1. **frequency=25**：这里设置了子波的频率为25 Hz。子波的频率决定了子波的主要振荡周期

我的感觉就是把一个尖峰裂开，变成下面的图

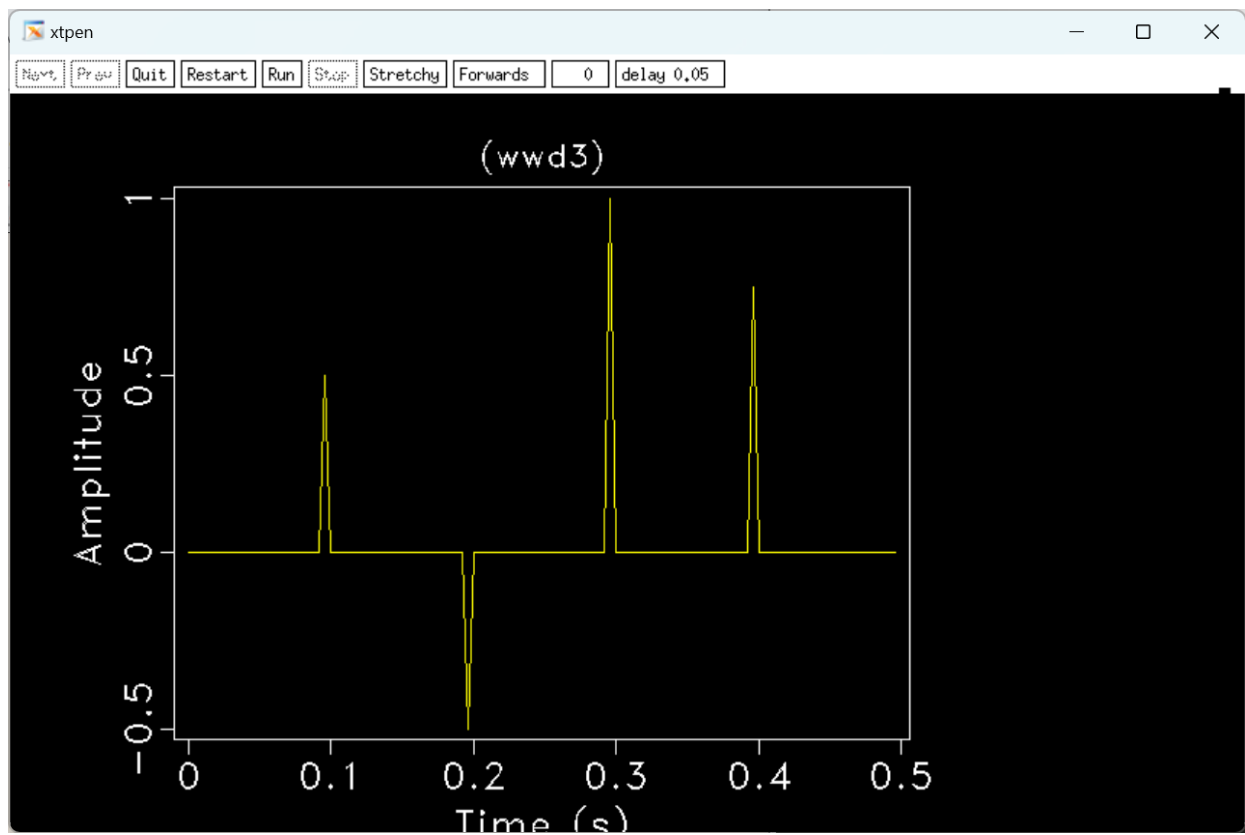
Ricker小波是高斯函数的负归一化二阶导数

$$\psi(t) = \frac{2}{\sqrt{3\sigma}\pi^{1/4}} \left(1 - \left(\frac{t}{\sigma} \right)^2 \right) e^{-\frac{t^2}{2\sigma^2}}$$

其图像为：



使用*Ricker*子波

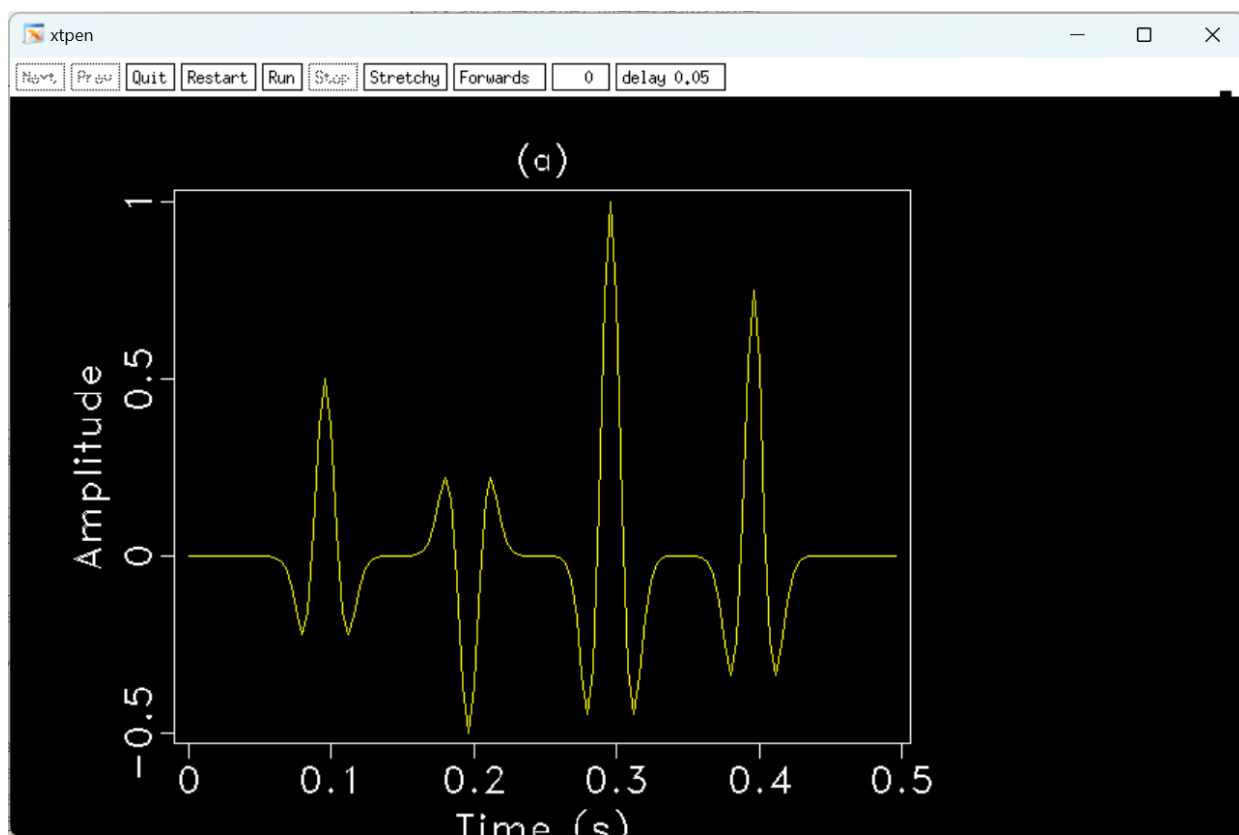


未使用*Ricker*子波

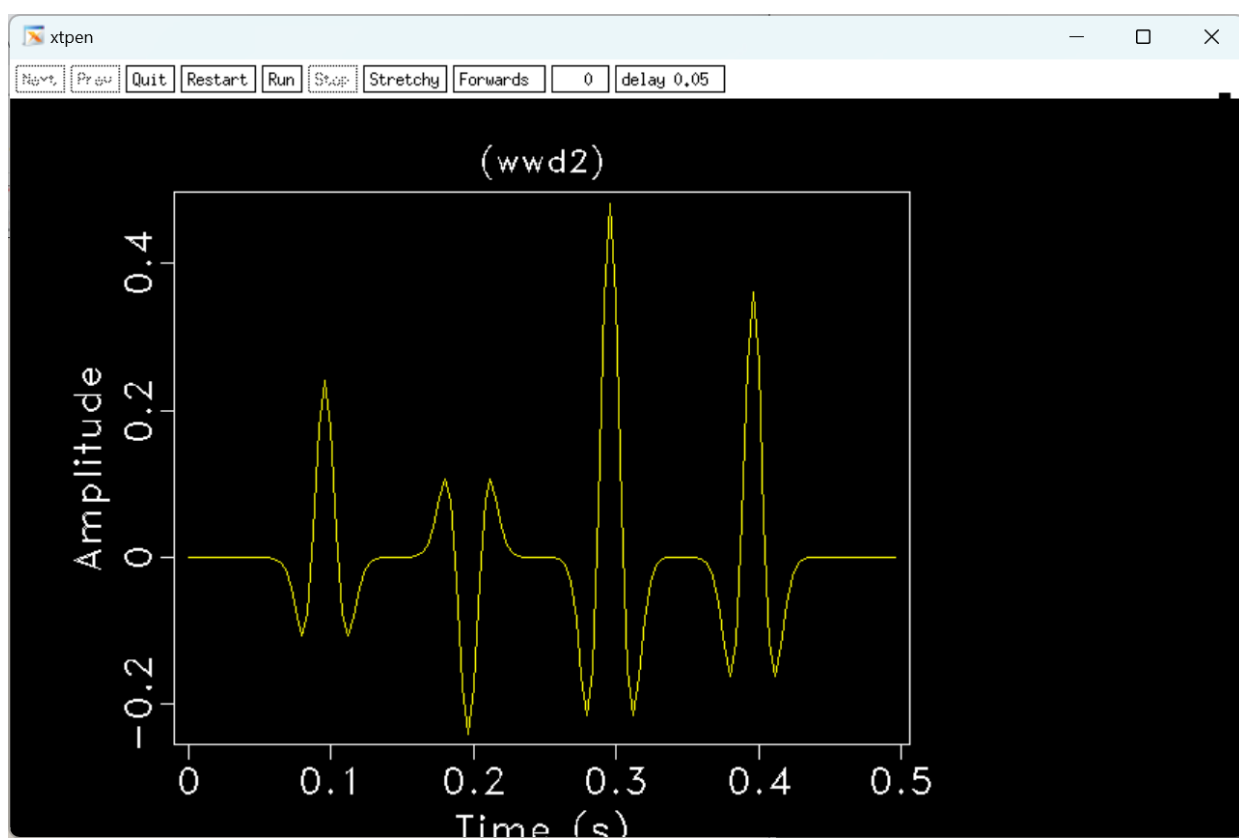
- FROM: <https://reproducibility.org/RSF/book/tccs/svscan/simi.html>

十、scale:缩放数据

1. `axis=1` :对数据的第二个维度进行缩放



使用`scale`对第二维度进行放缩

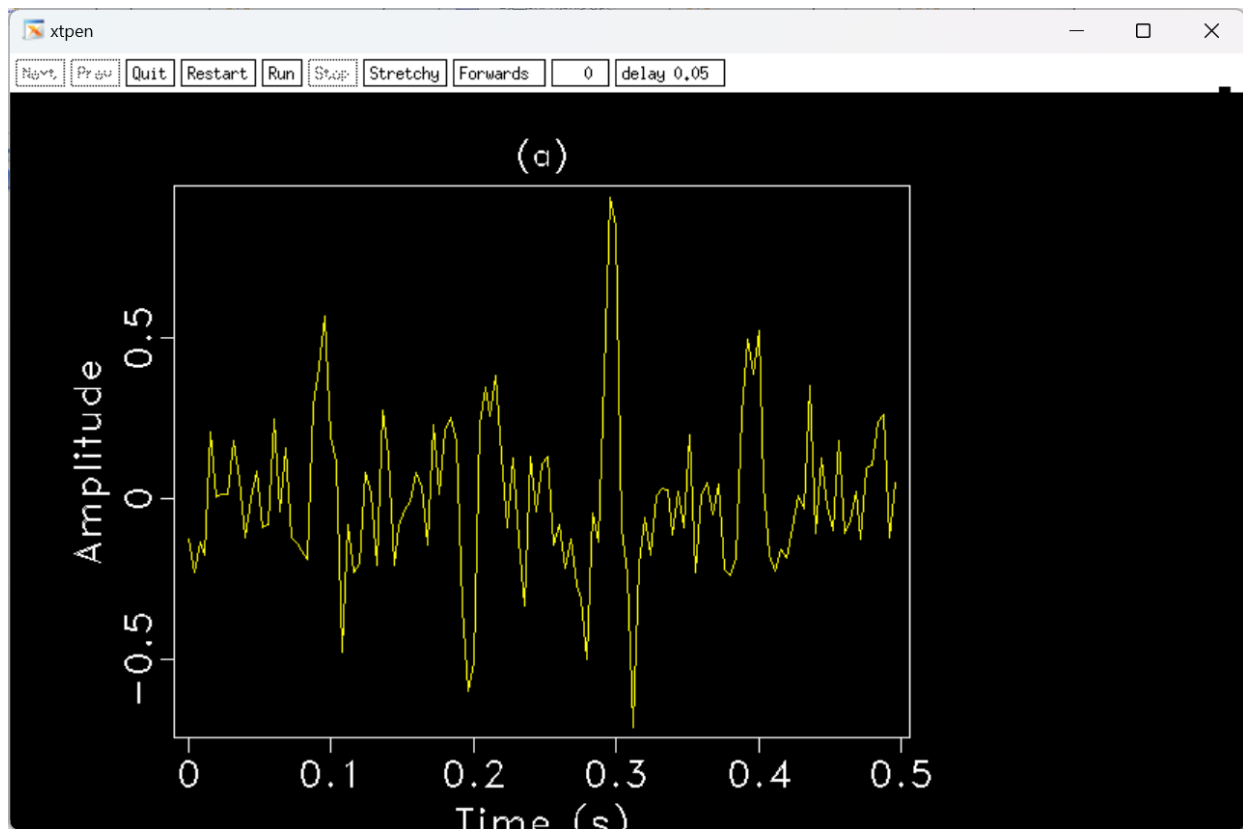


未使用`scale`

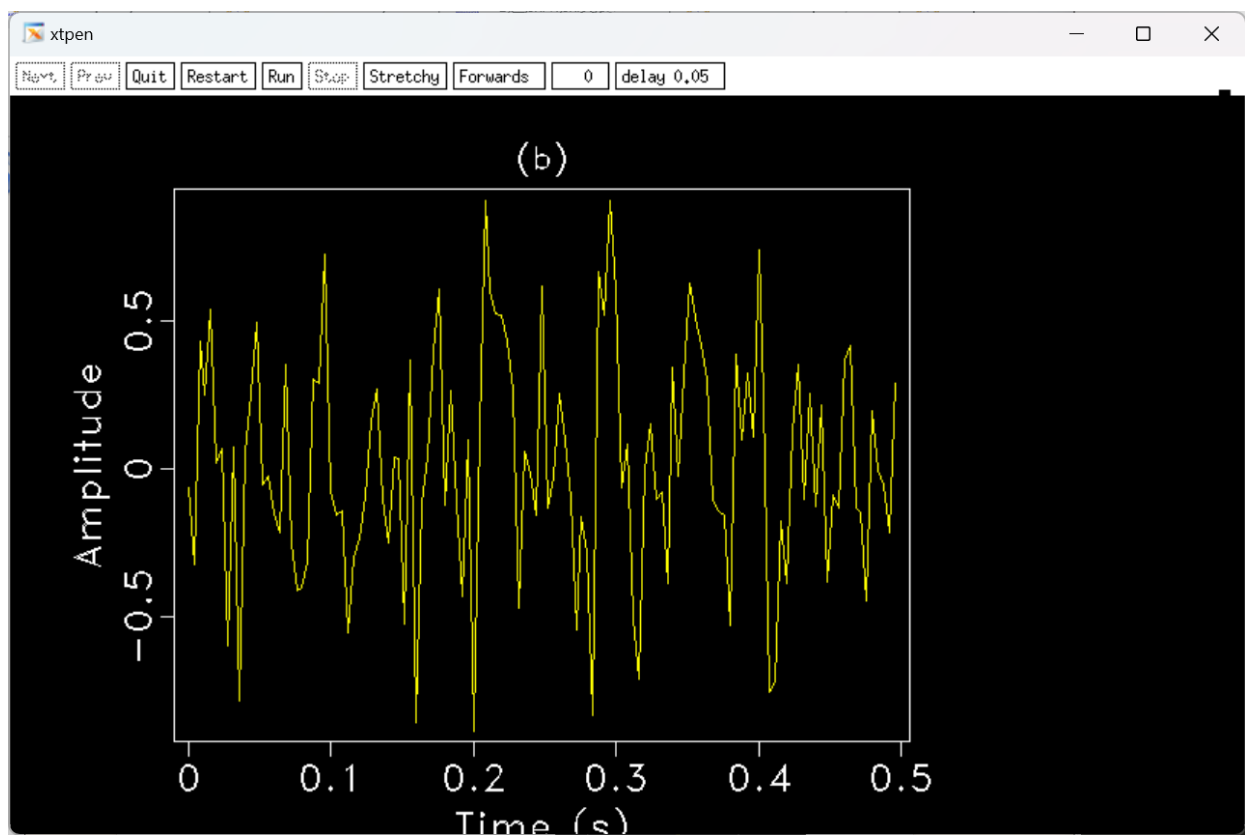
- FROM: <https://reproducibility.org/RSF/book/tccs/svscan/simi.html>

十一、similarity:度量两个数据集之间的局部相似性

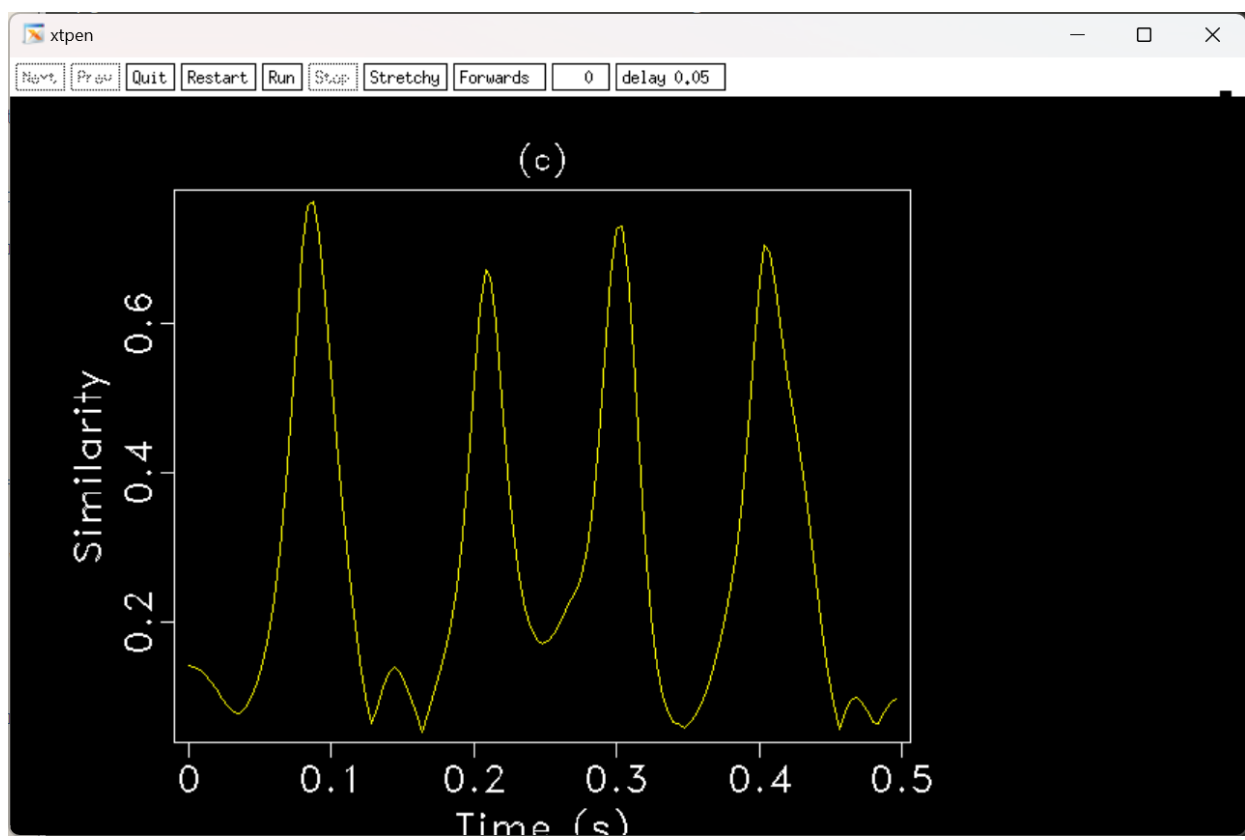
1. `other=${SOURCES[1]}` :指定了用于比较的第二个数据
2. `rect1=5, rect2=1` :用于相似度计算的平滑半径的大小参数



噪声较小的`trace1`



噪声较大



相似性分析

- FROM: <https://reproducibility.org/RSF/book/tccs/svscan/simi.html>

十二、inmo:模拟地震波在速度模型中的传播

1. `velocity=${SOURCES[1]}` 指定速度模型数据来源
2. `half=y` 表示只计算地震记录的前半部分

```
Flow('vrms',None,
     'math d1=0.004 n1=1001 o1=0 output="x1*125+2000+50*sin(9*x1)" ')

Flow('synt',None,
    ...

    spike d1=0.004 n1=1001 |
    noise rep=y seed=2006 |
    cut n1=100 |
    bandpass flo=4 fhi=20 |
    spray axis=2 n=256 d=25 o=-3200 label=Offset unit=m
    '')

Flow('cmpa','synt vrms',
     'inmo velocity=${SOURCES[1]} half=y | noise seed=2007 var=1e-10')
# 这段代码的主要目的是生成合成地震速度模型、合成地震数据，以及从速度模型中模拟地震记录数据
```

十三、blend:地震混合算子

1. `shot_time_in` :示射线的进入时间，即地震波从震源（地震枪或爆炸源）发出后到达地下界面的时间

2. **shot_time_out** :表示射线的出射时间，即地震波从地下界面反射回地表后的时间

```
## Apply dithering
Flow('dither','cmpa',
    ...
    window n1=1 |
    noise rep=y seed=122011 var=10 | math output="500*input"
    '')

Flow('shotttime1','cmpa','window n1=1 | math output=100*x1')
Flow('shotttime2','shotttime1 dither','add scale=1,1 ${SOURCES[1]}')

## Blend
Flow('cmp','cmpa cmpa shotttime1 shotttime2','blend
shot_time_in=${SOURCES[3]} shot_time_out=${SOURCES[2]} |add scale=1,1
${SOURCES[1]}')

...
生成地震速度模型数据 'vrms':
```

通过一个数学表达式生成合成的地震速度模型数据 'vrms'。速度模型通常用于模拟地震波传播。

生成合成地震数据 'synt':

创建名为 'synt' 的合成地震数据，包括脉冲序列、噪声、时间窗裁剪、带通滤波以及偏移信息的生成。这个过程模拟了合成地震数据的生成过程。

生成射线时间 'shotttime1' 和 'shotttime2':

生成 'shotttime1'，该数据包含与时间相关的射线时间。

生成 'shotttime2'，通过与 'shotttime1' 相加，模拟不同时间窗口内的射线时间。

混合地震数据 'cmp':

创建名为 'cmp' 的合成地震记录数据。

使用 blend 命令混合 'cmpa' 数据，同时考虑了 'shot_time_in' 和

'shot_time_out'，这允许在合成数据中模拟不同射线时间窗口内的地震记录。

最后，将 'cmpa' 数据与混合的数据相加。

总的来说，这段代码的主要作用是生成合成的地震速度模型数据和合成地震记录数据，并在后者中模拟射线时间窗口的影响。这些数据的过程通常用于地震学研究和分析，包括地下结构成像和地震波传播模拟。通过模拟不同时间窗口内的地震记录，研究人员可以更好地理解地震数据的时域特征。

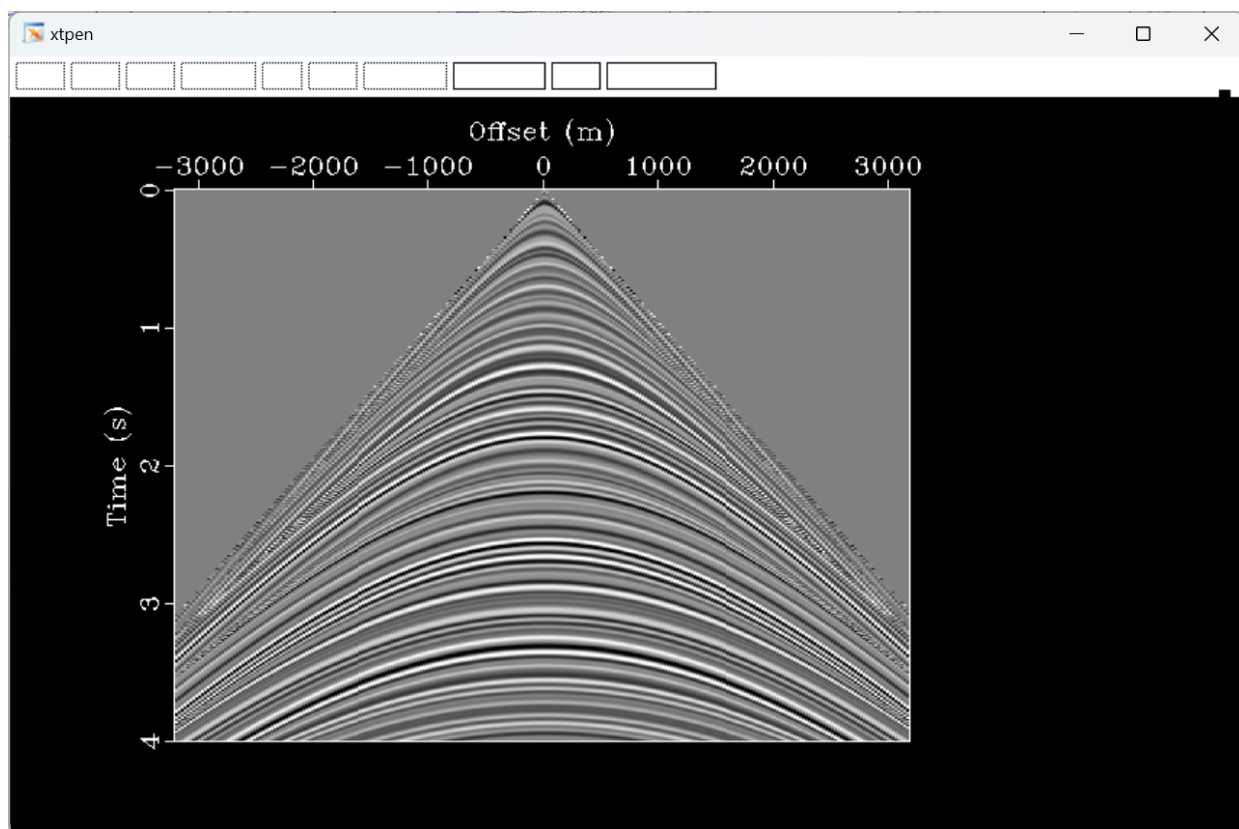
...

十四、clip:约束数据范围

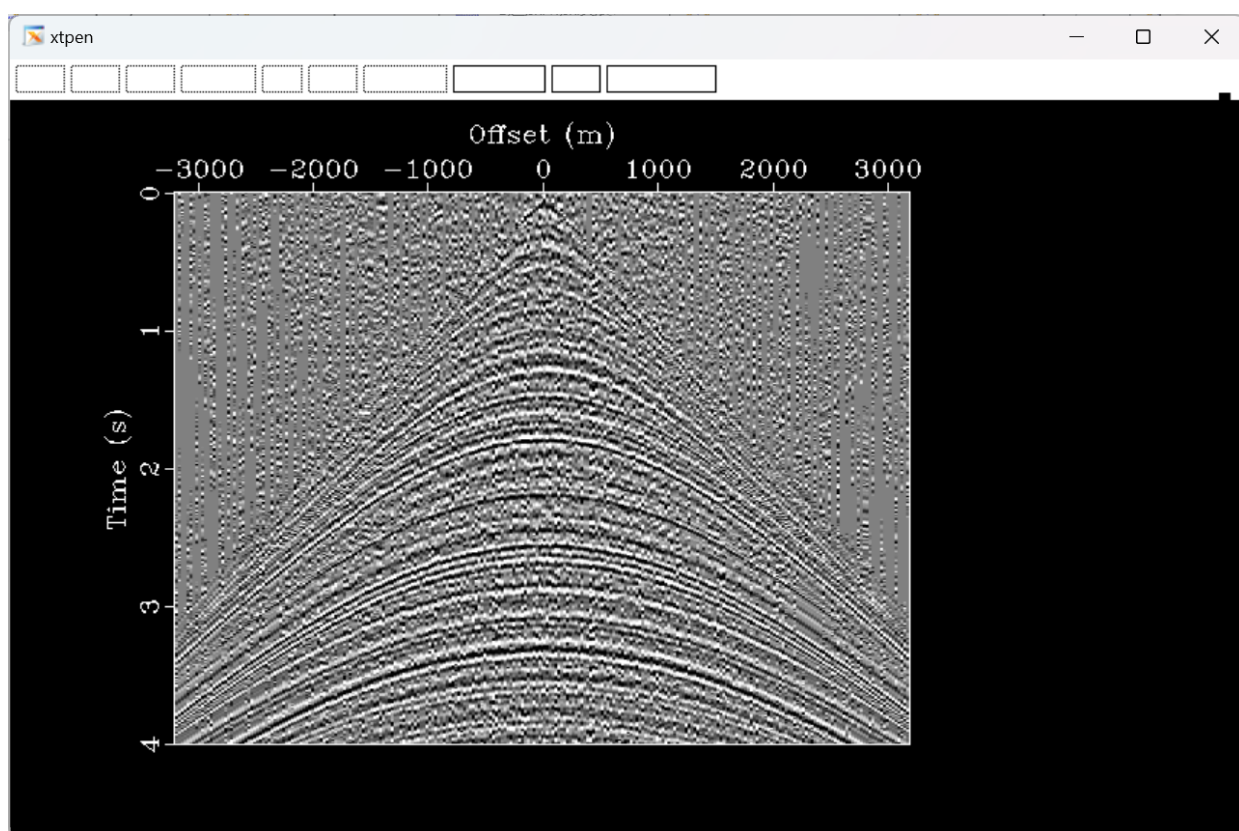
```
The output is
clip if input > clip
-clip if input < -clip
input if |input| < clip
```

1. clip=0.74 :clip值图像对比度

```
for case in ('a', ''):
    dat='cmp'+case
    Result(dat,'grey wanttitle=n clip=0.74 ' )
```



cmpa.vpl



cmp.vpl

- FROM: <https://reproducibility.org/RSF/book/tccs/svscan/simple.html>

十五、vscan:速度分析

half=y：表示只计算扫描图像的上半部分。

v0=2000：设置扫描的起始值为2000。

dv=10：设置扫描的步长为10。

nv=61：设置生成的扫描图像的数据点数为61。

type=%s：指定扫描的类型，这里使用 'semblance'。

nb=10：表示扫描的宽度为10

```
dat='cmp'
typ='semblance'
scn=typ+'scn'
Flow(scn,'cmp','vscan half=y v0=2000 dv=10 nv=61 type=%s nb=10 | clip2
lower=0' %typ)
Plot(scn+'0',scn,
    ...
    grey color=j allpos=y wanttitle=n pclip=100
    '')
```

...

可视化扫描图像：

使用 Plot 函数绘制扫描图像。

Plot(scn+'0', scn, 'grey color=j allpos=y wanttitle=n pclip=100') 命令用于以灰度图像的形式显示扫描图像，其中包括以下参数：

scn+'0'：指定要绘制的图像名称。

scn：指定要绘制的数据源名称。

grey：指定绘制的类型为灰度图像。

color=j：指定颜色方案为 "j"。

allpos=y：允许所有数据值都可可视化。

wanttitle=n：不显示图像标题。

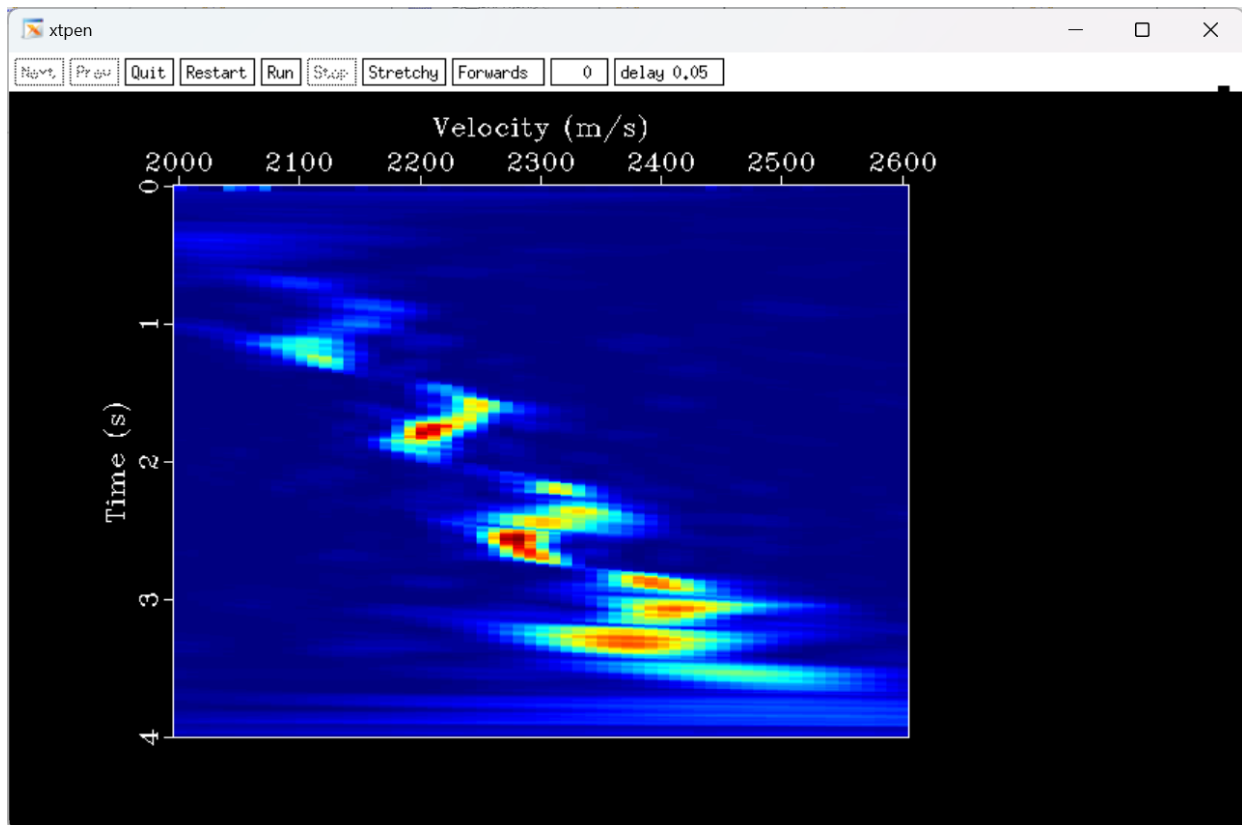
pclip=100：设置剪裁百分比为100，表示显示整个数据范围。

总的来说，这段代码的主要作用是生成地震数据的扫描图像，扫描类型为 'semblance'，然后将扫描图像以灰度图的形式进行可视化。这种可视化通常用于分析地震数据中的特定特征或信号，以便更好地理解地下结构和地震波传播的特性。

...

```
llp@VM-4-3-ubuntu:~/RSFSRC/book/tccs/svscan/simple$ sfin cmp.rsf
cmp.rsf:
  in="/var/tmp/tccs/svscan/simple/cmp.rsf@"
  esize=4 type=float form=native
  n1=1001      d1=0.004      o1=0      label1="Time" unit1="s"
  n2=256       d2=25        o2=-3200   label2="Offset" unit2="m"
  n3=1         d3=?         o3=?
      256256 elements 1025024 bytes
llp@VM-4-3-ubuntu:~/RSFSRC/book/tccs/svscan/simple$ sfin semblancescn.rsf
semblancescn.rsf:
  in="/var/tmp/tccs/svscan/simple/semblancescn.rsf@"
  esize=4 type=float form=native
  n1=1001      d1=0.004      o1=0      label1="Time" unit1="s"
  n2=61        d2=10        o2=2000   label2="Velocity" unit2="m/s"
  n3=1         d3=?         o3=?
      61061 elements 244244 bytes
```

semblancescn.rsf



semblancescn.vpl

- FROM: <https://reproducibility.org/RSF/book/tccs/svscan/simple.html>

十六、pick:从相似面中自动拾取

1. `rect1=50`: 定义了第一维度上的矩形窗口, 限定了选取的范围, 平滑半径
2. `rect2=10`: 定义了第二维度上的矩形窗口, 限定了选取的范围, 平滑半径
3. `vel0=2500`: 指定了速度阈值, 只选择速度大于等于 2500 的数据

```
pik=typ+'pik'  
Flow(pik,scn,'scale axis=2 | pick rect1=50 rect2=10 vel0=2500')  
Plot(pik,  
      ...  
      graph yreverse=y transp=y min2=2000 max2=2600 pad=n  
      Resultcol=7 Resultfat=7 wanttitle=n wantaxis=n  
      '')  
...
```

使用 Plot 函数绘制选取数据。

Plot(pik, 'graph yreverse=y transp=y min2=2000 max2=2600 pad=n Resultcol=7 Resultfat=7 wanttitle=n wantaxis=n') 命令用于绘制选取数据, 其中包括以下参数:

graph yreverse=y: 将 Y 轴翻转, 通常用于地震数据可视化。

transp=y: 将数据进行透明处理, 以使图像更易于阅读。

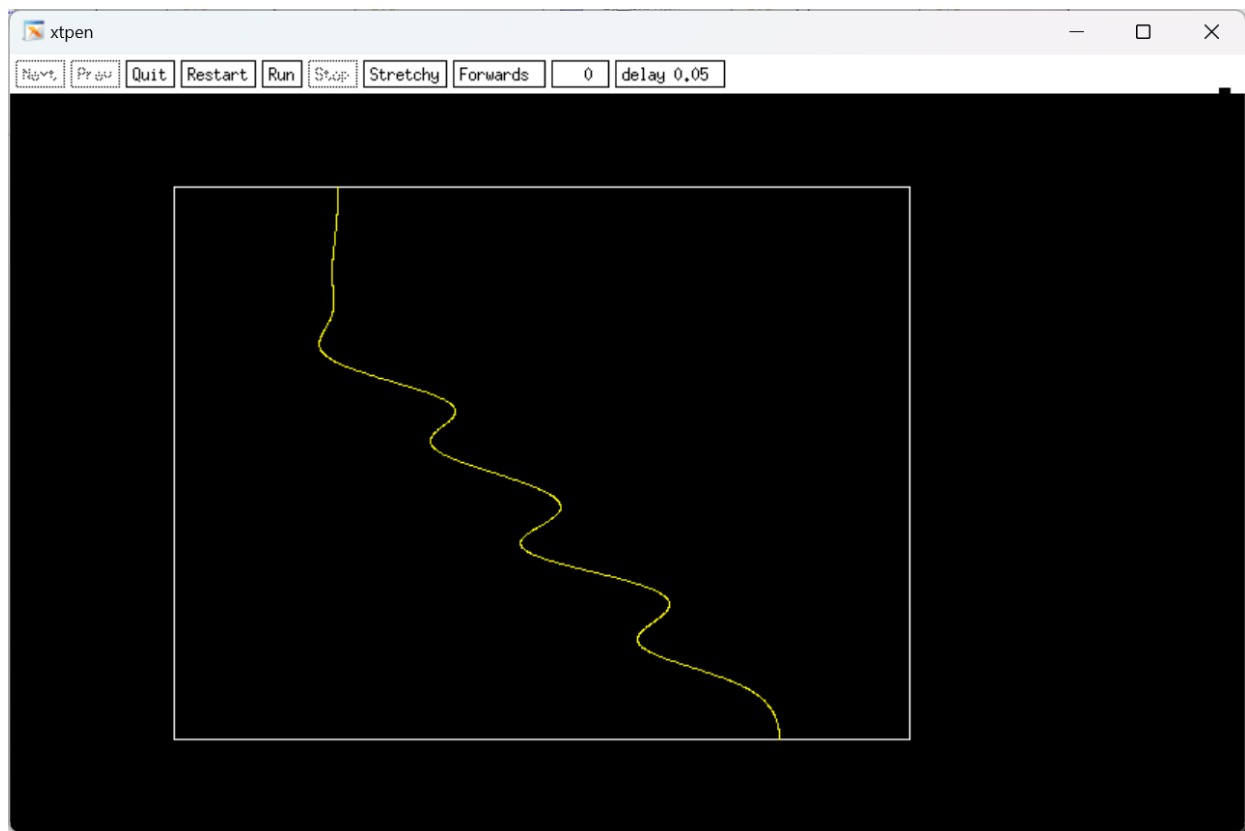
min2=2000 和 max2=2600: 指定 Y 轴范围, 控制可视化的垂直尺度。

pad=n: 指定不添加边距。

Resultcol=7 和 Resultfat=7: 定义颜色和线条粗细。

wanttitle=n 和 wantaxis=n: 禁止显示图像标题和坐标轴。

```
...
```



semblancepik.vpl

- FROM: <https://reproducibility.org/RSF/book/tccs/svscan/simple.html>