Seismologi Komputasi - Tugas 1

Farhan Hamid Lubis - 22319310
Rizky Adityo Prastama - 22319311

Pemodelan Raytracing - 1D Homogeneous Layered Earth Model

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
np.set_printoptions(suppress=True)
# %matplotlib inline
```

Step 1 : Input Parameter

Model yang dibuat adalah lapisan bumi homogen secara lateral dengan nilai velocity dan ketebalan masing-masing. Jumlah ray yang digunakan adalah 5. Sifat fisis untuk kasus model bumi berlapis homogen pada program ini disimpan dalam variabel sebagai berikut:

```
    * rays : integer
        Jumlah ray seismik
    * velo : 1D-array
        Nilai cepat rambat gelombang pada tiap lapisan
    * dz : 1D-array
        Ketebalan masing-masing lapisan
    * layers : integer
        Jumlah lapisan
```

In [2]:

```
rays    = 5
velo    = np.array([2300, 2250, 2200, 2150, 2100, 2050, 2000, 1950])
dz      = np.array([100, 100, 100, 100, 100, 100, 100, 100])
layers  = len(velo)
```

Step 2 : Pembentukan Sudut *Takeoff*

Pada tahap ini dilakukan looping untuk pembentukan sudut takeoff dari setiap ray. Dalam kasus ini penulis membuat masing-masing ray memiliki selisih 10 derajat. Variasi sudut *takeoff* disimpan dalam variabel berikut:

```
    * theta : 1D-array
        Variasi nilai sudut takeoff
```

In [3]:

```
theta = np.empty(shape = [0, rays])
for i in range(1, rays+1):
    t = i * 10
    theta = np.array([np.append(theta, t)])
print("theta : ",theta)
```

```
theta :  [[10. 20. 30. 40. 50.]]
```

Step 3 : Perhitungan Ray Parameter

Ketika bekerja pada bidang x dan z, maka komponen vektor slowness arah y (p_y) bernilai 0 dan menyisakan komponen x (p_x) dan z (p_z). Variasi velocity yang digunakan hanya ada pada arah vertikal, sehingga:

$$p_x = p = \sin i(z)/V(z)$$

dimana i adalah sudut takeoff dan $V(z)$ adalah nilai kecepatan pada kedalaman z. Dengan menggunakan persamaan eikonal $p^2 + p_z^2 = V^{-2}$, maka p_z dapat dihitung sebagai berikut:

$$p_z = \sqrt{V^{-2} - p^2}$$

Hasil perhitungan pada tahap ini disimpan dalam variabel berikut:

```
    * p : 1D-array
        Nilai horizontal slowness (px) untuk masing-masing ray
    * pz : 2D-array
        Nilai vertical slonwess (pz) masing-masing ray untuk tiap lapisan
```

In [4]:

```
p = np.empty(shape = [0, rays])
for i in range(rays):
    x = np.sin(np.deg2rad(theta[0, i])) / velo[0]
    p = np.array([np.append(p, x)])

pz = np.empty(shape = [layers, rays])
for j in range(rays):
    for i in range(layers):
        pz[i, j] = np.sqrt((1/(velo[i]))**(2) - (p[0, j])**2)
print("pz : \n",pz)
```

```
pz :
 [[0.00042818 0.00040856 0.00037653 0.00033306 0.00027947]
 [0.00043798 0.00041883 0.00038765 0.00034558 0.00029428]
 [0.00044823 0.00042953 0.00039919 0.00035848 0.00030932]
 [0.00045895 0.0004407  0.00041119 0.00037179 0.00032466]
 [0.00047017 0.00045238 0.00042367 0.00038555 0.00034033]
 [0.00048193 0.00046459 0.00043669 0.00039981 0.0003564 ]
 [0.00049427 0.00047738 0.00045027 0.0004146  0.00037292]
 [0.00050723 0.00049079 0.00046446 0.00042998 0.00038994]]
```

Step 4 : Perhitungan Lateral Displacement

Perpindahan gelombang secara lateral pada setiap kedalaman dapat dihitung dengan menggunakan rasio antara p dan p_z yang sudah diketahui. Dengan demikian:

$$\frac{dx}{dz} = \frac{p}{p_z} = \frac{p}{\sqrt{V^{-2} - p^2}}$$

$$dx = \frac{p}{\sqrt{V^{-2} - p^2}} dz$$

Hasil perhitungan pada tahap ini disimpan dalam variabel berikut:

```
    * depth : 1D-array
        Kedalaman masing-masing lapisan
    * dx : 2D-array
        Nilai lateral displacement masing-masing ray pada tiap lapisan
```

In [5]:

```
dx = np.empty(shape = [layers, rays])
for j in range(rays):
    for i in range(layers):
        dx[i, j] = p[0, j] * dz[i]/ pz[i, j]

a = np.array([np.zeros(rays)])
dx = np.append(a, dx, axis = 0)

for j in range(rays):
    for i in range(1, layers + 1):
        dx[i, j] = dx[i, j] + dx[i - 1, j]

total_depth = np.sum(dz)
dz = np.append(0, dz)
depth = np.array([np.zeros(len(dz))])
for i in range(1, len(dz)):
    depth[0, 0] = total_depth
    depth[0, i] = total_depth - (np.sum(dz[0:i + 1]))
```

Step 5 : Visualisasi

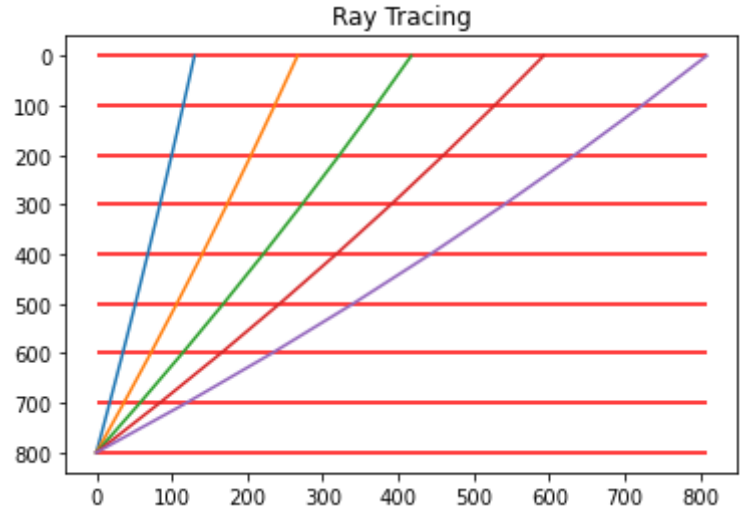
Berikut merupakan perintah untuk melakukan visualiasi ray tracing dengan sumber berada pada lapisan terakhir

In [ ]:

```
fig, ax = plt.subplots()
for i in range(rays):
    ax.plot(dx[:, i], depth[0, :])
ax.set_title("Ray Tracing")
for i in range(len(depth)):
    plt.hlines(depth[i], 0, np.amax(dx), colors='r')
plt.gca().invert_yaxis()
```

In [6]:

```
plt.show()
```



References

ČErvený V. (1989) Seismic ray theory. In: Geophysics. Encyclopedia of Earth Science. Springer, Boston, MA. https://doi.org/10.1007/0-387-30752-4_134 (https://doi.org/10.1007/0-387-30752-4_134)