1 Python correction

```
import numpy as np
import matplotlib.pyplot as plt

from skimage.io import imread
from skimage import data_dir
from skimage.transform import radon, rescale, rotate
from scipy.signal import convolve
import skimage.io
```

1.1 Acquisition simulation

The phantom image is first loaded and displayed.

```
# Read image
image = imread("phantom.png", as_gray=True)
image = image[:,:,0];
plt.imshow(image, cmap='gray');
```

The simulation of the projection is simply an addition of all gray-levels of the pixels, after rotating the image in order to simulate the rotation of the object (or of the sensor). See Fig.1.

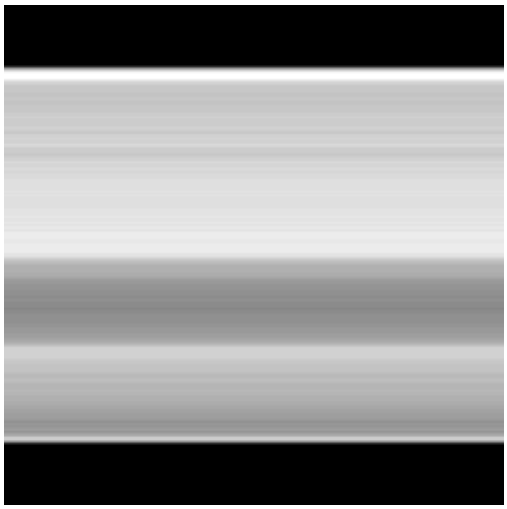

Figure 1: Simulation of the projection: contribution of a line D before rotation.

```
def simuProjection(I, theta):
    """
    simulation of the generation of a sinogram
    I : original image (phantom for example)
    theta: angles of projection
    """
    N = I.shape[1];
    M = len(theta);
    S = np.zeros((N,M));

    for i,ang in enumerate(theta):
        image1 = rotate(I, ang);
        S[:,i] = np.sum(image1, axis=1);

    return S;
```

1.2 Backprojection algorithm

The backprojection algorithm will sum-up all the contributions of each projection.

```python
def backprojection(P, theta, filtre):
    """
    Backprojection of
    P: image
    theta: list of projection angles
    filtre: bool, True if filtered
    """

    N = P.shape[0];
    R = np.zeros((N,N));

    # in case of filtered back-projection
    if filtre:
        h = RamLak(31);

    # loops over all angles
    for i,ang in enumerate(theta):
        proj = P[:,i];

        # filtered back-projection
        if filtre:
            proj = convolve(proj, h, mode='same');

        proj2 = np.matlib.repmat(proj, N, 1);
        proj2 = rotate(proj2, ang);
        R = R + proj2;
    return R.transpose();
```

The results is better in the case of a filtered backprojection. The RamLak function is provided and illustrated in Fig.2.

```python
def RamLak(width):
    """
    Ramlak filter of size width
    width must be odd
    """
    ramlak = np.zeros((2*width+1,));
    for indice, val in enumerate(np.arange(-width, width+1)):
      val = np.abs(val);

      if val==0: # center
         ramlak[indice]=np.pi/4;
      else:
        if val%2==1: # even indices
            ramlak[indice]=-1/(np.pi*val**2);
        else: # odd indices
            ramlak[indice]=0;

    return ramlak;
```

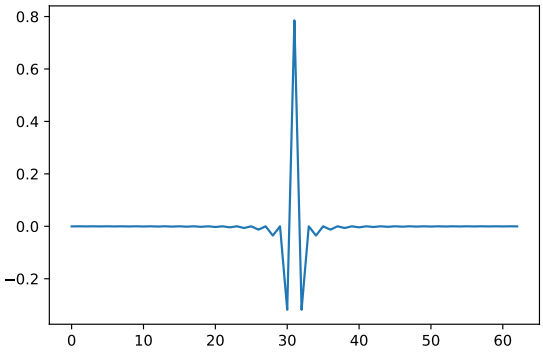

Figure 2: RamLak function.

The reconstruction of the original image is obtained by the following code:

```python
# Performs unfiltered backprojection
ubp = backprojection(P, theta, False);
plt.figure();
plt.imshow(ubp, cmap='gray');
plt.show()

# Performs filtered backprojection
fbp = backprojection(P, theta, True);
plt.figure();
plt.imshow(fbp, cmap='gray');
plt.show()
```

(a) Original phantom image. (b) Unfiltered backprojection. (c) Filtered backprojection.

Figure 3: Reconstruction by backprojection.