1 MATLAB correction

1.1 Acquisition simulation

The MATLAB built-in function is used to generate a phantom image.

```
% phantom image generation
I = phantom();

%% Projection with an angular step of 1 degree
angle = 1;
theta = 0:angle:180;
S=simuProjection(I, theta);
imshow(S, []);
```

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).

```
function S=simuProjection(I, theta)
% simulation of the generation of a sinogram
% I : original image (phantom for example)
% theta: angles of projection
taille=size(I);
S=zeros(taille(2),length(theta));

for i=1:length(theta)-1,
    image1=imrotate(I, theta(i), 'bilinear', 'crop');

    S(:,i)=sum(image1');
end
```

1.2 Backprojection algorithm

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

```
function R=backprojection(P, theta, filtre)
% Backprojection of a projected image P,
% at all angles 'theta'
% filtre: bool, applies filtering if True
N = size(P,1);
R =zeros(N);

% in case of filtered back-projection
h = RamLak(31);
```

```
10
% loops over all angles
12 for i=1:length(theta),
    proj = P(:,i);
14
    % filtered back-projection
16  if filtre==1
        proj = conv(proj, h, 'same');
18  end

20  proj2 = repmat(proj, 1, N);
    proj2 = imrotate(proj2, -theta(i), 'bilinear', 'crop');
22
    R = R + proj2;
24 end
```

The results is better in the case of a filtered backprojection. The RamLak function is provided and illustrated in Fig.1.

```
function [ramlak] = RamLak(width)
2 % Ramlak filter of size width
% width must be odd
4 k=-width:1:width;

6 for indice = 1:length(k);
  if(k(indice)==0) % valeur du centre
8    ramlak(indice)=pi/4;
  elseif(mod(k(indice),2)==1) % indices pairs
10   ramlak(indice)=-1/(pi*k(indice)^2);
  else % indices impairs
12   ramlak(indice)=0;
  end
14 end
```

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

```
%% reconstruction: simple back-projection
2 R1=backprojection(S, theta, 0);

4 imshow(R1, []);

6 %% Filtered back-projection
R2=backprojection(S, theta, 1);
8 imshow(R2, []);

10 %% matlab built-in functions
```

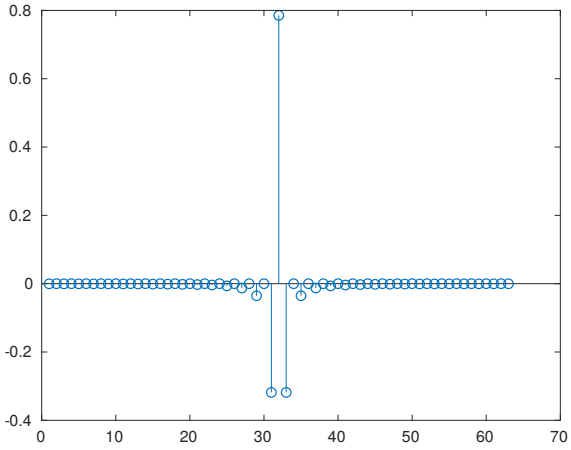


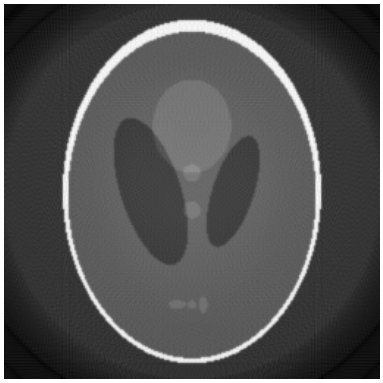
Figure 1: RamLak function.

```
s=radon(I, theta);
imshow(s, []);

r=iradon(s, theta);
figure();
imshow(r, []);
```

(a) Original phantom image.

(b) Unfiltered backprojection.



(c) Filtered backprojection.

Figure 2: Reconstruction by backprojection.