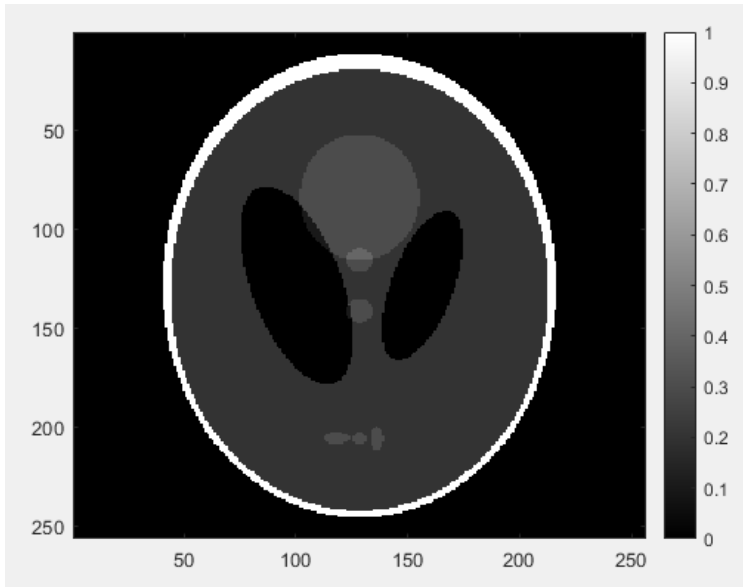


Aula prática 9

Reconstrução tomográfica

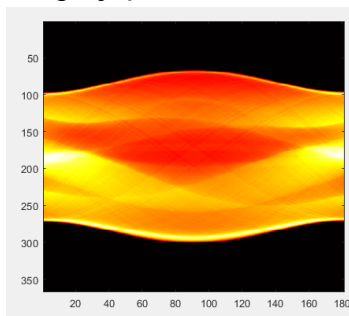
Responder às questões abaixo incluindo as imagens geradas no Matlab. Ao finalizar envie seu relatório juntamente com os scripts em arquivo Matlab ou Octave usados para fazer os exercícios. Por favor, faça comentários em seus scripts explicando brevemente o propósito de cada comando.

1. Crie uma matriz do *phantom* virtual Shepp-Logan com 256 x 256 pixels usando a função `I = phantom (256)`. Mostre a imagem do *phantom*.

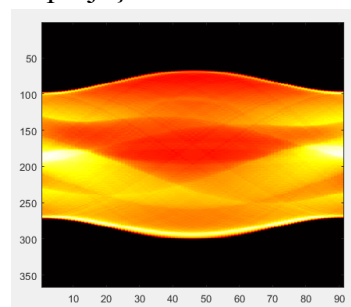


2. Use a função `radon` para obter as projeções do *phantom* para o caso de feixes paralelos. Adquiras as projeções de ângulos 0 a 180 graus. Construa três sinogramas variando o número de projeções: 45, 90 e 180. Mostre os três casos e comente sobre as diferenças entre os casos.

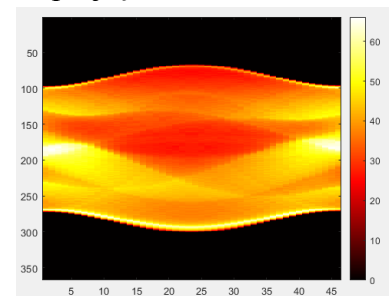
180 projeções:



90 projeções:

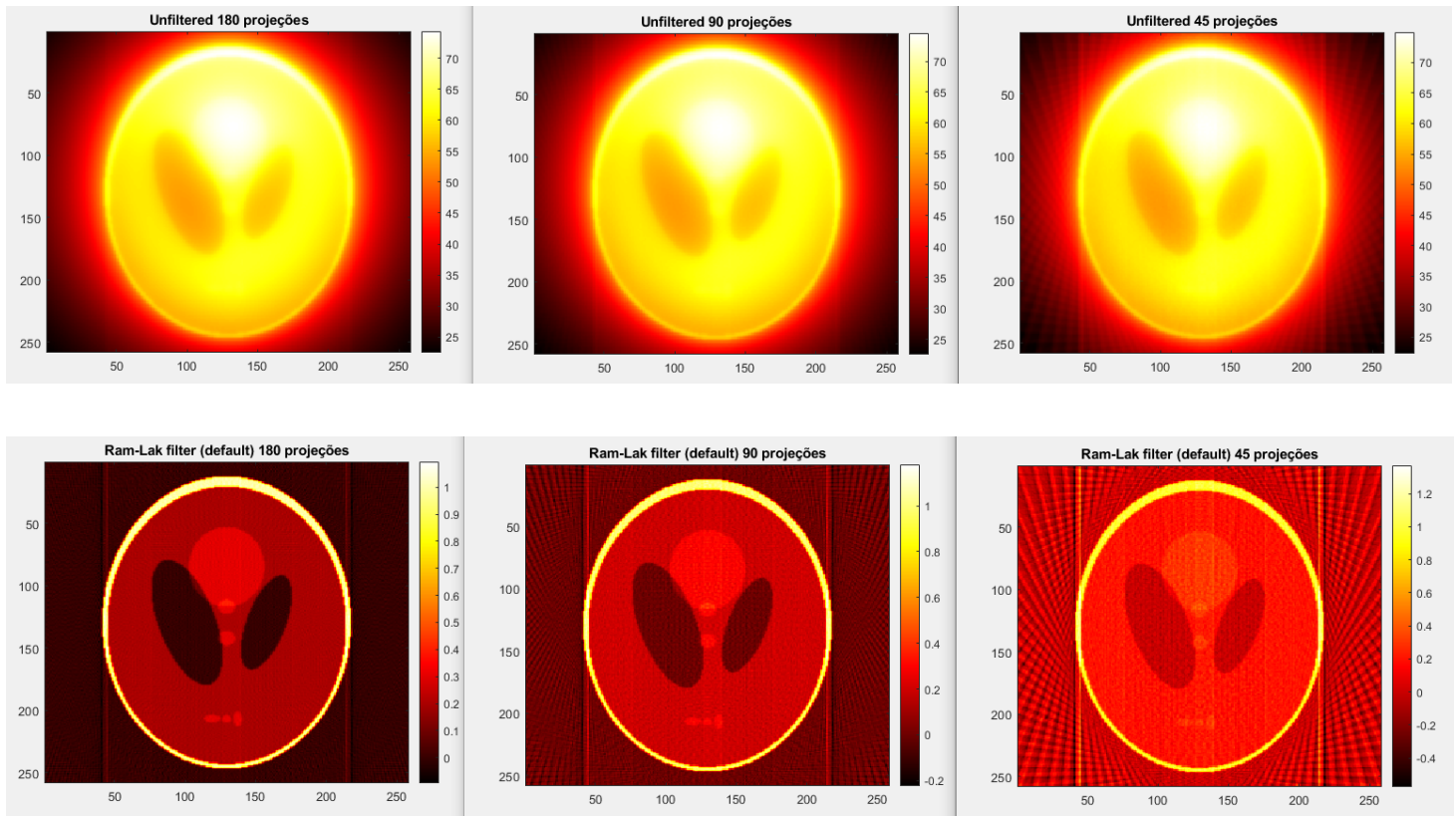


45 projeções:



Conforme o número de projeções vai diminuindo, a imagem vai ficando cada vez mais “pixelada”, diminuindo assim a resolução das projeções.

3. Use a função `iradon` para reconstruir a imagem do *phantom* Shepp-Logan a partir dos três sinogramas construídos no item 2.
 - a. Faça essa reconstrução sem usar filtro e usando o filtro 'Ram-Lak'. Comente as diferenças observadas entre as duas reconstruções para cada um dos 3 casos.

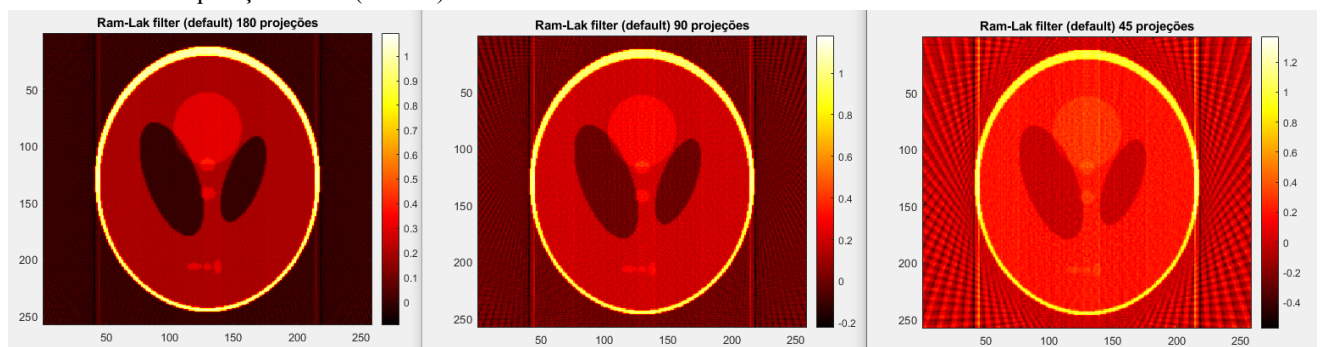


As reconstruções sem a filtragem são bem mais intensas (observar a escala “colorbar”) impossibilitando uma boa nitidez e identificação de algumas bordas ou delimitações no phantom, o que ocorre justamente o oposto para as reconstruções filtradas.

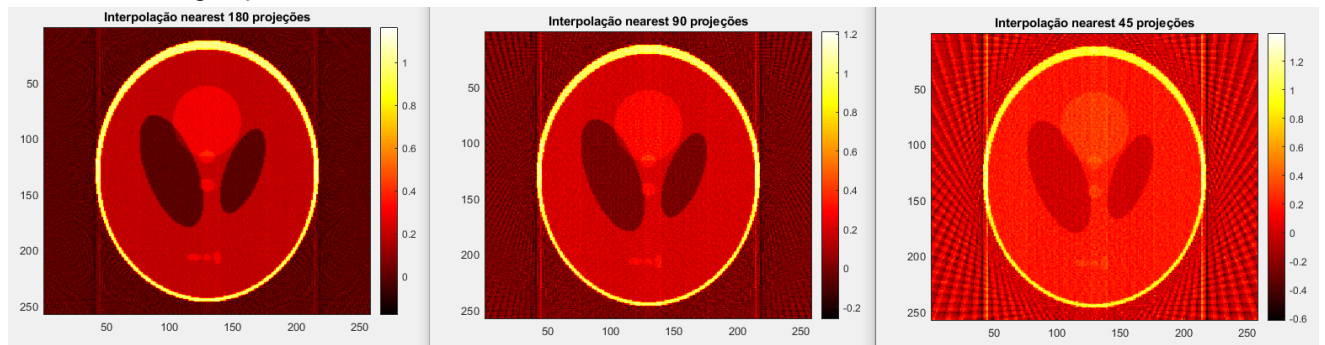
Na projeção filtrada, conforme o número de projeções diminui aparece uma maior discretização da imagem, evidenciando cada vez mais os “feixes” de reconstrução.

- b. Compare os resultados somente para o caso em que o filtro foi usado, mas variando o tipo de interpolação usado pela função `iradon`. Use as interpolações linear e pelos vizinhos mais próximos. Comente as diferenças observadas. Qual o papel da interpolação neste caso?

interpolação linear (default)

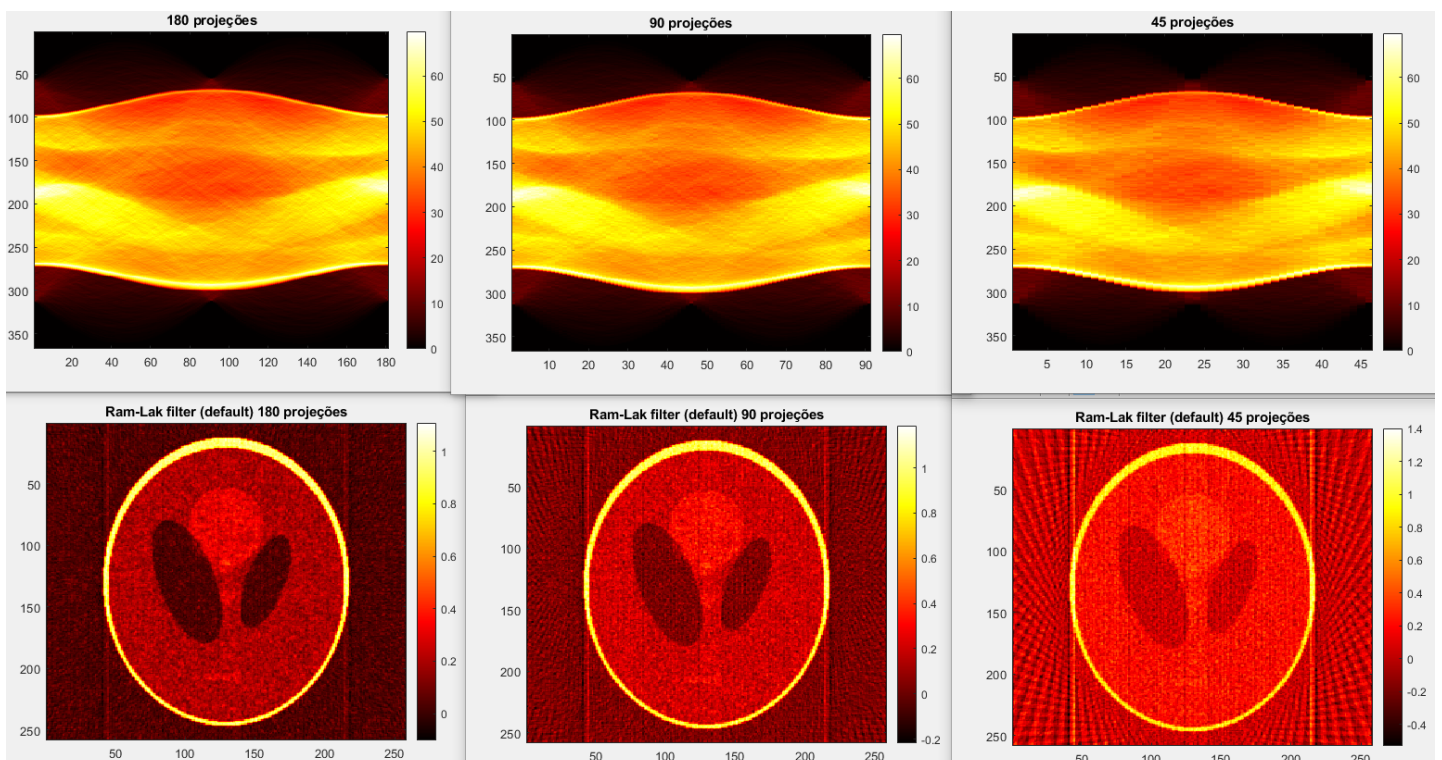
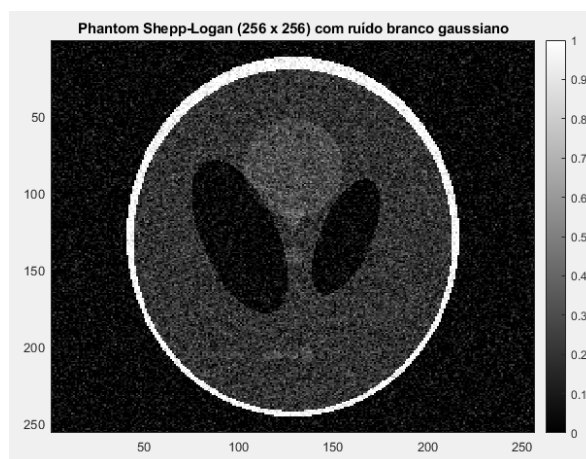


interpolação nearest

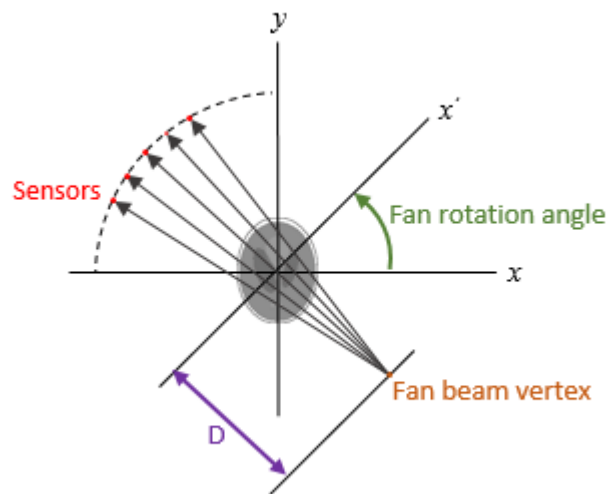


As diferenças observadas foram pouco significativas, porém é perceptível o papel da interpolação na reconstrução da imagem a partir das projeções após a aplicação do filtro.

4. Faça os mesmos passos dos itens 2 e 3, mas agora antes de aplicar a transformada de Radon, adicione ruído branco gaussiano à imagem do *phantom* Shepp-Logan. Use a função `imnoise`.

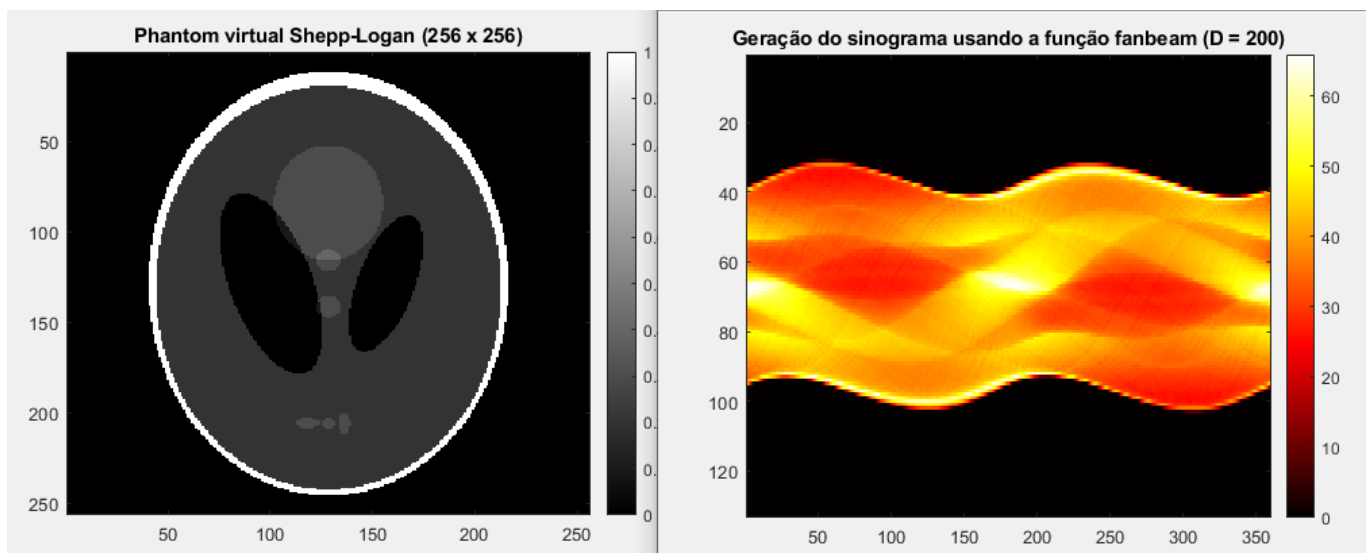


5. Explique o funcionamento para obter as projeções de uma imagem para o caso de feixes em “leque” usando a função fanbeam.



Os “feixes” para obter as projeções neste modo não são paralelos uns aos outros, mas possuem um ângulo que varrem um arco (“leque”), com sua origem ou *vertex* situado em uma distância especificada D , e seus sensores podem ser espaçados ao longo do arco circular (default) ou de forma linear.

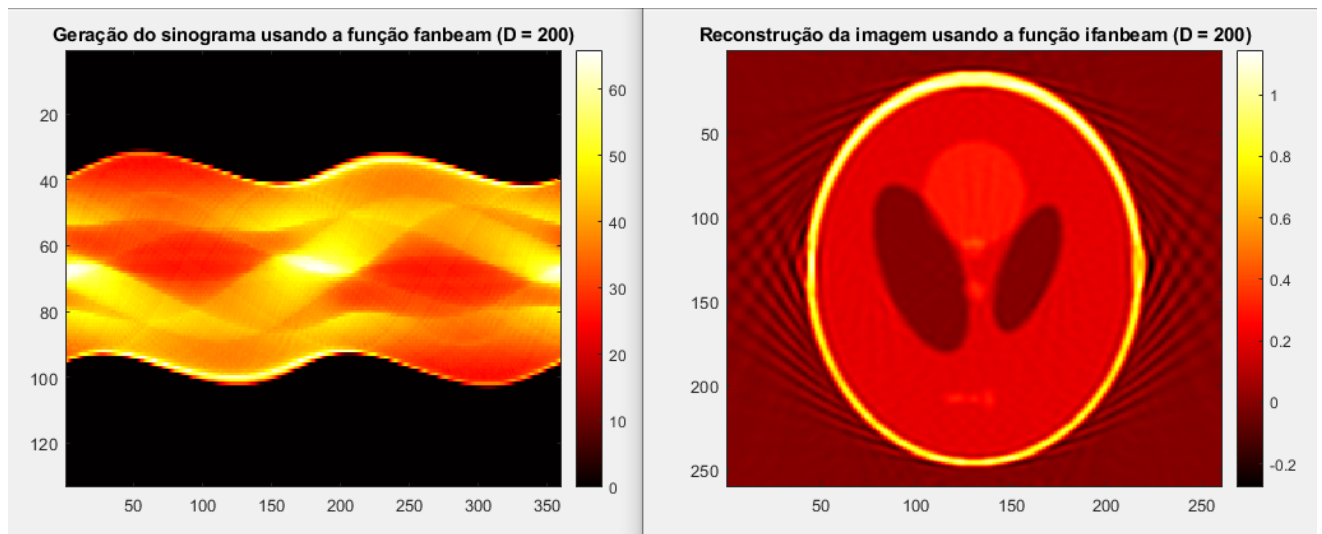
- a. Gerar sinogramas do phantom Shepp-Logan usando a função fanbeam.



6. Gerar imagens reconstruídas para feixes em “leque” usando a função ifanbeam.

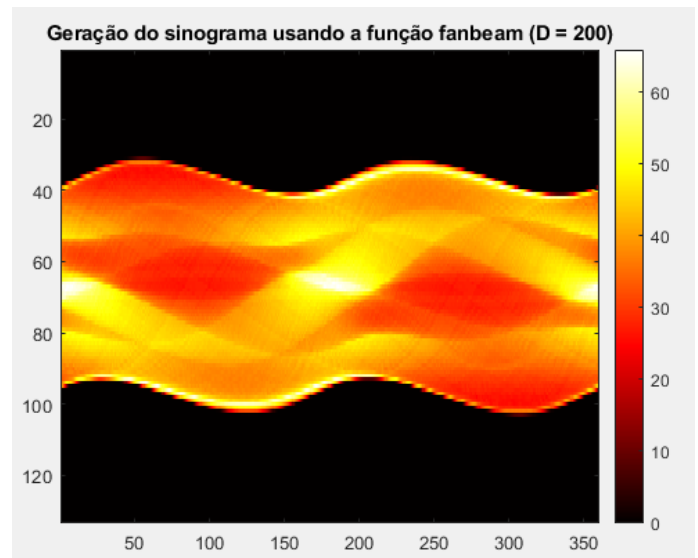
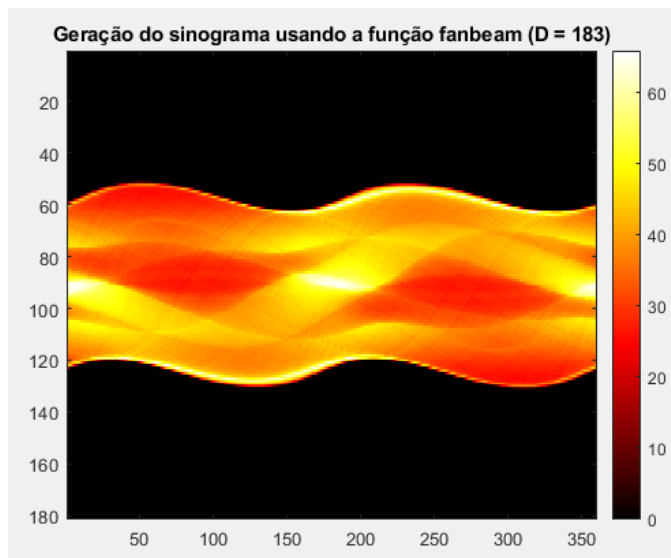
a. O que representa a variável D ?

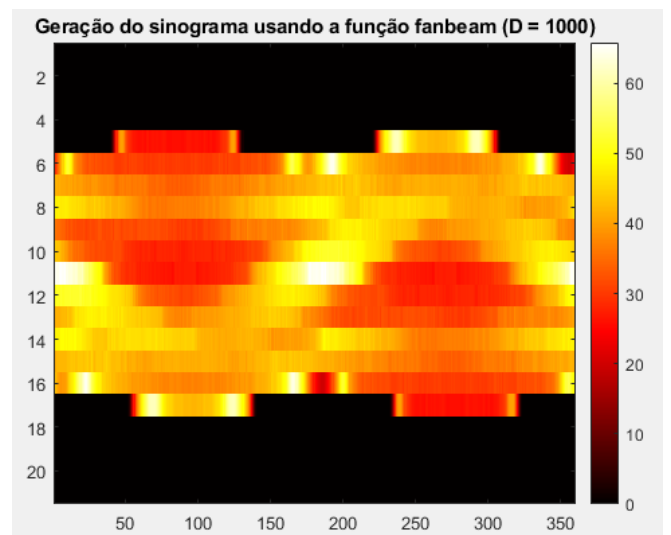
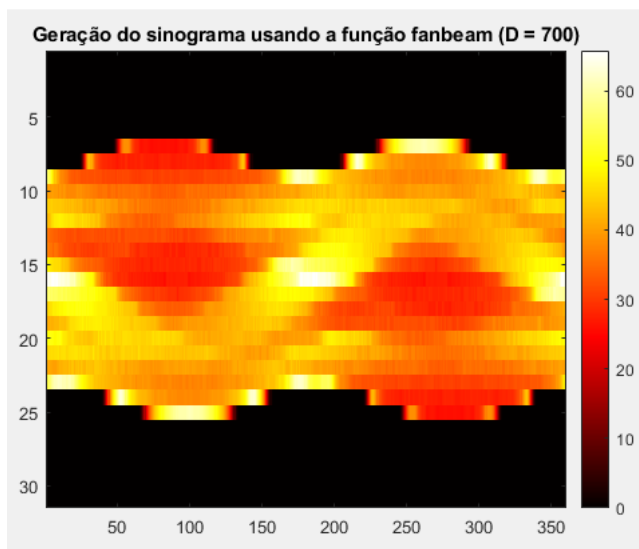
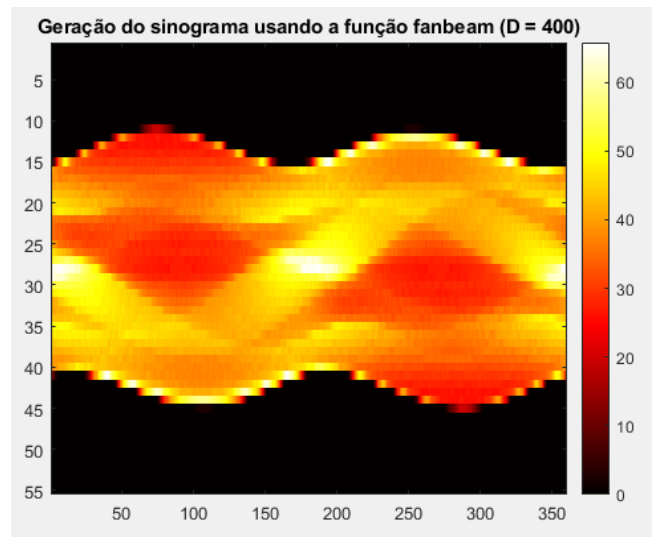
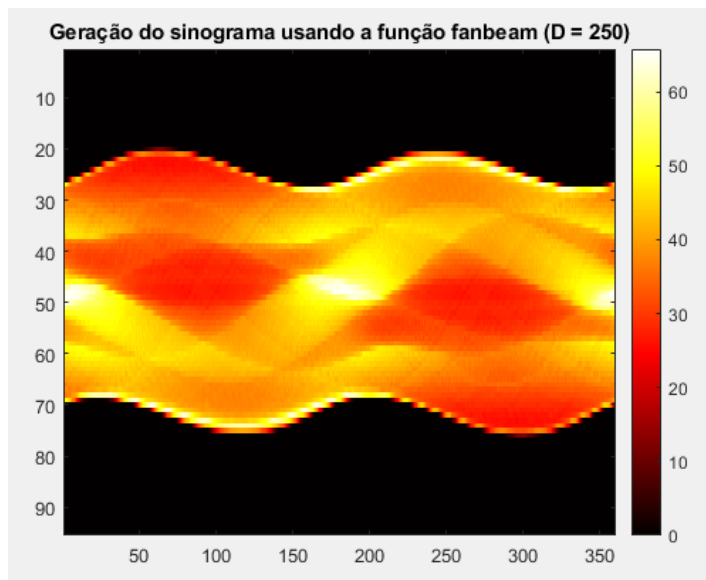
É a distância entre a origem da imagem até a fonte de geração dos “feixes” de projeção (*fan beam vertex*).



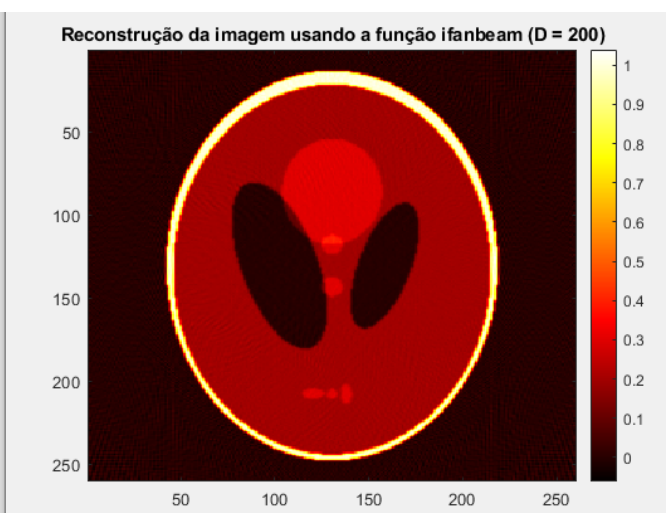
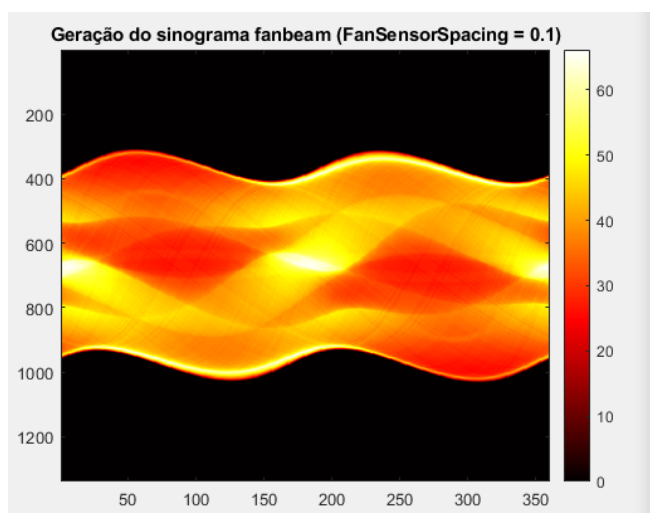
b. Variar o valor de D . O que acontece com a imagem reconstruída?

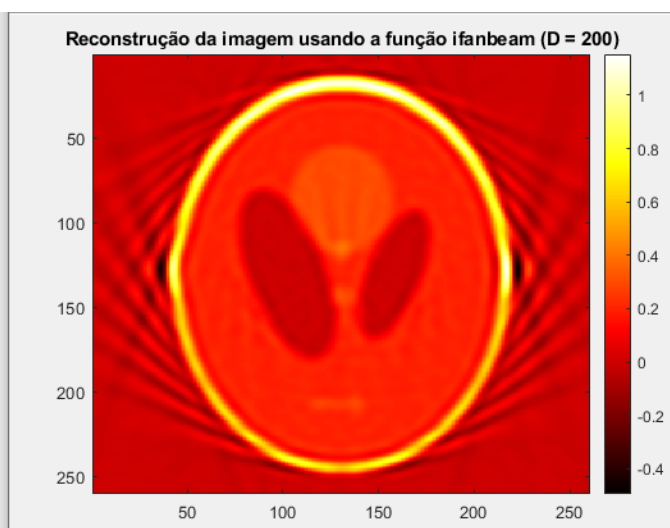
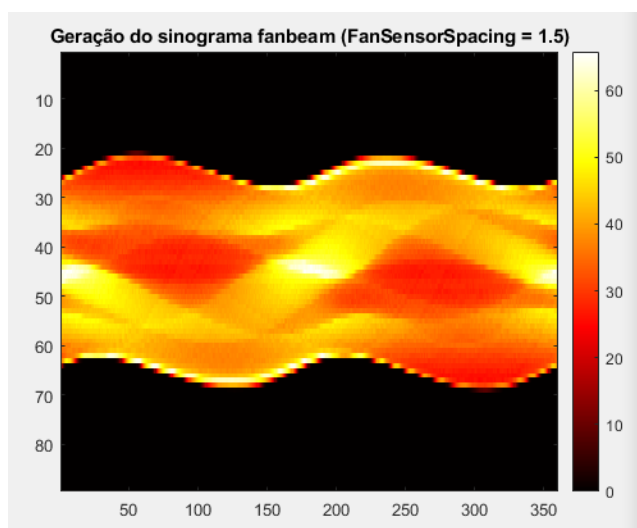
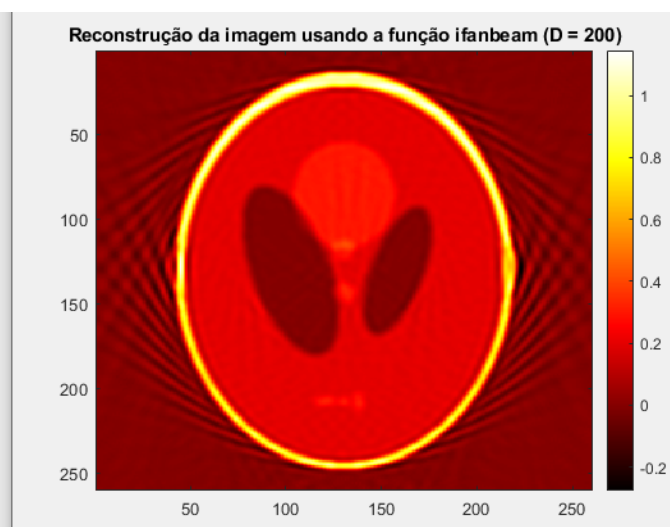
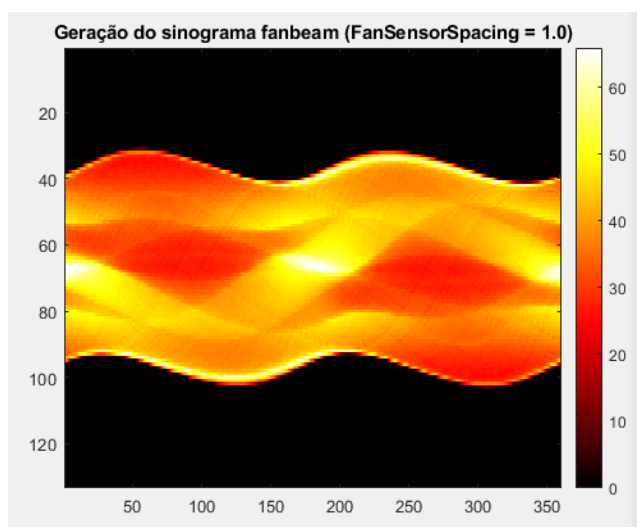
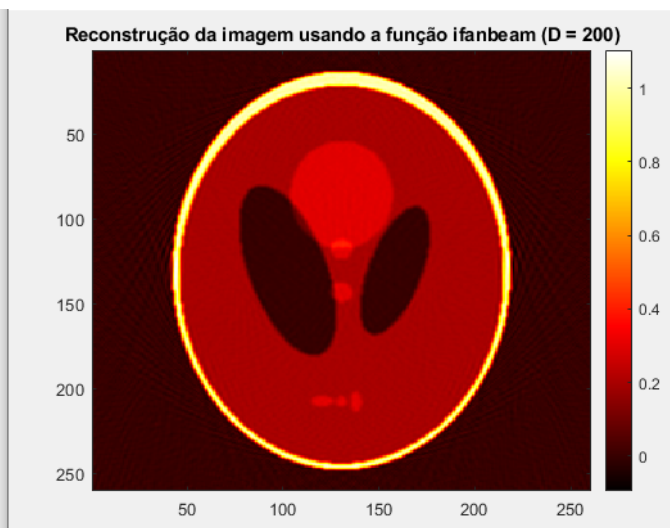
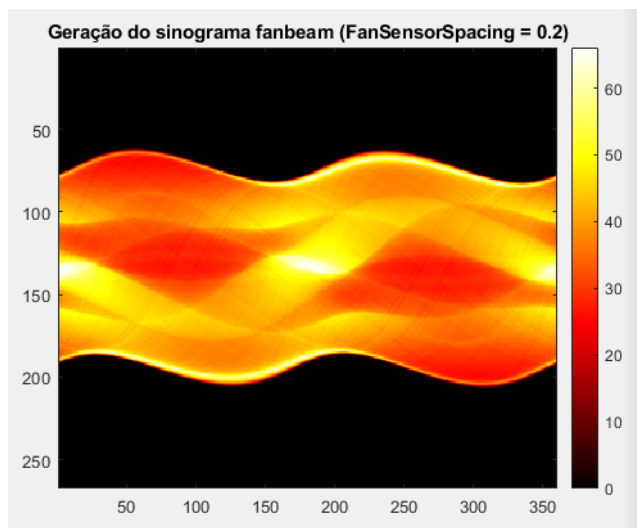
Como o valor do espaçamento entre os feixes de projeção estão fixos, quanto maior a distância D menor o número de feixes é projetado sobre a imagem, logo, o número de projeções vai diminuindo.

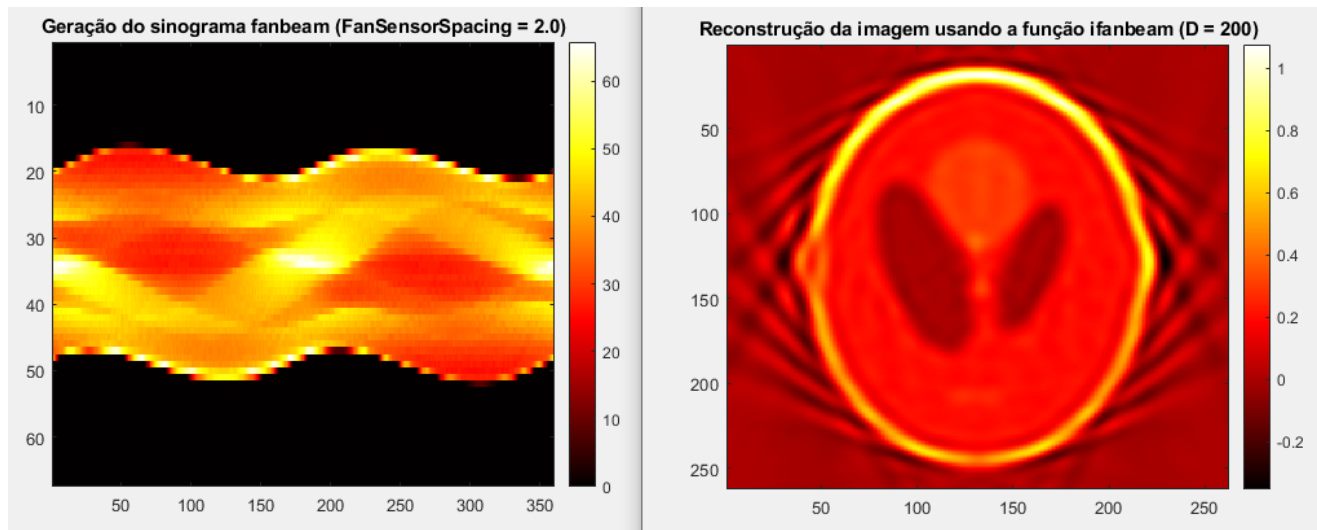




- c. O que representa o parâmetro 'FanSensorSpacing'? Varie o valor desse parâmetro entre 0.1 e 2 e descreva o que acontece com a imagem reconstruída.







- d. Nota: para variar o valor da variável D e do parâmetro 'FanSensorSpacing', é necessário que isso seja realizado tanto durante o processo para construir o sinograma quanto na reconstrução da imagem.

CÓDIGO FONTE:

```
%% Criando a matriz do phantom virtual Shepp-Logan com 256 x 256

% I = phantom(256);
% figure(100);
% imagesc(I);
% title("Phantom virtual Shepp-Logan (256 x 256)");
% colormap gray;
% colorbar;

%% Obtenção das projeções do phantom para ângulos de 0 a 180 graus

% % Número de projeções: 45, 90 e 180

% theta1 = 0:1:180;
% theta2 = 0:2:180;
% theta3 = 0:4:180;

% % Geração dos sinogramas
```



```

% R1 = radon(I,theta1);
% figure(1);
% imagesc(R1);
% title("180 projeções");
% colormap hot;
% colorbar;

% R2 = radon(I,theta2);
% figure(2);
% imagesc(R2);
% title("90 projeções");
% colormap hot;
% colorbar;

% R3 = radon(I,theta3);
% figure(3);
% imagesc(R3);
% title("45 projeções");
% colormap hot;
% colorbar;

%% Reconstrução da imagem do phantom a partir dos sinogramas gerados acima

% % unfiltered

% I1 = iradon(R1,theta1,'linear','none');
% figure(4);
% imagesc(I1);
% title("Unfiltered 180 projeções");
% colormap hot;
% colorbar;
%
% I2 = iradon(R2,theta2,'linear','none');
% figure(5);
% imagesc(I2);

```

```

% title("Unfiltered 90 projeções");
% colormap hot;
% colorbar;
%
% I3 = iradon(R3,theta3,'linear','none');
% figure(6);
% imagesc(I3);
% title("Unfiltered 45 projeções");
% colormap hot;
% colorbar;

% % Ram-Lak filter (default)

% I1 = iradon(R1,theta1);
% figure(4);
% imagesc(I1);
% title("Ram-Lak filter (default) 180 projeções");
% colormap hot;
% colorbar;
%
% I2 = iradon(R2,theta2);
% figure(5);
% imagesc(I2);
% title("Ram-Lak filter (default) 90 projeções");
% colormap hot;
% colorbar;
%
% I3 = iradon(R3,theta3);
% figure(6);
% imagesc(I3);
% title("Ram-Lak filter (default) 45 projeções");
% colormap hot;
% colorbar;

% % Ram-Lak filter (default) usando interpolação nearest
% % Obs: a interpolação linear (default) ja foi observada anteriormente

```

```

% I1 = iradon(R1,theta1,'nearest');
% figure(4);
% imagesc(I1);
% title("Interpolação nearest 180 projeções");
% colormap hot;
% colorbar;
%
% I2 = iradon(R2,theta2,'nearest');
% figure(5);
% imagesc(I2);
% title("Interpolação nearest 90 projeções");
% colormap hot;
% colorbar;
%
% I3 = iradon(R3,theta3,'nearest');
% figure(6);
% imagesc(I3);
% title("Interpolação nearest 45 projeções");
% colormap hot;
% colorbar;

%% Adicionado ruído branco gaussiano ao phantom e refeito os passos anteriores

% I = phantom(256);
%
% % adds zero-mean, Gaussian white noise with variance of 0.01 to grayscale image
% I.
% J = imnoise(I,'gaussian');
%
% figure(200);
% imagesc(J);
% title("Phantom Shepp-Logan (256 x 256) com ruído branco gaussiano");
% colormap gray;
% colorbar;
%
```

```

% % Número de projeções: 45, 90 e 180

%

% theta1 = 0:1:180;

% theta2 = 0:2:180;

% theta3 = 0:4:180;

%

% % Geração dos sinogramas

%

% R1 = radon(J,theta1);

% figure(1);

% imagesc(R1);

% title("180 projeções");

% colormap hot;

% colorbar;

%

% R2 = radon(J,theta2);

% figure(2);

% imagesc(R2);

% title("90 projeções");

% colormap hot;

% colorbar;

%

% R3 = radon(J,theta3);

% figure(3);

% imagesc(R3);

% title("45 projeções");

% colormap hot;

% colorbar;

%

% % Ram-Lak filter (default)

%

% I1 = iradon(R1,theta1);

% figure(4);

% imagesc(I1);

% title("Ram-Lak filter (default) 180 projeções");

% colormap hot;

```

```

% colorbar;

%

% I2 = iradon(R2,theta2);

% figure(5);

% imagesc(I2);

% title("Ram-Lak filter (default) 90 projeções");

% colormap hot;

% colorbar;

%

% I3 = iradon(R3,theta3);

% figure(6);

% imagesc(I3);

% title("Ram-Lak filter (default) 45 projeções");

% colormap hot;

% colorbar;


%% Sinogramas do phantom usando a função fanbeam


I = phantom(256);

% figure(100);

% imagesc(I);

% title("Phantom virtual Shepp-Logan (256 x 256)");

% colormap gray;

% colorbar;


% D = 200;

% % D = sqrt(size(I,1)^2 + size(I,2)^2);

% F = fanbeam(I,D);

% figure(7);

% imagesc(F);

% title('Geração do sinograma usando a função fanbeam (D = 200)');

% colormap hot;

% colorbar;

%

% % reconstrução usando ifanbeam

%

```



```

% I1 = ifanbeam(F,D);
% figure(8);
% imagesc(I1);
% title('Reconstrução da imagem usando a função ifanbeam (D = 200)');
% colormap hot;
% colorbar;

% % testando diferentes parâmetros de FanSensorSpacing (default = 1)
%
% D = 200;
% F = fanbeam(I,D,'FanSensorSpacing',2.0);
%
% figure(7);
% imagesc(F);
% title('Geração do sinograma fanbeam (FanSensorSpacing = 2.0)');
% colormap hot;
% colorbar;
%
% % reconstrução usando ifanbeam
%
% I1 = ifanbeam(F,D,'FanSensorSpacing',2.0);
% figure(8);
% imagesc(I1);
% title('Reconstrução da imagem usando a função ifanbeam (D = 200)');
% colormap hot;
% colorbar;

```