

1. Tomografia axial Computadorizada (CAT)

Pretende-se construir um programa que inverta uma imagem obtida por varrimento (sinograma). Mas como não temos dados vamos criá-los primeiro. O processo segue o que vimos nas aulas "teóricas". Começamos com uma imagem (o paciente!) e queremos fazer uma série de varrimentos que lêem essencialmente a absorção integrada ao longo da direcção particular desse varrimento. O integral é uma soma (integral de Riemann). Pode usar uma imagem à sua escolha (a preto e branco) ou usar a fornecida no Moodle (Shepp-Logan.png). Ou para começar criar uma imagem simples com 2 quadrados(ver no fim).

- (a) escreva uma rotina que, dada uma imagem (**numpy array**) e o número de varrimentos (diferentes ângulos entre 0 e π), nos dá um array que contém o sinograma (faça também o plot). Pontos a ter em conta:
 - i. o seu array de projecções vai receber mais pontos de um varrimento segundo a diagonal que segundo uma aresta da imagem. Convém manipular esta de modo a ter um número de pontos igual qualquer que seja a direcção do varrimento(sugestão: use uma máscara circular e preencha com zeros o resto);
 - ii. as coordenadas da imagem começam num canto (superior esquerdo?) mas o nosso varrimento deve referir-se ao centro da imagem.
 - iii. use funções do **scipy** como **rotate**, **interpolate**
- (b) agora queremos inverter o sinograma para recuperar a imagem (num contexto de imagiologia médica, a máquina de CAT dá-nos o sinograma, nunca tivemos a "imagem" e é esse o nosso objectivo). Comece por fazer uma simples inversão do processo, o que corresponde a uma retro projecção não-filtrada. Faça uma figura do resultado.
- (c) Queremos agora melhorar o resultado usando uma retro projecção filtrada, usando um filtro em rampa, como vimos na aula com o teorema da fatia central. Faça uma nova rotina, baseada na da alínea anterior, mas com a filtragem. Use as rotinas de FFT do **numpy** ou do **numpy**. Compare os resultados com os da alínea anterior.
- (d) Estude como varia o resultado final com o número de projecções.

[Importar dados de uma imagem para arrays numpy.](#)

Código para criar uma simples imagem num domínio de $S \times S$ pontos.

```
import numpy as np
# Construct a simple test target
S = 128
target = np.ones((S,S))
target[S/3:2*S/3,S/3:2*S/3] = 0.5
target[120:136,100:116] = 1.0

plt.figure()
plt.title("Target")
```

```
plt.imshow(target)
plt.show()
```
