Tomografia computadorizada

Exercício Computacional
MAP3122 - Quadrimestral 2021
Prof. Antoine Laurain

Este exercício computacional é individual. Veja as instruções detalhadas no final do texto.

1 Introdução

Problemas inversos são opostos aos *problemas diretos*. Informalmente, em um problema direto encontra-se um efeito de uma causa e, em um problema inverso, recebe-se o efeito e desejamos recuperar a sua causa. A situação mais comum que origina um problema inverso é a necessidade de interpretar medidas físicas indiretas de um objeto de interesse desconhecido. Por exemplo, na tomografia de raios-X, o problema direto é determinar as imagens que obteríamos de um corpo físico cuja estrutura interna conhecemos precisamente, usando raios-X. O problema inverso correspondente é reconstruir a estrutura interna de um corpo físico desconhecido a partir do conhecimento de imagens de raios-X tiradas de diferentes direções. Na figura 1 encontra-se um exemplo bidimensional: a fatia através de uma noz (esquerda) é a causa e a

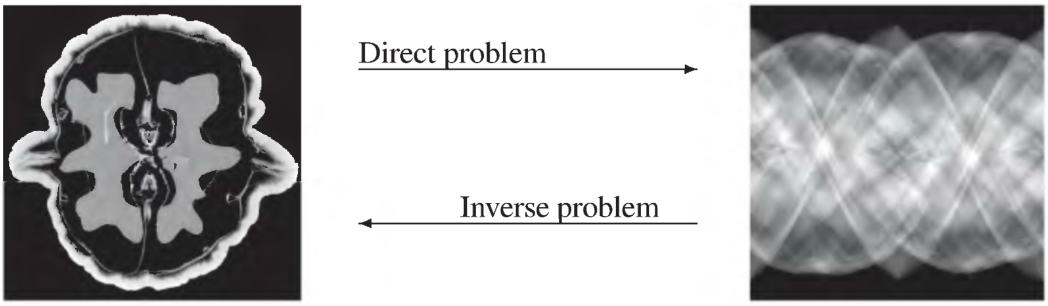

Figura 1: A imagem da fatia de um noz à esquerda é cortesia de Keijo Hamalainen e Aki Kallonen da Universidade de Helsinque, Finlândia (essa imagem vem de [1])

coleta de dados de raios-X (direita) é o efeito. Os dados tomográficos são mostrados na forma tradicional de sinograma.

Problemas diretos são em geral *bem-postos*. A noção de problema bem-posto foi introduzida por Jacques Hadamard (1865-1963). Um problema é bem-posto se ele satisfaz estas três condições:

H1) Existência: existe pelo menos uma solução.

H2) Unicidade: se existir uma solução, ela é única.

H3) Estabilidade: a solução deve depender continuamente dos dados.

O exemplo típico de problema direto é uma equação diferencial parcial (EDP) da física, tais como a equação da onda ou a equação do calor. De fato, para estes problemas, conhecendo as condições iniciais e as fontes, podemos calcular a solução única do problema.

Por outro lado, *problemas inversos* são frequentemente *mal postos*, no sentido que eles não satisfazem pelo menos uma das hipóteses acima. Por exemplo, pode existir um grande número de soluções, e neste caso é difícil saber qual destas soluções é a mais relevante para a aplicação. A razão pela qual estes problemas geralmente são mal postos é porque não temos informações suficientes para encontrar a causa do efeito que estamos observando. Esta falta de informação pode ter muitos motivos. Muitas vezes, é porque só podemos realizar um número limitado de medições: pode ser porque essas medições são caras ou porque a região onde é possível fazer medições é pequena. Mesmo se tivermos apenas dados parciais, gostaríamos de encontrar uma solução aproximada do problema inverso. Problemas inversos são alguns dos problemas matemáticos mais importantes da ciência, da engenharia, e da física, pois nos dizem sobre parâmetros que não podemos observar diretamente.

Neste EP vamos resolver uma versão simplificada de um problema inverso altamente relevante da medicina.

2 Descrição do problema de tomografia

A tomografia é uma técnica de processamento de imagem usada na medicina para visualizar estruturas anatômicas na forma de cortes. Um procedimento para fazer isso é projetar raios-X de muitos diferentes ângulos através do corpo, medir a força dos raios-X que passou pela imagem, e calcular como a imagem deve ser para cumprir com a saída de raios-X. A reconstrução de uma imagem desta forma é chamada de reconstrução tomográfica.

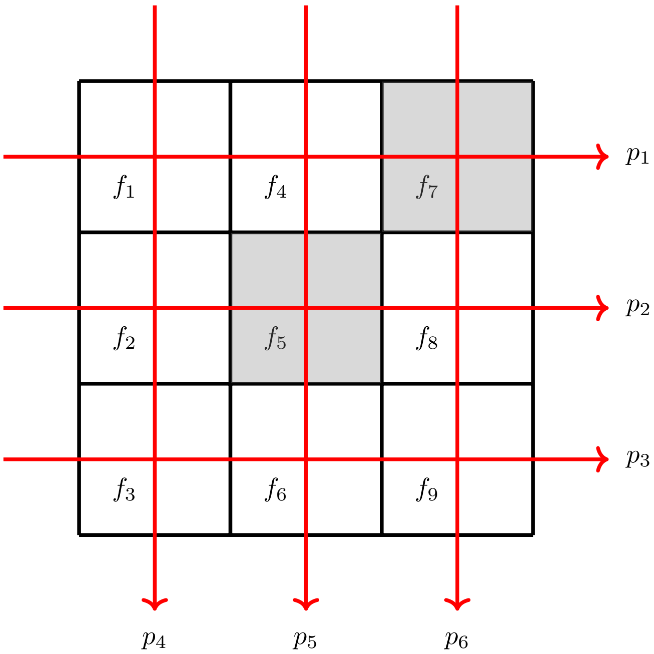

Figura 2: Ilustração de projeções horizontais e verticais numa imagem 3×3.

Uma imagem em tons de cinza de formato retangular pode ser modelada matematicamente como uma função $f : D \to \mathbb{R}$, onde $D \subset \mathbb{R}^2$ é um retângulo, com $f(x, y) \geq 0$ para todos $(x, y) \in D$. Para tratar f numericamente, precisamos primeiro discretizar f. Obtemos uma discretização sobrepondo uma grade sobre o domínio D, obtendo assim uma discretização de D em $n \times n$ pixeis (ver Figura 2). Os valores da imagem discretizada são considerados constantes dentro de cada célula da grade. Representamos estes valores da imagem discretizada como um vetor $(f_j)_{j=1}^{n^2}$, onde $f_j \geq 0$ são constantes. Até o fim deste exercício computacional, identificaremos f com o vetor $(f_j)_{j=1}^{n^2}$.

A Figura 2 mostra a discretização de f no caso particular $n = 3$ e como os valores f_j são ordenadas. Os raios-X são representados por linhas horizontais e verticais atravessando D. As medições correspondentes a estes raios-X são denotadas $(p_i)_{i=1}^{6}$. Cada medição p_i é a soma dos f_j nas células atravessadas pelo i-ésimo raio. Consequentemente, podemos expressar a relação entre os f_j e os p_i como um sistema de equações lineares:

$$\sum_{j=1}^{n^2} A_{ij} f_j = p_i, \qquad \text{para } i = 1, \ldots, 2n,$$

o que pode ser escrito na forma matricial:

$$Af = p \quad \text{com } A \in \mathbb{R}^{2n \times n^2} \text{ e } p \in \mathbb{R}^{2n}.$$

Por exemplo, no caso $n = 3$ da Figura 2, observamos que para a medição p_1 temos:

$$f_1 + f_4 + f_7 = p_1,$$

isto significa

$$A_{11} = 1, A_{12} = 0, A_{13} = 0, A_{14} = 1, A_{15} = 0, A_{16} = 0, A_{17} = 1, A_{18} = 0, A_{19} = 0.$$

Continuando a análise do caso $n = 3$ da Figura 2, chegamos à seguinte expressão do sistema linear $Af = p$, onde $A \in \mathbb{R}^{6\times 9}$, $f \in \mathbb{R}^9$ e $p \in \mathbb{R}^6$:

$$\begin{pmatrix} \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{pmatrix}.$$

Observamos que este sistema linear é subdeterminado, isso quer dizer que têm mais incógnitas f_i (9 incógnitas) do que equações (6 equações). Então se este sistema linear tiver uma solução, a solução não é necessariamente única.

Para definir um tipo de solução para este sistema subdeterminado, podemos definir o seguinte problema dos mínimos quadrados:

$$\min E(f) = \|Af - p\|^2 + \delta\|f\|^2 \text{ com respeito a } f = (f_j)_{j=1}^{n^2}, \tag{1}$$

onde $\|\cdot\|$ é a norma euclidiana e $\delta > 0$ é um escalar. A ideia atras da formulação (1) é o seguinte: para cada medição p dada, podem existir varias f soluções de $Af = p$, pois o sistema linear é subdeterminado, mas dentro deste conjunto de soluções f, o termo $\delta\|f\|^2$ permite (grosso modo) escolher a solução com menor norma euclidiana. Dizemos que $\delta\|f\|^2$ é um *termo de regularização.*

Vamos considerar um exemplo simples. Seja

$$f_\alpha = \begin{pmatrix} 1-\alpha \\ \alpha \\ 0 \\ \alpha \\ 1-\alpha \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \text{ então } Af_\alpha = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = p, \quad \text{para todos } \alpha \in [0, 1].$$

Este exemplo simples mostre que a equação $Af = p$ pode ter uma infinidade de soluções para o mesmo conjunto de medições p dado. Do outro lado, temos

$$\|f_\alpha\|^2 = 2(1-\alpha)^2 + 2\alpha^2 = 4\alpha^2 - 4\alpha + 2,$$

e assim é fácil verificar que o mínimo de $\|f_\alpha\|^2$ com respeito a α é atingindo por $\alpha = 1/2$. Então a função $f = (1/2, 1/2, 0, 1/2, 1/2, 0, 0, 0, 0)^\mathsf{T}$ tem a menor norma euclidiana entre todas as f_α soluções de $Af_\alpha = (1, 1, 0, 1, 1, 0)^\mathsf{T}$.

Podemos mostrar (ver o capítulo sobre método dos mínimos quadrados e sistemas sobredeterminados no curso) que a condição de otimalidade de primeira ordem para o problema de minimização (1) é dada pelo sistema normal seguinte:

$$(A^\mathsf{T} A + \delta I_{n^2}) f_\delta = A^\mathsf{T} p, \tag{2}$$

onde $I_{n^2} \in \mathbb{R}^{n^2\times n^2}$ é a matriz identidade, e A^T é a transposta de A. Observe que $A^\mathsf{T} A \in \mathbb{R}^{n^2\times n^2}$ é uma matriz quadrada. Então definimos a imagem f_δ solução do problema de tomografia como a solução do sistema regularizado (2).

2.1 Exercício 1

Neste exercício, a imagem original será sempre denotada $f^* \in \mathbb{R}^{n^2}$ e as medições correspondentes $p \in \mathbb{R}^{2n}$ satisfazem então $Af = p$, enquanto $f \in \mathbb{R}^{n^2}$ denotará a solução do problema de tomografia (2) com regularização. Observe que em geral temos $f \neq f^*$, mas o objetivo é que f seja o mais perto possível de f^*.

- Usando `numpy.linalg`, calcule o determinante de $A^\intercal A + \delta I_{n^2}$ para os valores $\delta = 0, 10^{-3}, 10^{-2}, 10^{-1}$ e para os três valores de n correspondentes às imagens f^* fornecidas (então são 12 determinantes para calcular, pode organizar os resultados em uma tabela). Como isso justifica o uso da regularização no problema (1)?

- Escreve um programa `tomo1.py` que fornece a imagem f solução do problema de tomografia, isto é, f é a solução da equação (2). Seu programa deve ter $f^* \in \mathbb{R}^{n^2}$ e as medições $p \in \mathbb{R}^{2n}$ correspondentes como entradas. Três pares (f^*, p) são fornecidos, a f^* será usada apenas para plotagens. Seu programa deve deduzir o n do par (f^*, p) fornecido, montar a matriz $A \in \mathbb{R}^{2n \times n^2}$ (a implementação de A deve funcionar para qualquer n, e não apenas para valores particulares de n), e plotar f e f^* lado a lado (os vetores f e f^* têm que ser transformados em matrizes para a plotagem; cuidado que esta transformação sempre tem que seguir a ordem da Figura 2). Use os três valores $\delta = 10^{-3}, 10^{-2}, 10^{-1}$ para o parâmetro de regularização.

Dica 1: A matriz A pode ser montada (no caso geral $A \in \mathbb{R}^{2n \times n^2}$) facilmente usando produtos de Kronecker. O produto de Kronecker de $B \in \mathbb{R}^{m_1 \times n_1}$ e $C \in \mathbb{R}^{m_2 \times n_2}$ é uma matriz em bloco $B \otimes C \in \mathbb{R}^{m_1 m_2 \times n_1 n_2}$ definida por

$$B \otimes C = \begin{pmatrix} BC_{11} & BC_{12} & \dots & BC_{1n_2} \\ BC_{21} & BC_{22} & \dots & BC_{2n_2} \\ \vdots & \vdots & \ddots & \ddots \\ BC_{m_2 1} & BC_{m_2 2} & \dots & BC_{m_2 n_2} \end{pmatrix}.$$

Aqui, cada bloco BC_{ij} é uma matriz de tamanho $m_1 \times n_1$. Pode usar a função `numpy.kron` para definir o produto de Kronecker.

Dica 2: Em caso de dificuldades para implementar o caso geral $A \in \mathbb{R}^{2n \times n^2}$, pode começar implementando o caso particular $n = 3$.

2.2 Exercício 2

O próximo passo é adicionar mais projeções à nossa imagem tomográfica. Conforme ilustrado na figura 3, usamos projeções horizontais, verticais e diagonais. O sistema linear que relaciona a imagem f com as medições p tem a forma:

$$Af = p \quad \text{com } A \in \mathbb{R}^{(6n-2) \times n^2} \text{ e } p \in \mathbb{R}^{6n-2}.$$

No caso particular $n = 3$, o sistema linear $Af = p$ é:

$$\begin{pmatrix}
\mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
\mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} \\
0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1}
\end{pmatrix}
\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{pmatrix}
=
\begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \\ p_9 \\ p_{10} \\ p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{15} \\ p_{16} \end{pmatrix}.$$

Observe que apesar de ter uma estrutura mais complicada que no Exercício 1, a matriz A continua sendo uma matriz em blocos, o que facilita a implementação dela.

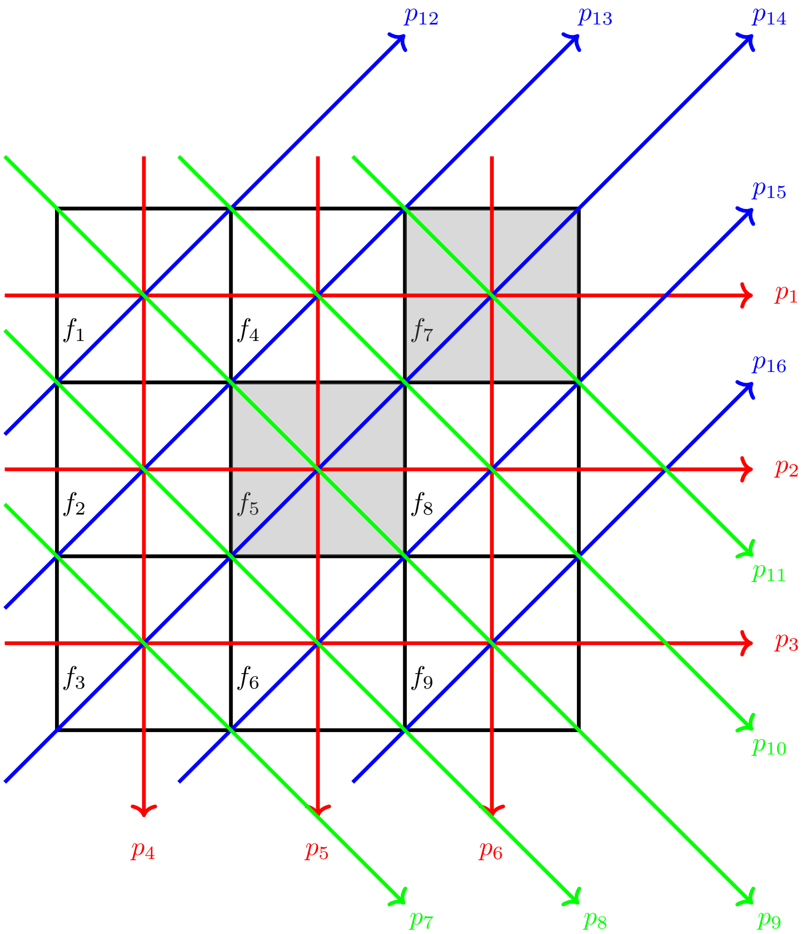

Figura 3: Ilustração de projeções horizontais, verticais e diagonais numa imagem 3×3.

- Usando `numpy.linalg`, calcule o determinante de $A^\mathsf{T} A + \delta I_{n^2}$ para os valores $\delta = 0, 10^{-3}, 10^{-2}, 10^{-1}$ e para os três valores de n correspondentes às imagens f^* fornecidas (então são 12 determinantes para calcular, pode organizar os resultados em uma tabela). Observamos uma diferença com o caso do Exercício 1?

- Escreve um programa `tomo2.py` que fornece a imagem f solução do problema de tomografia, isto é, f é a solução da equação (2). Seu programa deve ter $f^* \in \mathbb{R}^{n^2}$ e as medições $p \in \mathbb{R}^{6n-2}$ correspondentes como entradas. Três pares (f^*, p) são fornecidos, com as mesmas imagens f^* que no Exercício 1. Seu programa deve deduzir o n do par (f^*, p) fornecido, montar a matriz $A \in \mathbb{R}^{(6n-2) \times n^2}$ (a implementação de A deve funcionar para qualquer n, e não apenas para valores particulares de n), e plotar, lado a lado, f, f^*, e também a reconstrução correspondente do Exercício 1. Use os três valores $\delta = 10^{-3}, 10^{-2}, 10^{-1}$ para o parâmetro de regularização.

- Implemente no seu programa o calculo do erro de reconstrução L^2 relativo seguinte:

$$\text{erro} = 100 \times \frac{\|f - f^*\|}{\|f^*\|} = 100 \times \frac{\sqrt{\sum_{j=1}^{n^2} (f_j - f_j^*)^2}}{\sqrt{\sum_{j=1}^{n^2} (f_j^*)^2}}.$$

  Observe que este erro é uma porcentagem. Por exemplo se calcularmos erro $= 4,23$, isso quer dizer que a reconstrução f tem $4,23\%$ de diferença relativamente a imagem original f^* (observe que o erro pode ser maior que 100%). No seu relatório, apresente numa tabela os erros obtidos

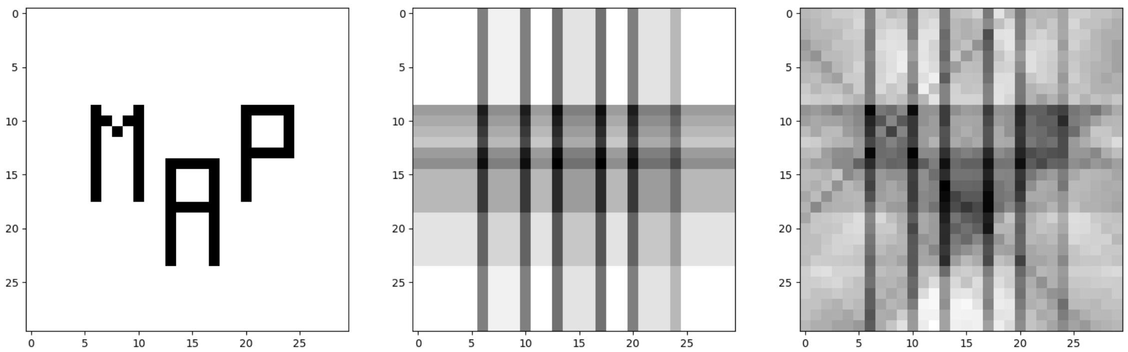

Figura 4: Exemplo de reconstrução. A imagem de esquerda é a figura f^* original. A imagem f no centro é a reconstrução usando apenas medições horizontais e verticais como no exercício 1. A imagem f a direita é a reconstrução usando medições horizontais, verticais e diagonais como no exercício 2. Devido ao número baixo de medições, a reconstrução usando apenas medições horizontais e verticais é de pessima qualidade e não podemos reconhecer as letras. Usando medições horizontais, verticais e diagonais a reconstrução melhora bastante, e começamos a reconhecer as letras, mas a reconstrução ainda é de baixa qualidade. Para obter uma melhor qualidade de reconstrução, são necessárias mais medidas.

para as três pares (f^*, p) fornecidas e os métodos de tomografia (os métodos do Exercício 1 e do Exercício 2). Comente os resultados da tabela.

- Um exemplo de reconstrução é dado na Figura 4.

3 Observações finais

O problema de reconstrução da imagem f^* a partir de medições p visto neste EP é uma versão bastante simplificada de um problema inverso altamente relevante da medicina. Este problema encontra-se frequentemente na literatura com o nome *tomografia computadorizada*.

Em uma aplicação mais realista, as projeções não são apenas horizontais, verticais e diagonais, mas são obtidas usando vários ângulos descrevendo uma volta completa em torno do paciente. Assim, obtemos muito mais informações que nestes dois exercícios e a qualidade da imagem reconstruída f fica bem melhor. Em contrapartida, o problema fica mais difícil do ponto de vista da matemática e computacional. Uma dificuldade adicional na prática é a presença de defeitos nas medições p tais como ruídos.

4 Instruções, observações e dicas

- **Este exercício computacional é individual.**
- O programa deverá ser escrito em Python (preferencialmente usando Python 3, se possível), usando o pacote `numpy`. O seu código deverá estar bem comentado e estruturado. A entrada e a saída deverão ser feitas de forma a ajudar o usuário a executar o programa e devem facilitar a análise dos resultados (por exemplo rodando o código desta forma: `python3 tomo1.py im1`). Se o seu programa precisa de arquivos de entrada, considere que os mesmos encontram-se na mesma pasta do executável, ou faça de forma que solicite o caminho/nome do arquivo ao usuário.
- O uso de bibliotecas não é permitido para resolver os sistemas lineares, você deve implementar seu próprio solver.
- Será usada a biblioteca `matplotlib` para as plotagens.
- As análises e resultados obtidos devem ser organizados em um relatório que deve minimamente discutir os problemas estudados e os resultados obtidos. A entrega deverá conter um relatório (no formato .pdf), contendo a análise do problema estudado e as figuras, e os códigos usados para as simulações computacionais (arquivos `tomo1.py` e `tomo2.py`). A entrega também deverá ser feita em um arquivo compactado único (por exemplo um arquivo `zip`).

- O uso de $\LaTeX$ para escrever o relatório é fortemente incentivado. Os relatórios escritos em Latex receberão um bônus de 5% da nota final.

- As três imagens f^* são dadas no formato `png` (chamam-se `im1.png`, `im2.png`, `im3.png`). As medições correspondentes p são dadas nos arquivos `p1.npy` para o Exercício 1, e nos arquivos `p2.npy` para o Exercício 2 (isto é, o `p1.npy` corresponde à projeções horizontais e verticais do f^*, enquanto `p2.npy` corresponde à projeções horizontais, verticais e diagonais do f^*). Você pode carregar estes arquivos usando por exemplo:

```
from numpy import load
p1 = load('p1.npy')
```

  O resultado é um vetor do tipo `numpy.array`.

Critérios de Correção

- Exercício 1 (3 pts) (Implementação correta da matriz A, resolução do sistema linear, figuras, interpretação dos resultados)
- Exercício 2 (5 pts) (Implementação correta da matriz A, resolução do sistema linear, figuras, interpretação dos resultados, cálculo do erro, figuras, discussão dos resultados obtidos)
- Código bem documentado: comentários, legibilidade. (1 pts)
- Qualidade do relatório (relevância dos comentários e apresentação geral). (1 pts)
- Uso de $\LaTeX$(+5% da nota final)
- Será verificado se o programa entregue roda e produz saídas consistentes com os resultados apresentados no relatório.
- Em caso de atraso de até 48h, -2 pontos. Após isso, o EP não será aceito.

Referências

[1] J. L. Mueller and S. Siltanen. *Linear and nonlinear inverse problems with practical applications*, volume 10 of *Computational Science & Engineering*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2012.