

---

Sobre a escolha da relaxação e ordenação das projeções no método de Kaczmarz com ênfase em implementações altamente paralelas e aplicações em reconstrução tomográfica

*Leonardo Bravo Estácio*

---



SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: \_\_\_\_\_

Sobre a escolha da relaxação e ordenação das projeções  
no método de Kaczmarz com ênfase em implementações  
altamente paralelas e aplicações em reconstrução  
tomográfica

**Leonardo Bravo Estácio**

***Orientador: Prof. Dr. Elias Salomão Neto Helou***

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

**USP – São Carlos**  
**Junho de 2014**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi  
e Seção Técnica de Informática, ICMC/USP,  
com os dados fornecidos pelo(a) autor(a)

E79s Estácio, Leonardo Bravo  
Sobre a Escolha da Relaxação e Ordenação das  
Projeções no Método de Kaczmarz com Ênfase em  
Implementações Altamente Paralelas e Aplicações em  
Reconstrução Tomográfica / Leonardo Bravo Estácio;  
orientador Elias Salomão Helou Neto. -- São Carlos,  
2014.  
72 p.

Dissertação (Mestrado - Programa de Pós-Graduação  
em Ciências de Computação e Matemática  
Computacional) -- Instituto de Ciências Matemáticas  
e de Computação, Universidade de São Paulo, 2014.

1. Kaczmarz. 2. Parâmetro de Relaxação. 3. Métodos  
Numéricos. 4. Algoritmos Paralelos. I. Helou Neto,  
Elias Salomão, orient. II. Título.

# Agradecimentos

---

---

Primeiramente agradeço os professores das disciplinas que cursei durante o desenvolvimento do mestrado em especial aos do LOt – Laboratório de Otimização do Instituto de Ciências Matemáticas e de Computação. Em especial ao meu orientador, Elias Salomão Helou Neto pela paciência e inspirações me proporcionadas e que possibilitaram, depois de muito estudo e questionamentos, o desenvolvimento dessa pesquisa ao longo destes dois anos.

Agradeço a minha namorada Viviane, meus pais Luiz Mário e Mônica, e irmãos Bruno e Igor, que mesmo longe sempre souberam estar presentes mesmo nos momentos mais difíceis e sempre aceitar as minhas decisões e escolhas.

Agradeço a meus amigos pelo companheirismo, reflexões, sugestões relacionadas ao projeto e pelas trocas de experiências.

Agradeço à FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo, pela concessão de bolsa e financiamento do projeto.

A todos aqueles que me incentivaram, acreditaram, apoiaram, enfim que de qualquer maneira, sabem de sua participação na realização deste trabalho, porém não foram citados, meu imenso e sincero **muito obrigado**.



*“Se você permanecer perdido na escuridão, por ela será engolido. Acenda uma luz e a ilumine.”*



O método de Kaczmarz é um algoritmo iterativo que soluciona sistemas lineares do tipo  $Ax = b$  através de projeções sobre hiperplanos bastante usado em aplicações que envolvem a Tomografia Computadorizada. Recentemente voltou a ser destaque após a publicação de uma versão aleatória apresentada por Strohmer e Vershynin em 2009 a qual foi provada possuir taxa de convergência esperada exponencial. Posteriormente, Eldar e Needell em 2011 sugeriram uma versão modificada do algoritmo de Strohmer e Vershynin, na qual a cada iteração é selecionada a projeção ótima a partir de um conjunto aleatório, utilizando para isto o lema de Johnson-Lindenstrauss. Nenhum dos artigos mencionados apresenta uma técnica para a escolha do parâmetro de relaxação, entretanto, a seleção apropriada deste parâmetro pode ter uma influência substancial na velocidade do método. Neste trabalho apresentamos uma metodologia para a escolha do parâmetro de relaxação, bem como implementações paralelas do algoritmo de Kaczmarz utilizando as ideias de Eldar e Needell. Nossa metodologia para seleção do parâmetro utiliza uma nova generalização dos resultados de Strohmer e Vershynin que agora leva em consideração o parâmetro  $\lambda$  de relaxação e, a partir daí, obtemos uma estimativa da taxa de convergência como função de  $\lambda$ . Escolhemos então, para uso no algoritmo, aquele que otimiza esta estimativa. A paralelização dos métodos foi realizada através da plataforma CUDA e se mostrou muito promissora, pois conseguimos, através dela, um ganho significativo na velocidade de convergência.



# Abstract

---

---

The Kaczmarz method is an iterative algorithm for finding the solution of a system of linear equations  $Ax = b$  by projecting onto the hyperplanes widely used in applications involving Computerized Tomography. It has been recently highlighted after the publication of a random version presented by Strohmer and Vershynin in 2009 that yields probably exponential convergence in expectation. Thereafter, Eldar and Needell in 2011 suggested a modified version of Strohmer and Vershynin algorithm, which at each iteration selects the optimal projection from a random set making use of the Johnson-Lindenstrauss lemma. None of the mentioned articles presents a technique for choosing the relaxation parameter, however, the proper selection of this parameter can achieve a substantial gain on the speed of the method. In this project we present a methodology for finding the relaxation parameter, as well as parallel implementations of Kaczmarz's Algorithm using the ideas of Eldar and Needell. Our methodology for parameter selection uses a new generalization on Strohmer and Vershynin's results which now regards the relaxation parameter  $\lambda$ . Thenceforward, we obtain an estimate of the convergence rate as a function of  $\lambda$ . Then we use this estimate in the algorithm the optimizer of this estimate. The parallelization of the methods has been implemented through the CUDA platform and appears to be very promising, since it delivers substantial gain in the convergence speed.



# Sumário

---

---

<b>Resumo</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização . . . . .	1
1.2 Contribuições . . . . .	3
<b>2 Aplicações em Tomografia Computadorizada</b>	<b>5</b>
2.1 Histórico e <i>Background</i> da Tomografia . . . . .	5
2.2 Princípios da Tomografia Computadorizada por Transmissão . . . . .	9
2.3 Tomografia por Emissão . . . . .	11
2.4 A Transformada de Radon e o Teorema da Fatia de Fourier . . . . .	13
2.5 Reconstrução de Imagens Tomográficas em um Modelo Discreto . . . . .	18
<b>3 Simulação do Traçado dos Feixes</b>	<b>23</b>
3.1 O Algoritmo de Siddon . . . . .	23
3.2 Exemplo do Traçado do Feixe . . . . .	25
3.3 Exemplo de simulação da Transformada de Radon . . . . .	28
<b>4 Técnica de Reconstrução Algébrica</b>	<b>33</b>
4.1 Método Clássico de Kaczmarz . . . . .	34
4.2 Método Kaczmarz Aleatório . . . . .	36
4.3 Método Kaczmarz Aleatório acelerado através do Lema de Johnson-Lindenstrauss	38
4.4 Método Kaczmarz Aleatório Paralelo . . . . .	39
4.4.1 Arquitetura CUDA . . . . .	41
<b>5 Sobre a Escolha do Parâmetro de Relaxação</b>	<b>45</b>
5.1 Taxa de Convergência Esperada Exponencial Considerando o Parâmetro de Relaxação . . . . .	45
5.2 Encontrando o Parâmetro de Relaxação . . . . .	48
<b>6 Simulações Numéricas</b>	<b>53</b>
6.1 Simulação Numérica da Estimativa . . . . .	53
6.2 Comparação do Algoritmo CGLS e o Algoritmo Aleatório de Strohmer . . . . .	61
6.3 Algoritmo Paralelo . . . . .	63

**7 Considerações Finais**

**65**

**Referências Bibliográficas**

**67**

---

# Lista de Figuras

---

---

2.1	Tomografia Linear . . . . .	6
2.2	Tomografia Axial Transversal . . . . .	6
2.3	Tomógrafo . . . . .	7
2.4	Primeira geração de escaners . . . . .	8
2.5	Segunda geração de escaners . . . . .	8
2.6	Terceira geração de escaners . . . . .	9
2.7	Radiografia Convencional . . . . .	10
2.8	Feixe de fótons atravessando um corpo . . . . .	10
2.9	PET - Tomografia por Emissão de Pósitrons . . . . .	12
2.10	SPECT - Tomografia por Emissão de Fóton Único . . . . .	12
2.11	Feixes paralelos variando ângulos $\in [0, 2\pi)$ . . . . .	13
2.12	Vetor $s$ perpendicular ao feixe $t$ . . . . .	14
2.13	Geometria do problema - Transformada de Radon . . . . .	14
2.14	Projeção $\mathcal{R}_\theta f(t)$ com ângulo $\theta = 135^\circ$ no <i>phantom</i> de Shepp-Logan . . . . .	15
2.15	Valor da Transformada de Radon aplicada ao <i>phantom</i> de Shepp-Logan com o ângulo $\theta = 135^\circ$ e o valor absoluto da Transformada de Fourier desta projeção . . . . .	17
2.16	Fatia Fourier - Transformada de Fourier aplicada em uma projeção da Transformada de Radon . . . . .	17
2.17	Matriz de <i>pixels</i> . . . . .	19
2.18	<i>phantom</i> de Shepp-Logan . . . . .	20
2.19	Sinogram do <i>phantom</i> de Shepp-Logan . . . . .	20
2.20	Reconstrução do <i>phantom</i> de Shepp-Logan. A primeira imagem foi reconstruída utilizando $p = 20.084$ projeções, a segunda utilizando $2p$ projeções, a terceira usando $3p$ projeções e a quarta utilizando $4p$ projeções. . . . .	21
3.1	Calculando as retas igualmente espaçadas traçadas no plano satisfazendo $x = 0 \leq i \leq m$ e $y = 0 \leq j \leq n$ . . . . .	25
3.2	Obtendo a interseção do feixe com as retas paralelas obtidas na Figura 3.1 . . . . .	25
3.3	Valores do conjunto $\alpha$ representam os escalares que quando multiplicados pelo vetor definido por $P_{ini}$ e $P_{fim}$ obtemos os pontos de interseção entre as retas paralelas e o feixe da Figura 3.2 . . . . .	26
3.4	Distâncias entre os pontos, dentro do intervalo definido, que representam cada um dos seguimentos de reta da Figura 3.3 . . . . .	26
3.5	Diferenças entre as coordenadas do plano e matriz . . . . .	27

3.6	Uma maneira de mapearmos as coordenadas cartesianas do feixe para as coordenadas da matriz é considerarmos que o feixe atravessando a área definida pelas retas paralelas é perpendicular ao ângulo do feixe original . . . . .	27
3.7	Mapeando as coordenadas do plano para as coordenadas da matriz utilizando o conjunto $\alpha_{mid}$ , cujo valores representam os escalares que quando multiplicados pelo vetor definido por $P_{ini}$ e $P_{fim}$ obtemos o ponto médio dos segmentos obtidos na Figura 3.2 . . . . .	28
3.8	Ponto central, magnitude do raio e magnitude da amplitude pertencentes a circunferência imaginária que passa em torno da área definida . . . . .	29
3.9	Retas paralelas e seus respectivos pontos que passam na tangente da circunferência imaginária e são perpendiculares ao ângulo $\theta$ do feixe . . . . .	29
3.10	Distância entre os feixes paralelos de um mesmo ângulo $\theta$ . . . . .	30
3.11	Feixes de uma Projecção da transformada de Radon discreta para $\theta = 30^0$ . . . . .	30
4.1	Representação geométrica do algoritmo clássico de Kaczmarz . . . . .	34
4.2	Representação geométrica do algoritmo clássico de Kaczmarz ordenando as equações do sistema de forma ótima . . . . .	35
4.3	Efeito do parâmetro de relaxação sobrelaxado e subrelaxado no Método de Kaczmarz clássico . . . . .	35
4.4	Representação geométrica do Método de Kaczmarz Aleatório . . . . .	36
4.5	Kaczmarz Projecção Ótima . . . . .	38
4.6	Arquitetura CUDA . . . . .	41
4.7	Modelo de memória da CUDA, as setas indicam a direção de acesso à memória. No caso das memórias constante e de textura, é permitido apenas a leitura. . . . .	42
6.1	Taxa de convergência do Método de Kaczmarz Aleatório em função da variação do parâmetro de relaxação e valor do chute inicial. . . . .	54
6.2	Figura gerada utilizando um chute inicial $x^0 = 100$ e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro $\ x^* - x^k\ _2$ ao final de 10000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de $[0, 1]$ utilizando (6.2) a fim de observar em qual $\lambda$ a função atingiu o seu mínimo e comparar com os mínimos obtidos minimizando as estimativas . . . . .	55
6.3	Figura gerada utilizando um chute inicial $x^0 = 100$ e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro $\ x^* - x^k\ _2$ ao final de 15000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de $[0, 1]$ utilizando (6.2) a fim de observar em qual $\lambda$ a função atingiu o seu mínimo e comparar com os mínimos obtidos minimizando as estimativas . . . . .	55
6.4	Figura gerada utilizando um chute inicial $x^0 = 10$ e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro $\ x^* - x^k\ _2$ ao final de 10000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de $[0, 1]$ utilizando (6.2) a fim de observar em qual $\lambda$ a função atingiu o seu mínimo e comparar com os mínimos obtidos minimizando as estimativas . . . . .	56
6.5	Figura gerada utilizando um chute inicial $x^0 = 10$ e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro $\ x^* - x^k\ _2$ ao final de 15000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de $[0, 1]$ utilizando (6.2) a fim de observar em qual $\lambda$ a função atingiu o seu mínimo e comparar com os mínimos obtidos minimizando as estimativas . . . . .	56

6.6	Figura gerada utilizando um chute inicial $x^0 = 0$ e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro $\ x^* - x^k\ _2$ ao final de 10000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de $[0, 1]$ utilizando (6.2) a fim de observar em qual $\lambda$ a função atingiu o seu mínimo e comparar com os mínimos obtidos minimizando as estimativas . . . . .	57
6.7	Figura gerada utilizando um chute inicial $x^0 = 0$ e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro $\ x^* - x^k\ _2$ ao final de 15000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de $[0, 1]$ utilizando (6.2) a fim de observar em qual $\lambda$ a função atingiu o seu mínimo e comparar com os mínimos obtidos minimizando as estimativas . . . . .	57
6.8	Comparação da taxa de convergência do Método Kaczmarz Aleatório de Strohmer e Vershynin em função da variação do parâmetro de relaxação na reconstrução do <i>phantom</i> de Shepp-Logan . . . . .	58
6.9	Valores de $\ x^k - x^*\ $ apresentando na Figura 6.8 e Valores de (5.20) com $\lambda \in (0, 2)$ . Ambos em escala $[0, 1]$ . . . . .	58
6.10	Comparação da taxa de convergência do Método de Kaczmarz Aleatório de Strohmer e Vershynin e o Método de Kaczmarz Aleatório Johnson-Lindenstrauss de Eldar e Needell em função da variação do parâmetro de relaxação para $x^0 = 1$ . . .	59
6.11	Valores de $\ x^k - x^*\ $ apresentando na Figura 6.10, valores de (5.20) e valores de (5.22) com $\lambda \in (0, 2)$ . Todos em escala $[0, 1]$ . . . . .	59
6.12	Comparação da taxa de convergência do Método de Kaczmarz Aleatório de Strohmer e Vershynin e o Método de Kaczmarz Aleatório Johnson-Lindenstrauss de Eldar e Needell em função da variação do parâmetro de relaxação para $x^0 = 10$ . .	60
6.13	Valores de $\ x^k - x^*\ $ apresentando na Figura 6.12, valores de (5.20) e valores de (5.22) com $\lambda \in (0, 2)$ . Todos em escala $[0, 1]$ . . . . .	60
6.14	Comparação da velocidade de convergência do algoritmo Kaczmarz aleatório de Strohmer e o método dos Gradientes Conjugados para equações lineares não simétricas. . . . .	61
6.15	Tempo gasto para convergência do algoritmo Kaczmarz Paralelo com critério de parada $\ x^k - x^*\  < 10^{-2}$ executado na máquina descrita no texto. . . . .	63
6.16	A primeira reconstrução foi realizada com 5021 projeções e levou 8.66 segundos para ser concluída. A segunda 10042 e 19.36 segundos. A terceira 15063 projeções e 29.97 segundos. E a quarta 20084 projeções e 40.67 segundos. Todas utilizando 120 <i>threads</i> . . . . .	64



# Introdução

## 1.1 Contextualização

A solução de sistemas lineares  $Ax = b$  com soluções factíveis, onde  $A \in \mathbb{R}^{m \times n}$ ,  $x \in \mathbb{R}^n$  e  $b \in \mathbb{R}^m$  aparece com frequência em métodos numéricos, seja de forma direta ou como ferramenta intermediária na obtenção da solução de problemas mais complexos. Apesar da aparente simplicidade do problema de encontrar  $x$  satisfazendo a igualdade  $Ax = b$ , diversos fatores contribuem para que o uso de métodos diretos tradicionais seja inviável em aplicações que envolvam a resolução de grandes sistemas de equações lineares esparsos e de grande uso de memória. Para estas aplicações, são utilizados métodos iterativos [14, 33, 38, 75].

O método de Kaczmarz [48], também conhecido na literatura como Técnica de Reconstrução Algébrica (ART - *Algebraic Reconstruction Technique*), é um algoritmo iterativo para solução de sistemas lineares  $Ax = b$ . Devido à sua simplicidade e velocidade, vem sendo usado em uma variedade de aplicações partindo da tomografia e indo até o processamento digital de sinais [31, 59, 78]. O algoritmo consiste em uma série de projeções alternadas realizadas nos hiperplanos definidos pelas equações do sistema.

Sejam  $a_1, \dots, a_m \in \mathbb{R}^n$  as linhas de  $A \in \mathbb{R}^{m \times n}$ . O método de Kaczmarz clássico escolhe a sequência de projeções de maneira cíclica. Considerando cada iteração como uma única projeção ortogonal, o método pode ser descrito por

$$x^{k+1} = x^k + \lambda \frac{b_i - \langle a_i, x^k \rangle}{\|a_i\|_2^2} a_i, \quad (1.1)$$

onde  $x^k$  é a  $k$ -ésima iteração,  $i = (k \bmod m) + 1$ ,  $\|\cdot\|_2$  denota a norma Euclidiana ou norma espectral para vetores ou matrizes e  $\lambda$  o parâmetro de relaxação.

Embora esta técnica tenha sido usada na prática por bastante tempo, estimativas quantitativas de sua taxa de convergência foram difíceis de serem obtidas [26, 35, 42]. Pelo algoritmo, podemos perceber que a taxa de convergência depende da ordenação das linhas da matriz  $A$  e, portanto, linhas mal ordenadas podem conduzir a uma lenta convergência. Para superar esta dificuldade, as linhas da matriz  $A$  podem ser selecionadas de uma maneira aleatória. Foi observado que esta

versão aleatória do algoritmo pode melhorar a taxa de convergência [43, 59], entretanto apenas recentemente resultados teóricos foram obtidos [62, 84, 85].

Uma variação do método clássico apresentada por Strohmer e Vershynin em [85] consiste em selecionar os hiperplanos para serem realizadas as projeções através de uma probabilidade proporcional a sua norma euclidiana. Sua abordagem pode ser descrita por

$$x^{k+1} = x^k + \lambda \frac{b_{p(k)} - \langle a_{p(k)}, x^k \rangle}{\|a_{p(k)}\|_2^2} a_{p(k)},$$

onde  $p(k)$  assume o valor  $i \in \{1, 2, \dots, m\}$  com probabilidade  $\frac{\|a_i\|_2^2}{\|A\|_F^2}$  e  $\|A\|_F$  denota a norma de Frobenius. Esta abordagem foi provada possuir limite exponencial na taxa de convergência dada por

$$\mathbb{E}\|x^k - x\|_2^2 \leq \left(1 - \frac{1}{\hat{\kappa}(A)}\right)^k \|x^0 - x\|_2^2,$$

onde  $\hat{\kappa}(A) = \frac{\|A\|_F}{\underline{\sigma}}$  (supondo  $\underline{\sigma} > 0$ ) é o número de condição escalado [25],  $\underline{\sigma}$  é o menor valor singular de  $A$ ,  $x^0$  é o valor do chute inicial e  $\mathbb{E}$  denota a esperança sobre a escolha das linhas. Isto assumindo que a matriz  $A$  é de posto completo de modo que  $\|A^{-1}\| \stackrel{def}{=} \inf \{M : M\|Ax\|_2 \geq \|x\|_2, \forall x\}$  esteja bem definido.

Após estes desenvolvimentos iniciais, foi possível acelerar a taxa de convergência do algoritmo através de uma estratégia de redução de dimensionalidade baseada no Lema de Johnson-Lindenstrauss [29, 47]. Esta adaptação possui um caráter essencialmente paralelo na qual diversas projeções são calculadas (em um espaço de dimensão reduzida), mas somente a que mais avança na direção da solução é utilizada.

Herman iniciou em 1978 [45] os estudos a respeito do parâmetro de relaxação provando que o algoritmo converge caso  $0 < \lambda < 2$ . A partir daí, muitos trabalhos [19, 29, 43, 44, 85] comentam que a taxa de convergência do método de Kaczmarz pode ser acelerada através da seleção de um bom parâmetro de relaxação. Entretanto, nenhum apresenta um critério para a sua seleção. A escolha de um parâmetro inapropriado pode reduzir a velocidade de convergência do algoritmo enquanto que bons valores para estes parâmetros resultam em um algoritmo altamente competitivo, em alguns casos sendo comparado favoravelmente com métodos bem estabelecidos como o dos Gradientes Conjugados [85].

Existem vários motivos que levam à necessidade de métodos iterativos para a solução de sistemas de equações lineares. Um deles é que em muitas aplicações, como na tomografia computadorizada [17, 43, 80], trabalhamos com uma matriz  $A$  que possui dimensões enormes, é esparsa e muito grande para ser armazenada na memória, assim, precisamos apenas armazenar os elementos não nulos em uma estrutura própria para este tipo de matriz.

Nestas matrizes esparsas, tanto o produto interno de vetores como o cálculo da multiplicação matriz vetor pode ser efetuado de maneira eficientemente linha por linha sem alterar seus elementos [75], e é desta forma de operar vários algoritmos iterativos. Assim, abordagens tradicionais, que alteram os coeficientes da matriz original, tornam-se impraticáveis e precisam ser substituídas por algoritmos iterativos. Uma vez que nesses casos existiria a necessidade de uma quantidade maior de memória para armazenar outros elementos, que deixariam de ser nulos na matriz original, quando utilizamos um método direto.

Na tomografia computadorizada, nosso objetivo é obter a imagem de uma secção transversal de um objeto de estudo através do conhecimento de suas integrais de linha ao longo de retas. Isto é, dados os valores de  $(\theta, t)$ , desejamos ser capaz de reconstruir  $f(x, y)$  utilizando a transformada

de Radon:

$$\mathcal{R}f(\theta, t) = \int_{\mathbb{R}} f(t \cos \theta - s \sin \theta, t \sin \theta + s \cos \theta) ds, \quad (1.2)$$

no Capítulo 2 Seção 2.4 comentamos sobre a transformada de Radon.

Há formas de se obter  $f$  analiticamente e podemos utilizar essas fórmulas para obter algoritmos diretos para a solução da Equação (1.2) [24, 59, 60]. De fato, os tomógrafos atuais utilizam quase que exclusivamente de métodos baseados em fórmulas fechadas de inversão da integral acima, mas tais algoritmos possuem diversos inconvenientes que podem impedir sua adoção em aparelhos mais modernos ou especializados [64], dentre os quais podemos citar a dificuldade em acomodar novas geometrias de aquisição (processos de escaneamento discutidos no Capítulo 2 Seção 2.1), modelar o efeito do ruído estatístico nas medidas [63] e gerenciamento de memória do método.

Outra abordagem é adotar uma discretização do espaço de possíveis imagens desde o começo da formulação do modelo de reconstrução. Como o operador de integral de linha acima é linear, a transformação que leva  $f(x, y)$  em  $\mathcal{R}f(\theta, t)$  utilizando a Equação (1.2) [59], quando discretizada, toma a forma de uma matriz e o problema reduz-se à solução do sistema linear

$$Ax = b, \quad (1.3)$$

onde  $A$  é uma matriz muito grande e esparsa. A principal vantagem desta abordagem é a ampla flexibilidade permitida na modelagem. Novas geometrias de aquisição são facilmente incorporadas na matriz do sistema, bem como outros efeitos presentes em uma aquisição tomográfica que são normalmente ignorados no modelo de integrais de linha. O processo de discretização do problema é discutido no Capítulo 2 Seção 2.5 e no Capítulo 3.

A reconstrução de imagens na Tomografia Computadorizada pode ser resumida no cálculo da inversa da transformada de Radon de maneira analítica ou através da solução de um sistema linear resultante da discretização do problema. Basicamente, podemos solucionar este problema através de algoritmos diretos, como o FBP – *Filtered Back-Projection* ou algoritmos iterativos, como o método de Kaczmarz. O FBP, que é baseado no Teorema do Corte Fourier (ver Capítulo 2 Seção 2.4), é o algoritmo mais usado atualmente na reconstrução de imagens, porém recentemente, ele vem perdendo bastante espaço devido a avanços nas tecnologias de *multi-slice* e feixes em cone [73]. Deste modo, os métodos iterativos, que se comportam de melhor maneira nesses problemas, voltaram a ser objeto de pesquisa.

O método de Kaczmarz vem sendo empregado com sucesso na solução de problemas tomográficos, o que é nossa motivação inicial para o seu estudo. Porém, dado o relativo sucesso de métodos analíticos (principalmente o FBP [59]) na reconstrução em tomógrafos por raios  $X$  convencionais, alguma motivação extra é necessária para justificar a pesquisa deste algoritmo em tomografia e ela encontra-se no fato de que o método serve como ferramenta para obtenção de algoritmos mais sofisticados de otimização, os quais permitem o uso de modelos melhores do que o simples sistema linear acima [19, 23, 40, 68, 82], de forma que quaisquer melhorias obtidas em seu desempenho possuem grande impacto na literatura.

## 1.2 Contribuições

A primeira contribuição do projeto foi sistematizar a escolha do parâmetro de relaxação no método de Kaczmarz com a ordenação aleatória de Sthromer e Vershynin [85, 86] de forma a acelerar a convergência do algoritmo. Para isto, partimos dos resultados apresentados no artigo [85], onde são exibidos experimentos em que o algoritmo converge mais rapidamente quando o tamanho de passo é escolhido corretamente, mas nenhum esforço é apresentando no sentido de

sistematizar esta escolha. E conseguimos encontrar uma forma de estimar o parâmetro baseando-se na taxa de convergência esperada do método.

A segunda contribuição foi aproveitar a abordagem de Eldar e Needel [29] e utilizá-la em implementações paralelas do algoritmo de Kaczmarz. Basicamente, o algoritmo apresentado em [29] modifica ligeiramente a ideia básica de [85] para que sejam escolhidas diversas linhas da matriz aleatoriamente, ao invés de apenas uma, e, dentre estas, a que mais avança na direção da solução é utilizada.

Por fim, unimos ambas contribuições e obtemos um algoritmo paralelo no qual o valor do parâmetro de relaxação é escolhido por nossa estimativa e apresentamos resultados de sua velocidade de convergência e utilidade na reconstrução de imagens. Porém, neste algoritmo paralelo, os parâmetros de relaxação podem não ser tão bem escolhidos como no Método de Kaczmarz Aleatório de Strohmer e Vershiny, uma vez que o algoritmo paralelo altera a probabilidade de seleção definida em [85].

---

# Aplicações em Tomografia Computadorizada

---

No presente Capítulo, apresentamos um breve histórico e *background* da Tomografia Computadorizada na Seção 2.1, a qual comenta suas origens e sua evolução ao decorrer do tempo. Os princípios fundamentais da Tomografia Computadorizada por transmissão são discutidos na Seção 2.2, onde definimos seu objetivo, as principais diferenças em relação a radiografia convencional e a dedução de seu problema teórico. Introduzimos os conceitos envolvidos na Tomografia por Emissão e sua diferença com relação a Tomografia por Transmissão na Seção 2.3. Na Seção 2.4, apresentamos a base matemática para a teoria de reconstrução de imagens tomográficas a partir de projeções e comentamos sobre o Teorema da Fatia Fourier. E, na Seção 2.5, damos uma rápida introdução sobre a formulação discreta do problema de reconstrução de imagens a partir de projeções e sobre o *phantom* de Shepp-Logan.

## 2.1 Histórico e *Background* da Tomografia

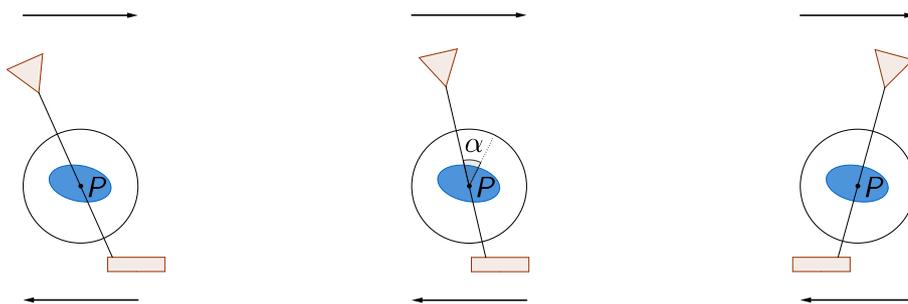
O surgimento da Tomografia Computadorizada (*TC*) foi um marco revolucionário para a radiologia tendo contribuído fortemente nos avanços voltados ao campo de diagnósticos radiológicos através de imagens [77]. A possibilidade de reprodução de uma imagem de uma seção transversal de um corpo tornou-se bastante popular em todo mundo desde a descoberta dos raios X em 1895 e posteriormente amplificada em 1972 com o desenvolvimento da tomografia computadorizada [56].

Wilhelm Röntgen em 1895 descobriu os raios X enquanto realizava experimentos com tubos de raios catódicos. Nestes experimentos, utilizou telas fluorescentes, que começavam a se iluminar quando eram atingidas pela luz emitida pelo tubo. Para sua surpresa, este efeito persistia mesmo quando o tubo era colocado dentro de uma caixa de papelão. Ele logo percebeu que este tubo não emitia apenas luz, mas também um novo tipo de radiação, que ele chamou de raios X por causa da sua estranha natureza. Este novo tipo de radiação não podia apenas atravessar a caixa mas também atenuava de maneira diferente em vários tipos de materiais e permitia, como a luz, ser capturada em um filme fotográfico. Isto abriu um grande leque de possibilidades para o seu uso na medicina. A primeira “Foto de Röntgen”, como foi chamada na época, de uma mão foi feita apenas alguns meses depois da descoberta dos raios X e a natureza dos raios X como pulsos eletromagnéticos

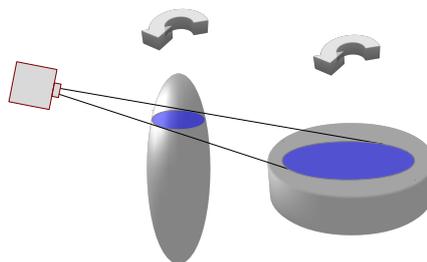
de radiação de curto alcance foi estabelecida por Max von Laue em 1912 [41]. A contribuição de Röntgen foi consagrada com o prêmio Nobel de física de 1901 [87].

Nos anos 60, através dos avanços da tecnologia dos computadores em geral, a tomografia computadorizada tornou-se viável, pois antes, as máquinas não poderiam dar conta do esforço computacional necessário para a reconstrução de imagens [15, 50]. Porém, as ideias que fundamentaram sua teoria foram traçadas muito antes disto, Johann Karl August Radon provou no ano de 1917, que caso obtivéssemos as intensidades das projeções de todos os ângulos de uma fatia bidimensional de uma função, ela poderia ser reconstruída a partir dessas projeções utilizando do que hoje conhecemos como Transformada de Radon [9, 71]. A Transformada de Radon é apresentada mais a frente, na Seção 2.4.

É importante dizer que antes da invenção da Tomografia Computadorizada, outros tipos de tomografia já existiam, como a Tomografia Linear e a Tomografia Axial Transversal [87]. Na Tomografia Linear (Figura 2.1), o tubo de raios X e o filme fotográfico se movem em velocidade constante com direções opostas. Este tubo, que se desloca com ângulo  $\alpha$ , fica sempre apontado para um ponto, chamado de ponto *fulcrum* ou ponto pivô. Ajustando as distâncias entre o tubo de raios X, o ponto e o filme fotográfico nos permite ajustar a área a ser escaneada. Na tomografia Axial Transversal, o filme fotográfico é posicionado de forma horizontal em frente ao objeto a ser escaneado e levemente abaixo do plano focal (Figura 2.2). O processo se dá rotacionando no eixo vertical o paciente e o filme fotográfico em uma mesma velocidade fixa enquanto o tubo de raios X permanece em um ponto estacionário.



**Figura 2.1:** Tomografia Linear



**Figura 2.2:** Tomografia Axial Transversal

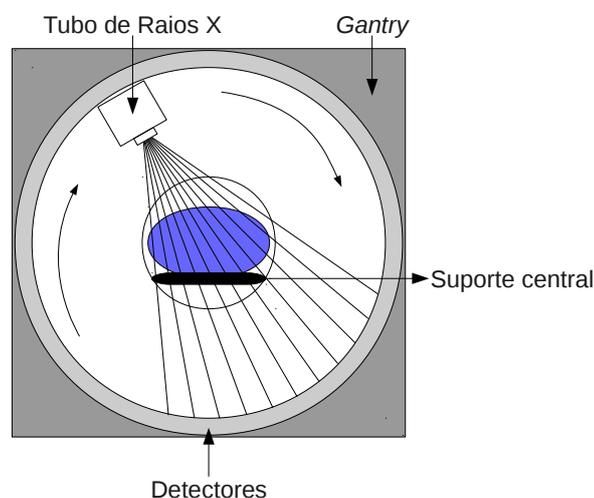
Nos métodos tradicionais, como a Tomografia Linear e a Tomografia Axial Transversal, os objetos atingidos pelos raios X que estão fora do plano focal (onde desejamos realizar a tomografia) ficam visíveis nas imagens resultantes destas tomografias, causando assim um efeito turvo. Já na TC, a reconstrução de uma imagem de uma seção transversal não são influenciadas pelos objetos que não pertencem a este plano focal, por esta razão, as imagens produzidas pela TC são muito mais nítidas e possuem, em geral, uma maior utilidade clínica [44].

Em 1973, o interesse de Allan M. Cormack pela tecnologia de raios X levou-o ao desenvolvimento de um método para calcular a atenuação dos feixes de raios X (raios se movimentam até um determinado ponto e mantêm uma direção constante) através de medidas de transmissão, esse método fundou as bases para o desenvolvimento do primeiro equipamento de tomografia computadorizada (tomógrafo). Cormack postulou, em sua pesquisa, que seria possível mostrar as pequenas diferenças de absorção em estruturas dos tecidos quando atravessados pelos raios X [50].

Uma implementação fundamentada na pesquisa de Cormack e Radon foi apresentada pelo engenheiro inglês Goldfrey N. Hounsfield em 1972, que foi reconhecido como o inventor do primeiro equipamento de TC [65]. Pela primeira vez foi possível, com um paciente situado entre um emissor de raios X e o detector, iniciar a transmissão de fótons de raios X em diferentes ângulos e, a partir dos valores de energia detectadas, calcular os coeficientes de absorção dos diversos tecidos através de um algoritmo matemático. Posteriormente, esses resultados foram apresentados em uma tela em tons de cinza, formando a imagem de uma secção plana do cérebro [18]. Em 1979, o Prêmio Nobel de medicina foi dividido entre Hounsfield e Cormack no reconhecimento à importante invenção que trouxe grandes avanços para o campo de diagnóstico por imagens médicas [15, 50].

A fabricante EMI (*Electric and Musical Industries*), uma empresa multinacional britânica do ramo fonográfico com sede em Londres – Inglaterra, fabricou o primeiro equipamento de TC comercialmente viável em 1972 [4, 15, 87]. Este aparelho foi utilizado para obter imagens do cérebro de um paciente e já permitia a diferenciação de tumores e hematomas. Seu lançamento causou euforia na comunidade médica e a popularidade dos tomógrafos começou a crescer no mundo todo [22]. O primeiro tomógrafo capaz de escanear o corpo inteiro foi desenvolvido em 1974 pelo professor da Universidade de Georgetown Robert Ledley mas a demora no processo para a reconstrução das imagens e sua qualidade passou a ser o principal problema destes novos equipamentos [65].

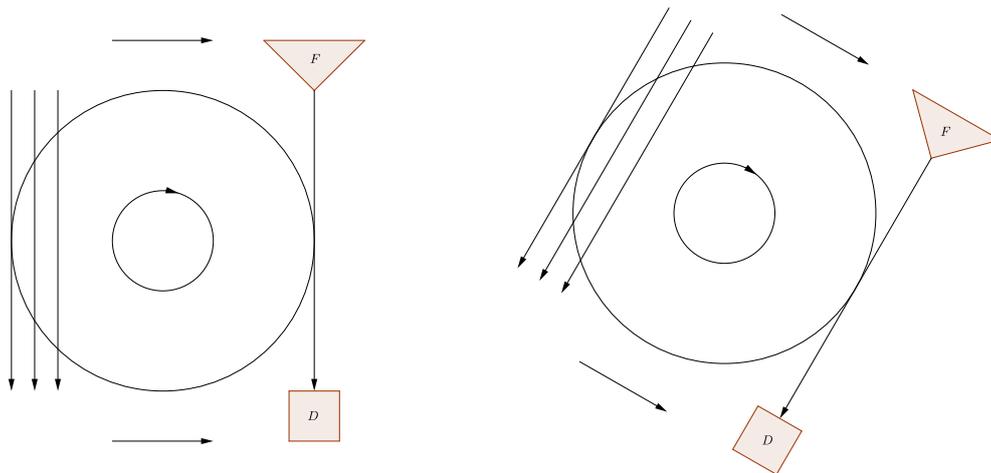
Um tomógrafo, independente da geração, é constituído por um mecanismo que move um tubo de raios X de forma circular e um conjunto de detectores (fixos ou móveis) denominado *gantry*. O processo de escaneamento de um objeto depende de seu modelo e programação, uma vez que está relacionado às movimentações do tubo de raios X sobre o suporte central e de como os feixes são detectados (Figura 2.3). O objeto a ser escaneado então é posto no suporte central ao *gantry* e é deslocado para o seu interior. Quando o *gantry* rotaciona-se ao redor do suporte central e transmite os feixes de raios X, as estruturas internas do objeto atenuam a radiação incidente e a quantidade de fótons do lado oposto são mensuradas através dos detectores. Processando esses valores obtidos através de um computador, obtemos uma imagem de tomografia computadorizada.



**Figura 2.3:** Tomógrafo

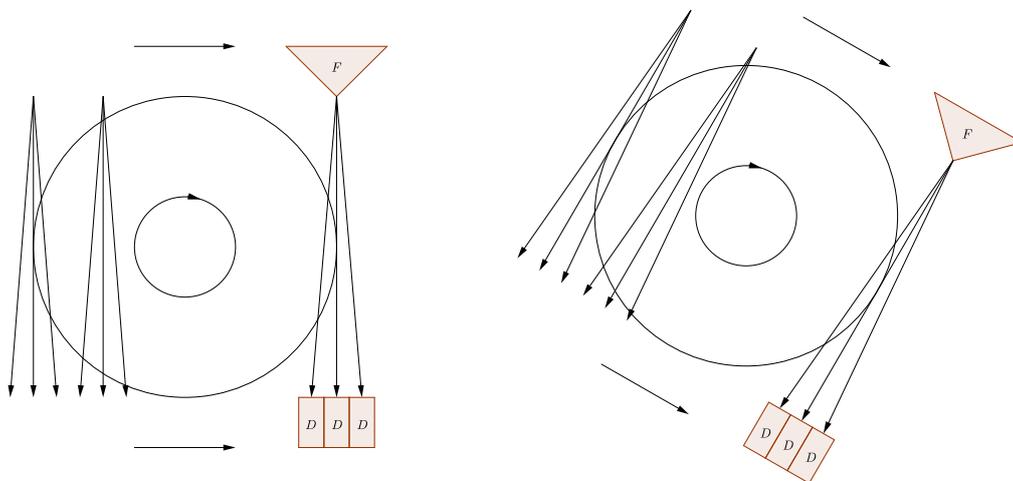
Modificações na estrutura dos tomógrafos vem sendo desenvolvidas desde a sua invenção objetivando melhores tempos de varredura e de reconstrução das imagens. Inicialmente, o tempo necessário para escanear um objeto e reconstruir apenas um único corte era de 8 minutos, porém poucos segundos são necessários para completar a mesma tarefa atualmente utilizando os melhores tomógrafos [56, 65].

Os tomógrafos que utilizam a primeira geração de processo de escaneamento realizam esta tarefa com apenas uma unidade detectora. Esta unidade se move em paralelo com o tubo de raios X e então o processo é repetido para ângulos diferentes, conforme apresentado pela Figura 2.4. Os tomógrafos desta geração demoram em média cerca de 5 minutos apenas para realizar o processo de escaneamento de um único corte [87].



**Figura 2.4:** Primeira geração de escaners

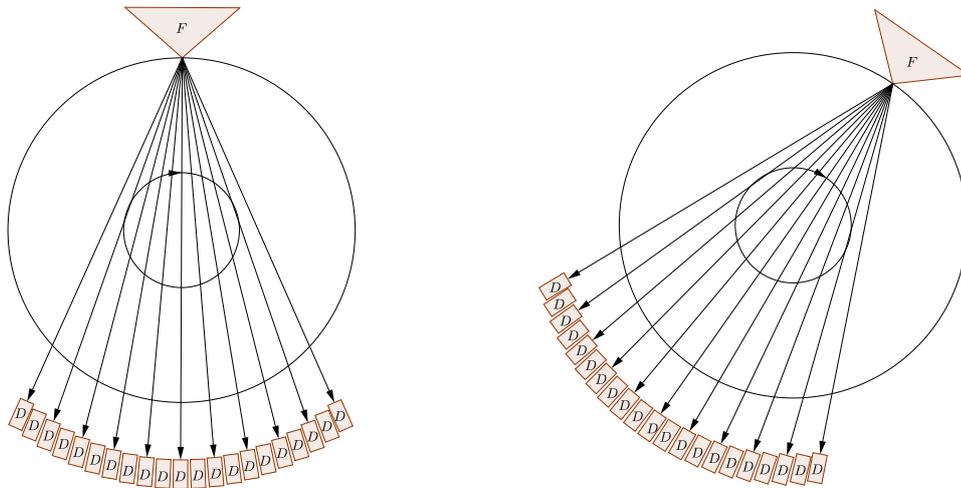
Os tomógrafos que utilizam a segunda geração de processo de escaneamento realizam esta tarefa com várias unidades detectoras. Semelhante a primeira geração, as unidades detectoras movem-se em paralelo com o tubo de raios X, porém vários feixes são detectados simultaneamente (Figura 2.5). Os tomógrafos desta geração demoram em média cerca de 20 segundos para realizar o processo de escaneamento de um único corte [87].



**Figura 2.5:** Segunda geração de escaners

Na terceira geração de tomógrafos o processo de escaneamento é feito em forma de leque e para cada feixe emitido existe um detector cobrindo a área atingida pelo feixe (Figura 2.6). Este

processo de escaneamento demora em média 0.5 segundos para concluir um único corte e é de longe o mais eficiente [87].



**Figura 2.6:** Terceira geração de scanners

A partir de 1998, com o objetivo de aumentar a velocidade de obtenção de cortes tomográficos, começaram a surgir os equipamentos com tecnologia multicortes ou *multislice* e feixes em cone. Esses equipamentos apresentam múltiplos conjuntos de detectores estrategicamente emparelhados, tornando possível a aquisição simultânea de vários cortes tomográficos [61, 87].

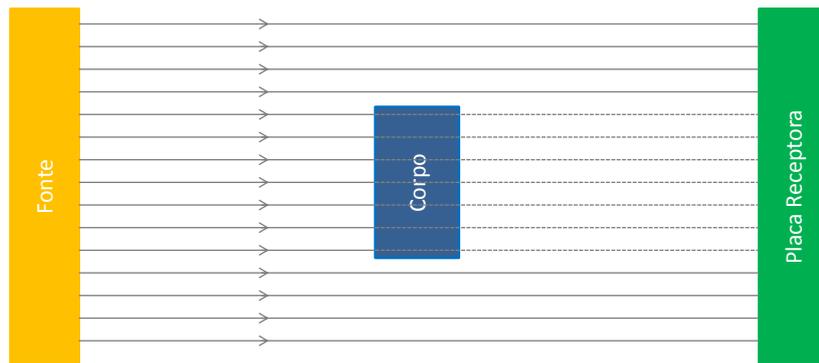
Começou então o que chamamos de guerra dos cortes tomográficos, “*the slice war*”. Nos últimos 10 anos, o número de cortes tomográficos simultâneos que podem ser obtidos através da rotação do *gantry* passaram de 4 para até 320 [55, 65, 72]. Assim, tornou-se possível a obtenção de imagens funcionais (em tempo real) do corpo humano, como batimentos do coração, e surgiram muitas técnicas e equipamentos específicos em TC, por exemplo: TC cardíaca, TC angiográfica, TC de feixe cônico e a fluoroscopia tomográfica [65].

No futuro, a tomografia computadorizada ainda irá continuar sendo a melhor opção quando precisamos visualizar a estrutura do esqueleto, dos pulmões e provavelmente do sistema digestivo [87]. Em outros contextos a TC irá permanecer como única alternativa, quando pacientes não puderem realizar exames como ressonância magnética devido a algum implante metálico.

Inicialmente, o foco das mudanças envolvendo a tecnologia de TC estavam em obter uma melhor velocidade na aquisição das imagens, isto é, reduzir o tempo no processo de escaneamento e de reconstrução das imagens, porém pouco se preocupava com a dosagem de raios X no paciente. Hoje no entanto, existe uma grande preocupação envolvendo a condução de doses menos elevadas de radiação visando garantir a proteção da saúde dos pacientes quando estamos desenvolvendo novas tecnologias [34, 70]. Vários recursos que buscam administrar a dose nos pacientes estão surgindo atualmente, além disso, muitas maneiras de reduzi-la estão sendo desenvolvidas e apresentadas na literatura acadêmica [72].

## 2.2 Princípios da Tomografia Computadorizada por Transmissão

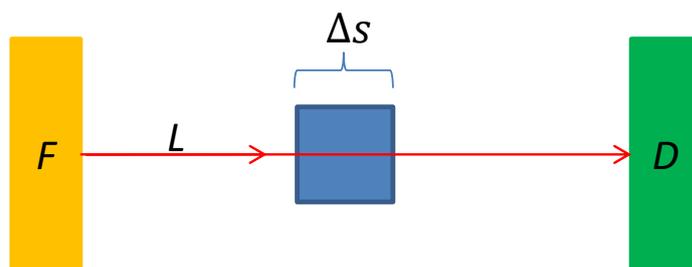
Na tomografia computadorizada por transmissão, o objetivo é obter a imagem de uma secção transversal de um objeto de estudo através do conhecimento de suas integrais de linha ao longo de retas. Mas, antes de descrever o modelo matemático da tomografia propriamente dita analisaremos



**Figura 2.7:** Radiografia Convencional

primeiro os princípios básicos da radiografia convencional. Nela, o paciente fica entre uma fonte de raios  $X$  e uma placa sensível. Ao serem atravessados pelos raios  $X$ , os tecidos dos órgãos mais densos (como o fígado) ou elementos mais pesados (como o cálcio dos ossos), absorvem ou desviam mais radiação que tecidos de órgãos menos densos (como o pulmão). Deste modo, nas regiões da placa receptora onde os tecidos mais densos ou elementos mais pesados forem projetados, a quantidade de radiação será diferente das demais, como podemos ver na Figura 2.7.

Observando a Figura 2.8, suponha, de uma maneira simplificada, que a fonte  $F$  seja pontual, o raio seja a reta  $L$ , o comprimento do caminho em que a reta  $L$  atravessa o corpo seja  $\Delta s$  e no lugar de uma placa sensível seja colocado um contador de fótons  $D$ . Caso haja muitos fótons detectados em  $D$ , significa que poucos foram absorvidos e portando o tecido na trajetória da reta  $L$  era pouco denso e, da mesma forma, poucos fótons detectados em  $D$  indicam tecidos muito densos. Assim, a diferença entre os fótons que foram emitidos em  $F$  e aqueles que foram detectados em  $D$  nos fornece a informação sobre a densidade ao longo do trajeto determinado por cada reta  $L$ . Porém, isto não nos dá informações necessárias para descrever a densidade em um ponto qualquer no interior do corpo.



**Figura 2.8:** Feixe de fótons atravessando um corpo

Diferente da radiografia convencional onde apenas uma projeção do corpo sobre a placa é feita, na tomografia computadorizada temos diversas projeções tomadas em direções diferentes no intuito de reconstruir as características de densidade do tecido em cada ponto ou pequena região no interior de um corpo.

Na tomografia computadorizada é conveniente restringir o corpo a apenas um plano, ou corte (“tomos” significa “corte” em grego). Deste modo, a incógnita passa a ser uma função  $f$  a valores reais, definida em um subconjunto do plano  $\mathbb{R}^2$ , que representa a propriedade do tecido que chamaremos “coeficiente linear de atenuação” (ou simplesmente “atenuação”).

A lei de Beer-Lambert [5, 53] relaciona a atenuação da luz em uma substância em função da quantidade de luz emitida, capacidade de absorção da substância e espessura do meio. Se

considerarmos a reta  $L$  na Figura 2.8 e garantirmos que o coeficiente linear de atenuação  $f(s)$  é constante em todo corpo, a variação relativa da intensidade ( $\frac{\Delta I}{I_0}$ ) é proporcional a atenuação multiplicada pelo tamanho do intervalo, ou seja:

$$\frac{\Delta I}{I_0} = f(s)\Delta s, \quad (2.1)$$

onde  $\Delta I$  são os fótons atenuados,  $I_0$  são os fótons emitidos e  $\Delta s$  é o comprimento do intervalo do corpo.

A partir de (2.1), porém desta vez considerando que o coeficiente linear de atenuação  $f(s)$  varie, subdividimos a região do corpo  $\Delta s$  que foi atravessada pelo feixe  $L$  em partes infinitesimais, chegando a:

$$\begin{aligned} - \int_{I_0}^{I_f} \frac{dI}{I} &= \int_L f(s)ds \\ - \ln|I| \Big|_{I_0}^{I_f} &= \int_L f(s)ds \\ -(\ln I_f - \ln I_0) &= \int_L f(s)ds \\ - \ln \frac{I_f}{I_0} &= \int_L f(s)ds \\ \varphi(L) &= \int_L f(s)ds, \end{aligned} \quad (2.2)$$

onde  $I_0$  é a quantidade inicial de fótons emitidos,  $I_f$  é a quantidade de fótons após ter passado pelo corpo,  $I$  é a intensidade do raio e  $f$  o coeficiente de atenuação local.

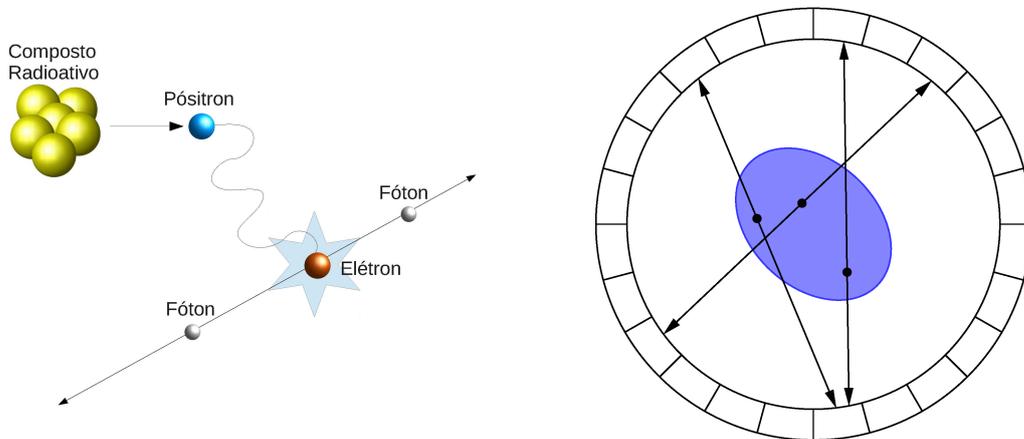
De (2.2) podemos deduzir que o problema teórico se resume a recuperar  $f$  conhecendo o valor de  $\varphi(L)$  para toda reta  $L$  que atravesse  $S$  através da inversa da Transformada de Radon [59, 67] (Seção 2.4). Na prática apenas há necessidade um número finito de feixes que atrevessem a região  $S$  para que possamos reconstruir uma imagem, mas este número pode ser bastante elevado. Nesse caso, espera-se que a reconstrução aproxime-se cada vez mais de  $S$  conforme aumentamos a quantidade de feixes incidentes [44].

## 2.3 Tomografia por Emissão

Nesta seção, apesar do foco do trabalho estar na Tomografia por Transmissão, apresentamos rapidamente os princípios da Tomografia por Emissão. Diferente da Tomografia Computadorizada, nesta modalidade  $f$  é a intensidade de emissão, e a buscamos determinar a partir de leituras de fótons emitidos para o exterior de um objeto de estudo quando aplicamos um composto radioativo em seu interior, conta com as duas mais conhecidas modalidades SPECT e PET (Tomografia por Emissão de Fóton Único e Tomografia por Emissão de Pósitrons) [69].

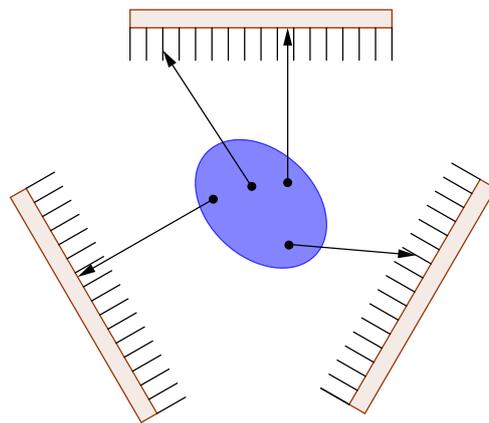
Na Tomografia por Emissão de Pósitrons, utilizamos um composto artificial que sofre um decaimento beta positivo emitindo um pósitron (anti-partícula do elétron com carga positiva). Este pósitron viaja pelo tecido por uma distância bastante curta [66], período no qual sofre uma redução em sua energia cinética e continua perdendo a aceleração até um ponto em que possa interagir com um elétron [87] e, quando isto ocorre, a reação produz dois fótons que viajam aproximadamente em direções opostas sendo esses fótons então detectados pelo aparelho de tomografia PET (Figura 2.3).

Esta técnica depende da detecção simultânea ou coincidente do par de fótons que viajam nestas direções opostas, e os que não são detectados simultaneamente (ou quase simultaneamente) são ignorados. Cada par de fótons, quando detectados, determinam uma reta e os dados do problema são o número de coincidências contadas em cada uma destas retas. O objetivo então passa a ser reconstruir a densidade de emissão, baseado no número de coincidências realizadas em cada ponto ou região específica do objeto de estudo.



**Figura 2.9:** PET - Tomografia por Emissão de Pósitrons

Na Tomografia por Emissão de Fóton Único, o composto utilizado é natural e quando sofre um decaimento, apenas gera um fóton que é detectado se conseguir passar com sucesso através dos colimadores (Figura 2.10). Colimadores são um conjunto de pequenos septos feitos de chumbo que impedem os fótons que não viajam de maneira ortogonal de chegarem nos detectores, deste modo, como muitos fótons são absorvidos por estes colimadores, as imagens reconstruídas através deste tipo de tomografia tendem a ser mais pobres e de pior qualidade quando comparadas com as geradas pelo PET [87].



**Figura 2.10:** SPECT - Tomografia por Emissão de Fóton Único

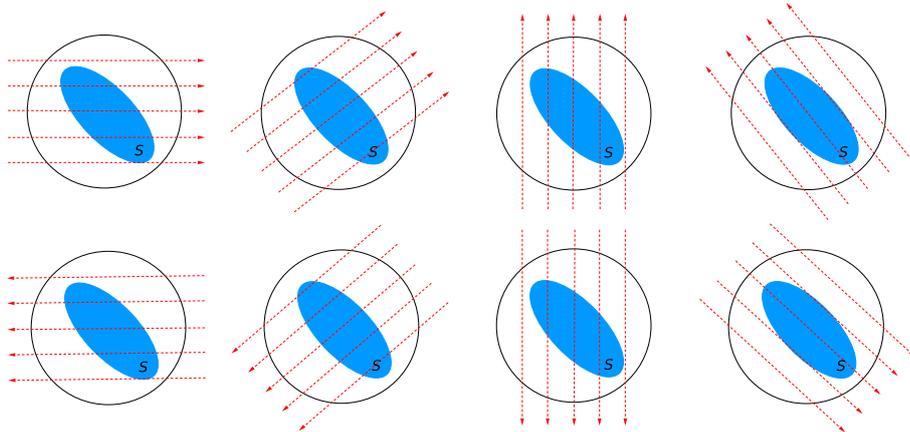
Utilizando o modelo matemático contínuo e não considerando fatores como atenuação, espalhamento, baixa estatística, etc, a tomografia por emissão também é representada pela Transformada de Radon, porém em sua forma atenuada [69]. Este tipo de tomografia atua principalmente no diagnóstico que busca obter informações sobre as atividades metabólicas do organismo de um paciente [69, 74], com isso podemos realizar exames precoces de vários tipos específicos de doenças, enquanto que alterações na anatomia (obtidas com a Tomografia por Transmissão), muitas

vezes, não se manifestam senão em estágios relativamente avançados de uma mesma doença, como é no caso de diversos tipos de câncer [74].

## 2.4 A Transformada de Radon e o Teorema da Fatia de Fourier

A teoria matemática da reconstrução de imagens a partir de projeções tem sua origem no trabalho de Johann Radon [71]. Com o objetivo de descobrir uma função  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  a partir de todas suas integrais de linha em um domínio previamente determinado, Radon propôs em seu artigo original uma formula fechada para o cálculo da solução deste problema inverso.

Anteriormente (Seção 2.2), apresentamos uma maneira de calcular a integral de linha de cada feixe  $L$  que atravessou determinado corte de um corpo tridimensional. Agora, precisamos considerar que vários desses feixes atravessaram o corte em ângulos  $\in [0, 2\pi)$ , conforme exemplificado na Figura 2.11. Note que, não necessariamente, estes feixes precisam estar em paralelo.



**Figura 2.11:** Feixes paralelos variando ângulos  $\in [0, 2\pi)$

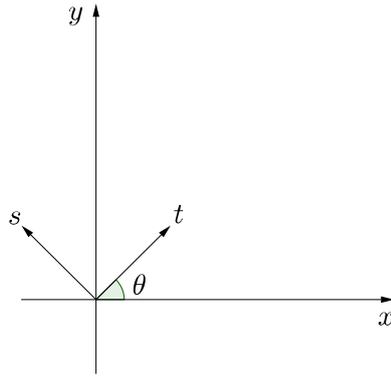
Nosso problema agora passa a ser determinar a taxa de atenuação em cada ponto  $\mathbf{x} = (x, y)$  pertencente a este corte à partir da soma das taxas de atenuação (calculadas pela Equação (2.2)) de cada feixe  $t$  (raio que parte de ponto e se move até outro com direção e sentido constante) com inclinação  $\theta$  percorrido. Olhando para o problema de forma contínua, precisamos recuperar  $f$ , através da Transformada de Radon definida pela equação:

$$\mathcal{R}f(\theta, t) = \int_{\mathbb{R}} f(t \cos \theta - s \sin \theta, t \sin \theta + s \cos \theta) ds, \quad (2.3)$$

onde também podemos denotar  $\mathcal{R}_\theta f(t) := \mathcal{R}f(\theta, t)$  ou, quando não houver ambiguidade com relação a  $f$ ,  $p_\theta(t) := \mathcal{R}_\theta f(t)$ . A função  $p_\theta = \mathcal{R}_\theta f$  é conhecida como projeção de  $f$  [63].

A Equação (2.3) é obtida através da matriz  $A_\theta$  (Equação (2.4)), uma matriz de rotação anti-horária para ajustar as coordenadas polares perpendiculares ao feixe  $t$  (Figura 2.12).

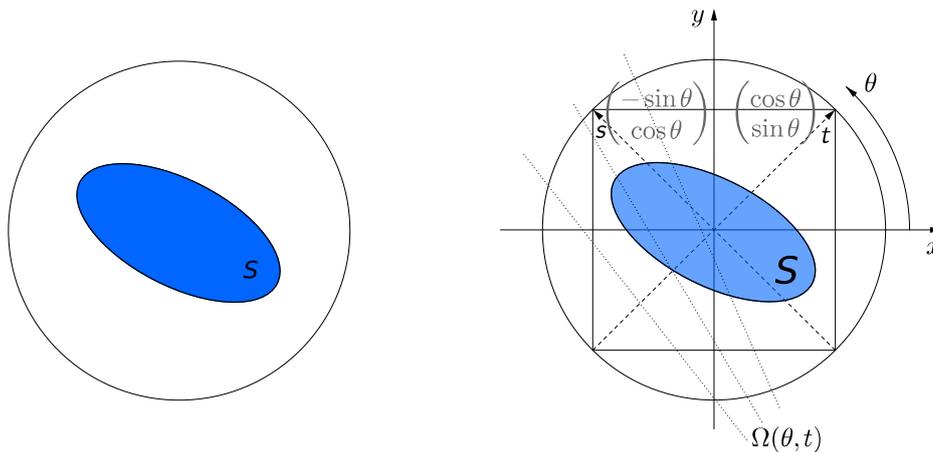
$$A_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (2.4)$$



**Figura 2.12:** Vetor  $s$  perpendicular ao feixe  $t$

$$\begin{aligned} \mathcal{R}f(\theta, t) &= \int_{\mathbb{R}} f\left(A_{\theta} \begin{bmatrix} t \\ s \end{bmatrix}\right) ds \\ &= \int_{\mathbb{R}} f(t \cos \theta - s \sin \theta, t \sin \theta + s \cos \theta) ds \end{aligned}$$

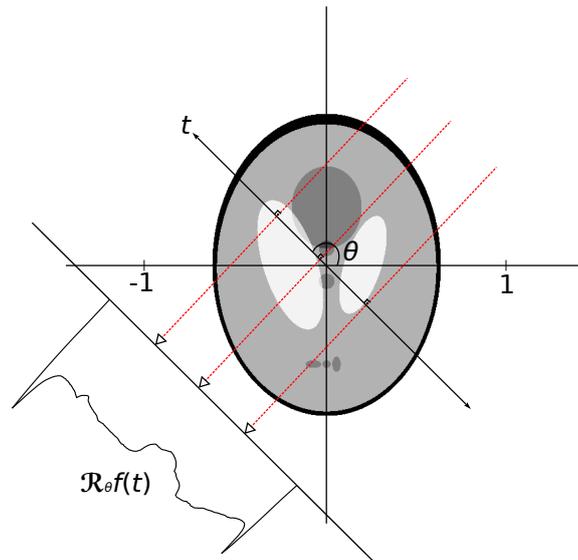
A Figura 2.13 ilustra a geometria do problema. Ao fixar um sistema de coordenadas cartesianas, e denotar  $\mathbf{x} = (x, y)$  um ponto deste sistema, então são conhecidas as integrais da função  $f$  ao longo das retas  $\Omega(\theta, t) = \{\mathbf{x} \in \mathbb{R}^2 : \mathbf{x}^t \cdot \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} = t\}$ . Variando o ângulo  $\theta \in [0, 2\pi)$  e o raio  $t \in [-1, 1]$ , a família de retas  $(\theta, t)$  cobre todo o interior do disco unitário e uma coleção de valores é obtida. Não há perda de generalidade em supor que  $f$  está definida em um disco unitário, caso contrário, basta efetuar uma mudança de variáveis. Isto é o mesmo que supor que o objeto tridimensional está contido dentro de um cilindro, que é o mesmo também a assumir que  $f$  tem suporte compacto.



**Figura 2.13:** Geometria do problema - Transformada de Radon

Observando a Figura 2.14 onde representamos a função  $f$  como uma imagem em escalas de tons de cinza, se compararmos a definição com os feixes e ângulos utilizados na imagem fica fácil de perceber que  $R_{\theta}f(t)$  é um gráfico em função de  $t$ , onde o ângulo  $\theta$  determina a inclinação do eixo  $t$  com relação à linha horizontal e a integral da função é efetuada sobre as retas perpendiculares (retas vermelhas tracejadas) a esse eixo. Outra propriedade que podemos observar é que não

necessariamente precisamos variar  $\theta \in [0, 2\pi)$ , pois estaríamos gerando informações redundantes, bastando apenas variar  $\theta \in [0, \pi)$ .



**Figura 2.14:** Projeção  $\mathcal{R}_\theta f(t)$  com ângulo  $\theta = 135^\circ$  no *phantom* de Shepp-Logan

Uma maneira para se obter a inversa da transformada de Radon  $\mathcal{R}^{-1}$ , a fim de reconstruir a função  $f$ , é através do Teorema da Fatia de Fourier, também conhecido como Teorema do Corte Central. Este teorema relaciona as projeções da Transformada de Radon com a Transformada de Fourier.

Na Transformada de Fourier realizamos a decomposição de uma função  $f$  em funções senoidais e cossenoidais de diferentes frequências e amplitudes específicas e, para que seja possível o cálculo desta transformada, é necessário que a função  $f$  seja integrável [11]. Tendo que  $f$  seja uma função definida em  $\mathbb{R}^n$ , dizemos que  $f$  é integrável [73] caso:

$$\|f(x)\|_1 = \int_{\mathbb{R}^n} |f(x)| dx < \infty. \quad (2.5)$$

A Transformada de Fourier de uma função  $f \in \mathbb{R}$  que satisfaz (2.5), é  $\hat{f}$  definida em  $\mathbb{C}$  pela integral:

$$\hat{f}(w) = \int_{\mathbb{R}} f(x) e^{-iwx} dx \quad (2.6)$$

onde  $i = \sqrt{-1}$ ,  $x$  e  $w \in \mathbb{R}$ .

Supondo que  $f \in \mathbb{R}$  é uma função que satisfaz (2.5) e que  $\hat{f}$  foi obtida através de  $f$ . Então, para calcularmos a Inversa da Transformada de Fourier em  $\mathbb{R}$ , fazemos:

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(w) e^{ixw} dw. \quad (2.7)$$

Uma imagem, que pode ser expressada como uma função de duas dimensões  $f(x, y)$ , pode ser analisada no domínio de Fourier, uma vez que satisfaz (2.5). A grande utilidade da Transformada de Fourier para a reconstrução de imagens vem do fato que  $f$  pode ser obtida de  $\hat{f}$  utilizando a Inversa da Transformada de Fourier.

As Equações (2.6) e (2.7) podem ser estendidas ao  $\mathbb{R}^n$ . Caso  $f$  satisfaça (2.5) em  $\mathbb{R}^n$ , a Transformada de Fourier de  $f$  e sua inversa são respectivamente:

$$\hat{f}(\omega) = \int_{\mathbb{R}^n} f(x) e^{-i\langle \omega, x \rangle} dx \quad (2.8)$$

e

$$f(x) = \frac{1}{(2\pi)^n} \int_{\mathbb{R}^n} \hat{f}(\omega) e^{i\langle x, \omega \rangle} d\omega \quad (2.9)$$

onde  $\langle \cdot, \cdot \rangle$  é o produto interno em  $\mathbb{R}^n$ ,  $x$  e  $\omega \in \mathbb{R}^n$ .

O Teorema da Fatia de Fourier afirma que uma linha com inclinação  $\theta$  passando pela origem da Transformada Bidimensional de Fourier de uma função  $f$  é o mesmo que a Transformada de Fourier Unidimensional da projeção  $p_\theta$ .

**Teorema 2.4.1. Teorema da Fatia de Fourier.** Para qualquer número real  $r$  e o vetor unitário  $\omega_\theta = (\cos \theta, \sin \theta)$ ,

$$\hat{f}(r\omega_\theta) = \hat{p}_\theta(r). \quad (2.10)$$

*Demonstração.*

Seja a Transformada de Radon de  $f$ :

$$\mathcal{R}f(\theta, t) = \int_{\mathbb{R}} f(t \cos \theta - s \sin \theta, t \sin \theta + s \cos \theta) ds, \quad (2.11)$$

e a Transformada de Fourier  $\hat{f}(\mathcal{R}f(\theta, t))$ :

$$\int_{\mathbb{R}} \int_{\mathbb{R}} f(t \cos \theta - s \sin \theta, t \sin \theta + s \cos \theta) ds e^{-i\langle r\omega_\theta, (t, s) \rangle} dt, \quad (2.12)$$

onde:

$$\begin{aligned} \langle r\omega_\theta, (t, s) \rangle &= [r \cos \theta \quad r \sin \theta] \cdot \begin{bmatrix} t \cos \theta - s \sin \theta \\ t \sin \theta + s \cos \theta \end{bmatrix} \\ &= r \cos \theta t \cos \theta - r \cos \theta s \sin \theta + r \sin \theta t \sin \theta + r \sin \theta s \cos \theta \\ &= rt \cos^2 \theta + rt \sin^2 \theta \\ &= rt (\cos^2 \theta + \sin^2 \theta) \\ &= rt. \end{aligned} \quad (2.13)$$

Podemos escrever (2.12) como:

$$\hat{f}(\mathcal{R}f(\theta, t)) = \int_{\mathbb{R}} \int_{\mathbb{R}} f(t \cos \theta - s \sin \theta, t \sin \theta + s \cos \theta) e^{-irt} ds dt. \quad (2.14)$$

Seja a Transformada de Fourier Bidimensional de uma função  $f \in \mathbb{R}^2$ :

$$\hat{f}(r \cos \theta, r \sin \theta) = \hat{f}(r\omega_\theta) = \int_{\mathbb{R}^2} f(x) e^{-i\langle r\omega_\theta, x \rangle} dx \quad (2.15)$$

Efetuamos uma rotação no sistema de coordenadas através através da matriz  $A_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ , considerando  $x = A_\theta \mathbf{t}$  e  $\mathbf{t} = \begin{bmatrix} t \\ s \end{bmatrix}$ :

$$\hat{f}(r \cos \theta, r \sin \theta) = \hat{f}(r\omega_\theta) = \int_{\mathbb{R}^2} f(t \cos \theta - s \sin \theta, t \sin \theta + s \cos \theta) e^{-i\langle r\omega_\theta, A_\theta t \rangle} dt. \quad (2.16)$$

Calculando  $\langle r\omega_\theta, A_\theta t \rangle$  da mesma forma que na Equação (2.13), obtemos:

$$\hat{f}(r\omega_\theta) = \int_{\mathbb{R}^2} f(t \cos \theta - s \sin \theta, t \sin \theta + s \cos \theta) e^{-irt} dt. \quad (2.17)$$

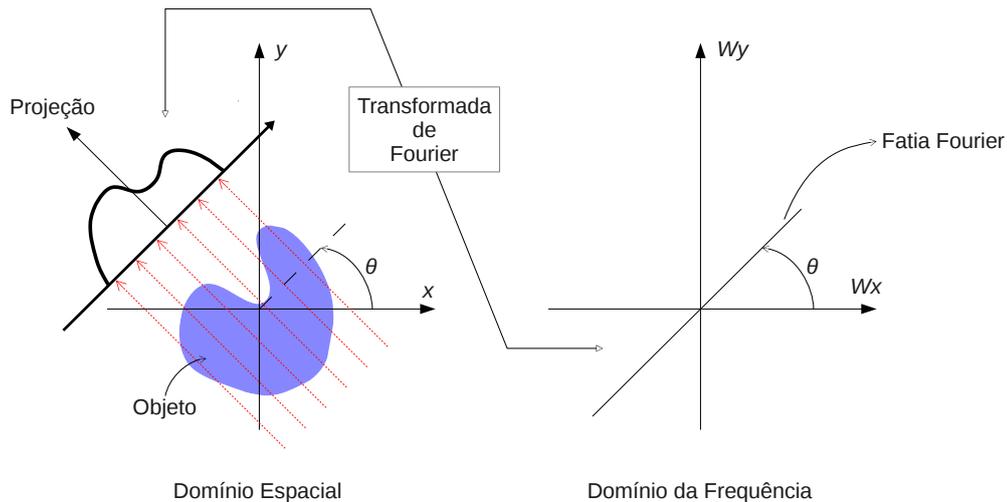
$$\therefore \int_{\mathbb{R}} \mathcal{R}f(\theta, t) e^{irt} dt = \hat{f}(r\omega_\theta). \quad \square$$

A Figura 2.15 mostra um exemplo onde primeiro calculamos a projeção referente ao ângulo  $\theta = 135^\circ$  (a mesma apresentada na Figura 2.14) e, logo após, calculamos o valor absoluto da Transformada de Fourier desta projeção.



**Figura 2.15:** Valor da Transformada de Radon aplicada ao *phantom* de Shepp-Logan com o ângulo  $\theta = 135^\circ$  e o valor absoluto da Transformada de Fourier desta projeção

A Figura 2.16 mostra que a Transformada de Fourier de uma projeção da Transformada de Radon com um ângulo  $\theta$  é uma fatia (“slice”) de mesmo ângulo no domínio das frequências da Transformada de Fourier Bidimensional,  $\hat{f}(w_x, w_y)$ , aplicada a função  $f(x, y)$ .



**Figura 2.16:** Fatia Fourier - Transformada de Fourier aplicada em uma projeção da Transformada de Radon

Utilizando o Teorema da Fatia Fourier (2.4.1), caso encontremos o valor de  $\hat{f}(\mathcal{R}f(\theta, t))$  para qualquer  $(\theta, t)$ , temos o valor da Transformada de Fourier Bidimensional  $\hat{f}(r\omega_\theta)$  em qualquer ponto

e assim, através do cálculo de  $\hat{f}^{-1}(\mathcal{R}f(\theta, t))$  utilizando (2.9) é possível recuperar  $f$ . Como na prática é impossível temos todas as projeções, consideramos apenas projeções finitas da Transformada de Radon e utilizamos o Método da Transformada Discreta de Fourier (DFT - *Discrete Fourier Transform*) ou o Método da Transformada Rápida de Fourier (FFT - *Fast Fourier transform*) para o seu cálculo.

Os métodos que obtêm amostras necessárias de uma função  $f$  no espaço da frequência por interpolação das amostras radiais e depois reconstróem esta mesma função através da Transformada Inversa de Fourier são conhecidos como Métodos de Fourier. A reconstrução de uma imagem através do uso desses métodos tendem a possuir uma qualidade inferior, quando comparamos, por exemplo, com métodos iterativos e por isso não possuem uma boa reputação [63].

A existência de uma formula fechada, como a descrita pelo Teorema (2.4.1) e Equação (2.9), está longe de resolver todos os problemas da Tomografia Computadorizada, pois não conhecemos o valor de  $\mathcal{R}f(\theta, t)$  para todo  $(\theta, t)$ , a Inversa da Transformada de Radon somente pode ser calculada de forma aproximada e ainda pelo fato do problema (2.2) ser mal posto ou mal condicionado, isto é, pequenas perturbação nos dados produzem grandes variações na solução [67]. Devido a estes e outros fatos dicutidos na próxima Seção 2.5 e mais adiante no Capítulo 4, vários métodos iterativos se mostram mais capacitados na execução desta tarefa, como é o caso do Método de Kaczmarz e dos Gradientes Conjugados.

## 2.5 Reconstrução de Imagens Tomográficas em um Modelo Discreto

Considerando que o corte do objeto que queremos realizar a tomografia esteja contido em uma área quadrada subdividida várias vezes em pequenos quadrados de tamanho constante, chamados *pixels* (*picture elements*), e aceitarmos que  $f$  seja constante em cada um deles. Podemos numerar de 1 a  $n$  os *pixels* desta área e chamar de  $x_j$  o valor de  $f$  no pixel  $j$ , transformando nossa incógnita, que antes era uma função  $f$  em  $\mathbb{R}^2$ , em um vetor  $x = (x_1, \dots, x_n)$  (Figura 2.17).

Tendo que o número de feixes que atravessam o objeto é finito, como acontece na realidade, e que  $L_1, \dots, L_m$  são estes raios cujas atenuações medimos através da equação (2.2), podemos chamar  $\varphi(L_i)$  (atenuação total do  $i$ -ésimo raio) de  $b_i$ . Desta maneira, esta discretização do problema irá possuir uma aproximação aceitável caso aumentemos o número de *pixels*, isto é, conforme diminuimos o tamanho de cada um. Na Seção 3.1 e 3.2 do Capítulo 3 explicamos como isto pode ser feito.

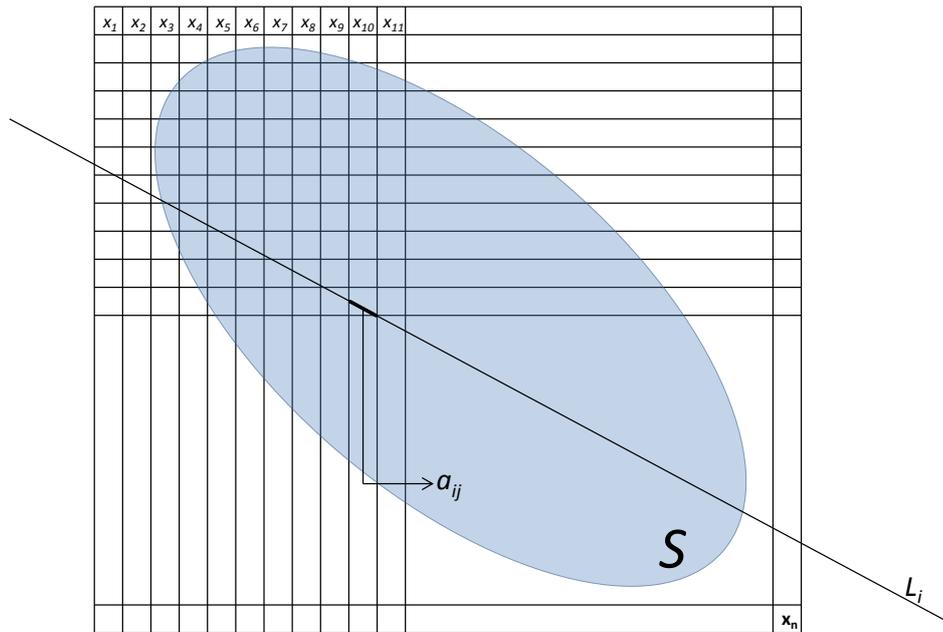
Deste modo, a integral em (2.2) se torna a soma das integrais sobre cada pixel atravessado pelo raio  $L$ . É imediato que esta equação possa ser escrita como:

$$\sum_{j=1}^n a_{ij}x_j = b_i \quad (1 \leq i \leq m), \quad (2.18)$$

onde  $a_{ij}$  é o comprimento da interseção do  $j$ -ésimo *pixel* com o feixe  $i$  ( $a_{ij}$  é zero se o raio  $i$  não passa pelo *pixel*  $j$ ). Na Seção 3.3 do Capítulo 3, explicamos detalhadamente como isso ocorre. Assim, em notação matricial temos:

$$Ax = b, \quad (2.19)$$

onde  $A = \{a_{ij}\} \in \mathbb{R}^{m \times n}$ ,  $b = \{b_i\} \in \mathbb{R}^m$ ,  $x = \{x_j\} \in \mathbb{R}^n$ . Se associarmos valores correspondentes a escala de tons de cinza a  $x_j$ , obteremos a imagem reconstruída ao acharmos a solução de (2.19).



**Figura 2.17:** Matriz de *pixels*

Uma característica marcante deste problema discretizado é possuir a matriz  $A$  esparsa e sobre-determinada (número de linhas muito maior que o número de colunas) resultante das seguintes afirmações. Primeiro, dado que o comprimento da interseção do  $j$ -ésimo *pixel* com o  $i$ -ésimo feixe ocorre apenas nos *pixels* por onde cada feixe passa e que nos demais índices da matriz  $A$  teremos elementos nulos. Segundo, como cada feixe representa uma equação do sistema e que a quantidade de feixes é muito maior do que a de *pixels*. Assim, a reconstrução de uma imagem em Tomografia Computadorizada utilizando este modelo discretizado requer encontrar soluções de grandes sistemas de equações lineares esparsos e sobre-determinados.

Colocando o problema na forma de  $Ax = b$  (Equação (2.19)), podemos utilizar variados métodos para calcular a sua resolução, como por exemplo o Método de Eliminação Gaussiana. Porém quando consideramos este problema na reconstrução de imagens ele se torna mais complexo, devido as características da matriz  $A$ . Para que possamos obter uma solução aceitavelmente aproximada e assim ter imagens nítidas com uma reconstrução, o número  $n$  de colunas de  $A$  deve ser grande, em média na ordem de  $10^4$ , o número de raios  $m$  costuma ser na ordem de  $10^5$ , assim, normante a matriz  $A$  possui  $10^9$  elementos [46].

Como explicamos acima, uma vez que a maior parte dos elementos de  $A$  é nula implica em uma baixa densidade (proporção de elementos nulos de uma matriz) e os métodos diretos para solução de sistema lineares alteram os elementos da matriz  $A$ . Caso tenhamos armazenados a matriz  $A$  em uma estrutura própria para matrizes esparsas, isto pode causar problemas. Uma das vantagens dos métodos iterativos é que eles não alteram os coeficientes de  $A$ .

Em todos os métodos iterativos para reconstrução de imagens nós precisamos decidir como inicializamos o processo, isto é, como escolhemos o valor de  $x^0$  para a primeira iteração. Kaufman [51] argumenta que um vetor com valores uniformes (todos elementos com o mesmo valor) é o ideal. Entretanto, como mencionado por Herman [43, 44], também é vantajoso utilizar um ponto inicial provido por um outro método alternativo de reconstrução de imagens. Neste trabalho seguimos a ideia de Kaufman e sempre inicializamos seu valor com elementos uniformes mas variando seus valores. Os respectivos valores usados são:  $x^0 = 0.\psi$ , em que  $\psi = [1, \dots, 1]' \in \mathbb{R}^n$  (denotado como  $x^0 = 0$ ),  $x^0 = 1.\psi$  (denotado como  $x^0 = 1$ ),  $x^0 = 10.\psi$  (denotado como  $x^0 = 10$ ) e  $x^0 = 100.\psi$  (denotado como  $x^0 = 100$ ), tais valores foram escolhidos para avaliar a influência

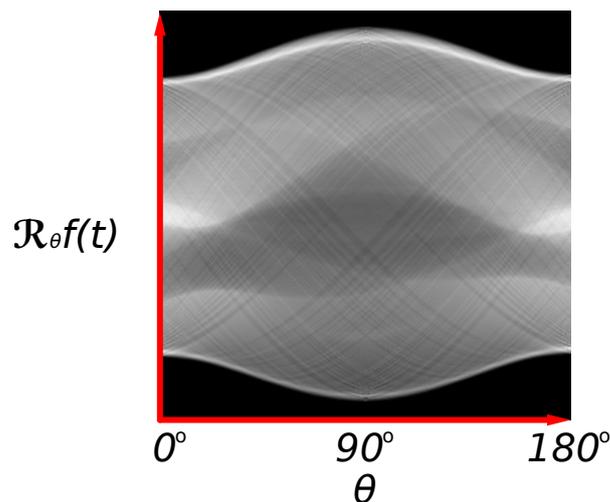
do chute inicial. O motivo de escolhermos tal abordagem, em detrimento aos mencionados por Herman, foi devido ao fato de estamos interessados somente na performance de um método específico.

O *phantom* de Shepp-Logan, Figura 2.18, é frequentemente usado em simulações em várias aplicações de imagens médicas. Com ele pode ser testada a capacidade de um algoritmo reconstruir secções transversais da cabeça humana na tomografia de raios-X [49]. Ele é formado por dez elipses com os parâmetros listados no trabalho original [79].



**Figura 2.18:** *phantom* de Shepp-Logan

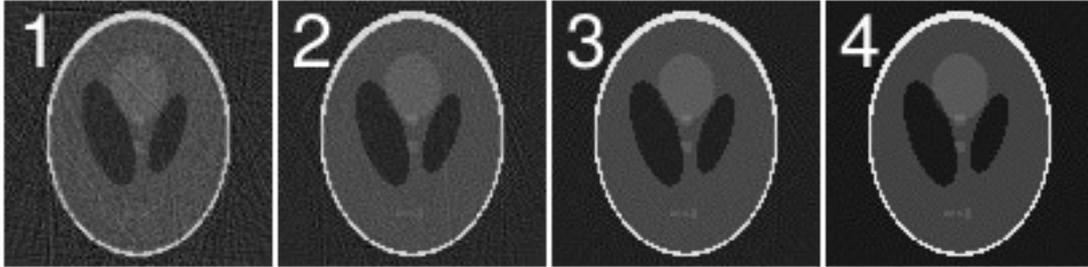
Utilizando a representação matricial da Figura 2.18 e o algoritmo descrito no Capítulo 3 pelo Algoritmo de Siddon, simulamos vários feixes paralelos com ângulos variando entre  $[0, \pi)$  atravessando a imagem com o intuito de reconstruí-la posteriormente utilizando o Método de Kaczmarz Aleatório de Strohmer e Vershynin [85]. Aplicando a Transformada de Radon nesta imagem, obtemos o seu sinograma (Figura 2.19), que representa o soma do caminho radiológico de cada feixe em um determinado ângulo. Também obtemos a matriz  $A$  esparsa e sobredeterminada, que representa o comprimento do feixe que atravessou cada *pixel*. Caso este sistema seja solucionado, estaremos reconstruindo a imagem original, neste caso, a Figura 2.18.



**Figura 2.19:** Sinogram do *phantom* de Shepp-Logan

Utilizando o Método de Kaczmarz Aleatório com  $\lambda = 1$  para solucionar o sistema linear obtido através da Transformada de Radon na Figura 2.18 com dimensões  $100 \times 100$  *pixels*, reconstruímos as imagens ilustradas pela Figura 2.20 utilizando um sinograma de  $150 \times 150$  *pixels* sem se preocupar em otimizar a quantidade de ângulos  $\theta \in [0, \pi)$  e a quantidade de feixes  $t$  por ângulo  $\theta$ .

A matriz usada para realizar as seguintes reconstruções é uma matriz  $A \in \mathbb{R}^{m \times n}$ , possui dimensões  $m = 20084$ ,  $n = 10000$  (não foram armazenadas linhas nulas), norma de Frobenius  $\|A\|_F = 1.22 \times 10^3$ , menor valor singular  $\underline{\sigma} = 0.15$ , maior valor singular  $\bar{\sigma} = 1.23 \times 10^2$ , número de condicionamento  $\kappa(A) = \bar{\sigma}/\underline{\sigma} = 8.43 \times 10^2$  e valor inicial de  $x^0 = 0$ .



**Figura 2.20:** Reconstrução do *phantom* de Shepp-Logan. A primeira imagem foi reconstruída utilizando  $p = 20.084$  projeções, a segunda utilizando  $2p$  projeções, a terceira usando  $3p$  projeções e a quarta utilizando  $4p$  projeções.

O método de Kaczmarz vem sendo empregado com sucesso na solução de problemas tomográficos, o que foi a nossa motivação inicial para o seu estudo. Porém, existem métodos analíticos que já realizam esta tarefa muito bem e alguma motivação extra é necessária para justificar sua pesquisa. Esta motivação se encontra no fato de que, utilizando este método, a matriz  $A$  pode ser armazenada de forma eficiente na memória, podemos facilmente acomodar novas geometrias de aquisição, modelar o efeito do ruído estatístico nas medidas e estudar sua paralelização efetiva.

Os maiores ganhos, talvez, estejam em seu gerenciamento de memória e computação paralela. Quanto maior a eficiência no gerenciamento de memória, podemos armazenar mais cortes simultâneos e, através do paralelismo, aumentar o número de cortes por unidade de tempo. Outro motivo, como dito anteriormente no Capítulo 1 Seção 1.1, é pelo fato de que o método serve como ferramenta para obtenção de algoritmos mais sofisticados de otimização, os quais permitem o uso de modelos melhores do que o simples sistema linear  $Ax = b$  [19, 23, 40, 68, 82], de forma que quaisquer melhorias obtidas em seu desempenho possuem grande impacto na literatura.



## Simulação do Traçado dos Feixes

Nos estudos de Tomografia Computadorizada por Transmissão (TCT - *Transmission Computed Tomography*), existe a necessidade da simulação do traçado dos feixes de raios X através do objeto a ser reconstruído, de forma a podermos utilizar/testar um algoritmo de reconstrução. Em uma simulação é necessário computar, por exemplo, o valor da Transformada de Radon de imagens discretas, a qual pode ser escrita, para um par  $(\theta, t)$  fixo, como:

$$b = \sum_{i,j} l(i,j)\mu(i,j), \quad (3.1)$$

onde  $\mu(i, j)$  é a densidade ou coeficiente de atenuação e  $l(i, j)$  é o comprimento da interseção do caminho percorrido pelo feixe com o *pixel*  $(i, j)$ .

Um algoritmo que podemos utilizar para realizar tais projeções é o Algoritmo de Siddon, que é descrito em [81].

### 3.1 O Algoritmo de Siddon

A ideia do Algoritmo de Siddon se baseia em considerarmos as interseções dos feixes nos *pixels* através da interseção dos feixes com as retas paralelas que definem as fronteiras destes *pixels*. Em uma matriz de *pixels*  $Z \in \mathbb{R}^{m \times n}$ , onde os elementos  $\in Z$  são denotados como  $z_{ij}$ , os conjuntos de planos paralelos igualmente espaçados podem ser escritos como:

$$\begin{aligned} x &= i, 0 \leq i \leq m \\ y &= j, 0 \leq j \leq n \end{aligned} \quad (3.2)$$

onde, por simplicidade, o espaço entre as retas paralelas adjacentes é assumido como unitário. A seguinte convenção é feita de modo que o *pixel* seja descrito por:

$$\{(x, y) \mid i \leq x < i + 1, j \leq y < j + 1\}.$$

Um raio de um ponto  $P_{ini} = (x_1, y_1)$  ao ponto  $P_{fim} = (x_2, y_2)$  é representado de forma paramétrica por:

$$\begin{aligned} x(\alpha) &= x_1 + \alpha(x_2 - x_1) \\ y(\alpha) &= y_1 + \alpha(y_2 - y_1) \end{aligned} \quad (3.3)$$

onde o parâmetro  $\alpha = 0$  em  $P_{ini}$  e  $\alpha = 1$  em  $P_{fim}$ . O algoritmo computa os valores de  $\alpha$  em cada interseção com os planos paralelos e os ordena de forma ascendente em um novo conjunto  $\{\alpha(0), \alpha(1), \dots, \alpha(n)\}$ . Deste modo, o comprimento da interseção  $l$  em um *pixel*  $(i, j)$  é feita por:

$$l(k) = [\alpha(k) - \alpha(k - 1)] L \quad \forall k = 1, \dots, n. \quad (3.4)$$

onde  $L$  é o comprimento total do feixe entre  $P_{ini}$  e  $P_{fim}$ ,

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

A identificação das coordenada do *pixel* cujo valor da interseção foi calculada por um valor  $k$  é feita por:

$$\begin{aligned} i(k) &= \lceil x_1 + \alpha_{mid}(x_2 - x_1) \rceil \\ j(k) &= \lceil y_1 + \alpha_{mid}(y_2 - y_1) \rceil \end{aligned} \quad (3.5)$$

onde  $\alpha_{mid}$  é calculado por:

$$\alpha_{mid} = \frac{1}{2} [\alpha(k - 1) + \alpha(k)]$$

Deste modo, a simulação do caminho radiológico efetuado pelo feixe determinado pelos pontos  $(x_1, y_1)$  e  $(x_2, y_2)$  na matriz  $Z$  é dado pelo Algoritmo de Siddon.

---

### - Algoritmo de Siddon

---

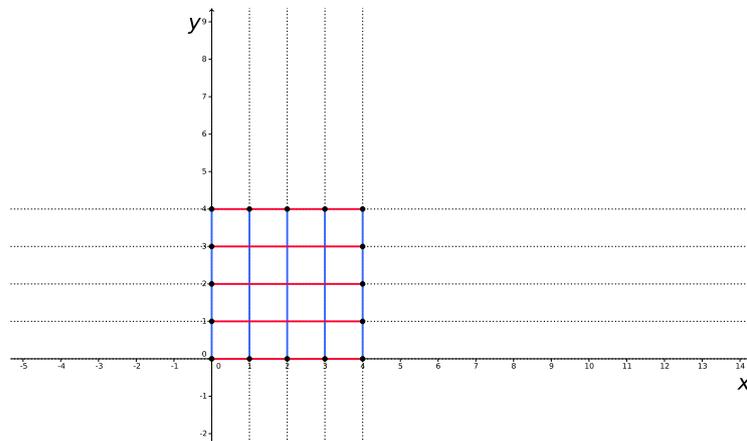
- 1: **function** SIDDON\_RAYTRACER( $Z, x_1, y_1, x_2, y_2$ )
  - 2:     Construir as linhas paralelas igualmente espaçadas  $x = 0 \leq i \leq m$  e  $y = 0 \leq j \leq n$
  - 3:     Calcular todas as interseções entre o segmento de reta definido pelos pontos  $(x_1, y_1)$  e  $(x_2, y_2)$  com as linhas paralelas no intervalo  $x = 0 \leq i \leq m$  e  $y = 0 \leq j \leq n$
  - 4:     Tendo que  $\alpha(1)$  seja 0 em  $(x_1, y_1)$  e para os valores subsequentes  $\alpha(2) \dots \alpha(p)$  seja a distância entre o ponto  $(x_1, y_1)$  e a interseção com as linhas igualmente espaçadas
  - 5:      $sum = 0$
  - 6:     **for**  $k \leftarrow 2$  **to**  $p$  **do**
  - 7:          $\alpha_{mid} = \frac{1}{2} (\alpha(k - 1) + \alpha(k))$
  - 8:          $i = \lceil x_1 + \alpha_{mid} * (x_2 - x_1) \rceil$
  - 9:          $j = \lceil y_1 + \alpha_{mid} * (y_2 - y_1) \rceil$
  - 10:          $sum = sum + z_{ij} * (\alpha(k) - \alpha(k - 1)) * L$
  - 11:     **return**  $sum$
-

## 3.2 Exemplo do Traçado do Feixe

Dada a matriz  $Z \in \mathbb{R}^{m \times n}$ , onde os elementos  $\in Z$  são denotados como  $z_{ij}$ ,  $m = 4$  e  $n = 4$  e um feixe iniciando no  $P_1 = (-0.75, -1)$  e terminando no ponto  $P_2 = (4.5, 6)$ , iremos exemplificar como este algoritmo de simulação do traçado do feixe funciona.

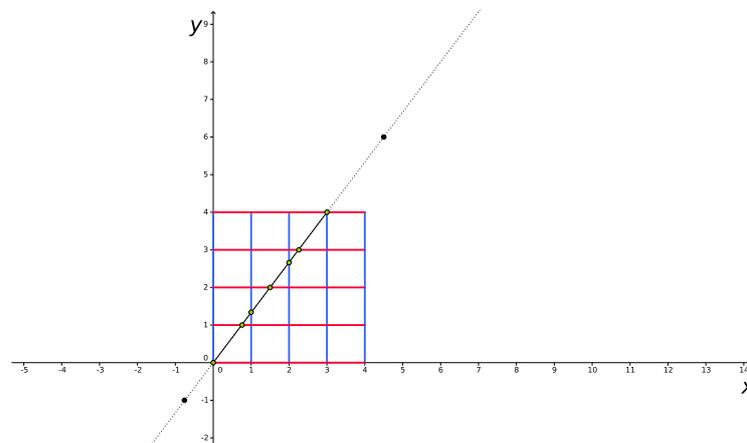
$$Z = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$$

O primeiro passo do algoritmo é construir retas igualmente espaçadas e paralelas ao eixo  $x$  e  $y$  no plano cartesiano (Figura 3.1). De forma que  $x = i, 0 \leq i \leq m$  são as retas paralelas ao eixo  $y$  e  $y = j, 0 \leq j \leq n$  são as retas paralelas ao eixo  $x$ .



**Figura 3.1:** Calculando as retas igualmente espaçadas traçadas no plano satisfazendo  $x = 0 \leq i \leq m$  e  $y = 0 \leq j \leq n$

Tendo as retas paralelas traçadas no plano, o próximo passo é calcular suas interseções com o feixe (Figura 3.2).



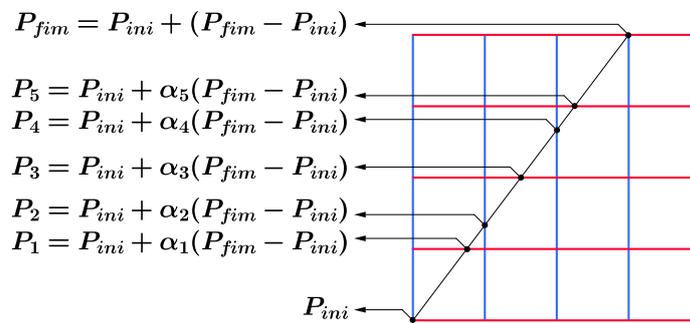
**Figura 3.2:** Obtendo a interseção do feixe com as retas paralelas obtidas na Figura 3.1

Como o feixe está partindo de  $P_1 = (-0.75, -1)$  e indo até  $P_2 = (4.5, 6)$ , o ponto  $P_{ini} = (0, 0)$  é resultado da primeira interseção que ocorre entre o feixe e uma das retas paralelas no intervalo delimitado por  $0 \leq x \leq m$  e  $0 \leq y \leq n$ . Calculando a distância entre  $P_{ini}$  com todos os demais pontos, obtemos o conjunto  $d = \{d_0, d_1, \dots, d_k\}$ , onde  $d_0 = 0$  e  $d_k$  é a distância do ponto  $P_{ini} = (0, 0)$  ao ponto  $P_{fim} = (3, 4)$ , o último a atravessar qualquer reta paralela dentro do intervalo definido.

Assim, podemos calcular o conjunto  $\alpha = \{0, \frac{d_1}{d_k}, \dots, \frac{d_{k-1}}{d_k}, 1\}$ , cujos valores representam os escalares que quando multiplicados pelo vetor definido pelos pontos  $P_{ini}$  e  $P_{fim}$ , podemos obter, partindo de  $P_{ini}$ , os pontos gerados pela interseção entre o feixe e as retas paralelas (Figura 3.3). Calculando os valores, obtemos respectivamente:

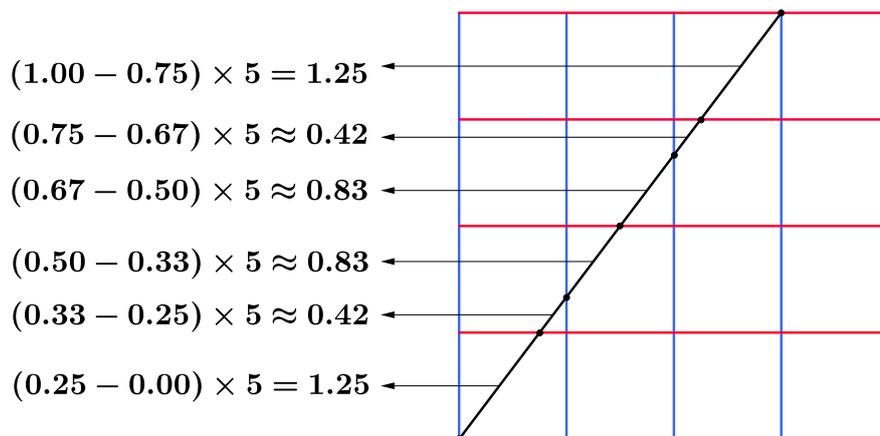
$$d = \{0, 1.25, 1.67, 2.5, 3.33, 3.75, 5\}$$

$$\alpha = \{0, 0.25, 0.33, 0.5, 0.67, 0.75, 1\}.$$



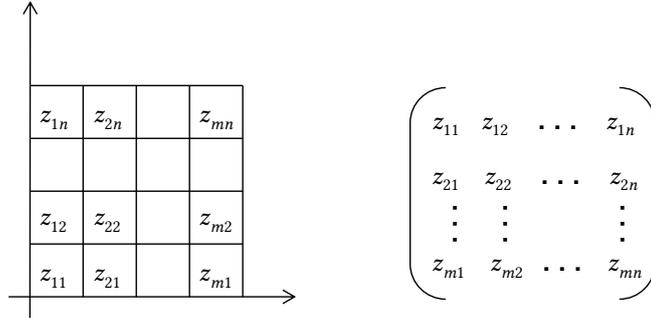
**Figura 3.3:** Valores do conjunto  $\alpha$  representam os escalares que quando multiplicados pelo vetor definido por  $P_{ini}$  e  $P_{fim}$  obtemos os pontos de interseção entre as retas paralelas e o feixe da Figura 3.2

Agora podemos calcular a distância entre os pontos que definem cada um dos seguimentos de reta através de  $[\alpha(k) - \alpha(k-1)] \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ , onde  $P_{ini} = (x_1, y_1)$  e  $P_{fim} = (x_2, y_2)$  (Figura 3.4).



**Figura 3.4:** Distâncias entre os pontos, dentro do intervalo definido, que representam cada um dos seguimentos de reta da Figura 3.3

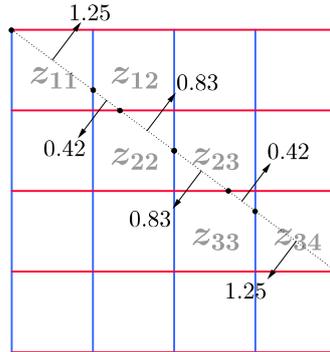
Note que existe uma diferença entre as coordenadas do plano cartesiano e as coordenadas de uma determinada matriz (Figura 3.5). Para mapear essas coordenadas, basta rotacionar a matriz



**Figura 3.5:** Diferenças entre as coordenadas do plano e matriz

$Z$  em  $90^\circ$  no sentido anti-horário ou considerar que o feixe que está atravessando esta matriz é perpendicular a este Figura 3.6.

$$Z = \begin{pmatrix} z_{11} & z_{12} & \dots & z_{1n} \\ z_{21} & z_{22} & \dots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & \dots & z_{mn} \end{pmatrix} \Rightarrow Z^* = \begin{pmatrix} z_{1n} & z_{2n} & \dots & z_{mn} \\ \vdots & \vdots & \ddots & \vdots \\ z_{12} & z_{22} & \dots & z_{m2} \\ z_{11} & z_{21} & \dots & z_{m1} \end{pmatrix} \quad (3.6)$$



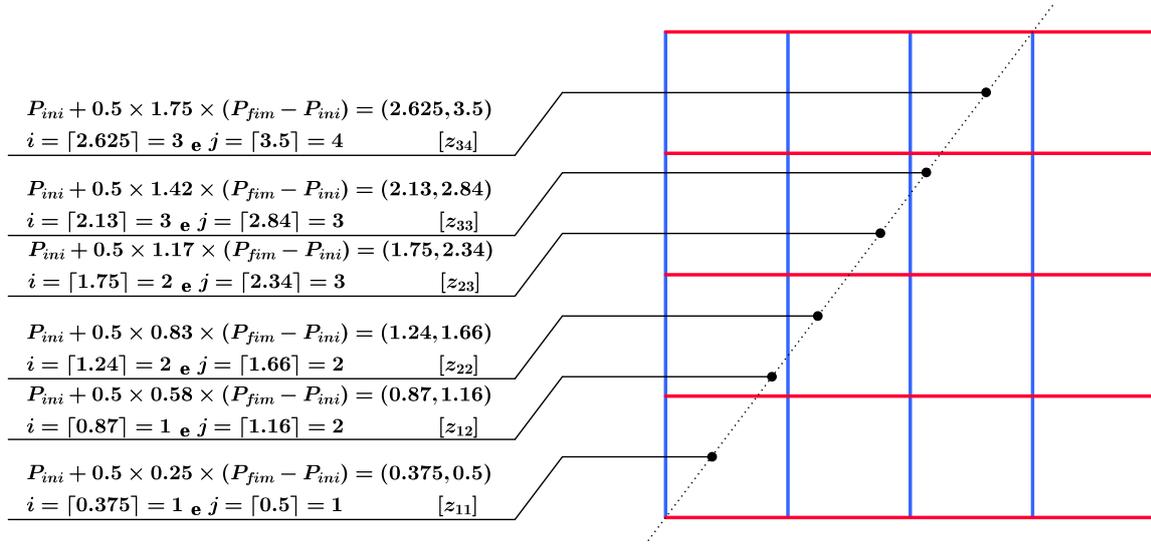
**Figura 3.6:** Uma maneira de mapearmos as coordenadas cartesianas do feixe para as coordenadas da matriz é considerarmos que o feixe atravessando a área definida pelas retas paralelas é perpendicular ao ângulo do feixe original

Calculamos agora o conjunto  $\alpha_{mid}$  por meio dos valores do conjunto  $\alpha$  através da equação:

$$\alpha_{mid}(k) = \frac{1}{2} [\alpha(k-1) + \alpha(k)].$$

Estes valores são escalares que quando multiplicados pelo vetor definido pelos pontos  $P_{ini}$  e  $P_{fim}$  e somados com o ponto  $P_{ini}$ , obtemos os pontos médios entre os segmentos (Figura 3.7). Respectivamente calculamos  $\alpha_{mid} = \{0.12, 0.29, 0.41, 0.58, 0.71, 0.87\}$  basta agora, para obtermos os índices da matriz, calcularmos:

$$\begin{aligned} \beta &:= \alpha_{mid}(k) \\ i(k) &= \lceil x_1 + \beta(x_2 - x_1) \rceil \\ j(k) &= \lceil y_1 + \beta(y_2 - y_1) \rceil. \end{aligned} \quad (3.7)$$



**Figura 3.7:** Mapeando as coordenadas do plano para as coordenadas da matriz utilizando o conjunto  $\alpha_{mid}$ , cujo valores representam os escalares que quando multiplicados pelo vetor definido por  $P_{ini}$  e  $P_{fim}$  obtemos o ponto médio dos segmentos obtidos na Figura 3.2

Como calculamos a distância entre os pontos que definiram cada um dos seguimentos de reta na Figura 3.4 e calculamos sobre quais índices  $i$  e  $j$  da matriz estes seguimentos estão passando (Figura 3.7), através de  $\sum_{k=2}^p z_{ij}^* (\alpha(k) - \alpha(k-1)) \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ , obtemos a soma-tória dos valores dos índices da matriz  $Z^*$  multiplicados pelo tamanho correspondente ao feixe que o atravessou. Numerando de 1 a  $m^* \times n^*$  os elementos da matriz  $Z^* \in \mathbb{R}^{m^* \times n^*}$  e tendo que  $x_{i \times n^* + j} := z_{ij}^*$  (Equação (3.8)). Obtendo a Equação (3.9).

$$Z^* = \begin{pmatrix} 4 & 8 & 12 & 16 \\ 3 & 7 & 11 & 15 \\ 2 & 6 & 10 & 14 \\ 1 & 5 & 9 & 13 \end{pmatrix} \rightarrow x = [4 \ 8 \ 12 \ 16 \\ 3 \ 7 \ 11 \ 15 \\ 2 \ 6 \ 10 \ 14 \\ 1 \ 5 \ 9 \ 13] \quad (3.8)$$

$$0x_1 + 0x_2 + 1.25x_3 + 0x_4 + 0x_5 + 0.83x_6 + 0.42x_7 + 0x_8 + 0.42x_9 \\ + 0.83x_{10} + 0x_{11} + 0x_{12} + 1.25x_{13} + 0x_{14} + 0x_{15} + 0x_{16} = 40 \quad (3.9)$$

Em uma simulação da Transformada de Radon, a Equação (3.9) representa uma equação do sistema linear esparsa que é gerada para cada  $(\theta, t)$ .

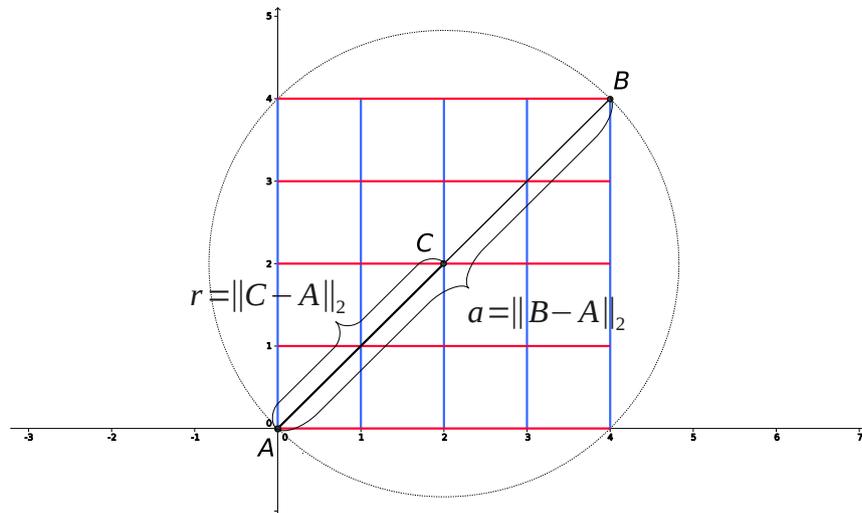
### 3.3 Exemplo de simulação da Transformada de Radon

A discretização da Transformada de Radon consiste no processo da transferência do modelo clássico da transformada para um modelo finito. O processo de discretização de um modelo é geralmente realizado como um primeiro passo para torná-lo adequado para avaliação numérica e implementação em computadores.

Para isto, definimos uma quantidade de feixes  $t$  que passam por cada ângulo  $\theta$ , assim como a quantidade de ângulos  $\theta$  usados. Uma vez que calculamos o traçado de um feixe na Seção 3.1, podemos agora simular vários destes feixes passando por estes ângulos variando entre  $[0, \pi)$ .

Utilizando-se da área delimitada pelas retas paralelas aos eixos  $x$  e  $y$  no intervalo  $0 \leq x \leq m$  e  $0 \leq y \leq n$  calculadas pelo algoritmo na Seção 3.1, o primeiro passo é definir um ponto central

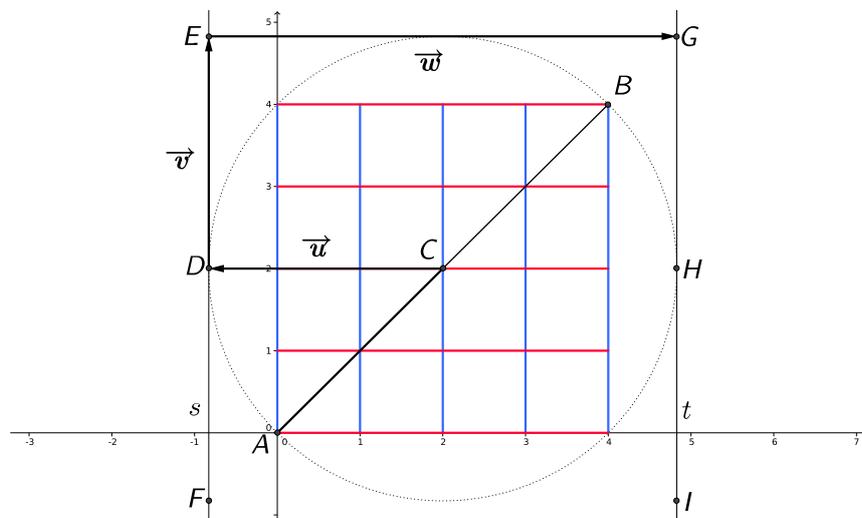
e depois obter a magnitude do raio e da amplitude pertencentes a circunferência imaginária que passa em torno desta área. Em uma matriz  $Z \in \mathbb{R}^{m \times n}$  o ponto central é  $C = (\frac{m}{2}, \frac{n}{2})$ , a magnitude do raio da circunferência imaginária é calculado através  $r = \|C - A\|_2$ , onde  $A = (0, 0)$  e a magnitude da amplitude é calculada através de  $a = \|B - A\|_2$ , onde  $B = (m, n)$  (Figura 3.8).



**Figura 3.8:** Ponto central, magnitude do raio e magnitude da amplitude pertencentes a circunferência imaginária que passa em torno da área definida

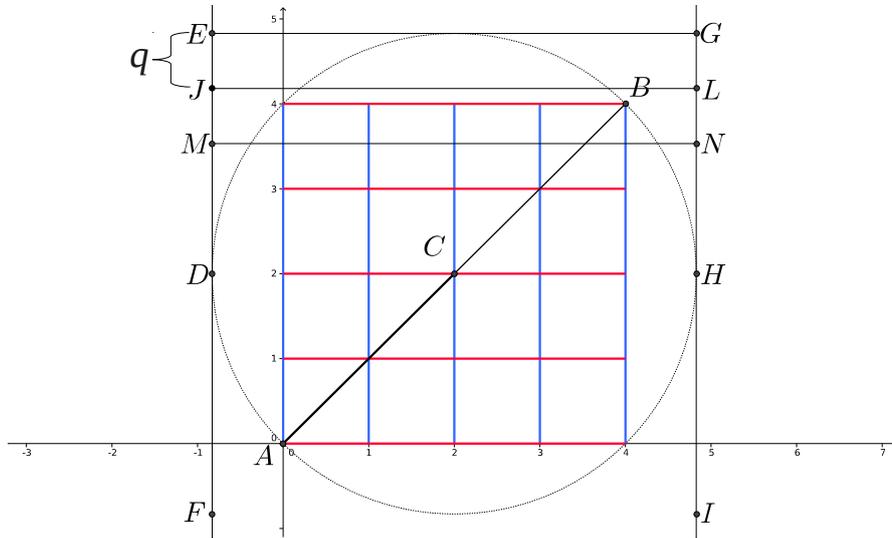
Como queremos traçar feixes no intervalo  $[0, \pi)$  escolhemos iniciar com o valor  $\theta = 0$ . Precisamos agora definir onde começa e onde termina cada feixe que atravessa esta área. Para isso, calculamos o ponto  $D = C - r(\cos \theta, \sin \theta)$ , que é o mesmo que somarmos o ponto  $C$  com o vetor  $\vec{u}$ , seguindo pelo ponto  $E = D + r(\cos(\theta + \frac{\pi}{2}), \sin(\theta + \frac{\pi}{2}))$  que é o mesmo que somarmos o ponto  $D$  com o vetor  $\vec{v}$  e  $F = D - r(\cos(\theta + \frac{\pi}{2}), \sin(\theta + \frac{\pi}{2}))$  (Figura 3.9).

Tendo que a reta  $t$  é paralela a  $s$  (Figura 3.9) e ambas passam pela tangente da circunferência imaginária, com o intervalo definido pelos pontos  $E$  e  $F$  na reta  $s$  podemos obter na reta  $t$  os pontos no intervalo entre  $G$  e  $I$  através da soma de qualquer ponto da reta  $s$  no intervalo definido com o vetor  $\vec{w} = 2(C - D)$ .



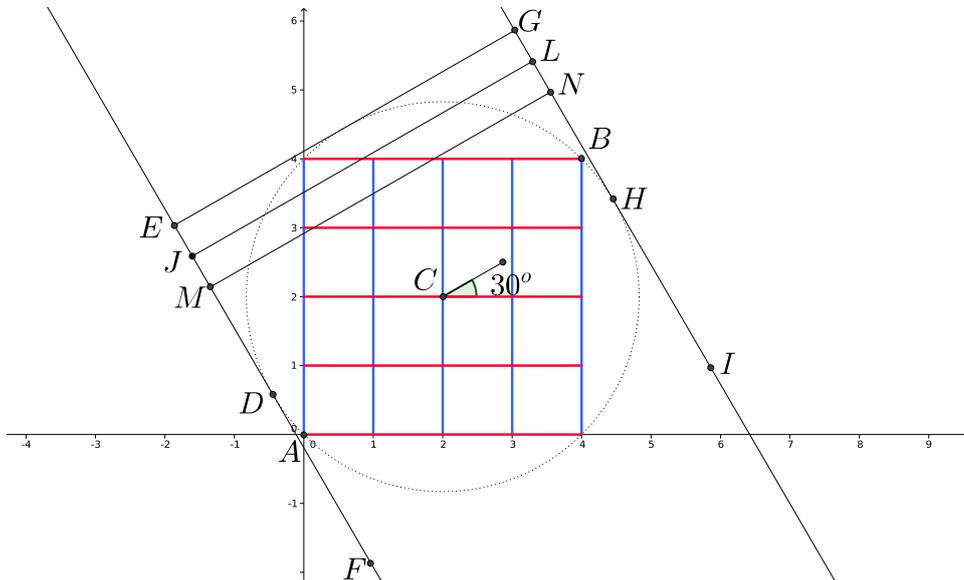
**Figura 3.9:** Retas paralelas e seus respectivos pontos que passam na tangente da circunferência imaginária e são perpendiculares ao ângulo  $\theta$  do feixe

Agora, digamos que queremos que uma quantidade  $k$  de feixes com ângulo  $\theta$  passe pela área delimitada, podemos calcular o escalar  $q = \frac{a}{k}$  que representa a distância entre cada feixe paralelo (Figura 3.10). Assim, o ponto inicial do  $k$ -ésimo feixe é definido por  $E - kq(\cos(\theta + \frac{\pi}{2}), \sin(\theta + \frac{\pi}{2}))$  e seu ponto final é definido por  $G - kq(\cos(\theta + \frac{\pi}{2}), \sin(\theta + \frac{\pi}{2}))$ .



**Figura 3.10:** Distância entre os feixes paralelos de um mesmo ângulo  $\theta$

Neste outro exemplo, apenas por questões ilustrativas, realizamos o mesmo processo anterior só que considerando o ângulo  $\theta = 30^\circ$ , obtendo a Figura 3.11.



**Figura 3.11:** Feixes de uma Projeção da transformada de Radon discreta para  $\theta = 30^\circ$

Como vimos anteriormente no Capítulo 2, quando aplicamos o Transformada de Radon sobre uma função, obtemos seu sinograma. No modelo discreto o tamanho do sinograma irá depender da precisão que queremos na discretização do problema. Por exemplo, caso utilizemos 128 feixes passando por cada ângulo  $\theta$  e utilizarmos 128 ângulos  $\theta$  diferentes variando entre  $[0, \pi)$ , obteremos um sinograma com dimensões  $\mathbb{R}^{128 \times 128}$ .

Através do processo descrito neste Capítulo, podemos gerar um sistema linear esparsa através da Transformada de Radon discretizada. Resolvendo este sistema linear obteremos uma aproxima-

ção da função na qual foi-se aplicada a transformada. Quanto maior for a quantidade de ângulos e feixes usados, quando solucionarmos este sistema, estaremos encontrando uma aproximação cada vez melhor da função original [8]. Este é um importante resultado que nos permite avaliar a reconstrução de imagens através de métodos para solução de sistemas lineares. No Capítulo 6 utilizamos esta transformada no *phantom* de Shepp-Logan para realizarmos testes em nosso método de escolha de parâmetros assim como em nosso algoritmo paralelo.



# Técnica de Reconstrução Algébrica

---

Um problema comum em diferentes áreas da matemática e da física consiste em encontrar um ponto de interseção dentre vários conjuntos convexos, conhecido como problema de factibilidade convexa. Sua formulação matemática, supondo que  $H$  seja um espaço de Hilbert e  $C_1, \dots, C_n$  são subconjuntos fechados com nenhuma interseção vazia, é definida por:

$$C = C_1 \cap \dots \cap C_n \neq \emptyset, \quad (4.1)$$

logo, o problema da factibilidade convexa passa a ser encontrar algum ponto  $C \neq \emptyset$  existente.

Quando usamos um modelo discreto, por exemplo para a reconstrução de imagens, cada conjunto  $C_i$  é um hiperplano definido por uma equação do sistema  $(x_i : \langle a_i, x^* \rangle = b_i)$  e podemos calcular uma solução exata ou aproximada deste problema através do uso de um algoritmo de projeção [3]. Os algoritmos de projeção, nos últimos anos, tem alcançando grandes avanços e já foram aplicados com sucesso em vários problemas, por exemplo, na reconstrução de imagens a partir de projeções [44, 60, 83, 87], no processamento de sinais [6, 16, 58], em equações Navier-Stokes [13, 57], entre outros. Estes métodos, usualmente, são obtidos através de diferentes abordagens: projeção em diferentes hiperplanos, minimização do próximo resíduo ou, quando a matriz do sistema linear for simétrica e positiva definida, por minimização de funções quadráticas [12].

Métodos de Projeções Alternadas (MAP - *Methods of Alternating Projections*), uma classe de algoritmos de projeção, são bastante utilizados para resolver o problema da factibilidade convexa (4.1). Porém, assim como o método dos gradientes ou dos sub-gradientes, MAP também pode ser lento, no entanto, caso tenhamos um método eficiente que escolha bem a ordem de suas projeções, ele pode se comportar de maneira competitiva [29, 43, 85].

O objetivo deste Capítulo é explicamos de uma maneira mais aprofundada os métodos que estamos buscando otimizar com o paralelismo e a escolha de parâmetros. O método de Kaczmarz, um tipo de MAP, é apresentado na Seção 4.1 em sua forma clássica. Na Seção 4.2 apresentamos sua versão aleatória proposta por Strohmer e Vershynin [85]. Na Seção 4.3 apresentamos o método de Eldar e Needell [28]. E na Seção 4.4 apresentamos nosso algoritmo paralelo e uma breve introdução à Plataforma de Computação Paralela CUDA.

## 4.1 Método Clássico de Kaczmarz

O método de Kaczmarz clássico, também conhecido como Técnica de Reconstrução Algébrica (ART - *Algebraic Reconstruction Technique*), é um algoritmo iterativo para solucionar sistemas do tipo  $Ax = b$  que age realizando operações nas linhas da matriz  $A$ . Ele efetua uma ‘varredura’ por todas estas linhas de uma maneira cíclica e sequencial. Nele, cada iteração depende da anterior e para que o processo de varredura em uma matriz  $A \in \mathbb{C}^{m \times n}$  complete um ciclo, são necessárias  $m$  iterações.

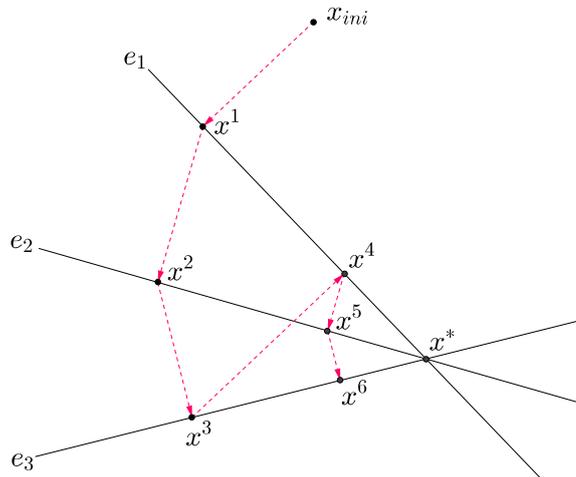
Este método, que tem uma longa história e rica literatura, originalmente foi proposto por Kaczmarz [48] e, de forma independente para uso na reconstrução de imagens, por Gordon, Bender e Herman [44]. Utilizando uma equação da matriz  $A$  por vez e atualizando a solução aproximada corrente ( $x^k$ ) a cada iteração o método converge para a solução deste sistema linear caso ele seja consistente [20]. E, caso o sistema seja inconsistente, o algoritmo converge até uma região próxima a interseção entre os hiperplanos e, não necessariamente, a uma solução de mínimos quadrados [27].

Sejam  $a_1, \dots, a_m \in \mathbb{R}^n$  as linhas de  $A$ . O método de Kaczmarz clássico realiza a cada iteração uma única projeção ortogonal, e é descrito por:

$$x^{k+1} = x^k + \lambda \frac{b_i - \langle a_i, x^k \rangle}{\|a_i\|_2^2} a_i^t, \quad (4.2)$$

onde  $x^k$  é a  $k$ -ésima iteração,  $i = (k \bmod m) + 1$ ,  $\|\cdot\|_2$  denota a norma Euclidiana ou norma espectral para vetores ou matrizes e  $\lambda$  o parâmetro de relaxação.

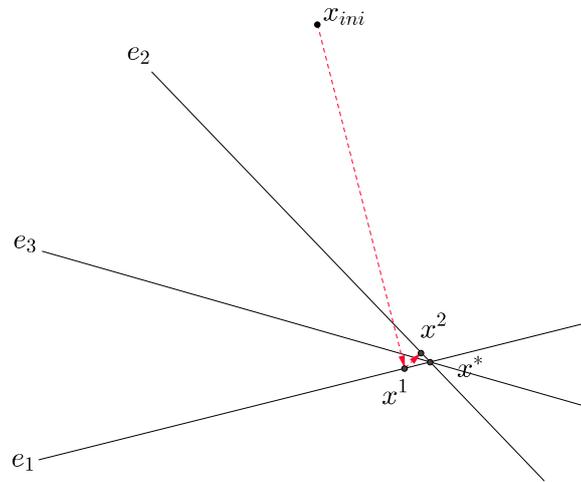
A representação geométrica do método é exemplificada pela Figura 4.1, onde utilizamos projeções ortogonais exatas (valor do parâmetro de relaxação  $\lambda = 1$ ) e verificamos a convergência do algoritmo. Nesta Figura 4.1 e nas seguintes (4.2, 4.4 e 4.5), os hiperplanos definidos pelas equações do sistema linear  $Ax = b$  estão representados como  $e_1, e_2$  e  $e_3$ .



**Figura 4.1:** Representação geométrica do algoritmo clássico de Kaczmarz

No artigo [43], Herman menciona que a ordem na qual os dados (hiperplanos da matriz  $A$ ) são selecionados para calcular a próxima iteração do algoritmo possui um grande efeito no desempenho do método. E ainda, artigos como [54] comentam que resultados na literatura apontam que uma escolha aleatória destes hiperplanos seja a melhor prática a ser empregada. Porém, caso ordenemos as linhas de forma que o menor ângulo formado entre o hiperplano escolhido pela iteração atual e o próximo seja o maior possível (no máximo até  $90^\circ$ ) e, além disso, para o cálculo da

primeira iteração escolhermos o hiperplano que maximiza  $\|x^1 - x^0\|_2$ , podemos melhorar a taxa de convergência de forma significativa, mas com um grande custo computacional (o que torna este processo inviável) (Figura 4.2).

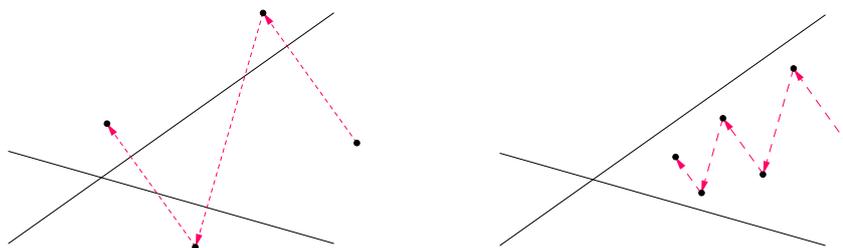


**Figura 4.2:** Representação geométrica do algoritmo clássico de Kaczmarz ordenando as equações do sistema de forma ótima

Também em [43] é mostrado que se ajustarmos o parâmetro de relaxação  $\lambda$  obtemos uma outra forma de otimizar o método. A dificuldade da escolha deste parâmetro está na dependência da ordem que os hiperplanos da matriz  $A$  são selecionados, no número de iterações que pretendemos utilizar para calcular a solução do sistema e na distância entre o  $x^0$  inicial e a solução.

Podemos considerar que o parâmetro de relaxação  $\lambda$  seja constante ou variável, mas Herman comenta em [43] que com a escolha de um parâmetro de relaxação constante foram obtidos melhores resultados. Outros livros também mencionam sobre a ordenação das equações da matriz  $A$  e sobre os parâmetros de relaxação, como [30, 49], mas nenhum deles comenta de fato como estes parâmetros devem ser selecionados, se não de forma manual.

A Figura 4.3 mostra os casos sobrerelaxado e subrelaxado do método. O caso sobrerelaxado é aquele em que o parâmetro de relaxação  $\lambda > 1$  e o caso subrelaxado o valor de  $\lambda < 1$ .



**Figura 4.3:** Efeito do parâmetro de relaxação sobrerelaxado e subrelaxado no Método de Kaczmarz clássico

Enquanto as condições para a convergência do método já estão bem estabelecidas, úteis resultados teóricos a respeito da taxa de convergência do método não foram demonstrados. E isto se dá principalmente devido ao que comentamos a respeito da ordenação dos hiperplanos do sistema, sem considerar ainda os parâmetros de relaxação. O que gostaríamos de obter é uma taxa de convergência baseada no número de condicionamento da matriz  $A$  a fim de podermos comparar sua eficiência com a de outros métodos presentes na literatura [85].

## 4.2 Método Kaczmarz Aleatório

Para driblar a dificuldade de estimar a taxa de convergência do Método de Kaczmarz devido a ordenação das linhas da matriz  $A$ , Strohmer and Vershynin [85] propuseram um método aleatório e provaram possuir taxa de convergência esperada exponencial. O método proposto possui uma probabilidade fixa para escolha das equações do sistema e converge de acordo com o número de condição escalado de  $A$ ,  $\hat{\kappa}(A) = \frac{\|A\|_F}{\sigma}$ , onde  $\sigma$  é o menor valor singular de  $A$ . Strohmer e Vershynin também postularam que seu método superava todos os outros métodos conhecidos quando empregados em sistemas extremamente sobredeterminados [85].

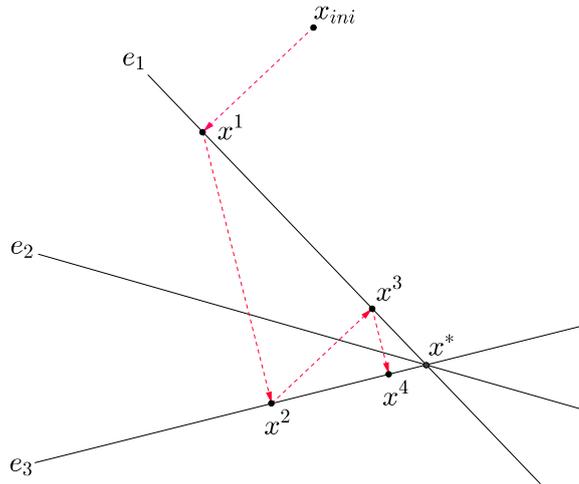
A ideia para sua realização partiu das observações realizadas em várias simulações numéricas onde a taxa de convergência do método de Kaczmarz era acelerada de forma significativa quando o algoritmo escolhia as linhas para realizar as projeções de uma maneira aleatória, ao invés de seguir a ordem cíclica. Strohmer e Vershynin propuseram uma versão específica de aleatoriedade, que busca selecionar as linhas da matriz  $A$  com uma probabilidade proporcional ao quadrado de sua norma euclidiana.

Tendo como  $Ax = b$  o sistema linear de equações e tendo como  $x^0$  o ponto inicial arbitrário, para  $k = 0, 1, \dots$ , o método de Kaczmarz Aleatório é definido por:

$$x^{k+1} = x^k + \lambda \frac{b_{p(k)} - \langle a_{p(k)}, x^k \rangle}{\|a_{p(k)}\|_2^2} a_{p(k)}, \quad (4.3)$$

onde  $p(k)$  assume o valor  $i \in \{1, 2, \dots, m\}$  com probabilidade  $\frac{\|a_i\|_2^2}{\|A\|_F^2}$  e  $\|A\|_F$  denota a norma de Frobenius.

A representação geométrica do método é exemplificada pela Figura 4.4, onde utilizamos projeções ortogonais exatas (valor do parâmetro de relaxação  $\lambda = 1$ ) e verificamos a convergência do algoritmo. Note que a reta  $e_2$  de fato não precisa ser selecionada nenhuma vez para que o método encontre a solução do sistema.



**Figura 4.4:** Representação geométrica do Método de Kaczmarz Aleatório

Como comentado em [29] e [86], este método de seleção das linhas não é ótimo em geral. Porém, a motivação de se selecionar as linhas com esta probabilidade encontra-se no fato que desta maneira, pode-se provar a taxa de convergência esperada do método. Além disso, é uma estratégia computacionalmente eficiente pois é necessário realizar o seu cálculo uma única vez e seus valores são conhecidos com exatidão. A concepção desta probabilidade partiu da ideia de pré-condicionar

a matriz  $A$  através do escalamento de suas linhas. Entretanto encontrar o pré-condicionador ótimo para um sistema do tipo  $Ax = b$  é um problema de otimização mais complexo do que calcular a inversa de  $A$  [86]. Desta forma foi necessário uma alternativa computacionalmente mais barata e menos precisa.

A taxa de convergência para o algoritmo aleatório de Strohmer (4.3) possui um limite exponencial de

$$\mathbb{E}\|x^k - x^*\|_2^2 \leq \left(1 - \frac{1}{\hat{\kappa}(A)}\right)^k \|x^0 - x^*\|_2^2,$$

onde  $\hat{\kappa}(A) = \frac{\|A\|_F}{\underline{\sigma}}$ ,  $\underline{\sigma}$  é o menor valor singular de  $A$ ,  $x^0$  é uma estimativa inicial arbitrária e  $\mathbb{E}$  denota a esperança sobre a escolha das linhas. Isto assumindo que a matriz  $A$  é de posto completo de modo que  $\|A^{-1}\| \stackrel{def}{=} \inf \{M : M\|Ax\|_2 \geq \|x\|_2, \forall x\}$  esteja bem definido.

Neste método aleatório quando  $A$  é bem condicionada (como uma regra geral, se o número de condição  $\kappa(A) = 10^k$ , então podemos perder até  $k$  dígitos de precisão quando solucionamos o sistema linear utilizando um método direto [21]) implica em uma convergência com  $O(n)$  iterações. Uma vez que cada iteração consiste em uma única projeção (que leva  $O(n)$  para ser calculada), podemos concluir que com o tempo  $O(n^2)$  espera-se que o método convirja. Com este resultado espera-se que o método se comporte de uma melhor maneira do que o método de Eliminação Gaussiana que leva tempo de  $O(mn^2)$  para convergir. Quanto maior o sistema levado em consideração, mais distante esta comparação se torna.

É claro que antes de tudo, precisamos saber (ou aproximar) as normas euclidianas de cada linha da matriz  $A$  antes de iniciarmos o algoritmo, o que leva  $O(nm)$ . Mas estas normas, em muitos casos, podem ser conhecidas a priori. Por exemplo, todas elas podem ser iguais a 1 (como o caso das matrizes de Vandermonde provenientes de aproximações trigonométricas) ou quando estão concentrando ao redor de um valor constante (como o caso de algumas matrizes aleatórias) [85, 86].

O algoritmo Kaczmarz Strohmer pode ser descrito em linguagem estruturada da seguinte forma, note que neste algoritmo utilizamos um número de projeções fixas, porém podemos estipular a condição de parada baseada em um erro mínimo.

---

### - Kaczmarz Strohmer

---

```

1: function KACZMARZ_STROHMER( $A, b, x, \lambda, proj$ )
2:    $k \leftarrow 0$ 
3:    $x^0 \leftarrow x$ 
4:   while  $k < proj$  do
5:     seleciona  $i$  com a probabilidade  $\frac{\|a_i\|_2^2}{\|A\|_F^2}$ 
6:      $a_i \leftarrow A_i$   $\triangleright a_i$  recebe um vetor linha com os elementos da linha  $i$  da matriz  $A$ 
7:      $x^{k+1} \leftarrow x^k + \lambda \frac{b_i - a_i x^k}{\|a_i\|_2^2} a_i^t$ 
8:      $k \leftarrow k + 1$ 
9:   return  $x^k$ 

```

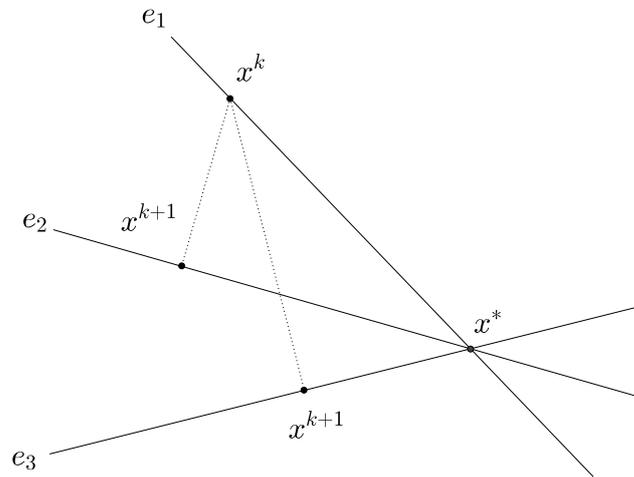
---

Strohmer e Vershynin em suas simulações numéricas e resultados teóricos demonstraram em [85] que até para sistemas moderadamente sobredeterminados seu algoritmo poderia convergir de maneira mais rápida do que o Método dos Gradientes Conjugados. Principalmente na solução de um modelo para o problema da reconstrução de funções limitadas por banda a partir de um conjunto não uniforme apresentado em [31]. A resolução deste problema através do Método de Kaczmarz Aleatório se mostrou bastante promissora para esta probabilidade usada e também o

utilizamos em nossas simulações numéricas no Capítulo 6. O uso desta probabilidade para seleção das linhas resolve parcialmente o problema da ordenação das linhas da matriz, pois como foi dito dito, esta probabilidade não é a ótima. No entanto, quando  $A$  é bem condicionada, o método se comporta de maneira eficiente.

### 4.3 Método Kaczmarz Aleatório acelerado através do Lema de Johnson-Lindenstrauss

Eldar e Needell em [29], a fim de melhorar a velocidade de convergência do algoritmo de Strohmer e Vershynin [85], propuseram uma nova abordagem que sugere uma maneira diferente para seleção das linhas da matriz  $A$  para realizar as projeções. Uma vez que as projeções no algoritmo de Strohmer são feitas de maneira ortogonal, podemos ver através da Figura 4.5 que a projeção ótima na  $k$ -ésima iteração é a que maximiza  $\|x^{k+1} - x^k\|_2$ . Porém, calcular a norma euclidiana entre vetores requer  $O(n)$  operações e claramente não podemos efetuar muitos cálculos adicionais por iteração.



**Figura 4.5:** Kaczmarz Projeção Ótima

A abordagem de Eldar e Needell sugere projetar a matriz  $A$  em um espaço de dimensão reduzido de forma que as características das linhas de  $A$  não sejam tão afetadas. Assim, a projeção ótima ( $\max \|x^{k+1} - x^k\|_2$ ) é calculada nesse espaço de dimensão reduzida, porém só a que mais avança no sentido da solução do sistema é escolhida. Após escolher a equação neste espaço de dimensão reduzida, o algoritmo a utiliza para fazer a projeção utilizando a matriz  $A$  e calculando o valor de  $x^{k+1}$ .

Uma maneira de reduzir a dimensão de um problema é através de uma matriz aleatória  $\Phi$  com elementos selecionados de forma independente e igualmente distribuídos a partir da distribuição normal de Gauss (como utilizado em [29]). Deste modo, com a matriz  $\Phi \in \mathbb{R}^{n \times d}$  gerada aleatoriamente com distribuição Gaussiana podemos reduzir a dimensão da matriz  $A \in \mathbb{R}^{m \times n}$  em uma matriz  $\bar{A} \in \mathbb{R}^{m \times d}$  com uma alta probabilidade de preservar a geometria do problema através do cálculo:

$$\bar{A}_{m \times d} = A_{m \times n} \times \Phi_{n \times d}. \tag{4.4}$$

A ideia chave deste mapeamento aleatório surge do lema de Johnson-Lindenstrauss [47]. Este lema afirma que um pequeno conjunto de pontos em um espaço de grande dimensão pode ser

mapeado em um espaço de dimensão muito menor de tal modo que as distâncias entre os pontos sejam praticamente preservadas. Uma demonstração deste resultado pode ser vista em [22]. Outras formas de reduzir a dimensão do problema podem ser observadas em [1, 2].

Eldar e Needell afirmam que este algoritmo modificado possui uma taxa de convergência para a solução do sistema  $Ax = b$  no pior caso semelhante ao Método de Kaczmarz Aleatório de Strohmer e Vershynin. O que é verdade, pois no pior caso ele utiliza o primeiro índice deste conjunto selecionado através da probabilidade imposta por Strohmer e Vershynin. Este fato pode ser observado no Algoritmo descrito logo abaixo.

Na prática, é esperado que ele convirja com um tempo menor que o algoritmo de Strohmer e Vershynin quando considerarmos uma grande redução de dimensionalidade da matriz  $A$ , uma vez que o produto de  $\Phi_{d \times n}$  por  $x^k$  é realizado a cada iteração do algoritmo. Como, neste caso,  $d \ll n$  e cada iteração custa  $O(pdn)$  ( $p$  é o número de linhas selecionadas para o cálculo da qual anda mais em relação a solução do sistema), espera-se que o algoritmo convirja em  $O(pdn^2)$  operações. Entretanto caso  $d$  seja muito pequeno a taxa de convergência será semelhante a do algoritmo de Strohmer e Vershynin mas com um custo computacional maior. Eldar e Needell sugerem em seu artigo que a redução de dimensão seja feita com  $d = \log n$ .

O Algoritmo Kaczmarz Aleatório Johnson-Lindenstrauss utilizando a redução de dimensão (4.4) pode ser escrito em linguagem estruturada da seguinte forma.

---

#### - Algoritmo Kaczmarz Aleatório Johnson-Lindenstrauss

---

```

1: function KACZMARZ_ALEATORIO_JL( $A, b, x, d, \lambda, proj$ )
2:    $k \leftarrow 0$ 
3:    $\Phi \leftarrow$  matriz Gaussianas aleatória com dimensão  $d \times n$ 
4:    $\bar{A} \leftarrow A \times \Phi^t$ 
5:    $x^0 \leftarrow x$ 
6:   while  $k < proj$  do
7:      $v \leftarrow$  seleciona o índice de  $p$  linhas de  $\bar{A}$  com a probabilidade  $\frac{\|a_i\|_2^2}{\|A\|_F^2}$ 
8:     for  $j \leftarrow 1$  até  $p$  do  $\triangleright$  calcula qual  $v_j$  caminha mais em relação a solução do sistema
9:        $\alpha_{v_j} \leftarrow \bar{A}_{v_j} \triangleright \alpha_{v_j}$  recebe um vetor linha com os elementos da linha de índice  $v_j$  da
matriz  $\bar{A}$ 
10:        $\gamma_j \leftarrow \frac{|b_{v_j} - \langle \alpha_{v_j}, \Phi x^k \rangle|}{\|\alpha_{v_j}\|_2}$ 
11:        $q \leftarrow$  índice onde  $\max\{\gamma_1, \dots, \gamma_p\}$  é atingido
12:        $\gamma_{v_q}^* \leftarrow \frac{|b_{v_q} - \langle a_{v_q}, x^k \rangle|}{\|a_{v_q}\|_2}$ 
13:        $\gamma_{v_1}^* \leftarrow \frac{|b_{v_1} - \langle a_{v_1}, x^k \rangle|}{\|a_{v_1}\|_2}$ 
14:       if  $\gamma_{v_q}^* > \gamma_{v_1}^*$  then
15:          $v_q \leftarrow v_1$ 
16:          $x^{k+1} \leftarrow x^k + \lambda \frac{b_{v_q} - a_{v_q} x^k}{\|a_{v_q}\|_2^2} a_{v_q}^t$ 
17:          $k \leftarrow k + 1$ 
18:   return  $x^k$ 

```

---

## 4.4 Método Kaczmarz Aleatório Paralelo

Como discutido por Eldar e Needell em [29] no caso em que as projeções são feitas de maneira exata, é fácil ver que a projeção ótima a ser selecionada é a que maximiza  $\|x^{k+1} - x^k\|_2$ . Mas, infe-

lizmente, o custo do cálculo de cada uma dessas normas é impraticável caso feito sequencialmente. No entanto, este processo pode ser facilmente paralelizado.

Nossa ideia foi selecionar  $p$  linhas da matriz  $A$  seguindo a probabilidade imposta por Strohmer e Vershynin em [85] e calcular através de processamento paralelo os valores das normas  $\|x^{k+1} - x^k\|_2$  utilizando a ideia de Eldar e Needell [29], porém sem considerar a redução de dimensionalidade. Para isto, utilizamos a Plataforma de Computação Paralela CUDA, armazenamos a matriz  $A$  utilizando memória de textura e para cada uma das  $p$  linhas selecionadas calculamos os valores de  $\|x^{k+1} - x^k\|_2$  em diferentes *threads* estruturadas em uma única *grid* da GPU.

Consideramos duas implementações paralelas, na primeira calculamos os valores de  $x^{k+1}$  nas *threads* e armazenamos seus valores em memória com o intuito de futuramente utiliza-los como a próxima iteração do algoritmo (caso maximizassem a norma  $\|x^{k+1} - x^k\|_2$ ). Porém esta estratégia não se mostrou muito eficaz já que o custo para armazenar todos os valores de  $x^{k+1}$  de cada *thread* se mostrou maior do que apenas armazenar os índices usados para o cálculo de cada  $x^{k+1}$  e suas respectivas normas. Por fim utilizamos as *threads* apenas para calcular e armazenar os valores das normas de  $\|x^{k+1} - x^k\|_2$  e os índices que geraram estas normas. Note que utilizando essa abordagem, não precisamos calcular o valor de  $x^{k+1}$ , devido ao seguinte resultado:

$$\begin{aligned}
x^{k+1} &= x^k + \frac{b_i - \langle a_i, x^k \rangle}{\|a_i\|_2^2} a_i^t \\
\|x^{k+1} - x^k\|_2 &= \left\| x^k + \frac{b_i - \langle a_i, x^k \rangle}{\|a_i\|_2^2} a_i^t - x^k \right\|_2 \\
\|x^{k+1} - x^k\|_2 &= \left\| \frac{b_i - \langle a_i, x^k \rangle}{\|a_i\|_2^2} a_i^t \right\|_2 \\
\|x^{k+1} - x^k\|_2 &= \frac{|b_i - \langle a_i, x^k \rangle|}{\|a_i\|_2}.
\end{aligned} \tag{4.5}$$

O Algoritmo Kaczmarz Paralelo pode ser escrito em linguagem estruturada da seguinte forma.

---

#### - Algoritmo Kaczmarz Paralelo

---

```

1: function KACZMARZ_STROHMER_PARALELO( $A, b, x, \lambda, proj$ )
2:    $k \leftarrow 0$ 
3:    $x^0 \leftarrow x$ 
4:   while  $k < proj$  do
5:      $v \leftarrow$  seleciona o índice de  $p$  linhas de  $A$  com a probabilidade  $\frac{\|a_i\|_2^2}{\|A\|_F^2}$ 
6:     for  $j \leftarrow 1$  até  $p$  do  $\triangleright$  calcula em paralelo qual  $v_j$  caminha mais em relação a solução do sistema
7:        $a_{v_j} \leftarrow A_{v_j}$   $\triangleright a_{v_j}$  recebe um vetor linha com os elementos da linha de índice  $v_j$  da matriz  $A$ 
8:        $norms_j \leftarrow \frac{|b_{v_j} - \langle a_{v_j}, x^k \rangle|}{\|a_{v_j}\|_2}$ 
9:        $q \leftarrow$  índice onde  $\max\{norms_1, \dots, norms_p\}$  é atingido
10:       $x^{k+1} \leftarrow x^k + \lambda \frac{b_{v_q} - \langle a_{v_q}, x^k \rangle}{\|a_{v_q}\|_2^2} a_{v_q}^t$ 
11:       $k \leftarrow k + 1$ 
12:   return  $x^k$ 

```

---

Uma abordagem semelhante porém considerando a redução de dimensionalidade também foi implementada, mas utilizando o algoritmo descrito acima obtivemos os melhores resultados.

Resultados de simulações utilizando este algoritmo são realizadas no Capítulo 6 Seção 6.3. A seguir, na Subseção 4.4.1 damos uma breve introdução a arquitetura CUDA utilizada para o desenvolvimento deste algoritmo paralelo.

#### 4.4.1 Arquitetura CUDA

Inicialmente, as GPUs foram criadas para processar *pixels* e *vértices* de forma paralela. Quando começaram a ser usadas para o desenvolvimento de aplicações mais genéricas como em [10, 36, 37, 52], passaram a ser consideradas como unidades de processamento paralelo de baixo custo.

Tendo em vista que as GPUs oferecem esse enorme potencial computacional, a NVIDIA em fevereiro de 2007 tornou pública uma nova plataforma de software denominado CUDA (*Compute Unified Device Architecture*). Com a CUDA, é possível explorar o potencial computacional das GPUs, até então restrito a computação gráfica, para a computação em geral.

Antes de desenvolver uma aplicação utilizando GPUs deve-se analisar o problema e verificar se ele pode tirar proveito do paralelismo. Alguns fatores importantes para a avaliação do problema são:

- Intensidade numérica do problema;
- Coalescência de memória.

A intensidade numérica se refere ao trabalho que as *threads* irão executar, por exemplo, um período muito longo de processamento de cálculos com poucos acessos à memória. A Coalescência de memória é importante para aproveitar uma característica do modelo de memória do CUDA, que permite a leitura de vários elementos de memória ao mesmo tempo.

O desenvolvimento de aplicações em CUDA conta com um conjunto de bibliotecas fornecidas pela NVIDIA que trabalham como intermediárias entre a aplicação e o dispositivo. Não é possível que a aplicação comunique-se diretamente com o hardware, tal comunicação deve ser feita através do driver da CUDA, de bibliotecas de tempo de execução (runtime) ou de bibliotecas de alto nível, todas fornecidas pela NVIDIA. Tal organização de dispositivos e bibliotecas pode ser vista na Figura 4.6.

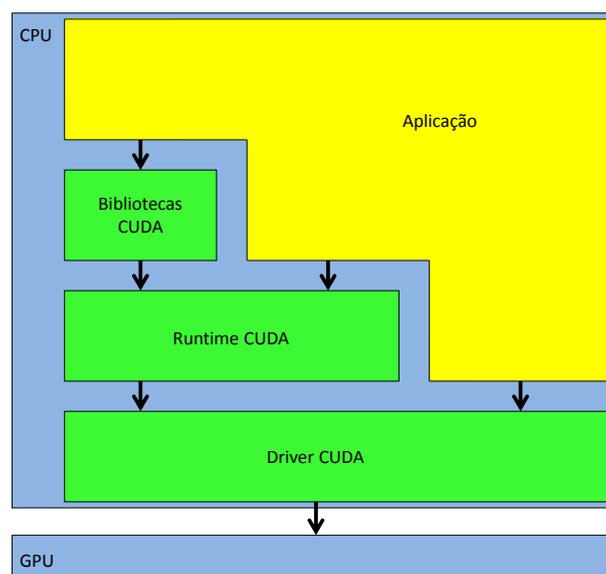


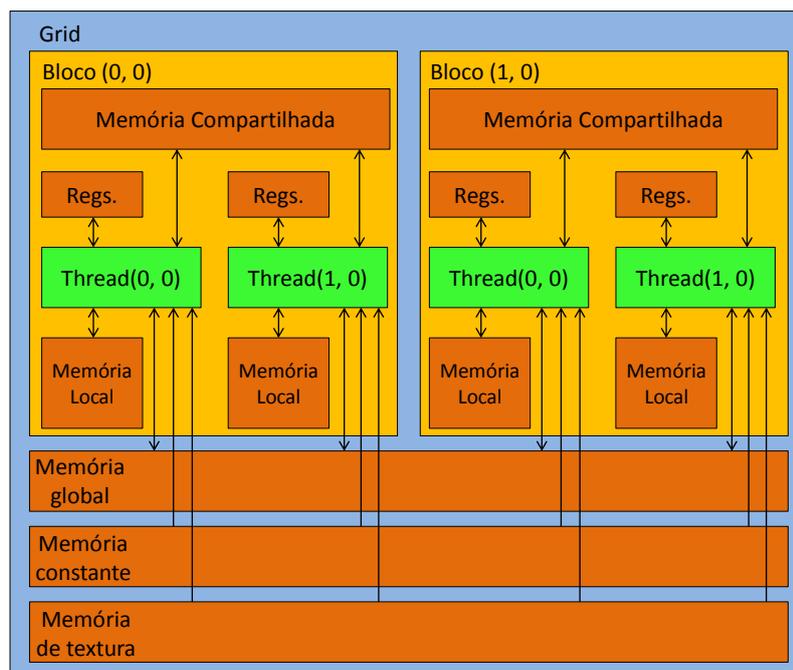
Figura 4.6: Arquitetura CUDA

O algoritmo é executado em paralelo através milhares de *threads*, cada uma delas executa o mesmo conjunto de instruções de forma independente em dados diferentes. Um bloco é um grupo de *threads* que compartilham dados através de memória compartilhada e sincronizam sua execução para coordenar os acessos de memória. Estes pontos de sincronização são especificadas dentro do código e agem como uma barreira onde todas as *threads* em um bloco ficam suspensas até serem sincronizadas. Um *grid* é uma organização de blocos de *threads*. Dentro de cada *grid*, existe um número de blocos definido pelo programador.

A CUDA possui um modelo de memória em vários níveis como visto na Figura 4.7. Esse modelo de memória permite que sejam feitas algumas otimizações em seu acesso, como utilizar a memória constante ou de textura para dados que apenas serão lidos.

A vantagem de se utilizar a memória de textura é quando queremos tirar proveito da forma em que os dados estão armazenados. Quando realizarmos uma leitura, dados adjacentes também serão lidos, podendo estar arranjada em  $\mathbb{R}$ ,  $\mathbb{R}^2$  ou  $\mathbb{R}^3$ . Por exemplo: no caso em que os dados estão armazenados em  $\mathbb{R}^2$  como matrizes, os dados adjacentes em todas as direções da posição acessada também serão lidos.

A memória constante pode ser escrita apenas a partir do código executado em CPU, possui uma velocidade de leitura maior que as demais e seu tamanho é muito reduzido, por isso ela é mais adequada para armazenar, por exemplo, parâmetros de execução.



**Figura 4.7:** Modelo de memória da CUDA, as setas indicam a direção de acesso à memória. No caso das memórias constante e de textura, é permitido apenas a leitura.

Usando CUDA, a GPU passa a ser visto como um co-processador da CPU permitindo que tarefas sejam atribuídas a GPU enquanto a CPU continua a execução normal da aplicação. Apesar das vantagens apresentadas, algumas limitações ainda são encontradas na arquitetura: funções recursivas não são suportadas, orientação a objetos é parcialmente suportada e apenas em dispositivos mais novos e a largura de banda do barramento de dados e a latência entre CPU e GPU podem provocar gargalo na transferência de dados.

Como discutido em [28], o método de Kaczmarz cíclico pode não ser o melhor algoritmo possível para a arquitetura de GPU. De fato, é muito barato criar *threads* para atingir ganhos em rotinas

BLAS (*Basic Linear Algebra Subprogram*), porém, o problema deve ser grande o suficiente para que o tempo de instanciar uma *thread*, transferir os dados da memória principal e para a memória da GPU e qualquer outra interrupção necessária, não sejam maiores que os ganhos. Elble *et al.* [28] ainda afirma que devido a este fato e que as únicas operações que possam ser paralelizadas são o produto vetorial e a adição de vetores, o algoritmo apresentaria um desempenho superior na GPU apenas quando o sistema linear for muito grande e denso. Porém, nas implementações que realizamos através da arquitetura CUDA não paralelizamos o algoritmo em si, mas sim como ele processa a estratégia de seleção de linhas.



# Sobre a Escolha do Parâmetro de Relaxação

Neste Capítulo apresentamos a metodologia usada para estimar o parâmetro de relaxação no método aleatório de Strohmer e Vershynin [85, 86], escolha esta baseada na taxa de convergência esperada do método. Na Seção 5.1, fazemos uma generalização dos resultados obtidos em [85, 86], porém desta vez considerando o parâmetro de relaxação. Posteriormente usamos o resultado obtido na Seção 5.1 e encontramos uma forma de estimar o valor do parâmetro de relaxação  $\lambda$  na Seção 5.2.

## 5.1 Taxa de Convergência Esperada Exponencial Considerando o Parâmetro de Relaxação

Em um trabalho recente [85], o problema de encontrar a taxa de convergência esperada para o método de Kaczmarz para a solução de sistemas lineares consistentes sobredeterminados foi resolvida para o caso em que a ordem das projeções é escolhida aleatoriamente seguindo uma distribuição proporcional à norma das linhas da matriz de coeficientes  $A$ . A demonstração está baseada no seguinte resultado <sup>1</sup>:

**Lema 5.1.1.** *Seja  $Z_k := a_{i(k)}/\|a_{i(k)}\|_2$  e  $Ax^* = b$ , então o algoritmo (4.2) satisfaz:*

$$\|x^{k+1} - x^*\|_2^2 = \|x^k - x^*\|_2^2 + (\lambda^2 - 2\lambda) \langle Z_k, x^k - x^* \rangle^2.$$

*Demonstração.*

$$\begin{aligned} x^{k+1} &= x^k - \lambda \frac{\langle a_{i(k)}, x^k \rangle - b_{i(k)}}{\|a_{i(k)}\|_2^2} a_{i(k)}^t \\ &= x^k - \lambda \left( \left\langle \frac{a_{i(k)}}{\|a_{i(k)}\|_2}, x^k \right\rangle - \frac{b_{i(k)}}{\|a_{i(k)}\|} \right) \frac{a_{i(k)}^t}{\|a_{i(k)}\|_2}. \end{aligned}$$

<sup>1</sup>Os autores mostram o resultado para o caso  $\lambda = 1$ , mas generalização é natural e pouco altera a demonstração

Uma vez que  $\langle a_{i(k)}, x^* \rangle = b_{i(k)}$ , então:

$$\begin{aligned} x^{k+1} &= x^k - \lambda (\langle Z_k, x^k \rangle - \langle Z_k, x^* \rangle) Z_k^t \\ &= x^k - \lambda \langle Z_k, x^k - x^* \rangle Z_k^t. \end{aligned} \quad (5.1)$$

Portanto:

$$\begin{aligned} \|x^{k+1} - x^*\|_2^2 &= \|x^k - x^* - \lambda \langle Z_k, x^k - x^* \rangle Z_k^t\|_2^2 \\ &= \|x^k - x^*\|_2^2 - 2\lambda \langle Z_k, x^k - x^* \rangle^2 + \lambda^2 \langle Z_k, x^k - x^* \rangle^2 \\ &= \|x^k - x^*\|_2^2 + (\lambda^2 - 2\lambda) \langle Z_k, x^k - x^* \rangle^2. \end{aligned} \quad \square$$

A fim de tentarmos melhorar o efeito do parâmetro  $\lambda$  na convergência, consideramos uma iteração anterior do algoritmo utilizando novamente o Lema (5.1.1), chegando a:

$$\|x^{k+1} - x^*\|_2^2 = \|x^{k-1} - x^*\|_2^2 + (\lambda^2 - 2\lambda) [\langle Z_{k-1}, x^{k-1} - x^* \rangle^2 + \langle Z_k, x^k - x^* \rangle^2]. \quad (5.2)$$

Por outro lado, usando (5.1):

$$\begin{aligned} \langle Z_k, x^k - x^* \rangle &= \langle Z_k, x^k - x^{k-1} + x^{k-1} - x^* \rangle \\ &= \langle Z_k, x^k - x^{k-1} \rangle + \langle Z_k, x^{k-1} - x^* \rangle \\ &= \langle Z_k, -\lambda \langle Z_{k-1}, x^{k-1} - x^* \rangle Z_{k-1} \rangle + \langle Z_k, x^{k-1} - x^* \rangle \\ &= -\lambda \langle Z_{k-1}, x^{k-1} - x^* \rangle \langle Z_k, Z_{k-1} \rangle + \langle Z_k, x^{k-1} - x^* \rangle. \end{aligned} \quad (5.3)$$

Substituindo a Equação (5.3) em (5.2), chegamos a:

$$\begin{aligned} \|x^{k+1} - x^*\|_2^2 &= \|x^{k-1} - x^*\|_2^2 + (\lambda^2 - 2\lambda) [\langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \\ &\quad + (-\lambda \langle Z_{k-1}, x^{k-1} - x^* \rangle \langle Z_k, Z_{k-1} \rangle + \langle Z_k, x^{k-1} - x^* \rangle^2)]. \end{aligned} \quad (5.4)$$

Assim, a fim de minimizar  $\|x^{k+1} - x^*\|_2^2$  teríamos que encontrar o minimizador de um polinômio de grau 4 em  $\lambda$ , uma tarefa trivial dado que os coeficientes de tais polinômios possam ser calculados, o que é o caso. Note que  $\langle Z_{k-1}, x^{k-1} - x^* \rangle$  seria necessário para o algoritmo de qualquer forma e  $\langle Z_k, Z_{k-1} \rangle$  pode ser pré-calculado. Enquanto  $\langle Z_k, x^{k-1} - x^* \rangle$  não seria calculado normalmente pelo método, notamos que a quantidade necessária  $\langle Z_k, x^k - x^* \rangle$  pode ser obtida facilmente desse valor, de forma que o cálculo dos coeficientes não adicionaria muito ao esforço computacional.

Infelizmente, calcular o valor de  $\lambda$  por esta abordagem não se mostrou uma prática vantajosa e o parâmetro escolhido tende a estar muito próximo de 1. Porém, consideramos apenas poucos passos do algoritmo ( $x^k$  e  $x^{k-1}$ ), entretanto aumentar o número de iterações consideradas não é praticável, já que torna os cálculos muito custosos.

Uma outra forma de tentarmos estimar o efeito a longo prazo da relaxação pode ser obtido quando  $Z_k$  são variáveis aleatórias independentes tais que:

$$P \left( Z_k = \frac{a_{i(k)}}{\|a_{i(k)}\|_2} \right) = \frac{\|a_{i(k)}\|_2^2}{\|A\|_F^2},$$

esta probabilidade de seleção segue a estratégia adotada por Strohmer e Vershynin no algoritmo 4.3.

**Lema 5.1.2.** Utilizando a probabilidade  $P\left(Z_k = \frac{a_{i(k)}}{\|a_{i(k)}\|_2}\right) = \frac{\|a_{i(k)}\|_2^2}{\|A\|_F^2}$ , podemos obter a taxa de convergência esperada do Método de Kaczmarz Aleatório de Strohmer e Vershynin levando em consideração o parâmetro de relaxação  $\lambda \in (0, 2)$ :

$$\mathbb{E}\|x^{k+1} - x^*\|_2^2 \leq (1 + (\lambda^2 - 2\lambda)\hat{\kappa}^{-2}(A)) \mathbb{E}\|x^k - x^*\|_2^2, \quad (5.5)$$

onde  $\hat{\kappa}(A) = \frac{\|A\|_F}{\underline{\sigma}}$  é o número de condição escalado e  $\underline{\sigma}$  é o menor valor singular da matriz  $A$ .

*Demonstração.*

Seja  $\underline{\sigma}$  o menor valor singular da matriz  $A$ , então:

$$\min_{v \neq 0} \frac{\|Av\|_2}{\|v\|_2} = \underline{\sigma}.$$

Ou seja, para qualquer  $y \neq 0$ :

$$\begin{aligned} \frac{\|Ay\|_2^2}{\|y\|_2^2} &\geq \underline{\sigma}^2 \\ \sum_{i=1}^m \frac{|\langle a_i, y \rangle|^2}{\|y\|_2^2} &\geq \underline{\sigma}^2 \\ \sum_{i=1}^m |\langle a_i, y \rangle|^2 &\geq \frac{\|y\|_2^2}{\underline{\sigma}^{-2}} \\ \sum_{i=1}^m \|a_i\|_2^2 \left| \left\langle \frac{a_i}{\|a_i\|_2}, y \right\rangle \right|^2 &\geq \frac{\|y\|_2^2}{\underline{\sigma}^{-2}}. \end{aligned}$$

Multiplicando ambos os lados da inequação por  $\frac{1}{\|A\|_F^2}$ , obtemos:

$$\begin{aligned} \sum_{i=1}^m \frac{\|a_i\|_2^2}{\|A\|_F^2} \left| \left\langle \frac{a_i}{\|a_i\|_2}, y \right\rangle \right|^2 &\geq \frac{\|y\|_2^2}{\|A\|_F^2 \underline{\sigma}^{-2}} \\ \sum_{i=1}^m \frac{\|a_i\|_2^2}{\|A\|_F^2} \left| \left\langle \frac{a_i}{\|a_i\|_2}, y \right\rangle \right|^2 &\geq \hat{\kappa}^{-2}(A) \|y\|_2^2. \end{aligned}$$

Considerando a probabilidade  $P\left(Z_k = \frac{a_{i(k)}}{\|a_{i(k)}\|_2}\right) = \frac{\|a_{i(k)}\|_2^2}{\|A\|_F^2}$ , temos:

$$\mathbb{E}|\langle Z_k, y \rangle|^2 \geq \hat{\kappa}^{-2}(A) \|y\|_2^2.$$

Dado que  $y$  seja obtido através de  $x^k - x^*$ :

$$\begin{aligned} \mathbb{E}|\langle Z_k, x^k - x^* \rangle|^2 &\geq \hat{\kappa}^{-2}(A) \|x^k - x^*\|_2^2 \\ \mathbb{E} \frac{|\langle Z_k, x^k - x^* \rangle|^2}{\|x^k - x^*\|_2^2} &\geq \hat{\kappa}^{-2}(A). \end{aligned} \quad (5.6)$$

Reescrevendo a equação do Lema (5.1.1), temos:

$$\begin{aligned}
\|x^{k+1} - x^*\|_2^2 &= \|x^k - x^*\|_2^2 + (\lambda^2 - 2\lambda) \langle Z_k, x^k - x^* \rangle^2 \\
&= \|x^k - x^*\|_2^2 + \|x^k - x^*\|_2^2 (\lambda^2 - 2\lambda) \left\langle Z_k, \frac{x^k - x^*}{\|x^k - x^*\|_2} \right\rangle^2 \\
&= \left( 1 + (\lambda^2 - 2\lambda) \left\langle Z_k, \frac{x^k - x^*}{\|x^k - x^*\|_2} \right\rangle^2 \right) \|x^k - x^*\|_2^2.
\end{aligned} \tag{5.7}$$

Agora utilizando a probabilidade  $P\left(Z_k = \frac{a_{i(k)}}{\|a_{i(k)}\|_2}\right) = \frac{\|a_{i(k)}\|_2^2}{\|A\|_F^2}$  e  $\mathbb{E}\left\langle Z_k, \frac{x^k - x^*}{\|x^k - x^*\|_2} \right\rangle^2 \geq \hat{\kappa}^{-2}(A)$  (obtida em (5.6)), avaliamos a esperança condicional a respeito da escolha de  $Z_k$  fixando-se os vetores  $Z_0, \dots, Z_{k-1}$  ( $\mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}}$ ) e obtendo os vetores  $x^1, \dots, x^k$ . Notando que  $\lambda \in (0, 2)$  implica  $\lambda^2 - 2\lambda < 0$ , escrevemos (5.7) da seguinte forma:

$$\begin{aligned}
\mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 &= \left( 1 + \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} (\lambda^2 - 2\lambda) \left\langle Z_k, \frac{x^k - x^*}{\|x^k - x^*\|_2} \right\rangle^2 \right) \|x^k - x^*\|_2^2 \\
&\leq (1 + (\lambda^2 - 2\lambda) \hat{\kappa}(A)^{-2}) \|x^k - x^*\|_2^2.
\end{aligned}$$

Considerando agora a esperança completa em ambos os lados, chegamos ao resultado:

$$\mathbb{E} \|x^{k+1} - x^*\|_2^2 \leq (1 + (\lambda^2 - 2\lambda) \hat{\kappa}(A)^{-2}) \mathbb{E} \|x^k - x^*\|_2^2. \quad \square$$

Um resultado semelhante ao Lema (5.1.2) é obtido em [85], a diferença é que agora estamos levando o parâmetro de relaxação em consideração. Obviamente, não se pode ver nenhuma utilidade para a relaxação aqui uma vez que o ótimo seria atingido com  $\lambda = 1$ .

## 5.2 Encontrando o Parâmetro de Relaxação

Utilizando uma generalização dos resultados obtidos pelo Lema (5.1.2) na Seção 5.1 deste Capítulo, obtemos:

**Lema 5.2.1.** *Suponha que  $Z_k$  segue a lei de probabilidade  $P\left(Z_k = \frac{a_{i(k)}}{\|a_{i(k)}\|_2}\right) = \frac{\|a_{i(k)}\|_2^2}{\|A\|_F^2}$ , então temos:*

$$\mathbb{E} \langle Z_k, x \rangle \langle Z_k, y \rangle = \frac{1}{\|A\|_F^2} \langle Ax, Ay \rangle.$$

*Demonstração.*

$$\begin{aligned}
\mathbb{E} \langle Z_k, x \rangle \langle Z_k, y \rangle &= \sum_{i=1}^m \frac{\|a_i\|_2^2}{\|A\|_F^2} \left\langle \frac{a_i}{\|a_i\|_2}, x \right\rangle \left\langle \frac{a_i}{\|a_i\|_2}, y \right\rangle \\
&= \frac{1}{\|A\|_F^2} \sum_{i=1}^m \langle a_i, x \rangle \langle a_i, y \rangle \\
&= \frac{1}{\|A\|_F^2} \langle Ax, Ay \rangle. \quad \square
\end{aligned}$$

O que fizemos para encontrar as estimativas para o parâmetro de relaxação  $\lambda$  foi utilizar o Lema 5.2.1 em (5.4) e avaliar primeiro a esperança condicional a  $Z_0, \dots, Z_{k-1}$  ( $\mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}}$ ) e então

condicional a  $Z_0, \dots, Z_{k-2}$  ( $\mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}}$ ), estimar as quantidades comparadas a  $\|x^{k-1} - x^*\|_2^2$  e depois utilizar os resultados para estimar  $\|x^{k+1} - x^*\|_2^2$  comparando a  $\|x^{k-1} - x^*\|_2^2$ .

Partindo de (5.4) e calculando a distributiva de  $(-\lambda \langle Z_{k-1}, x^{k-1} - x^* \rangle \langle Z_k, Z_{k-1} \rangle + \langle Z_k, x^{k-1} - x^* \rangle)^2$ , podemos chegar a:

$$\begin{aligned} \|x^{k+1} - x^*\|_2^2 &= \|x^{k-1} - x^*\|_2^2 + (\lambda^2 - 2\lambda) [\langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \\ &\quad + \lambda^2 \langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \langle Z_k, Z_{k-1} \rangle^2 \\ &\quad - 2\lambda \langle Z_{k-1}, x^{k-1} - x^* \rangle \langle Z_k, Z_{k-1} \rangle \langle Z_k, x^{k-1} - x^* \rangle \\ &\quad + \langle Z_k, x^{k-1} - x^* \rangle^2]. \end{aligned} \quad (5.8)$$

Passando a esperança condicional dos dois lados na Equação (5.8), fixando-se a escolha dos vetores  $Z_0, \dots, Z_{k-1}$  para realizar as projeções e obter os vetores  $x^1, \dots, x^k$ . Avaliamos a esperança condicional  $\mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}}$  sobre a escolha de  $Z_k$ .

$$\begin{aligned} \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 &= \|x^{k-1} - x^*\|_2^2 \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} 1 \\ &\quad + (\lambda^2 - 2\lambda) [\langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} 1 \\ &\quad + \lambda^2 \langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \langle Z_k, Z_{k-1} \rangle^2 \\ &\quad - 2\lambda \langle Z_{k-1}, x^{k-1} - x^* \rangle \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \langle Z_k, Z_{k-1} \rangle \langle Z_k, x^{k-1} - x^* \rangle \\ &\quad + \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \langle Z_k, x^{k-1} - x^* \rangle^2]. \end{aligned} \quad (5.9)$$

Aplicando-se o Lema (5.2.1) na Equação (5.9), obtemos:

$$\begin{aligned} \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 &= \|x^{k-1} - x^*\|_2^2 + (\lambda^2 - 2\lambda) [\langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \\ &\quad + \lambda^2 \langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \frac{\|AZ_{k-1}\|_2^2}{\|A\|_F^2} \\ &\quad - 2\lambda \langle Z_{k-1}, x^{k-1} - x^* \rangle \frac{\langle Z_{k-1}, A^t A (x^{k-1} - x^*) \rangle}{\|A\|_F^2} \\ &\quad + \frac{\|A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^2}]. \end{aligned} \quad (5.10)$$

Para futuramente aplicarmos o Lema (5.2.1) em  $\langle Z_{k-1}, x^{k-1} - x^* \rangle^2$  de uma maneira simplificada, encontramos o maximizador de  $\frac{\|AZ_{k-1}\|_2^2}{\|A\|_F^2}$  e o chamamos de  $\varrho$ . Isto torna a Equação (5.10), na seguinte Inequação:

$$\begin{aligned} \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 &\geq \|x^{k-1} - x^*\|_2^2 + (\lambda^2 - 2\lambda) [\langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \\ &\quad + \lambda^2 \varrho \langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \\ &\quad - \frac{2\lambda}{\|A\|_F^2} \langle Z_{k-1}, x^{k-1} - x^* \rangle \langle Z_{k-1}, A^t A (x^{k-1} - x^*) \rangle \\ &\quad + \frac{\|A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^2}], \end{aligned} \quad (5.11)$$

onde  $\varrho = \max \frac{\|AZ_{k-1}\|_2^2}{\|A\|_F^2}$ .

Agora, passando a esperança condicional dos dois lados na Inequação (5.11), fixando-se a escolha dos vetores  $Z_0, \dots, Z_{k-2}$  para realizar as projeções e obter os vetores  $x^1, \dots, x^{k-1}$ . Avaliamos a esperança condicional  $\mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}}$  a respeito da escolha de  $Z_{k-1}$ .

$$\begin{aligned} \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 &\geq \|x^{k-1} - x^*\|_2^2 \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} 1 \\ &\quad + (\lambda^2 - 2\lambda) \left[ \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} \langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \right. \\ &\quad + \lambda^2 \varrho \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} \langle Z_{k-1}, x^{k-1} - x^* \rangle^2 \\ &\quad - \frac{2\lambda}{\|A\|_F^2} \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} \langle Z_{k-1}, x^{k-1} - x^* \rangle \langle Z_{k-1}, A^t A (x^{k-1} - x^*) \rangle \\ &\quad \left. + \frac{\|A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^2} \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} 1 \right]. \end{aligned} \quad (5.12)$$

Aplicando-se o Lema (5.2.1) na Inequação (5.12), obtemos:

$$\begin{aligned} \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 &\geq \|x^{k-1} - x^*\|_2^2 \\ &\quad + (\lambda^2 - 2\lambda) \left[ \frac{\|A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^2} \right. \\ &\quad + \lambda^2 \varrho \frac{\|A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^2} - 2\lambda \frac{\|A^t A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^4} \\ &\quad \left. + \frac{\|A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^2} \right]. \end{aligned} \quad (5.13)$$

Agora a partir de (5.13), colocamos os termos semelhantes em evidência obtendo (5.14).

$$\begin{aligned} \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 &\geq \|x^{k-1} - x^*\|_2^2 \\ &\quad + (\lambda^2 - 2\lambda) \left( 2 + \lambda^2 \varrho - 2\lambda \frac{\|A^t A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^2 \|A (x^{k-1} - x^*)\|_2^2} \right) \frac{\|A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^2} \\ &\quad \geq \|x^{k-1} - x^*\|_2^2 \\ &\quad + \frac{(\lambda^2 - 2\lambda)}{\|A\|_F^2} \left( 2 + \lambda^2 \varrho - 2\lambda \frac{\|A^t A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^2 \|A (x^{k-1} - x^*)\|_2^2} \right) \|A (x^{k-1} - x^*)\|_2^2. \end{aligned} \quad (5.14)$$

Na Inequação (5.14), como o valor de  $-2\lambda \frac{\|A^t A (x^{k-1} - x^*)\|_2^2}{\|A\|_F^2 \|A (x^{k-1} - x^*)\|_2^2}$  é negativo, podemos subestimar seu valor sem quebrar a restrição de inferioridade do lado direito, uma vez que  $(\lambda^2 - 2\lambda)$  sempre é negativo. Como o rank  $(A \in \mathbb{R}^{m \times n}) = n$  e  $\min_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \sigma_n$  é o menor valor singular  $\sigma$  e sabendo que ao calculamos  $A^t A$  apenas estamos elevando ao quadrado os valores singulares de  $A$ . Podemos escrever (5.14) como:

$$\begin{aligned} \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 &\geq \|x^{k-1} - x^*\|_2^2 \\ &\quad + \frac{(\lambda^2 - 2\lambda)}{\|A\|_F^2} \left( 2 + \lambda^2 \varrho - 2\lambda \frac{\sigma^4 \|x^{k+1} - x^*\|_2^2}{\|A\|_F^2 \sigma^2 \|x^{k+1} - x^*\|_2^2} \right) \|A (x^{k-1} - x^*)\|_2^2, \end{aligned} \quad (5.15)$$

agora, apenas simplificando (5.15), obtemos:

$$\begin{aligned} \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 &\geq \|x^{k-1} - x^*\|_2^2 \\ &+ \frac{(\lambda^2 - 2\lambda)}{\|A\|_F^2} \left( 2 + \lambda^2 \underline{\rho} - 2\lambda \frac{\underline{\sigma}^2}{\|A\|_F^2} \right) \|A\|_2 \|x^{k-1} - x^*\|_2^2, \end{aligned} \quad (5.16)$$

onde  $\frac{\|A^t A(x^{k-1} - x^*)\|_2^2}{\|A(x^{k-1} - x^*)\|_2^2}$  foi subestimado por  $\underline{\sigma}^2$ , o menor valor singular da matriz  $A$  ao quadrado.

Em (5.16), como  $\|A(x^{k-1} - x^*)\|_2^2$  é positivo, podemos superestimar seu valor.

$$\begin{aligned} \mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 &\geq \|x^{k-1} - x^*\|_2^2 \\ &+ \frac{(\lambda^2 - 2\lambda)}{\|A\|_F^2} \left( 2 + \lambda^2 \underline{\rho} - 2\lambda \frac{\underline{\sigma}^2}{\|A\|_F^2} \right) \|A\|_2^2 \|x^{k-1} - x^*\|_2^2 \\ &\geq \|x^{k-1} - x^*\|_2^2 \\ &+ \frac{\|A\|_2^2 (\lambda^2 - 2\lambda)}{\|A\|_F^2} \left( 2 + \lambda^2 \underline{\rho} - 2\lambda \frac{\underline{\sigma}^2}{\|A\|_F^2} \right) \|x^{k-1} - x^*\|_2^2 \\ &\geq \|x^{k-1} - x^*\|_2^2 \\ &+ \frac{\|A\|_2^2 (\lambda^2 - 2\lambda)}{\|A\|_F^4} (2\|A\|_F^2 + \lambda^2 \rho - 2\lambda \underline{\sigma}^2) \|x^{k-1} - x^*\|_2^2 \end{aligned} \quad (5.17)$$

onde  $\rho := \max_{i=1 \dots m} \frac{\|A a_i\|_2^2}{\|a_i\|_2^2}$ . Assim, (5.17) pode ser escrito como:

$$\mathbb{E}_{\{Z_0, \dots, Z_{k-2}\}} \mathbb{E}_{\{Z_0, \dots, Z_{k-1}\}} \|x^{k+1} - x^*\|_2^2 \geq \left[ 1 + \frac{\|A\|_2^2 (\lambda^2 - 2\lambda)}{\|A\|_F^4} (2\|A\|_F^2 + \lambda^2 \rho - 2\lambda \underline{\sigma}^2) \right] \|x^{k-1} - x^*\|_2^2. \quad (5.18)$$

Agora passando a esperança completa em ambos os lados a Inequação (5.18) obtemos:

$$\mathbb{E} \|x^{k+1} - x^*\|_2^2 \geq \left[ 1 + \frac{\|A\|_2^2 (\lambda^2 - 2\lambda)}{\|A\|_F^4} (2\|A\|_F^2 + \lambda^2 \rho - 2\lambda \underline{\sigma}^2) \right] \mathbb{E} \|x^{k-1} - x^*\|_2^2, \quad (5.19)$$

ou seja, para estimar o parâmetro de relaxação  $\lambda$  neste caso, precisamos calcular:

$$\min_{\lambda \in (0,2)} (\lambda^2 - 2\lambda) (2\|A\|_F^2 + \lambda^2 \rho - 2\lambda \underline{\sigma}^2). \quad (5.20)$$

Analogamente, considerando desta vez  $\gamma = \min \frac{\|A Z_{k-1}\|_2}{\|A\|_F}$  na equação (5.10), chegamos ao resultado:

$$\mathbb{E} \|x^{k+1} - x^*\|_2^2 \leq \left[ 1 + \frac{(\lambda^2 - 2\lambda)}{\hat{\kappa}^2(A) \|A\|_F^2} (2\|A\|_F^2 + \lambda^2 \delta - 2\lambda \|A\|_2^2) \right] \mathbb{E} \|x^{k-1} - x^*\|_2^2, \quad (5.21)$$

onde  $\delta := \min_{i=1 \dots m} \frac{\|A a_i\|_2^2}{\|a_i\|_2^2}$  e  $\hat{\kappa}^2(A) = \frac{\|A\|_F^2}{\underline{\sigma}^2}$ . Ou seja, para estimar o parâmetro de relaxação  $\lambda$  neste caso, precisamos calcular:

$$\min_{\lambda \in (0,2)} (\lambda^2 - 2\lambda) (2\|A\|_F^2 + \lambda^2 \delta - 2\lambda \|A\|_2^2). \quad (5.22)$$

Para chegar em ambos os resultados, tanto em (5.20) como em (5.22), utilizamos valores superestimados e subestimados para tornar os cálculos necessários na obtenção de  $\lambda$  menos custosos e desta maneira não podemos esperar muita precisão. A minimização (5.20) é otimista em relação a taxa de convergência do algoritmo e, desta forma, calcula o parâmetro de relaxação  $\lambda$  acreditando que o algoritmo terá uma boa convergência. Por outro lado, a minimização (5.22) é pessimista neste mesmo aspecto. Em geral minimizando (5.20) obtemos parâmetros  $\lambda > 1$  e minimizando (5.22) obtemos  $\lambda < 1$ . No Capítulo 6 fizemos simulações para estudar o comportamento de (5.20) e (5.22) e apresentamos os resultados experimentais obtidos.

# Simulações Numéricas

Este Capítulo tem o objetivo de avaliar as duas estimativas, (5.20) e (5.22), encontradas na Seção 5.2 do Capítulo anterior. E além disso, avaliar o desempenho de nossa implementação paralela do método discutida no Capítulo 4 Seção 4.4.

Na Seção 6.1, a fim de efetuar simulações e mensurar o desempenho das estimativas, utilizamos diferentes matrizes sobredeterminadas ( $A \in \mathbb{R}^{m \times n}$  com  $m \gg n$ ) de posto completo variando o número de iterações, chute inicial e parâmetro de relaxação e apresentamos seus resultados através de gráficos. Após isto, na Seção 6.2 comparamos o desempenho do Método de Kaczmarz Aleatório de Strohmer e Vershynin com o Método dos Gradientes Conjugados para Matrizes não Simétricas.

Na Seção 6.3, efetuamos simulações no algoritmo paralelo considerando o tempo para solucionar um sistema linear variando o parâmetro de relaxação  $\lambda$  e o tempo para reconstruir uma imagem do *phantom* de Shepp-Logan.

## 6.1 Simulação Numérica da Estimativa

Para as primeiras simulações, cogitamos os polinômios trigonométricos discutidos por Feichtinger e Gröchenig em [32, 39] e também utilizado por Strohmer e Vershynin [85] para avaliação de seu algoritmo. A reconstrução de funções limitadas por banda a partir de amostras não uniformes  $\{f(t_k)\}$  é um problema clássico na análise de Fourier com uma vasta gama de aplicações [7]. Como um modelo eficiente de dimensão finita apropriado para o tratamento para do problema de amostragem não uniforme, o modelo consiste em solucionar o sistema linear:

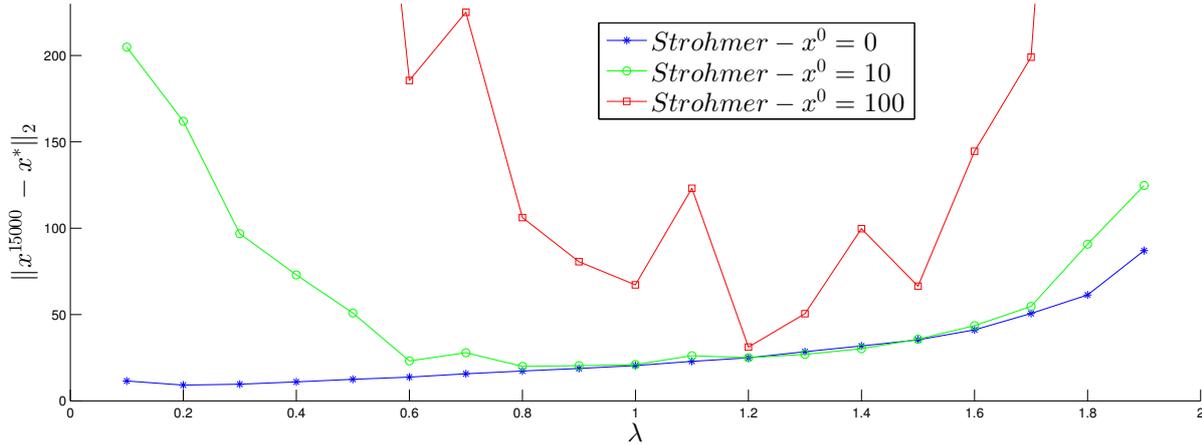
$$Ax = b, \quad \text{onde } A_{j,k} = \sqrt{w_j} e^{2\pi i k t_j}, \quad b_j = \sqrt{w_j} f(t_j), \quad (6.1)$$

onde  $\{t_j\}_{j=1}^m$  são valores uniformemente distribuídos,  $w_j = \frac{t_{j+1} - t_{j-1}}{2}$ , com  $j = 1, \dots, m$ .  $x = \{x_k\}_{k=-r}^r \in \mathbb{C}^{2r+1}$  é um vetor de valores complexos arbitrários,  $f(t_j) = \sum_{k=-r}^r x_k e^{2\pi i k t_j} \forall j = 1, \dots, m$  e  $i = \sqrt{-1}$ .

Nesta primeira simulação, utilizando o Método de Kaczmarz Aleatório no sistema linear (6.1), fixamos o número de iterações (*proj*)  $k = 15000$  e analisamos o erro  $\|x^k - x^*\|_2$  para valores do

parâmetro de relaxação  $\lambda \in (0, 2)$  em diferentes valores iniciais de  $x^0 = 0$ ,  $x^0 = 10$  e  $x^0 = 100$  (definidos no final da Página 19). Obtendo a Figura 6.1.

A matriz  $A \in \mathbb{C}^{m \times n}$  e o vetor  $b \in \mathbb{C}^m$  que utilizamos possuem dimensões  $m = 5000$ ,  $n = 1001$ , construídos a partir de  $r = 500$  ( $n = 2r + 1$ ) e valores de  $t_j$  gerados através de uma função geradora de números pseudo-aleatórios uniformemente distribuídos dentro do intervalo  $[0, 1]$  e ordenados de forma ascendente. Esta matriz possui norma de Frobenius  $\|A\|_F = 38.73$ , menor valor singular  $\underline{\sigma} = 0.24$ , maior valor singular  $\bar{\sigma} = 22.37$  e número de condicionamento  $\kappa(A) = \bar{\sigma}/\underline{\sigma} = 93.26$ .



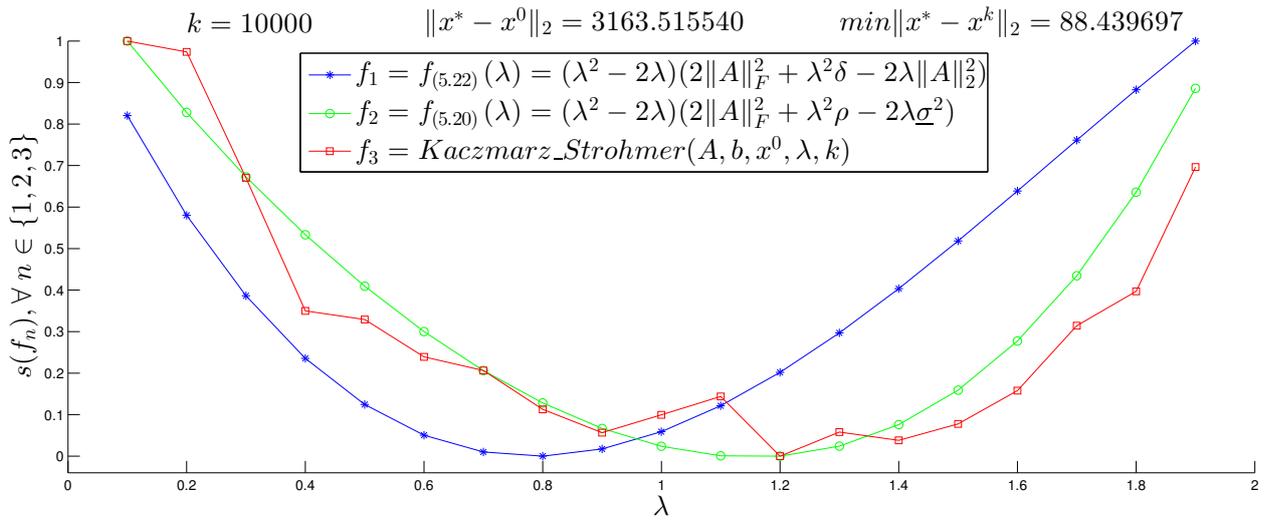
**Figura 6.1:** Taxa de convergência do Método de Kaczmarz Aleatório em função da variação do parâmetro de relaxação e valor do chute inicial.

Analisando a Figura 6.1, podemos ver que mantendo  $k$  fixo, o valor do  $\lambda$  ótimo varia de acordo com o valor de  $x^0$  inicial. Temos que para  $x^0 = 0$  o valor do  $\lambda$  ótimo no intervalo  $[0.1, 0.3]$ , para  $x^0 = 10$  o valor do  $\lambda$  está no intervalo  $[0.8, 0.9]$  e para  $x^0 = 100$  o valor do  $\lambda$  ótimo no intervalo  $[1.2, 1.3]$  – quaisquer valores dentro destes intervalos já seriam boas aproximações. Alguns dos intervalos que foram mencionados estão difíceis de serem enxergados na Figura 6.1 por causa da diferença entre as escalas das funções, porém, nos gráficos seguintes podemos ver estes intervalos com uma maior clareza.

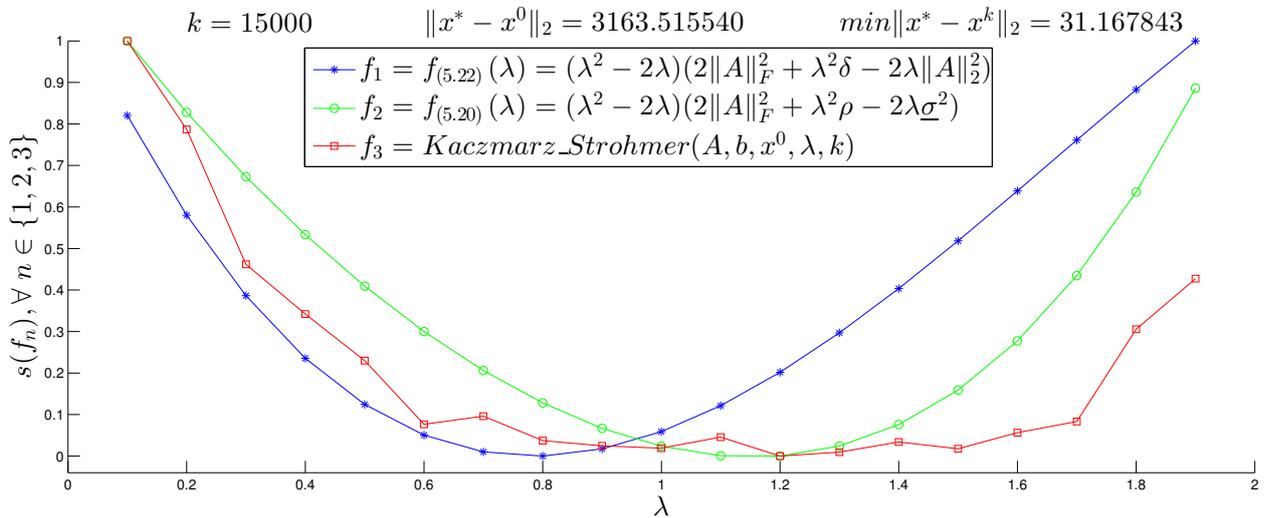
Utilizando a fórmula (6.2), podemos colocar em escala  $[0, 1]$  os valores de  $\|x^k - x^*\|_2$  apresentados na Figura 6.1 e também os de nossas estimativas apresentadas em (5.20) e (5.22) dado  $\lambda \in (0, 2)$ .

$$s(f(x)) = \frac{f(x) - \inf_{(0,2)} f(x)}{\sup_{(0,2)} f(x) - \inf_{(0,2)} f(x)} \quad (6.2)$$

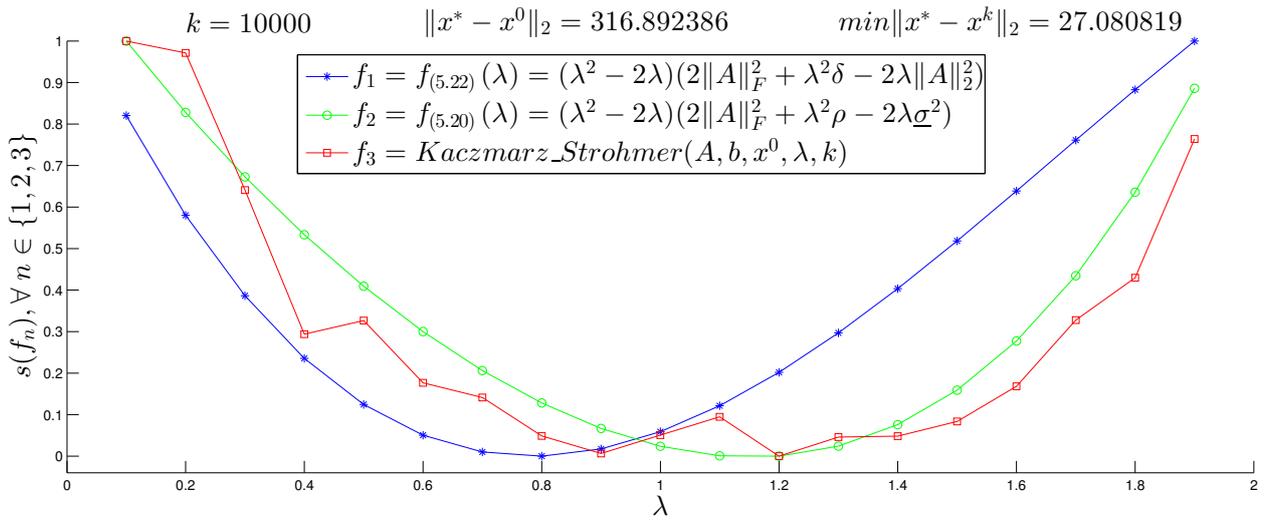
Partindo da Figura 6.2 e indo até a Figura 6.7 mostramos os resultados obtidos quando aplicamos o Método de Kaczmarz Aleatório neste problema e avaliamos o erro  $\|x^k - x^*\|_2$  (utilizando (6.2)) e os mínimos de nossas estimativas (5.20) e (5.22) para  $k = 10000$  e  $k = 15000$  em função da variação do parâmetro de relaxação, número de iterações e valores para  $x^0$  iniciais.



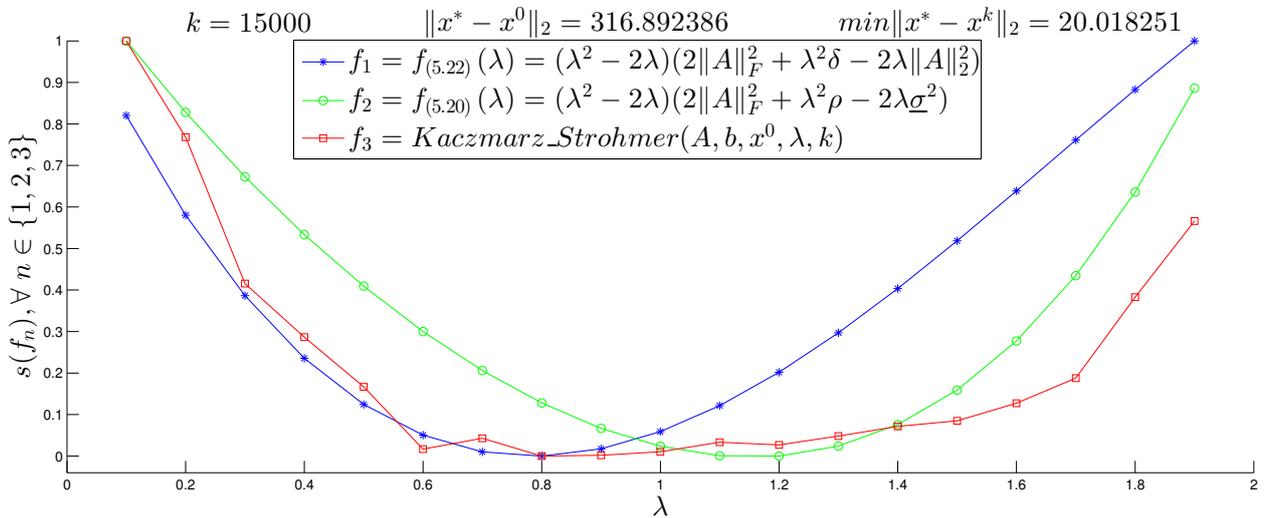
**Figura 6.2:** Figura gerada utilizando um chute inicial  $x^0 = 100$  e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro  $\|x^* - x^k\|_2$  ao final de 10000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de  $[0, 1]$  utilizando (6.2) a fim de observar em qual  $\lambda$  a função atingiu o seu mínimo e comprar com os mínimos obtidos minimizando as estimativas



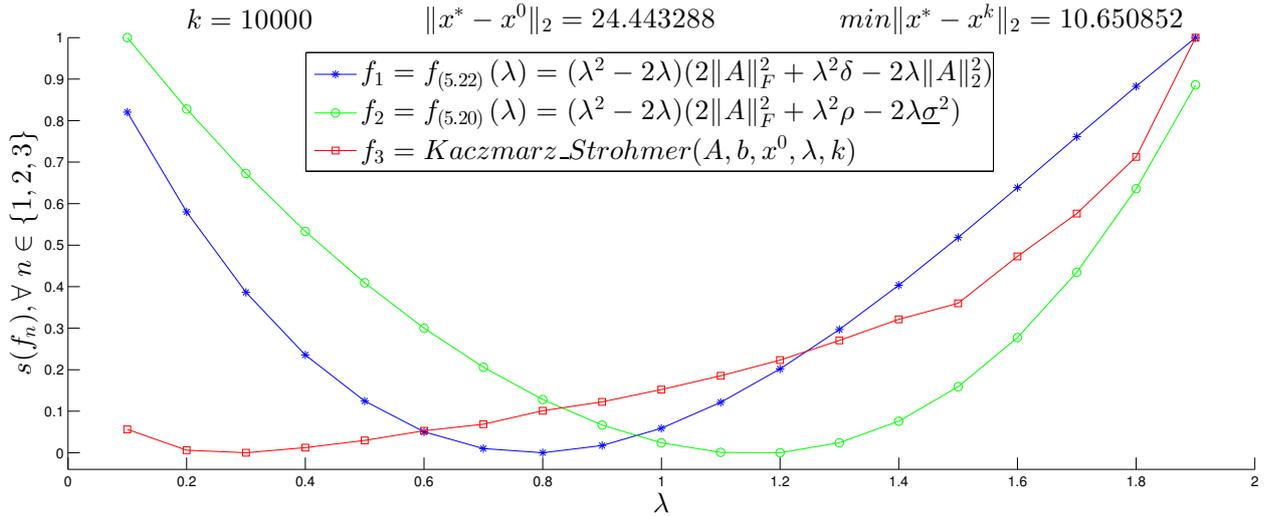
**Figura 6.3:** Figura gerada utilizando um chute inicial  $x^0 = 100$  e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro  $\|x^* - x^k\|_2$  ao final de 15000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de  $[0, 1]$  utilizando (6.2) a fim de observar em qual  $\lambda$  a função atingiu o seu mínimo e comprar com os mínimos obtidos minimizando as estimativas



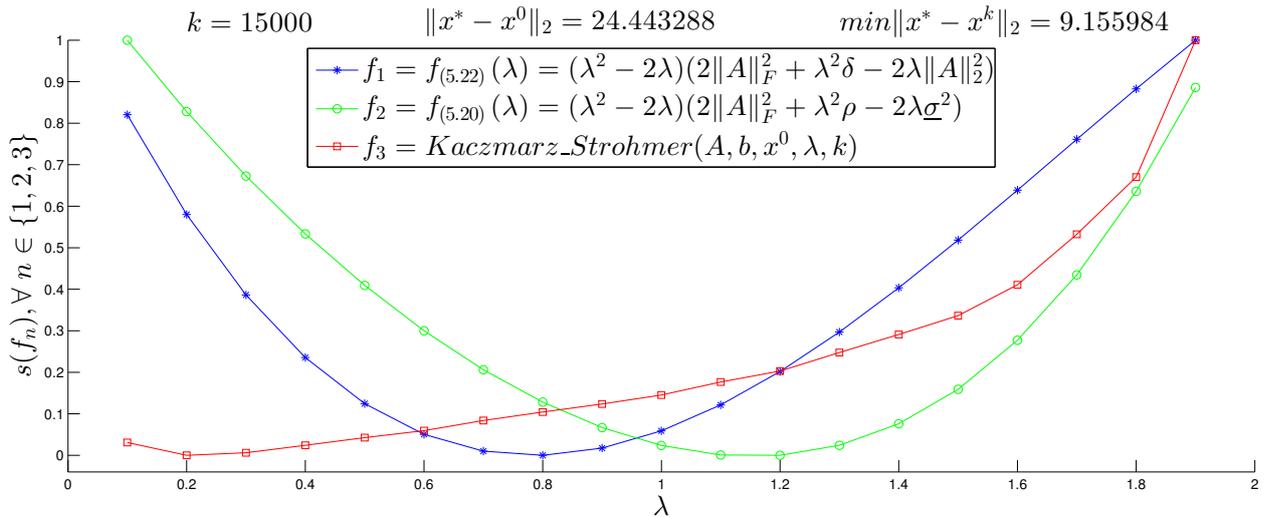
**Figura 6.4:** Figura gerada utilizando um chute inicial  $x^0 = 10$  e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro  $\|x^* - x^k\|_2$  ao final de 10000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de  $[0, 1]$  utilizando (6.2) a fim de observar em qual  $\lambda$  a função atingiu o seu mínimo e comprar com os mínimos obtidos minimizando as estimativas



**Figura 6.5:** Figura gerada utilizando um chute inicial  $x^0 = 10$  e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro  $\|x^* - x^k\|_2$  ao final de 15000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de  $[0, 1]$  utilizando (6.2) a fim de observar em qual  $\lambda$  a função atingiu o seu mínimo e comprar com os mínimos obtidos minimizando as estimativas



**Figura 6.6:** Figura gerada utilizando um chute inicial  $x^0 = 0$  e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro  $\|x^* - x^k\|_2$  ao final de 10000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de  $[0, 1]$  utilizando (6.2) a fim de observar em qual  $\lambda$  a função atingiu o seu mínimo e comparar com os mínimos obtidos minimizando as estimativas



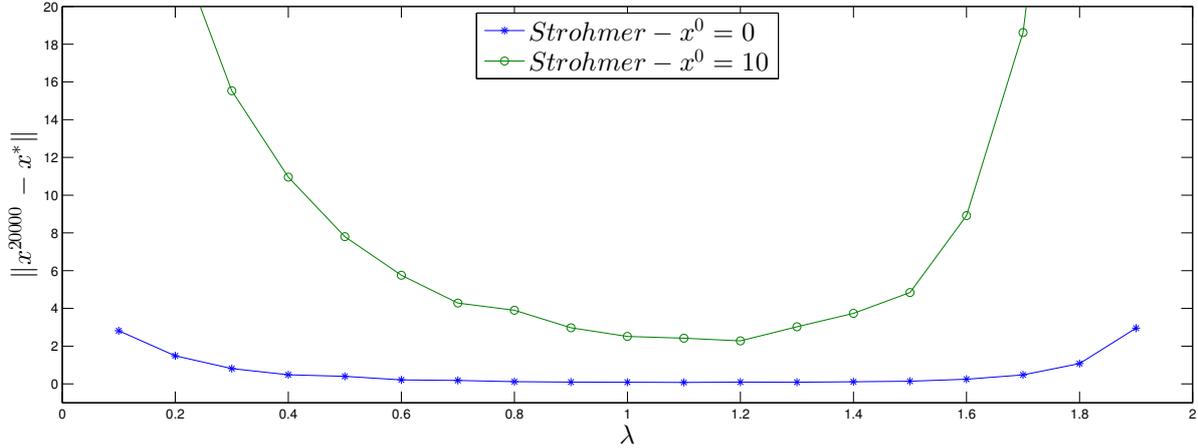
**Figura 6.7:** Figura gerada utilizando um chute inicial  $x^0 = 0$  e executando o Método de Kaczmarz Aleatório no problema (6.1) com o intuito de avaliar o erro  $\|x^* - x^k\|_2$  ao final de 15000 iterações e as estimativas (5.20) e (5.22). Utilizamos todas as funções na escala de  $[0, 1]$  utilizando (6.2) a fim de observar em qual  $\lambda$  a função atingiu o seu mínimo e comparar com os mínimos obtidos minimizando as estimativas

Podemos concluir que para este problema, se utilizarmos nossas estimativas, podemos obter valores muito próximos aos dos parâmetros de relaxação ótimos quando utilizamos os valores iniciais  $x^0 = 10$  e  $x^0 = 100$ . Mas, apesar da nossa estimativa obter valores melhores que do que o  $\lambda = 1$  quando utilizamos  $x^0 = 0$ , o seu valor está muito distante do ótimo. Os valores obtidos minimizando as estimativas foram  $\lambda = 1.1538$  por (5.20) e  $\lambda = 0.7848$  por (5.22).

Em geral, em todos os sistemas que estimamos os parâmetros, quando a distância entre o chute inicial e a solução do sistema  $\|x^0 - x^*\|_2$  for pequena, nossa sugestão é utilizar a estimativa (5.22) e, por outro lado, caso esta distância seja grande, utilizamos (5.20). Uma outra forma de avaliar

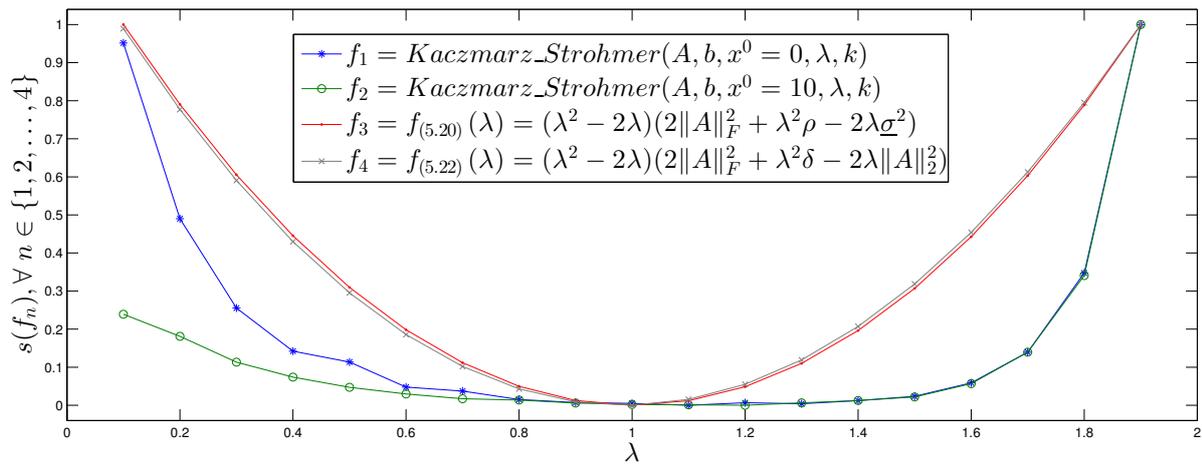
esta distância sem conhecer o valor de  $x^*$  é através do resíduo  $\|b - Ax^0\|_2$ , no entanto estas são perguntas que ficaram sem respostas e que são temas para pesquisas futuras.

Em uma segunda simulação utilizamos o Método de Kaczmarz de Strohmer para reconstruir um *phantom* Shepp-Logan de  $40 \times 40$  pixels utilizando um sinograma de  $256 \times 256$  pixels e obtemos a Figura 6.8 avaliando o erro da iteração  $k = 20000$  para chutes iniciais  $x = 0$  e  $x = 10$ . A matriz  $A \in \mathbb{R}^{m \times n}$  possui dimensões  $m = 57557$ ,  $n = 1600$ ,  $\|A\|_F^2 = 1.71 \times 10^6$ ,  $\underline{\sigma} = 19.56$ ,  $\bar{\sigma} = 2.09 \times 10^2$  e o número de condicionamento  $\kappa(A) = \bar{\sigma}/\underline{\sigma} = 10.68$ .



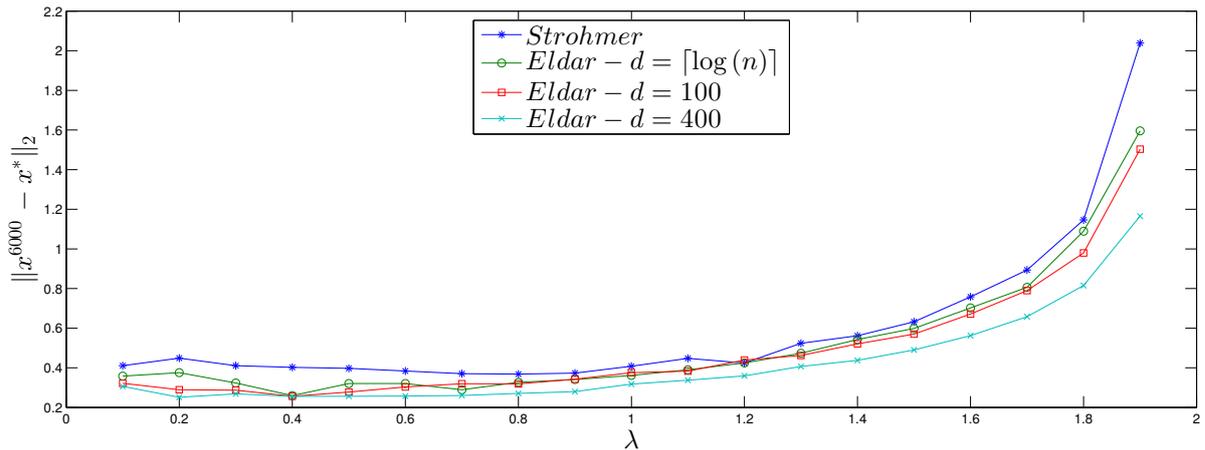
**Figura 6.8:** Comparação da taxa de convergência do Método Kaczmarz Aleatório de Strohmer e Vershynin em função da variação do parâmetro de relaxação na reconstrução do *phantom* de Shepp-Logan

Infelizmente, para esta matriz, calcular a estimativa não se provou ser uma prática promissora. Neste problema se já utilizássemos projeções exatas, não precisaríamos calcular o valor de  $\lambda$  através de nossas estimativas, como podemos ver no gráfico em escala de  $[0, 1]$  na Figura 6.9. Realizamos vários testes utilizando tanto *phantoms* de Shepp-Logan como sinogramas de tamanhos diferentes e constatamos que o valor do parâmetro de relaxação calculado por ambas estimativa tendem a estar próximos de 1. No entanto, os parâmetros estimados não fogem do ótimo.

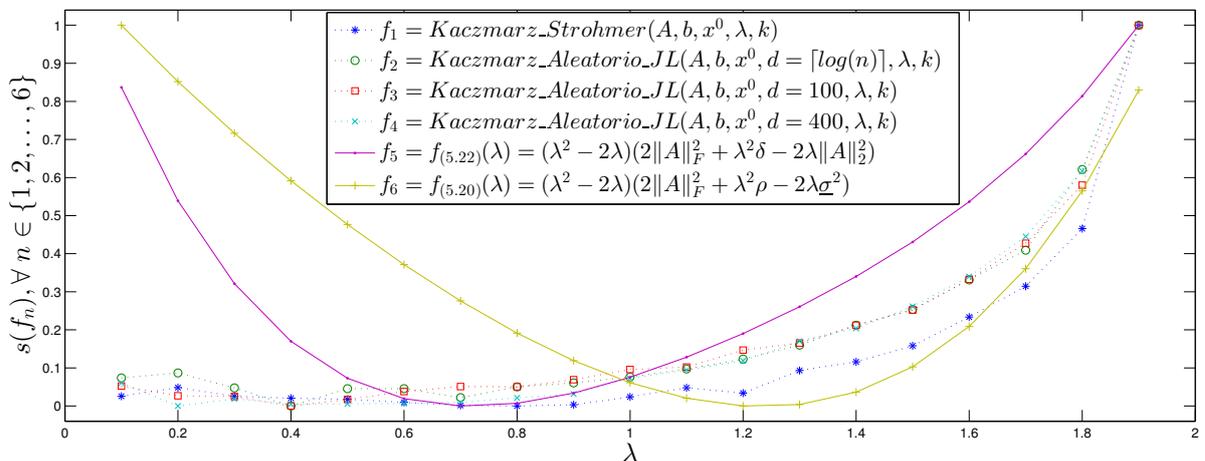


**Figura 6.9:** Valores de  $\|x^k - x^*\|$  apresentando na Figura 6.8 e Valores de (5.20) com  $\lambda \in (0, 2)$ . Ambos em escala  $[0, 1]$ .

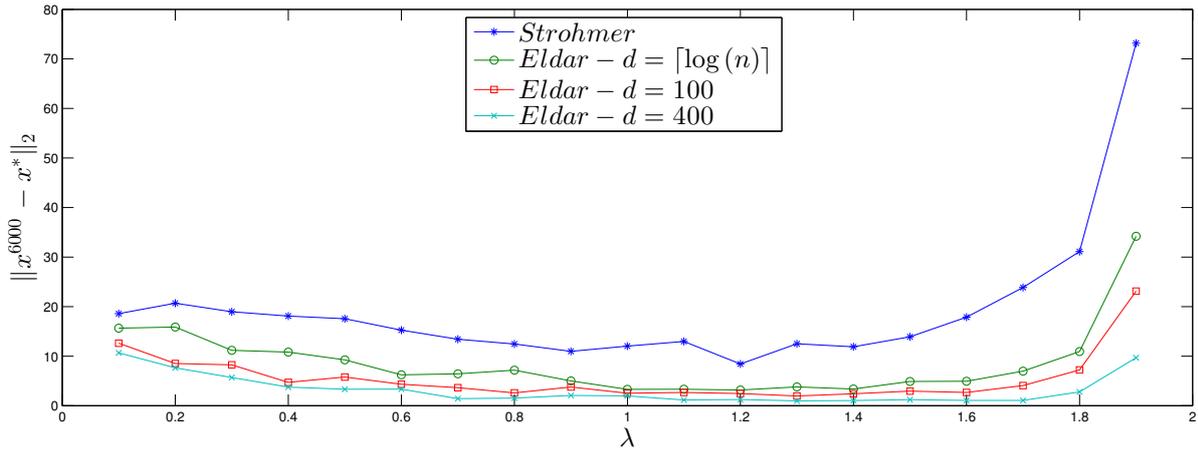
Agora utilizando o método de Kaczmarz de Strohmer e Vershynin e o de Eldar e Needell realizamos os mesmos testes anteriores para  $x = 1$  e  $x = 10$ , obtendo as Figuras 6.10, 6.11, 6.12 e 6.13. Onde utilizamos uma matriz  $A \in \mathbb{C}^{m \times n}$  e um vetor  $b \in \mathbb{C}^m$  de dimensões  $m = 5000$ ,  $n = 500$ ,  $\|A\|_F = 1.29 \times 10^3$ , menor valor singular  $\underline{\sigma} = 19.82$ , maior valor singular  $\bar{\sigma} = 1.12 \times 10^3$ , número de condicionamento  $\kappa(A) = \bar{\sigma}/\underline{\sigma} = 56.43$ , a parte real dos elementos está no intervalo  $[0, 1]$ , a parte imaginária dos elementos está no intervalo  $[0, i]$  e o número de linhas selecionadas para o cálculo da solução que mais se aproxima da solução do problema é 10. Nestes gráficos, as valores de  $d$  referem-se a redução de dimensionalidade, conforme discutido no Capítulo 4 Seção 4.3.



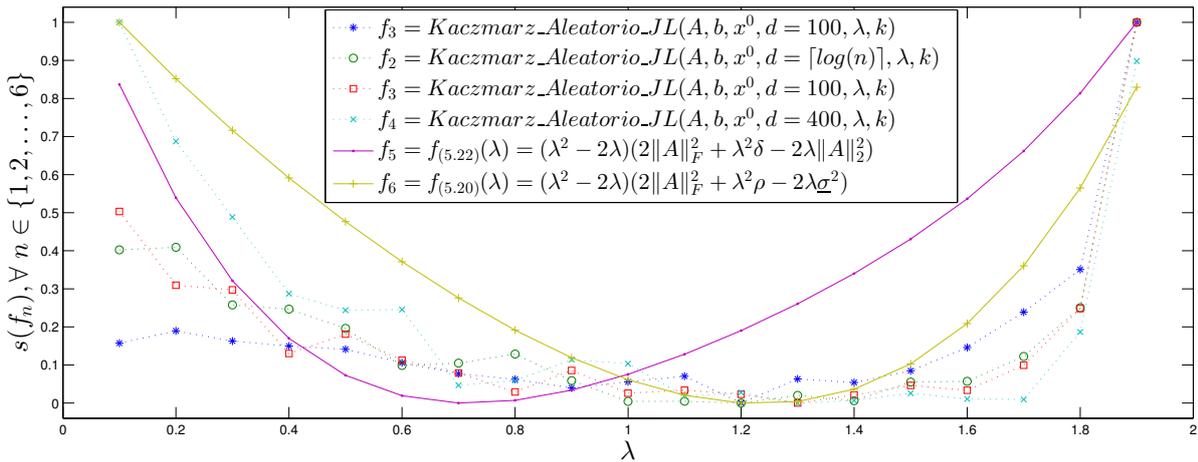
**Figura 6.10:** Comparação da taxa de convergência do Método de Kaczmarz Aleatório de Strohmer e Vershynin e o Método de Kaczmarz Aleatório Johnson-Lindenstrauss de Eldar e Needell em função da variação do parâmetro de relaxação para  $x^0 = 1$



**Figura 6.11:** Valores de  $\|x^k - x^*\|$  apresentando na Figura 6.10, valores de (5.20) e valores de (5.22) com  $\lambda \in (0, 2)$ . Todos em escala  $[0, 1]$



**Figura 6.12:** Comparação da taxa de convergência do Método de Kaczmarz Aleatório de Strohmer e Vershynin e o Método de Kaczmarz Aleatório Johnson-Lindenstrauss de Eldar e Needell em função da variação do parâmetro de relaxação para  $x^0 = 10$



**Figura 6.13:** Valores de  $\|x^k - x^*\|$  apresentando na Figura 6.12, valores de (5.20) e valores de (5.22) com  $\lambda \in (0, 2)$ . Todos em escala  $[0, 1]$ .

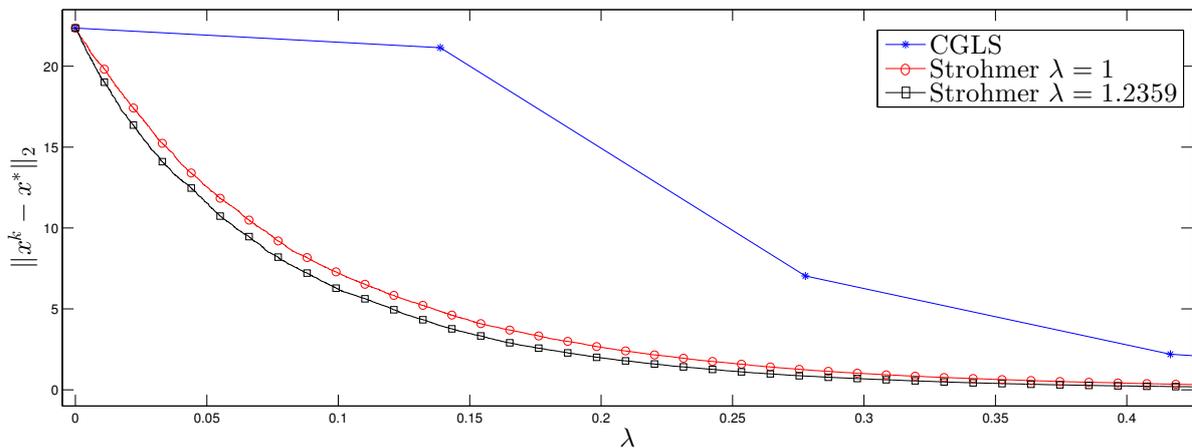
Como anteriormente, o  $\lambda_1 = 0.7195$  calculado pela estimativa (5.22) se comportou de maneira melhor quando a distância entre o cuto inicial e a solução do sistema são pequenas e  $\lambda_2 = 1.2359$  por (5.20) caso contrário. Além disto, os resultados exibidos pela Figura 6.11 e 6.13 mostra que quando variamos o valor de  $\lambda$  o método de Kaczmarz de Strohmer e Vershynin se comporta de maneira diferente do que o método de Eldar e Needell.

O motivo disto acontecer é que, apesar da escolha dos hiperplanos ser realizada através da probabilidade de Strohmer e Vershynin, durante a execução do algoritmo de Eldar e Needell, não um, mas vários hiperplanos são selecionados a fim de realizar o cálculo do passo que mais avança em relação a solução do problema (em um espaço de dimensões reduzidas). Realizar esta seleção dentro deste conjunto altera a probabilidade de escolha das linhas e consequentemente nossa estimativa fica baseada em valores incorretos.

## 6.2 Comparação do Algoritmo CGLS e o Algoritmo Aleatório de Strohmer

Utilizando o método de Kaczmarz aleatório de Strohmer e Vershynin e o dos Gradientes Conjugados para Equações Lineares não Quadradas (CGLS) na mesma matriz usada para gerar as Figuras 6.10, 6.11, 6.12 e 6.13 (resultados da simulação no algoritmo de Eldar e Needell), porém modificando o vetor  $b = 0$  para tornar o sistema homogêneo e calcularmos a distância entre o chute inicial  $x^0 = 1$  e a solução do sistema  $x^* = 0$  de forma mais simples. Assim, temos a matriz  $A \in \mathbb{C}^{m \times n}$  que possui dimensões  $m = 5000$ ,  $n = 500$ ,  $\|A\|_F = 1.29 \times 10^3$ , menor valor singular  $\underline{\sigma} = 19.82$ , maior valor singular  $\bar{\sigma} = 1.12 \times 10^3$ , número de condicionamento  $\kappa(A) = \bar{\sigma}/\underline{\sigma} = 56.43$ , a parte real dos elementos está no intervalo  $[0, 1]$ , a parte imaginária dos elementos está no intervalo  $[0, i]$ .

Executamos ambos os algoritmos e analisamos o tempo de convergência do método através do erro  $\|x^k - x^*\|_2$  utilizando, no método de Strohmer e Vershynin, projeções com o valor do parâmetro igual a  $\lambda = 1$  e também com valor calculado pela nossa estimativa (5.20) (que neste caso se comportou de melhor forma)  $\lambda = 1.2358$  (Figura 6.14).



**Figura 6.14:** Comparação da velocidade de convergência do algoritmo Kaczmarz aleatório de Strohmer e o método dos Gradientes Conjugados para equações lineares não simétricas.

Analisando a Figura 6.14 vemos que apesar de reduzir bastante o erro em cada iteração o algoritmo CGLS se mostrou bem mais custoso do que o método de Kaczmarz de Strohmer e Vershynin. Além disso, vemos o ganho na velocidade de convergência utilizando a estimativa calculada minimizando (5.20).

O Método dos Gradientes Conjugados para Matrizes não Quadradas que utilizamos para esta comparação de desempenho é descrito por Saunders em [76] e está descrito na próxima página.

---

**- Método dos Gradientes Conjugados para Matrizes não Simétricas**

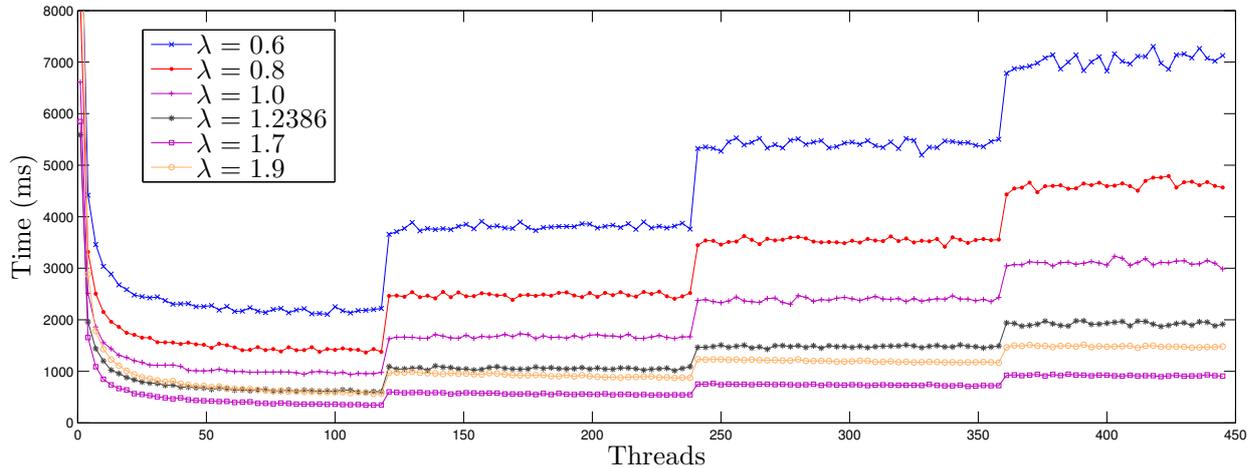
---

```
1: function CGLS( $A, x_0, b, k, max\_iteracoes, tolerancia$ )
2:    $r = b - A \times x$ ;
3:    $s = A^t \times r$ ;
4:    $p = s$ ;
5:    $norm_{s_0} = \|s\|_2$ ;
6:    $gamma = (norm_{s_0})^2$ ;
7:    $norm_x = \|x\|_2$ ;
8:    $flag = 0$ ;
9:    $k = 0$ ;
10:  while  $k < max\_iteracoes$  and  $flag == 0$  do
11:     $q = A \times p$ ;
12:     $delta = \|q\|_2^2$ ;
13:    if  $delta == 0$  then
14:       $delta = eps$ ;  $\triangleright$  O menor número que pode ser adicionado a 0 e ser considerado
      diferente de 0.
15:       $alpha = gamma/delta$ ;
16:       $x = x + alpha \times p$ ;
17:       $r = r - alpha \times q$ ;
18:       $s = A^t \times r$ ;
19:       $norm_s = \|s\|_2$ ;
20:       $gamma_1 = gamma$ ;
21:       $gamma = (norm_s)^2$ ;
22:       $beta = gamma/gamma_1$ ;
23:       $p = s + beta \times p$ ;
24:       $norm_x = \|x\|_2$ ;
25:      if  $norm_s \leq norm_{s_0} \times tolerancia$  or  $norm_x \times tolerancia \geq 1$  then
26:         $flag = 1$ ;
27:       $k = k + 1$ ;
return  $x$ 
```

---

## 6.3 Algoritmo Paralelo

A Figura 6.15 mostra o resultado do tempo de execução do Algoritmo Kaczmarz Paralelo apresentado no Capítulo 4 Seção 4.4 para diferentes quantidades de threads e valores de  $\lambda$ . Essa simulação feita em uma máquina Intel Core i7 X 990 @ 3.47GHz, 250GB RAM e GPU Tesla C2050/C2070, mostra que existe um gargalo ao se utilizar uma quantidade maior que 120 *threads* e, aumentando este número, diminuimos a velocidade de convergência.



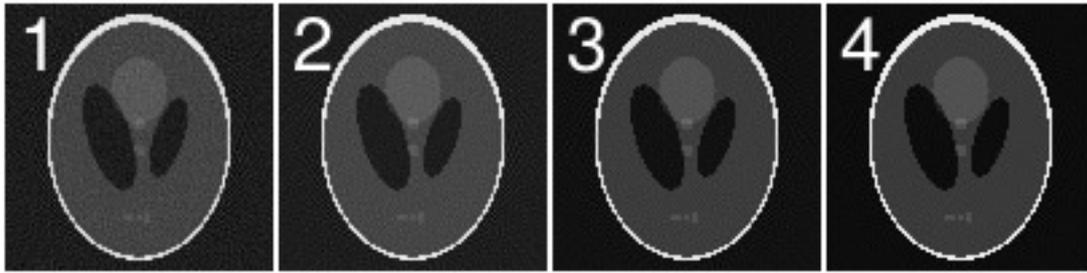
**Figura 6.15:** Tempo gasto para convergência do algoritmo Kaczmarz Paralelo com critério de parada  $\|x^k - x^*\| < 10^{-2}$  executado na máquina descrita no texto.

A matriz  $A \in \mathbb{R}^{m \times n}$  usada para fazer a simulação apresentada na Figura 6.15 possui dimensões  $m = 5000$ ,  $n = 500$ , os valores de seus elementos reais foram gerados através de uma função geradora de números pseudo-aleatórios uniformemente distribuídos dentro do intervalo  $[0, 1]$  e utilizamos um chute inicial  $x^0 = 100$ . Minimizando (5.20) em função de  $\lambda$ , obtemos o parâmetro de relaxação  $\lambda = 1.2386$ .

Analisando a Figura 6.15 vemos que quando a quantidade de *threads* ultrapassa o valor de 120, o algoritmo começa a convergir de forma mais lenta, e o mesmo acontecimento se repete com 240 e 360 *threads*. Isto está relacionado com a quantidade de *threads* simultâneas que a GPU consegue processar. Esta GPU pode processar até 448 *threads* simultâneas, mas não com tanta eficiência que trabalha com um pouco mais que  $\frac{1}{4}$  delas.

Como discutimos anteriormente, um dos motivos para existirem tantos valores de  $\lambda$  diferentes que fazem o algoritmo convergir mais rápido do que o calculado por nossa estimativa é que apesar da escolha dos hiperplanos ser realizada através da probabilidade de Strohmer e Vershynin, durante a execução do algoritmo paralelo vários hiperplanos são selecionados a fim de realizar o cálculo do passo que mais avança em relação a solução do problema em cada *thread*. Após calcular e selecionar do hiperplano que maximiza  $\|x^{k+1} - x^k\|_2$  dentro deste conjunto, sua probabilidade é alterada e consequentemente nossa estimativa fica baseada em valores incorretos.

A Figura 6.16 mostra o resultado do tempo de execução do algoritmo Kaczmarz paralelo para reconstruir um *phantom* de Shepp-Logan de  $100 \times 100$  *pixels*.



**Figura 6.16:** A primeira reconstrução foi realizada com 5021 projeções e levou 8.66 segundos para ser concluída. A segunda 10042 e 19.36 segundos. A terceira 15063 projeções e 29.97 segundos. E a quarta 20084 projeções e 40.67 segundos. Todas utilizando 120 *threads*

A matriz usada para realizar as reconstruções da Figura 6.16 é uma matriz  $A \in \mathbb{R}^{m \times n}$ , possui dimensões  $m = 20084$ ,  $n = 10000$  (não foram armazenadas linhas nulas), norma de Frobenius  $\|A\|_F = 1.22 \times 10^3$ , menor valor singular  $\underline{\sigma} = 0.15$ , maior valor singular  $\bar{\sigma} = 1.23 \times 10^2$ , número de condicionamento  $\kappa(A) = \bar{\sigma}/\underline{\sigma} = 8.43 \times 10^2$  e valor inicial de  $x^0 = 0$ . Para esta matriz o valor de  $\lambda$  calculado pela estimativa 5.22 é 0.994896513806671 e pela estimativa 5.20 é 1.000220844201059, porém utilizamos  $\lambda = 1$  para realizar estas reconstruções.

## Considerações Finais

---

Neste trabalho encontramos estimativas para o parâmetro de relaxação e efetuar uma efetiva implementação paralela do Método de Kaczmarz. Em nossas simulações numéricas, mostramos que o Método de Kaczmarz Aleatório de Strohmer e Vershynin converge de forma mais rápida com o valor de parâmetro escolhido após minimizar (5.20) ou (5.22), isto é, caso comparemos com o mesmo algoritmo utilizando o valor de parâmetro  $\lambda = 1$ . Nossa escolha entre utilizar (5.20) ou (5.22) está baseada na distância entre o valor de inicial e a solução do sistema  $\|x^0 - x^*\|$  ou no resíduo  $\|b - Ax^0\|_2$  quando  $x^*$  não é conhecido. Entretanto ainda não conseguimos definir a noção de grande ou pequeno para  $\|Ax^0 - b\|_2$  ou  $\|x^0 - x^*\|_2$  a fim de selecionar qual estimativa deve ser usada.

Como estas estimativas encontradas foram baseadas na probabilidade da seleção das linhas do Método de Kaczmarz Aleatório de Strohmer e Vershynin. Elas podem não obter valores tão precisos de  $\lambda$  a fim de serem utilizadas em nosso algoritmo paralelo ou no Método de Kaczmarz de Eldar e Needel, uma vez que eles alteram a probabilidade de seleção das linhas.

Em trabalhos futuros o que pode ser feito é uma análise destas estimativas utilizando mais algumas aproximações. Uma vez que o valor de  $\max_{i=1\dots m} \frac{\|Aa_i\|_2^2}{\|a_i\|_2^2}$  e o valor de  $\min_{i=1\dots m} \frac{\|Aa_i\|_2^2}{\|a_i\|_2^2}$  são muito custosos para serem calculados, podemos aproximá-los. No caso deste máximo, podemos utilizar o índice da linha da matriz  $A$  que maximiza  $\|a_i\|_2^2$  e, no caso do mínimo, podemos utilizar o índice da linha que minimiza  $\|a_i\|_2^2$  (uma vez que este processo já é requerido para o cálculo da probabilidade da seleção de linhas).

Além disso, o que pode ser feito a respeito da escolha do parâmetro de relaxação no algoritmo paralelo é tentar encontrar uma nova estimativa baseada em uma probabilidade que se encaixa em sua estratégia de seleção de linhas. E ainda, precisamos de uma forma eficiente de calcular ou estimar o menor valor singular para estas matrizes sobredeterminadas e fundamentar a maneira de selecionar qual estimativa ((5.20) ou (5.22)) deve ser usada para melhorar a velocidade de convergência.



---

# Referências Bibliográficas

---

---

- [1] AILON, N.; CHAZELLE, B. The fast johnson-lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, v. 39, n. 1, p. 302–322, 2009.
- [2] AILON, N.; LIBERTY, E. An almost optimal unrestricted fast johnson-lindenstrauss transform. *ACM Transactions on Algorithms (TALG)*, v. 9, n. 3, p. 21, 2013.
- [3] BAUSCHKE, H. H.; BORWEIN, J. M. On projection algorithms for solving convex feasibility problems. *SIAM Rev.*, v. 38, n. 3, p. 367–426, 1996.
- [4] BECKMANN, E. C. CT scanning the early days. *British Journal of Radiology*, v. 79, n. 937, p. 5–8, 2006.
- [5] BEER, A. Bestimmung der absorption des rothen lichts in farbigen flüssigkeiten. *Annalen der Physik und Chemie*, v. 86, p. 78–88, 1852.
- [6] BEHRENS, R. T.; SCHARF, L. L. Signal processing applications of oblique projection operators. *IEEE Transactions on Signal Processing*, p. 1413–1424, 1994.
- [7] BENEDETTO, J. J.; FERREIRA, P. J. *Modern sampling theory: mathematics and applications*. Springer, 2001.
- [8] BEYLKIN, G. Discrete radon transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 35, n. 2, p. 162–172, 1987.
- [9] BITELLI, T. *Física e Dosimetria das Radiações*. 2nd ed. São Paulo: Editora Atheneu, 2006.
- [10] BOLZ, J.; FARMER, I.; GRINSPUN, E.; SCHRÖÖDER, P. Sparse matrix solvers on the gpu: conjugate gradients and multigrid. *ACM Trans. Graph.*, v. 22, n. 3, p. 917–924, 2003.
- [11] BRACEWELL, R. *The Fourier Transform and Its Applications*. 3 ed. McGraw-Hill Science/Engineering/Math, 1999.
- [12] BREZINSKI, C. Projection methods for linear systems. *Journal of Computational and Applied Mathematics*, v. 77, n. 1–2, p. 35 – 51, 1997.
- [13] BROWN, D. L.; CORTEZ, R.; MINION, M. L. Accurate projection methods for the incompressible navier-stokes equations. *J. Comput. Phys*, v. 168, p. 464–499, 2001.
- [14] BURDEN, R. L.; FAIRES, D. J. *Numerical Analysis*. 7 ed. Brooks Cole, 2000.

- [15] BUZUG, T. M. *Computed Tomography From Photon Statistics to Modern Cone-Beam CT*. 1–490 p., 2008.
- [16] BYRNE, C. A unified treatment of some iterative algorithms in signal processing and image reconstruction. *Inverse Problems*, v. 20, n. 1, p. 103, 2004.
- [17] CARSON, R. E.; LANGE, K. A Statistical Model for Position Emission Tomography: The EM Parametric Image Reconstruction Algorithm. *Journal of the American Statistical Association* *11*, v. 80, n. 389, p. 20+, 1985.
- [18] CARVALHO, A. C. P. História da tomografia computadorizada. *Revista da Imagem, São Paulo*, v. 29, p. 61–66, 2007.
- [19] CENSOR, Y.; EGGERMONT, P.; GORDON, D. Strong underrelaxation in kaczmarz’s method for inconsistent systems. *Numerische Mathematik*, v. 41, p. 83–92, 1983.
- [20] CENSOR, Y.; ZENIOS, S. A. *Parallel optimization: Theory, algorithms, and applications*. Oxford University Press, USA, 1997.
- [21] CHENEY, E.; KINCAID, D. *Numerical mathematics and computing*. Cengage Learning, 2012.
- [22] DASGUPTA, S.; GUPTA, A. An elementary proof of the Johnson-Lindenstrauss lemma. 1999.
- [23] DE, C.; HALTMEIER, M.; LEITAO, A.; SCHERZER, O. On steepest-descent-Kaczmarz methods for regularizing systems of nonlinear ill-posed equations. *Appl. Math. Comput.*, v. 202, n. 2, p. 596–607, 2008.
- [24] DEFRISE, M.; NOO, F.; CLACKDOYLE, R.; KUDO, H. Truncated hilbert transform and image reconstruction from limited tomographic data. *Inverse Problems*, v. 22, n. 3, p. 1037, 2006.
- [25] DEMMEL, J. W. The probability that a numerical analysis problem is difficult. *Mathematics of Computation*, v. 50, n. 182, p. 449–480, 1988.
- [26] DEUTSCH, F.; HUNDAL, H. The rate of convergence for the method of alternating projections, ii. *Journal of Mathematical Analysis and Applications*, v. 205, n. 2, p. 381 – 405, 1997.
- [27] EGGERMONT, P.; HERMAN, G.; LENT, A. Iterative algorithms for large partitioned linear systems, with applications to image reconstruction. *Linear Algebra and its Applications*, v. 40, n. 0, p. 37 – 67, 1981.
- [28] ELBLE, J. M.; SAHINIDIS, N. V.; VOUZIS, P. Gpu computing with kaczmarz’s and other iterative algorithms for linear systems. *Parallel Comput.*, v. 36, n. 5-6, p. 215–231, 2010.
- [29] ELДАР, Y. C.; NEEDELL, D. Acceleration of randomized kaczmarz method via the johnson—lindenstrauss lemma. *Numer. Algorithms*, v. 58, n. 2, p. 163–177, 2011.
- [30] EPSTEIN, C. L. *Introduction to the mathematics of medical imaging*. 2 ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2007.

- [31] FEICHTINGER, H. G.; CENKER, C.; MAYER, M.; STEIER, H.; STROHMER, T. New variants of the pocs method using affine subspaces of finite codimension with applications to irregular sampling, p. 299–310. 1992.
- [32] FEICHTINGER, H. G.; GRÖCHENIG, K. Theory and practice of irregular sampling. In: BENEDETTO, J.; FRAZIER, M., eds. *Wavelets: Mathematics and Applications*, Studies in Advanced Mathematics, NuHAG;Classical, CRC Press, 1994, p. 305–363 (*Studies in Advanced Mathematics*, ).
- [33] FRANCO, N. B. *Cálculo Numérico*. 1 ed. São Paulo: Pearson Prentice Hall, 2006.
- [34] FURQUIM, T.; NERSISSIAN, D. Y. Estudos de otimização de dose e qualidade de imagem em processos de transição tecnológica em mamografia. *Revista Brasileira de Física Médica*, v. 4, n. 3, p. 11–14, 2011.
- [35] GALÁNTAI, A. On the rate of convergence of the alternating projection method in finite dimensional spaces. *Journal of Mathematical Analysis and Applications*, v. 310, n. 1, p. 30 – 44, 2005.
- [36] GALOPPO, N.; GOVINDARAJU, N. K.; HENSON, M.; MANOCHA, D. Lu-gpu: Efficient algorithms for solving dense linear systems on graphics hardware. In: *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, SC '05, Washington, DC, USA: IEEE Computer Society, 2005, p. 3– (SC '05, ).
- [37] GÖDDEKE, D.; STRZODKA, R.; TUREK, S. Performance and accuracy of hardware-oriented native-, emulated-and mixed-precision solvers in fem simulations. *Int. J. Parallel Emerg. Distrib. Syst.*, v. 22, n. 4, p. 221–256, 2007.
- [38] GOLUB, G. H.; VAN LOAN, C. F. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*. 3rd ed. The Johns Hopkins University Press, 1996.
- [39] GRÖCHENIG, K. Irregular sampling, toeplitz matrices, and the approximation of entire functions of exponential type. *Math. Comp.*, v. 68, p. 749–765, 1999.
- [40] HALTMEIER, M.; LEITÃO, A.; RESMERITA, E. On regularization methods of em-kaczmarz type. *Inverse Problems*, v. 25, n. 7, p. 075008, 2009.
- [41] HAMMOND, C. *The basics of crystallography and diffraction*. 3rd ed. International Union of Crystallography : Oxford University Press, 2009.
- [42] HANKE, M.; NIETHAMMER, W. On the acceleration of kaczmarz’s method for inconsistent linear systems. *Linear Algebra and its Applications*, v. 130, n. 0, p. 83 – 98, 1990.
- [43] HERMAN, G.; MEYER, L. Algebraic reconstruction techniques can be made computationally efficient [positron emission tomography application]. *Medical Imaging, IEEE Transactions on*, v. 12, n. 3, p. 600–609, 1993.
- [44] HERMAN, G. T. *Image reconstruction from projections: the fundamentals of computerized tomography*. 2nd ed. Springer, 2009.
- [45] HERMAN, G. T.; LENT, A.; LUTZ, P. H. Relaxation methods for image reconstruction. *Commun. ACM*, v. 21, n. 2, p. 152–158, 1978.

- [46] IUSEM, A. N.; PIERRO, A. R. Fundamentos matemáticos da tomografia computadorizada: Métodos de expansão em séries. *Revista Matemática Universitária*, v. 11, p. 53–65, 1990.
- [47] JOHNSON, W.; LINDENSTRAUSS, J. Extensions of Lipschitz mappings into a Hilbert space. In: *Conference in modern analysis and probability (New Haven, Conn., 1982)*, v. 26 de *Contemporary Mathematics*, American Mathematical Society, p. 189–206, 1984.
- [48] KACZMARZ, S. Angenäherte Auflösung von Systemen linearer Gleichungen. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, v. 35, p. 355–357, 1937.
- [49] KAK, A. C.; SLANEY, M. *Principles of Computerized Tomographic Imaging*. New York: IEEE Press, 1988.
- [50] KALENDER, W. A. *Computed Tomography: Fundamentals, System Technology, Image Quality, Applications*. Wiley-VCH, 2006.
- [51] KAUFMAN, L. Implementing and accelerating the em algorithm for positron emission tomography. *IEEE transactions on medical imaging*, v. 6, n. 1, p. 37–51, 1987.  
Disponível em: <http://dx.doi.org/10.1109/tmi.1987.4307796>
- [52] KRÜGER, J.; WESTERMANN, R. Linear algebra operators for gpu implementation of numerical algorithms. *ACM Trans. Graph.*, v. 22, n. 3, p. 908–916, 2003.
- [53] LAMBERT, J. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Germany: Eberhardt Klett, 1760.
- [54] LEWITT, R. M. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Physics in Medicine and Biology*, v. 37, n. 3, p. 705, 1992.  
Disponível em: <http://stacks.iop.org/0031-9155/37/i=3/a=015>
- [55] MAHESH, M.; HEVEZI, J. M. Slice wars vs dose wars in multiple-row detector ct. *J Am Coll Radiol*, v. 6, n. 3, 2009.
- [56] MCROBBIE, D. W. Radiation exposure in computed tomography: Fundamentals, influencing parameters, dose assessment, optimisation, scanner data, terminology. *Physics in Medicine and Biology*, v. 46, n. 5, p. 1593, 2001.
- [57] MINION, M. L. Semi-implicit projection methods for incompressible flow based on spectral deferred corrections. *Applied Numerical Mathematics*, v. 48, n. 3-4, p. 369–387, 2004.
- [58] MOON, T. K.; STIRLING, W. C. *Mathematical Methods and Algorithms for Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 2000.
- [59] NATTERER, F. *The mathematics of computerized tomography*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001.
- [60] NATTERER, F.; WUBBELING, F. *Mathematical methods in image reconstruction*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001.
- [61] NÓBREGA, A. I. *Manual de Tomografia Computadorizada*. Editora Atheneu/Centro Universitário São Camilo, 2005.
- [62] NEEDELL, D. Randomized kaczmarz solver for noisy linear systems. *BIT Numerical Mathematics*, v. 50, p. 395–403, 2010.

- [63] NETO, E. S. H. *Algoritmos incrementais com aplicações em tomografia computadorizada*. Doutorado, Universidade Estadual de Campinas, 2009.
- [64] PAN, X.; SIDKY, E. Y.; VANNIER, M. Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction? *Inverse Problems*, v. 25, n. 12, p. 123009+, 2009.
- [65] PASCHOAL, C. M. M. *Caracterização de dispositivos eletrônicos para dosimetria em tomografia computadorizada*. Doutorado, Universidade Federal de Sergipe, 2012.
- [66] PHELPS, M. E. *PET: physics, instrumentation, and scanners*. Springer, 2006.
- [67] PIERRO, A. R. Fundamentos Matemáticos da Tomografia Computadorizada: Métodos de Transformação. *Revista Matemática Universitária*, v. 11, p. 53–65, 1990.
- [68] PIERRO, A. R.; HELOU NETO, E. S. From convex feasibility to convex constrained optimization using block action projection methods and underrelaxation. *International Transactions in Operational Research*, v. 16, n. 4, p. 495–504, 2008.
- [69] PIERRO, A. R. D. Problemas matemáticos em tomografia por emissão. Notas de Aula, 2003.
- [70] PODGORSK, E. B. *Radiation Oncology Physics: A Handbook for Teachers And Students*. International Atomic Energy Agency (IAEA), 2005.
- [71] RADON, J. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Akad. Wiss.*, v. 69, p. 262–277, 1917.
- [72] REHANI, M. M. Radiation protection in newer imaging technologies. *Radiation Protection Dosimetry*, v. 139, n. 1-3, p. 357–362, 2010.
- [73] REZVANI, N. *Iterative reconstruction algorithms for polyenergetic x-ray computerized tomography*. Doutorado, University of Toronto, 2012.
- [74] ROBILOTTA, C. C. A tomografia por emissão de pósitrons: uma nova modalidade na medicina nuclear brasileira. *Revista Panam Salud Publica*, v. 20, n. 2/3, p. 134–142, 2006.
- [75] SAAD, Y. *Iterative methods for sparse linear systems*. 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003.
- [76] SAUNDERS, M.; HANSEN, P. C.; BLEICHRODT, F. CGLS: CG method for  $Ax = b$  and Least Squares. [On-line].  
Disponível em: <http://www.stanford.edu/group/SOL/software/cgls/>
- [77] SEERAM, E. *Computed Tomography: Physical Principles, Clinical Applications, and Quality Control*. Saunders, 2001.
- [78] SEZAN, K. M.; STARK, H. Applications of convex projection theory to image recovery in tomography and related areas, p. 415–462. 1987.
- [79] SHEPP, L.; LOGAN, B. F. The fourier reconstruction of a head section. *IEEE Transactions on Nuclear Science*, v. NS21, n. 3, p. 21–43, 1974.

- [80] SHEPP, L. A.; VARDI, Y. Maximum Likelihood Reconstruction for Emission Tomography. *Medical Imaging, IEEE Transactions on*, v. 1, n. 2, p. 113–122, 1982.
- [81] SIDDON, R. L. Fast calculation of the exact radiological path for a three-dimensional CT array. *Medical Physics*, v. 12, p. 252–255, 1985.
- [82] SIDKY, E. Y.; XIAOCHUAN, P. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine and Biology*, v. 53, n. 17, p. 47–77, 2008.
- [83] SONKA, M.; FITZPATRICK, M. J., eds. *Handbook of Medical Imaging Volume 2. Medical Image Processing and Analysis*. Bellingham, Washington: SPIE Press, 2000.
- [84] STROHMER, T.; VERSHYNIN, R. A randomized solver for linear systems with exponential convergence. In: DÍAZ, J.; JANSEN, K.; ROLIM, J.; ZWICK, U., eds. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, v. 4110 de *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 499–507, 2006.
- [85] STROHMER, T.; VERSHYNIN, R. A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.*, v. 15, n. 2, p. 262–278, 2009.
- [86] STROHMER, T.; VERSHYNIN, R. Comments on the randomized kaczmarz method. *Journal of Fourier Analysis and Applications*, v. 15, p. 437–440, 2009.
- [87] SUETENS, P. *Fundamentals of medical imaging*. Cambridge University Press, 2009.