

UNIVERSIDADE FEDERAL DO PARANÁ

HELTON WU

MATHEUS BORGES BORBA DOS SANTOS

VINICIUS DE OLIVEIRA DIAS

Restauração de Imagens

Curitiba, PR

2025

HELTON WU
MATHEUS BORGES BORBA DOS SANTOS
VINICIUS DE OLIVEIRA DIAS

Restauração de Imagens

Trabalho apresentado à Universidade Federal do Paraná, à disciplina CMM204 MAT 1 – Programação Não Linear, como requisito para o desenvolvimento do projeto de Resolução de imagens via otimização não linear e análise do desempenho.

Universidade Federal do Paraná (UFPR)
Departamento de Matemática
Graduação em Matemática Industrial

Orientador: Prof(a). Dr(a). Evelin Heringer Manoel Krulikovski

Curitiba, PR
2025

SUMÁRIO

1	INTRODUÇÃO	3
2	METODOLOGIA	4
2.1	Formulação do Problema	4
2.2	Demonstração da Convexidade	4
2.2.1	Demonstração	4
2.3	Preparação da Imagem	5
2.4	Modelagem da Degradação	6
2.5	Interpretação da Convolução	6
2.6	Métodos de Otimização Implementados	8
2.6.1	Gradiente Descendente	8
2.6.2	Gradientes Conjugados (CGNE)	8
2.7	Quase-Newton L-BFGS	9
2.8	Métricas de Avaliação	9
2.9	Configuração Experimental	10
3	RESULTADOS E DISCUSSÃO	11
3.1	Análise dos Resultados Obtidos	11
3.2	Visualização de resultados	12
3.3	Implicações Práticas	14
4	CONCLUSÃO	16
	REFERÊNCIAS	18

1 INTRODUÇÃO

O problema de restauração de imagens desempenha um papel central em diversas áreas do processamento digital, especialmente quando a imagem adquirida passa por algum tipo de degradação decorrente de limitações físicas, ópticas ou mecânicas do processo de formação. Entre os tipos mais comuns de deterioração encontra-se o *borramento (blur)*, que pode surgir devido ao movimento relativo entre a cena e o sensor, instabilidades do dispositivo de captura ou restrições inerentes ao sistema óptico.

Nesse contexto o problema de restauração de imagens, com foco específico em desborrar (deblurring), a degradação é modelada como um problema inverso linear da forma

$$Ax = b,$$

em que A corresponde ao operador de convolução associado ao filtro responsável pelo borramento, x é a imagem ideal desconhecida e b representa a imagem borrada observada.

A tarefa de reconstrução da imagem nítida pode ser formulada como um problema de otimização convexa, cuja função de custo adotada é

$$\min_{x \in \Omega} \frac{1}{2} \|Ax - b\|^2,$$

onde Ω denota o espaço de imagens viáveis. Na prática, contudo, a imagem original x é desconhecida e o operador A constitui apenas uma aproximação do processo real de degradação, de modo que o problema se reduz à desconvolução, cujo objetivo é estimar a melhor aproximação possível de x a partir da observação b .

A natureza tipicamente mal-condicionada desse problema impede o uso direto da inversão explícita de A , o que torna necessário recorrer a métodos iterativos de otimização numérica capazes de produzir aproximações sucessivas e estáveis da solução. Neste trabalho, foram implementadas e analisadas três abordagens clássicas: descida do gradiente, gradiente conjugado e o método quasi-Newton L-BFGS.

2 METODOLOGIA

2.1 Formulação do Problema

Nesta seção apresentamos a metodologia adotada para resolver o problema de desborrar de imagens. O objetivo é reconstruir uma imagem nítida x desconhecida a partir de uma versão borrada b , modelada por

$$b = Ax,$$

onde A representa o operador de borramento obtido pela convolução com um kernel Gaussiano. O processo de reconstrução é formulado como o problema de otimização, onde queremos minimizar

$$J(x) = \frac{1}{2} \|Ax - b\|_2^2$$

2.2 Demonstração da Convexidade

Uma função $F : \mathbb{R}^n \rightarrow \mathbb{R}$ é convexa se, para quaisquer $x, y \in \mathbb{R}^n$ e $t \in [0,1]$:

$$F(tx + (1-t)y) \leq tF(x) + (1-t)F(y)$$

2.2.1 Demonstração

Seja $J(x) = \frac{1}{2} \|Ax - b\|^2$. Queremos mostrar que J é convexa.

Sejam $x, y \in \mathbb{R}^n$ e $t \in [0,1]$. Defina $z = tx + (1-t)y$.

Calculemos $J(z)$:

$$J(z) = \frac{1}{2} \|A[tx + (1-t)y] - b\|^2 = \frac{1}{2} \|tAx + (1-t)Ay - b\|^2$$

Observe que:

$$tAx + (1-t)Ay - b = tAx + (1-t)Ay - (t + (1-t))b = t(Ax - b) + (1-t)(Ay - b)$$

Portanto:

$$J(tx + (1-t)y) = \frac{1}{2} \|t(Ax - b) + (1-t)(Ay - b)\|^2$$

Como a função $\phi(w) = \|w\|^2$ é convexa, vale a desigualdade

$$\|tu + (1-t)v\|^2 \leq t\|u\|^2 + (1-t)\|v\|^2 \quad \text{para quaisquer } u, v.$$

Aplicando-a com $u = Ax - b$ e $v = Ay - b$, obtemos:

$$\|t(Ax - b) + (1-t)(Ay - b)\|^2 \leq t\|Ax - b\|^2 + (1-t)\|Ay - b\|^2.$$

Multiplicando por $\frac{1}{2}$:

$$\frac{1}{2}\|t(Ax - b) + (1-t)(Ay - b)\|^2 \leq t \cdot \frac{1}{2}\|Ax - b\|^2 + (1-t) \cdot \frac{1}{2}\|Ay - b\|^2$$

Ou seja:

$$J(tx + (1-t)y) \leq tJ(x) + (1-t)J(y)$$

O que prova que J é convexa.

Assim, demonstramos que $J(\mathbf{x})$ é convexa, validando teoricamente o uso dos métodos de otimização implementados e garantindo que a solução ótima corresponde ao mínimo global \mathbf{x}^* .

2.3 Preparação da Imagem

As imagens utilizadas foram carregadas por meio da biblioteca `scikit-image`. Em seguida, foram convertidas para tons de cinza e normalizadas no intervalo $[0,1]$, produzindo a imagem verdadeira x_{true} . Esse pré-processamento assegura que as operações matemáticas subsequentes sejam consistentes.



Figura 1 – Imagem 1 utilizada.

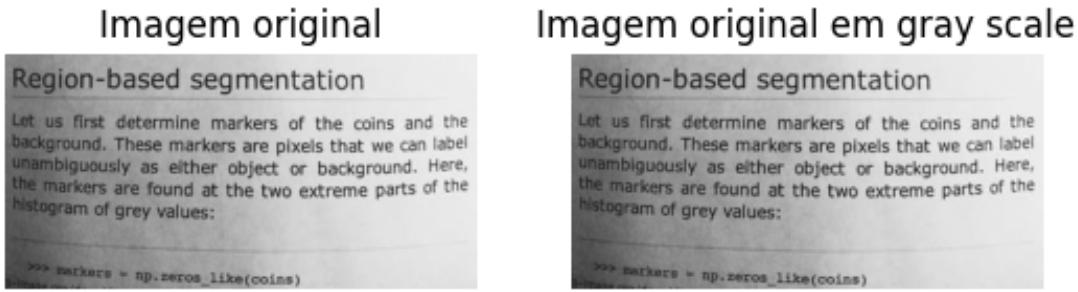


Figura 2 – Imagem 2 utilizada.

2.4 Modelagem da Degradação

Para simular o processo de borramento, utilizamos um kernel Gaussiano bidimensional com desvio padrão σ . O kernel é dado por:

$$G(i,j) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right),$$

e normalizado para ter soma igual a 1.

A imagem borrada é então calculada pela operação de convolução:

$$b = G * x_{\text{true}},$$

onde $*$ denota a operação de convolução bidimensional, definida como:

$$(G * x)(i,j) = \sum_{m=-k}^k \sum_{n=-k}^k G(m,n) \cdot x(i-m, j-n),$$

onde k determina o tamanho do kernel, feito com a seguinte operação $k = 6\sigma + 1$

2.5 Interpretação da Convolução

A operação de convolução pode ser interpretada como um processo de *deslizamento* do kernel sobre a imagem, onde cada pixel da imagem resultante é calculado como uma média ponderada dos pixels vizinhos da imagem original. Os pesos são determinados pelos valores do kernel Gaussiano, dando maior importância aos pixels centrais e menor importância aos pixels distantes.

- **Suavização:** O kernel Gaussiano atua como um filtro passa-baixa, suavizando variações abruptas de intensidade
- **Preservação:** Mantém a estrutura geral da imagem enquanto reduz detalhes finos
- **Implementação:** Utilizou-se `convolve2d` do SciPy com condições de corte simétricas

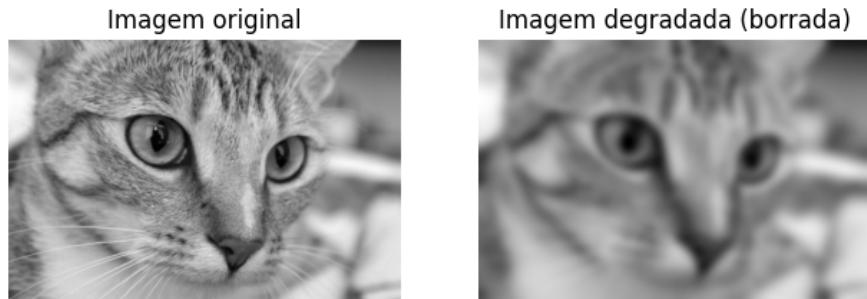


Figura 3 – Imagem original (esquerda) e imagem borrada (direita) com kernel Gaussiano $\sigma = 5$.

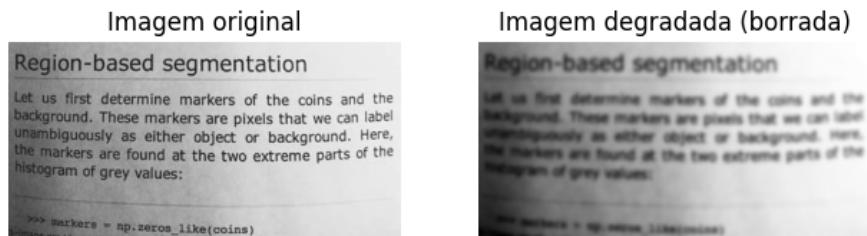


Figura 4 – Imagem original (esquerda) e imagem borrada (direita) com kernel Gaussiano $\sigma = 2$.

2.6 Métodos de Otimização Implementados

2.6.1 Gradiente Descendente

O método do Gradiente Descendente atualiza iterativamente a solução através de:

$$x_{k+1} = x_k - t_k \nabla J(x_k),$$

onde o gradiente é calculado por:

$$\nabla J(x) = A^T(A(x) - b),$$

e o tamanho do passo t_k é determinado por busca linear exata:

$$t_k = \frac{\|\nabla J(x_k)\|_2^2}{\|A(\nabla J(x_k))\|_2^2}.$$

Esta escolha de passo ótimo aproveita a estrutura quadrática do problema, garantindo o máximo decréscimo da função objetivo a cada iteração.

2.6.2 Gradientes Conjugados (CGNE)

O método CGNE resolve o sistema das equações normais:

$$A^T Ax = A^T b,$$

utilizando direções conjugadas. O algoritmo implementado segue os seguintes passos:

1. Inicialização: $x_0 = 0$, $r_0 = A^T b$, $p_0 = r_0$
2. Para $k = 0, 1, 2, \dots$ até convergência:

$$\alpha_k = \frac{\|r_k\|_2^2}{\|Ap_k\|_2^2}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k - \alpha_k A^T Ap_k$$

$$\beta_k = \frac{\|r_{k+1}\|_2^2}{\|r_k\|_2^2}$$

$$p_{k+1} = r_{k+1} + \beta_k p_k$$

A principal vantagem computacional reside na evitabilidade de formar explicitamente a matriz $A^T A$, operando apenas com produtos matriz-vetor através de convoluções.

2.7 Quase-Newton L-BFGS

O método L-BFGS constrói uma aproximação de baixa memória da inversa da matriz Hessiana. Para uma função objetivo $f(x)$, a direção de busca é calculada por:

$$d_k = -H_k \nabla f(x_k),$$

$$x_{i+1} = x_i + d_k$$

onde H_k é uma aproximação da inversa da Hessiana atualizada pela fórmula BFGS limitada.

A atualização utiliza os m pares de vetores mais recentes $\{s_i, y_i\}$, onde:

$$s_i = x_{i+1} - x_i$$

$$y_i = \nabla f(x_{i+1}) - \nabla f(x_i)$$

A implementação utiliza a rotina `scipy.optimize.minimize` com o método L-BFGS-B, que incorpora limites nas variáveis (não utilizados neste trabalho).

2.8 Métricas de Avaliação

A qualidade da reconstrução foi avaliada utilizando o **PSNR (Peak Signal-to-Noise Ratio)**, uma métrica amplamente usada para quantificar a similaridade entre a imagem original e a reconstruída, definida por:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right),$$

onde MAX é o valor máximo possível da imagem (1.0 para imagens normalizadas) e MSE é o erro quadrático médio entre a imagem original e a reconstruída:

$$\text{MSE} = \frac{1}{N} \|x_{\text{true}} - x_{\text{rec}}\|_2^2.$$

A implementação da função é feita utilizando a biblioteca `scikit-image`.

2.9 Configuração Experimental

Todos os experimentos foram implementados em Python utilizando as bibliotecas NumPy, SciPy, e scikit-image. Os critérios de parada estabelecidos foram tolerância de 10^{-5} para a norma do gradiente ou residual, com limite máximo de 1000 iterações.

Para monitoramento do progresso, amostras das imagens intermediárias foram armazenadas a cada 10 iterações para todos os métodos.

3 RESULTADOS E DISCUSSÃO

3.1 Análise dos Resultados Obtidos

Os métodos foram avaliados quantitativamente através da métrica PSNR em diferentes estágios da otimização. As Tabelas 1 e 2 resumem o desempenho final de cada método das imagens 1 e 2 respectivamente.

Tabela 1 – Desempenho comparativo dos métodos de otimização para imagem 1

Método	PSNR Final (dB)	Iterações	PSNR Borrada (dB)	Ganho (dB)
GD	29.54	1000	25.90	3.64
CGNE	30.78	1000	25.90	4.88
L-BFGS	29.92	170	25.90	4.02

Tabela 2 – Desempenho comparativo dos métodos de otimização para imagem 2

Método	PSNR Final (dB)	Iterações	PSNR Borrada (dB)	Ganho (dB)
GD	21.75	1000	19.08	2.67
CGNE	24.74	1000	19.08	5.66
L-BFGS	23.26	320	19.08	4.18

3.2 Visualização de resultados



Figura 5 – Comparativo Imagem 1 pelo método da descida do gradiente



Figura 6 – Comparativo Imagem 1 pelo método do gradiente conjulgado



Figura 7 – Comparativo Imagem 1 pelo método L-BFGS

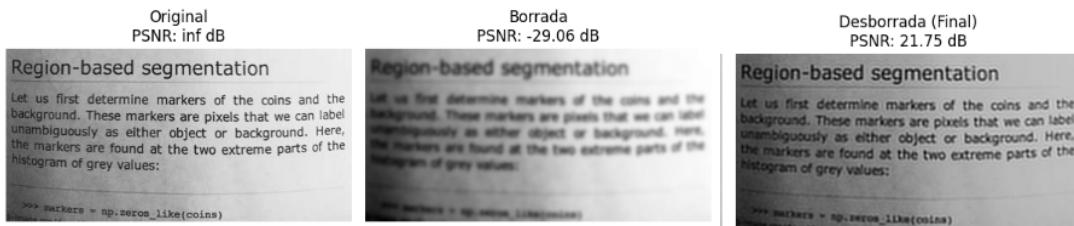


Figura 8 – Comparativo Imagem 2 pelo método da descida do gradiente

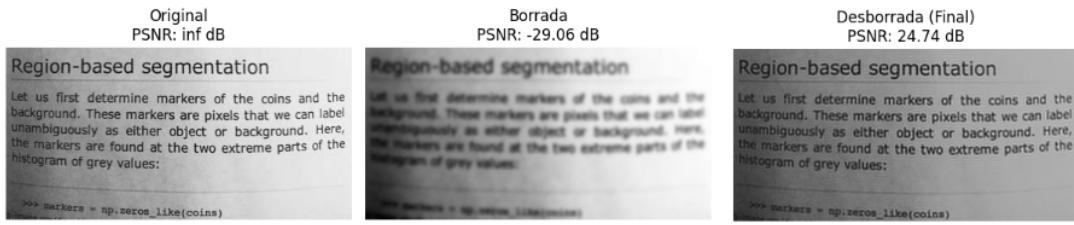


Figura 9 – Comparativo Imagem 2 pelo método do gradiente conjulgado

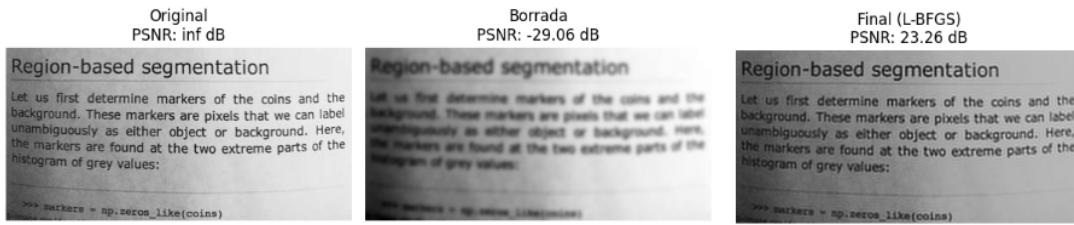


Figura 10 – Comparativo Imagem 2 pelo método L-BFGS

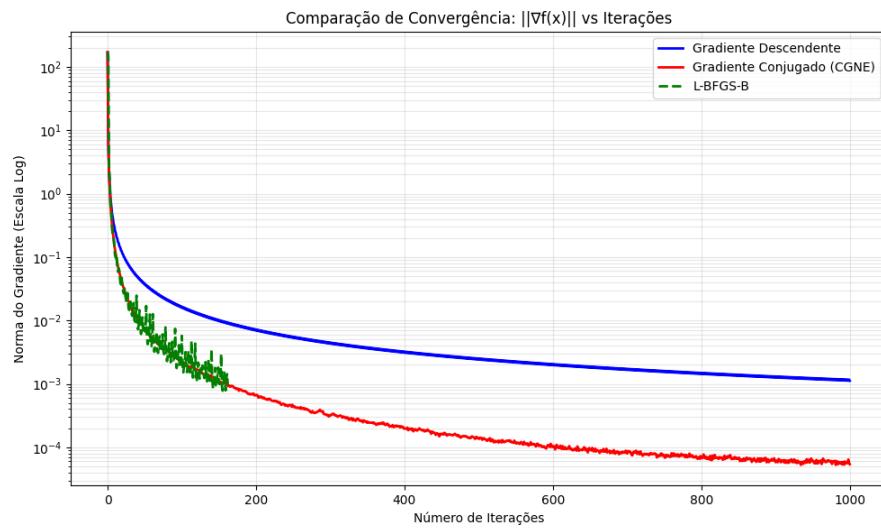


Figura 11 – Gráfico Número de iterações x Norma do Gradiente em escala logarítmica para a imagem 1

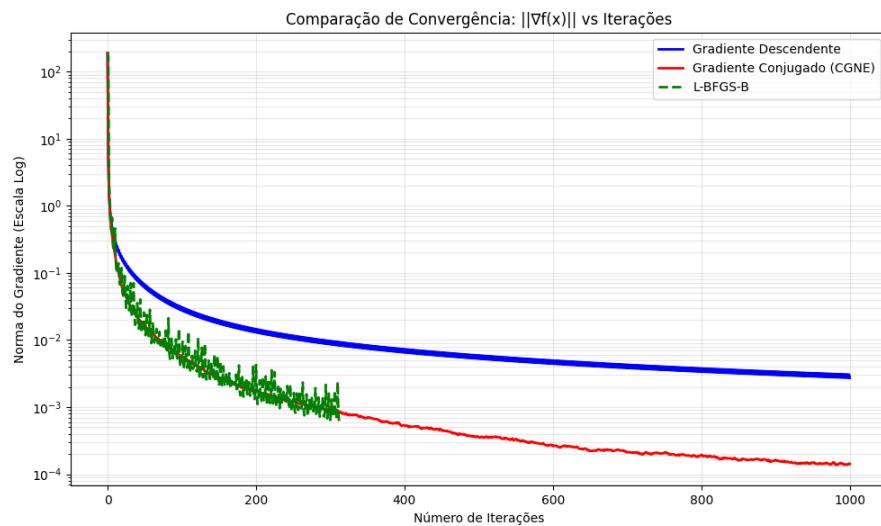


Figura 12 – Gráfico Número de iterações x Norma do Gradiente em escala logarítmica para a imagem 2

3.3 Implicações Práticas

Os resultados sugerem recomendações para aplicações práticas:

- **CGNE:** Melhor escolha quando a qualidade final é prioritária

- **L-BFGS:** Ideal quando há restrições computacionais ou de tempo
- **GD:** Apenas para problemas simples ou como referência inicial

4 CONCLUSÃO

Os métodos iterativos analisados neste trabalho permitiram elucidar, de maneira abrangente, como diferentes estratégias de otimização se comportam diante do problema de restauração de imagens degradadas por borramento Gaussiano. Observou-se que cada método possui características próprias que o tornam mais adequado a determinados contextos computacionais ou a distintas exigências de qualidade.

O método CGNE destacou-se como a abordagem que alcançou a reconstrução de maior fidelidade, apresentando os maiores valores de PSNR e revelando as estruturas originais com menor grau de distorção. Esse desempenho reforça a conhecida aptidão dos métodos baseados em equações normais para tratar problemas inversos mal condicionados, sobretudo quando o operador de degradação consiste em uma convolução Gaussiana que atenua de forma pronunciada as componentes de alta frequência. A robustez do CGNE, mesmo em ambientes desfavoráveis, confirma seu papel consolidado como referência clássica na resolução de problemas de desconvolução.

O método L BFGS, por sua vez, sobressaiu pela notável eficiência computacional. Sua rápida convergência e o reduzido número de iterações necessárias para atingir resultados satisfatórios tornam essa técnica especialmente adequada a aplicações de grande porte ou a sistemas com recursos computacionais limitados. A capacidade de aproximar a Hessiana de maneira parcimoniosa, recorrendo apenas a informações de primeira ordem, constitui uma virtude relevante em cenários de alta dimensionalidade, como é o caso do processamento de imagens.

O método do Gradiente Descendente, embora apresente desempenho inferior segundo as métricas quantitativas, preserva utilidade pedagógica ao expor com clareza os mecanismos fundamentais dos métodos de otimização. Seu comportamento mais irregular e sua dependência sensível da escolha do passo ilustram de maneira exemplar as dificuldades vinculadas à solução de problemas inversos mal postos.

Os resultados obtidos confirmam que a formulação convexa empregada é matematicamente consistente e apropriada: todos os métodos produziram ganhos

substanciais de qualidade quando comparados à imagem borrada original. Convém observar, contudo, que pequenas variações nos parâmetros experimentais, como o desvio padrão do kernel Gaussiano, as condições de contorno ou os critérios de parada, podem alterar de modo significativo a qualidade final da reconstrução, o que evidencia a sensibilidade característica de problemas inversos.

1. O método CGNE obteve a melhor qualidade de reconstrução (PSNR: 30.78 dB para a imagem 1 e 24.74dB para a imagem 2)
2. O L-BFGS demonstrou a melhor eficiência computacional (160 iterações para imagem 1 e 320 iterações para a imagem 2)
3. O Gradiente Descendente apresentou o pior desempenho geral
4. Todos os métodos melhoraram significativamente a qualidade em relação à imagem borrada
5. Os resultados validam a formulação convexa do problema de restauração

REFERÊNCIAS

NOCEDAL, Jorge; WRIGHT, Stephen. *Numerical Optimization*. 2. ed. New York: Springer, 2006. Informações e preview disponíveis em: <<https://link.springer.com/book/10.1007/978-0-387-40065-5>>.

STRANG, Gilbert. *Introduction to Linear Algebra*. 4. ed. Wellesley–Cambridge Press, 2009. Página oficial do autor com materiais: <<https://math.mit.edu/~gs/linearalgebra/>>.

RIBEIRO, Ademir Alves; KARAS, Elizabeth Wegner. *Otimização contínua: aspectos teóricos e computacionais*. São Paulo: Cengage Learning, 2014. <<https://www.ime.unicamp.br/~sandra/MS629/handouts/livro28jul.pdf>>

VAN DER WALT, S. et al. scikit-image: image processing in Python. *PeerJ*, v. 2, p. e453, 2014. Disponível em: <<https://scikit-image.org/>>.