Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 12, n. 1, 2026.

Trabalho apresentado no XLIV CNMAC, Fundação Getulio Vargas - Rio de Janeiro - RJ, 2025.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Método BFGS: Um Estudo Comparativo entre as Buscas Armijo e Wolfe na Minimização da Função de Rosenbrock

João M. L. Luiz[1], Marcio A. de A. Bortoloti[2]
Universidade Estadual do Sudoeste da Bahia, Vitória da Conquista, BA

Os métodos quase-Newton buscam alcançar as boas taxas de convergência do método de Newton, mas sem a necessidade de fazer cálculos de segundas derivadas e inversões de matrizes. Essas operações demandam um alto custo computacional, o que pode diminuir o desempenho do método. Nesse contexto destacamos o método Broyden-Fletcher-Goldfarb-Shanno (BFGS) que caracteriza-se por construir aproximações para as inversas das matrizes hessianas. Formalmente, esse método pode ser organizado como o algoritmo a seguir.

Algoritmo 1: MÉTODO BFGS

1 Tome um ponto inicial $x^0 \in \mathbb{R}^n$, uma matriz simétrica definida positiva Q_0 e $\epsilon > 0$.

2 Defina $k = 0$.

3 Repita enquanto $\|f'(x^k)\| > \epsilon$.

4 $d^k := -Q_k f'(x^k)$.

5 $x^{k+1} := x^k + \alpha_k d^k$, com $\alpha_k > 0$.

6 $r^k := x^{k+1} - x^k, \quad s^k := f'(x^{k+1}) - f'(x^k)$.

7 Atualize Q_{k+1} por

$$Q_{k+1} = Q_k + \frac{(r^k - Q_k s^k)(r^k)^\top + r^k(r^k - Q_k s^k)^\top}{\langle r^k, s^k\rangle} - \frac{\langle r^k - Q_k s^k, s^k\rangle r^k (r^k)^\top}{\langle r^k, s^k\rangle^2}$$

8 Defina $x^k = x^k + \alpha_k d^k$.

No passo 5 do algoritmo acima o comprimento de passo α_k é determinado por uma busca linear. Neste trabalho analisaremos o método BFGS equipado com as buscas de Armijo e Wolfe.

A busca de Armijo determina $\alpha_k > 0$ tal que

$$f(x^k + \alpha_k d^k) \leq f(x^k) + \sigma\alpha_k \langle f'(x^k), d^k\rangle, \tag{1}$$

onde $\sigma \in (0, \frac{1}{2})$. Observamos que essa busca é bem definida, ou seja, dada uma direção de descida d^k, sempre existe $\alpha_k > 0$ que satisfaz a desigualdade (1) (veja em [1]). Além disso ressaltamos que essa busca garante um decrescimento da sequência $\{f(x^k)\}$.

[1]joaomatheusliraluiz499@gmail.com
[2]mbortoloti@uesb.edu.br

A busca de Wolfe é caracteriza pela introdução de um critério de curvatura na busca de Armijo, essa busca determina o maior real postivo $\alpha_k > 0$ (como visto em [1]) que satisfaz

$$f(x^k + \alpha_k d^k) \leq f(x^k) + \sigma_1 \alpha_k \langle f'(x^k), d^k \rangle, \tag{2}$$

$$\langle f'(x^k + \alpha_k d^k), d^k \rangle \geq \sigma_2 \langle f'(x^k), d^k \rangle, \tag{3}$$

onde $0 < \sigma_1 < \sigma_2 < \frac{1}{2}$.

Para os testes numéricos, consideramos o problema de minimizar a função $f : \mathbb{R}^n \to \mathbb{R}$ dada por

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2],$$

conhecida como função de Rosenbrock. Para esse problema, utilizamos o método BFGS equipado com as regras de Armijo e Wolfe com o objetivo de estudar o desempenho de cada uma dessas buscas. Nossos testes consideraram o tempo de CPU e o número de iterações como fatores de comparação entre as buscas. O método foi implementado utilizando a linguagem de programação Julia [2]. Os códigos estão disponíveis livremente no endereço `https://github.com/petimatematica/bfgs`.

Nos testes do BFGS com Armijo, consideramos $\sigma = 0,4$ e para os testes com Wolfe consideramos $\sigma_1 = 0,3$ e $\sigma_2 = 0,4$. Para cada $n \in \{10, 20, \ldots, 90, 100\}$ foram tomados 10 chutes iniciais aleatoriamente escolhidos. Os testes mostraram que o método BFGS, equipado com a busca de Wolfe, apresentou melhor desempenho em relação ao mesmo método com busca de Armijo no número de iterações. Por outro lado, no tempo de CPU, o método com busca de Wolfe teve resultados ligeiramente melhores. No entanto, essa diferença não foi significativa.

Agradecimentos

O autor João Matheus Lira Luiz agradece ao Programa de Educação Tutorial Institucional da Universidade Estadual do Sudoeste da Bahia (PETI/UESB) pelo apoio financeiro

Referências

[1] A. Izmailov e M. Solodov. **Otimização, volume 2: métodos computacionais**. IMPA, 2018.

[2] B. Lauwens. **Introdução à programação em Julia**. `https://juliaintro.github.io/`. Acesso em: 15 mar. 2025. 2025.