

Universidade Federal do Amazonas
Instituto de Ciências Exatas
Programa de Pós-Graduação em Matemática
Mestrado em Matemática Aplicada

Aplicação dos Métodos Quase-Newton em Otimização Intervalar

Sérgio Kardec Soares Batista

Manaus - AM
Março de 2020

Universidade Federal do Amazonas
Instituto de Ciências Exatas
Programa de Pós-Graduação em Matemática
Mestrado em Matemática

Aplicação dos Métodos Quase-Newton em Otimização Intervalar

por

Sérgio Kardec Soares Batista

sob a orientação do

Prof. Dr. Sandro Dimy Barbosa Bitar

Manaus - AM
Março de 2020

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

B333a Batista, Sérgio Kardec Soares
Aplicação dos Métodos Quase-Newton em Otimização Intervalar /
Sérgio Kardec Soares Batista. - 2020.
59 f. : il., color. ; 31 cm.

Orientador(a): Sandro Dimy Barbosa Bitar.
Dissertação (mestrado) - Universidade Federal do Amazonas,
Programa de Pós-Graduação em Matemática, Manaus, 2020.

1. Otimização Intervalar. 2. Função Intervalar Paramétrica. 3.
Diferença de Hukuhara. 4. Direção de Descida. 5. Cálculo
Intervalar. I. Bitar, Sandro Dimy Barbosa. II. Universidade Federal
do Amazonas. Programa de Pós-Graduação em Matemática. III.
Título

Aplicação dos Métodos Quase-Newton em Otimização Intervalar

por

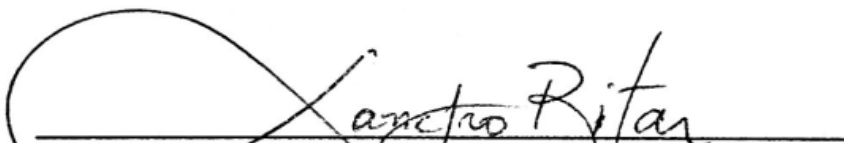
Sérgio Kardec Soares Batista ¹

Dissertação apresentada ao Programa de Pós-Graduação em Matemática da Universidade Federal do Amazonas como requisito parcial para obtenção do título de Mestre em Matemática.

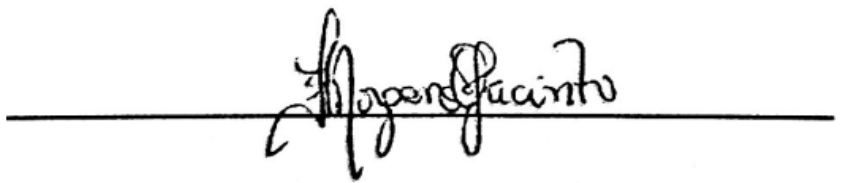
Área de Concentração: Matemática

Aprovada em 20/04/2020

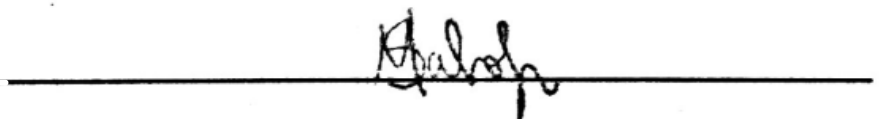
Banca Examinadora:



Prof. Dr. Sandro Dimy Barbosa Bitar (Presidente)
Universidade Federal do Amazonas – DM/UFAM



Prof. Dr. Flávia Morgana de Oliveira Jacinto (Membro)
Universidade Federal do Amazonas – DM/UFAM



Prof. Dr. Mário Salvatierra Júnior (Membro Externo)
Universidade Federal do Amazonas – ICOMP/UFAM

*Dedico este trabalho aos
meus pais Francisco Sérgio e
Edcilane Soares.*

Agradecimentos

Primeiramente a Deus, a minha mãe Edcilane Soares e a meu pai Francisco Sérgio pelo apoio, pelo carinho, pela confiança, pelo amor incondicional.

Agradeço aos meus amigos da Matemática: Airton Freitas, Albert Amaral, André Matos, Davi Misturini, Erivelta, Fábio Henrique, Jéssica Valim, Matheus Hudson, Umbelito, Patrick Sena, Rafhael Aguiar, Sara, Yasmin Brilhante, Wington Vital e os demais nos quais formei amizades ao longo do curso.

Aos professores de Graduação em Matemática da UFAM: Alfredo Wagner, Inês Padilha, Mário Salvatierra, Michel Pinho, Nilomar Oliveira, Raul Mesquita, Sandro Bitar, Stefan Ehbauer, Valtemir por conhecimentos transmitidos. Agradeço também ao professor Max de Souza por ter me orientado no PIBIC durante a minha graduação, com o qual foi de grande relevância no meu crescimento na Matemática, o que me fez estudar a Matemática no nível mais elevado, adquirindo novos conhecimentos, novas ferramentas.

Ao professor Sandro Dimy Barbosa Bitar pela orientação, pela ajuda na elaboração deste trabalho e por ter me acompanhado ao longo do Mestrado.

Agradeço ao professor Mário Salvatierra Júnior e a professora Flávia Morgana de Oliveira Jacinto por terem aceitado participar da minha banca.

Agradeço aos professores do Programa de Pós-Graduação em Matemática: Dragomir Tsonev, Inês Padilha, Jorge Fernandes, José Nazareno, Marcos Aurélio, Mikhail Neklyudov, Rosilene, Thiago Rodrigo que contribuíram para a minha formação acadêmica e o meu desenvolvimento na Matemática, no qual me ajudou a obter ferramentas para a elaboração deste trabalho.

Aos meus amigos de Mestrado: André Matos, Ayana Santana, Cristiano Souza, Davi Misturini, Edfram, Fábio Henrique, Flávia, Fernando, Gabriel Araújo, Maristela Barbosa, Wington Vital. Aos amigos do IMPA Caio, Gabriel Ernandes, Lucas Pereira, Marielson Dias e Rodrigo.

Agradeço também aos que não citei nessa lista de agradecimentos, que contribuíram também para minha formação na Matemática.

À FAPEAM pelo apoio financeiro.

“Afirmações extraordinárias
necessitam de evidências
extraordinárias”.

(Carl Sagan)

Resumo

Nessa dissertação realizou-se um estudo sobre Otimização Intervalar, assumindo que o valor da Função Objetivo é um intervalo. Para tal problema, seguiu-se um desenvolvimento teórico para munir o espaço intervalar de operações básicas fundamentais como adição subtração e multiplicação. Alguns Teoremas e Corolários, considerados essenciais à compreensão dos principais resultados, foram abordados. Condições para a existência de solução de um problema de otimização intervalar são apresentadas através de um problema auxiliar em espaço finito dimensional real, onde a função de valor intervalar é levado a um valor real através de uma função auxiliar. Ao final do trabalho apresenta-se uma nova direção de descida via método quasi-newton (versão BFGS) cuja função objetivo é integrada ao longo dos intervalos dos seus coeficientes intervalares, sendo o problema de otimização irrestrito e, além disso, apresenta-se também um algoritmo genérico baseado em busca direcional para esse problema.

Palavras-chave: Otimização Intervalar, Direção de Descida, Função Intervalar Paramétrica, Diferença de Hukuhara, Cálculo Intervalar.

Abstract

In this dissertation, a study on Interval Optimization was carried out, assuming that the value of Purpose function is an interval. For such a problem, a theoretical development followed to equip the interval space with an ordered body structure. Some Theorems and Corollaries, considered essential to the understanding of the main results, were addressed. Conditions for the solution of an optimization problem

intervals are presented through an auxiliary problem in real dimensional finite space, where the function of interval value is brought to a real value through an auxiliary function. At the end of the work, a new direction of descent via the quasi-newton method (version BFGS) whose objective function is integrated over the intervals of its interval coefficients, the optimization problem being unrestricted and, in addition, an algorithm is also presented generic based on directional search for this problem.

Keywords: Interval Optimization, Descent Direction, Parametric Interval Function, Interval Calculus..

Sumário

Introdução	1
1 Preliminares	4
1.1 Fundamentos da aritmética intervalar	4
2 O Problema	12
2.1 Otimização Intervalar	12
2.1.1 Propriedades da Função Intervalar	12
2.1.2 O Problema Geral	17
2.1.3 Método Quase-Newton (versão BFGS)	21
3 Algumas simulações	23
3.1 Testando Método de Newton	23
3.2 Testando Método Quase-Newton	28
4 Considerações Finais	34
4.1 A Primeira Simulação	34
4.2 A Segunda Simulação	35
4.3 Uma Nova Abordagem	35
5 Conclusão	37
6 Apêndice	39
6.1 Implementação dos Algoritmos	39
Referências Bibliográficas	49

Introdução

Um tema recorrente na matemática aplicada são problemas de minimização e maximização (otimização em geral), onde busca se minimizar ou maximizar uma função através da escolha não-arbitrária dos valores das variáveis da função dentro de um conjunto factível (nesse trabalho esse conjunto será o $\mathbb{R}^n$). A função que deseja-se otimizar pode medir um certo ganho ou custo e supondo que queira-se tomar uma decisão racional então o objetivo pode ser maximizar ou minimizar (respectivamente). Acontece que na vida real o custo dos insumos de modo geral variando no tempo, ainda mais se o custo for o próprio tempo em si (para realização de uma tarefa por exemplo), no caso, o tempo médio para realização de um serviço pode variar num certo intervalo e pensando nesse tipo de problema, uma das primeiras propostas foi a otimização intervalar (por volta de 1960 Moore [5], bem antes dos modelos bayesianos ou fuzzy) onde considera-se que os coeficientes da função objetivo variam em certos intervalos pré estabelecido que devem garantir que apesar desse tais custo serem incertos (suponha um problema de minimização), os custos reais estarão compreendido nesses intervalos, diminuindo assim a chance de erro, dado a incerteza dos custos. Há diversos casos que se pode aplicar e elucidar melhor a necessidade de adotar intervalos nos coeficientes da função um caso desses será visto num exemplo posterior de custo de estoque, trata-se de um problema clássico de otimização onde se deseja estocar uma quantidade ótima de um certo insumo em função de sua necessidade para atender uma certa demanda mas essa tal demanda pode ser uma incerteza (que até então não é levado em conta na literatura de base) mas que agora podemos via otimização intervalar considerar os casos para os quais pode-se aplicar o que em seguida será desenvolvido para acomodar o que já se entende como ponto ótimo.

Óbvio que para essa "nova" área da otimização precisa que seja definido posteriormente como se soma, subtrai, multiplica intervalos e outras diversas operações matemáticas serão reestabelecidas nessa dissertação afim de que o leitor se situe em que ambiente será trabalhado tais elementos aritméticos e até elementos do cálculo serão levemente modificados para se adequar ao produto final que é um algoritmo iterativo que busca via direção de descida que será definida posteriormente como a de newton e quase-newton afim de encontrar um ponto optimal-eficiente que minimize uma função de custo geral (desde que seja duplamente diferenciável) dentro de um cenário irrestrito.

O feito desse trabalho foi utilizar o método quase-newton para solucionar esse problema pois tem um ótimo desempenho na sua versão BFGS invés de buscar uma solução para o problema via método de newton clássico, como Ghosh [7] implementa, pois e além do quase-newton ter ótimo desempenho também exige menos do computador, em comparação ao método de newton clássico pois esse exige que se inverta uma matriz hessiana a cada iteração no ponto para descobrir o próximo ponto, e sabe-se que inverter matriz tem alto custo computacional dependendo da matriz pode ser até ser impossível para o computador. Dessa forma propõe-se o método quase-newton que não tem essa necessidade que na maioria dos casos gerais inviabilisaria a busca pelo ponto ótimo.

O problema otimização comum basicamente se trata de encontrar $x^* \in \Omega \subset \mathbb{R}^n$ tal que $F(x^*) \leq F(x) \forall x \in \Omega$ onde $F(x)$ é uma função de classe C^2 bem definida $F : \mathbb{R}^n \rightarrow \mathbb{R}$ agora o caso intervalar tem diversas alterações, como a imagem da função intervalar, que é o conjunto de todos intervalos possíveis ($I(\mathbb{R})$) daí o primeiro problema seria definir uma ordem pelo menos parcial para poder comparar dois elemento do conjunto de $\{F(x)|x \in \mathbb{R}^n\}$ assim matematicamente trata se de encontrar $x^* \in \mathbb{R}^n$ tal que $F(x^*) \preceq F(x) \forall x$ com $\preceq$ sendo uma ordem parcial para $I(\mathbb{R})$ e nesse formato será resolvido três exemplos, dois abstratos e um como já citado nessa mesma introdução, é um problema de estoque, e note que se trata mesmo de um problema de otimização pois se uma empresa não estoca, se tem prejuízo pois se não houver o que vender obviamente não há faturamento (faturamento que se perde e ainda o cliente que se perde muita das vezes para sempre) e se estoca demais também tem prejuízo pois há custo de armazenagem e custo de desperdícios e etc, assim note que essa função custo de estoque é coerciva ao longo dos quadrantes positivos assim é possível garantir a existência de um ponto ótimo intermediario (ou seja, deve existir um $x^* \in \mathbb{R}^n$ tal que $C(x^*) \leq C(x) \forall x$ onde $C(x)$ é a função custo de estocagem. Assim sendo o trabalho será dividido da seguinte maneira, no primeiro capítulo serão estabelecidas as operações básicas, dentre elas operações aritméticas e explicar com mais cuidado como seria uma função intervalar e reescrever ela em função da combinação convexa dos extremos de cada intervalo dos coeficientes da função em questão, em seguida no capítulo 2 será visto conceitos mais aprofundados como o que seria uma derivada de uma função desse tipo, e será definida uma nova maneira de subtrair dois intervalos pois notará-se que a diferença usual pode causar problema em derivadas de funcionais lineares intervalares simples (é um funcional porém os coeficientes desse funcional são intervalos) que a principio deveria ser diferenciável tal como é para o caso "normal" e para burlar esse problema será definida a tal diferença de Hukuhara citada no título dessa dissertação, já no capítulo 3 será explorado os métodos para resolver esse problema e bem como suas comparações, no capítulo 4 será a resolução dos exemplo pelos dois métodos (newton que já foi feito por Ghosh [7] e quase-newton) depois exibirá-se o algoritmo utilizado e por

fim avaliará-se a abordagem foi vantajosa em algum sentido.

Capítulo 1

Preliminares

Neste capítulo apresenta-se as definições, resultados e exemplos da teoria básica e aplicada sobre o tema espaço intervalar que serão utilizados no decorrer desta dissertação. Faz-se uma revisão breve dos conceitos de operações elementares relacionadas com aritmética intervalar. Pretende-se destacar os principais resultados de Ghosh[7] vinculados à derivada de Hukurara. Nesse sentido, apresenta-se algumas definições e resultados que poderão ser aprofundados em [5, 4, 6]

1.1 Fundamentos da aritmética intervalar

Inicialmente serão definidas as operações aritméticas no espaço intervalar. Tais operações foram definidas por Moore [5] e Panda [1]

Operações aritméticas com intervalos

Os intervalos fechados da reta real $\mathbb{R}$ são definidos da seguinte maneira:

$$[a, b] = \{x \in \mathbb{R} | a \leq x \leq b \forall a, b \in \mathbb{R}\}$$

Definição 1.1. *Considere um intervalo real fechado $X = [a, b]$. Um número intervalar é um intervalo fechado da reta, ou seja, consiste do conjunto $\{x \in \mathbb{R} | a \leq x \leq b\}$. O conjunto de todos os X é chamado de $I(\mathbb{R})$.*

[?] Assim sintetizou Silveira [3] que por sua vez usou Moore [5]

Um número intervalar é também chamado simplesmente de intervalo. As extremidades de um número intervalar X serão denotadas por $\overline{X}$ e $\underline{X}$. Assim,

(2.4), 2.4 (??), ??

$$X = [\underline{X}, \overline{X}] \tag{1.1}$$

Dois números intervalares X e Y são ditos iguais se eles são os mesmos conjuntos. Isso acontece se suas extremidades, respectivas, são iguais:

$$Y = X \Leftrightarrow \overline{X} = \overline{Y} \text{ e } \underline{X} = \underline{Y} \quad (1.2)$$

Como já dito, a ideia de trabalhar no universo intervalar $I(\mathbb{R})$ aí se pode generalizar o conceito de parâmetros fixos em funções objetivo, mas é claro que em alguns casos esses parâmetros de funções que podem estar medindo custo ou lucro por exemplo, pode se tratar de um número fixo (como por exemplo preços fixos) para esse caso particular ainda assim podemos enxergar como um intervalo, no caso, um intervalo degenerado que está definido a seguir.

Definição 1.2. $X \in I(\mathbb{R})$ é dito degenerado se $\underline{X} = \overline{X}$. Tal intervalo contém um único número real x . Por convenção, aceita-se a identificação de um intervalo degenerado $[x, x]$ com o número real x . Nesse sentido, pode-se escrever a seguinte identidade

$$x = [x, x]. \quad (1.3)$$

Como definido em Moore [?].

Da definição de intervalo degenerado tem que $\mathbb{R} \subset I(\mathbb{R})$

Definição 1.3. O conjunto abstrato de intervalos invertidos é denotado por $\overline{I(\mathbb{R})} = \{[a, b] | [b, a] \in I(\mathbb{R})\}$

A interseção de dois intervalos X e Y é vazia se $\underline{X} < \overline{Y}$ ou $\underline{Y} < \overline{X}$. Nesse caso escrevemos

$$X \cap Y = \emptyset$$

indicando que X e Y não tem pontos em comum. Quando $X \cap Y \neq \emptyset$, prova-se que a interseção ainda é um intervalo:

$$X \cap Y = [\max\{\underline{X}, \underline{Y}\}, \min\{\overline{X}, \overline{Y}\}] \quad (1.4)$$

Nesse caso, a união de X e Y também é um intervalo:

$$X \cup Y = [\min\{\underline{X}, \underline{Y}\}, \max\{\overline{X}, \overline{Y}\}] \quad (1.5)$$

Em geral, a união de dois intervalos não é um intervalo. Entretanto, a envoltória intervalar $X \sqcup Y$ de dois intervalos, que é definida por

$$X \sqcup Y = [\min\{\underline{X}, \underline{Y}\}, \max\{\overline{X}, \overline{Y}\}]. \quad (1.6)$$

é sempre um intervalo. Além disso, tem-se

$$X \cup Y \subseteq X \underline{\cup} Y \quad (1.7)$$

para quaisquer intervalos X e Y

Para elucidação desse conceito tomar-se-á o seguinte exemplo:

Se $X = [-1, 0]$ e $Y = [1, 2]$, então $X \underline{\cup} Y = [-1, 2]$. Apesar de $X \underline{\cup} Y$ ser um conjunto desconexo (ou seja, não é um intervalo), a relação (1.7) é mantida.

A respeito do intervalo $X = [\underline{X}, \overline{X}]$ pode-se extrair os seguintes números:

Definição 1.4. A medida $\omega(X)$ de um intervalo X é definida por:

$$\omega(X) = \overline{X} - \underline{X} \quad (1.8)$$

Definição 1.5. O valor absoluto de X , denotado por $|X|$, é o máximo dos valores absolutos de suas extremidades:

$$|X| = \max\{|\overline{X}|, |\underline{X}|\} \quad (1.9)$$

Note que $|x| \leq |X| \quad \forall x \in X$

Definição 1.6. O ponto médio $m(X)$ de X é dado por

$$m(X) = \frac{1}{2}(\overline{X} + \underline{X}). \quad (1.10)$$

A seguir tem-se alguns exemplos para fixação das definições anteriores.

Exemplo 1.1. Sejam $X = [0, 2]$ e $Y = [-1, 1]$. A interseção e a união de X e Y são os intervalos $X \cap Y = [\max\{0, -1\}, \min\{2, 1\}] = [0, 1]$, e $X \cup Y = [\min\{0, -1\}, \max\{2, 1\}] = [-1, 2]$.

Temos $m(X) = \omega(X) = 2$ e, por exemplo, $|X| = \max\{0, 2\} = 2$ O ponto médio de Y é $m(Y) = 0$.

A noção de intervalo degenerado, como definido acima, permite ver o conjunto dos intervalos fechados como uma extensão do conjunto dos números reais pois dado um número real x , temos a associação

$$x \mapsto [x, x] \quad (1.11)$$

que leva os números reais a intervalos degenerados que associa números reais a intervalos degenerados. Agora será definido as operações aritméticas básicas entre intervalos. O ponto chave nessas definições é o fato de que operar intervalos recai inicialmente sobre operações clássicas em conjuntos como bem observa Silveira [3].

Definição 1.7. *Sejam $X = [\underline{X}, \overline{X}]$, $Y = [\underline{Y}, \overline{Y}]$ dois intervalos. A soma de X e Y é o conjunto*

$$X + Y = \{x + y : x \in X, y \in Y\}. \quad (1.12)$$

que é um intervalo. De fato,

$$\underline{X} \leq x \leq \overline{X} \text{ e } \underline{Y} \leq y \leq \overline{Y}.$$

Pode-se ver pela soma das inequações que $x + y \in X + Y$ e $\underline{X} + \underline{Y} \leq x + y \leq \overline{X} + \overline{Y}$. Consequentemente, a fórmula

$$X + Y = [\underline{X} + \underline{Y}, \overline{X} + \overline{Y}] \quad (1.13)$$

pode ser usada para calcular (1.12)

Exemplo 1.2. *Sejam $X = [0, 2]$ e $Y = [-1, 1]$ como no Exemplo 1.1.2. Então*

$$X + Y = [0, 2] + [-1, 1] = [0 + (-1), 2 + 1] = [-1, 3].$$

A diferença de dois intervalos X e Y é o conjunto

$$X - Y = \{x - y : x \in X, y \in Y\} \quad (1.14)$$

Temos

$$\underline{X} \leq x \leq \overline{X} \text{ e } -\overline{X} \leq -x \leq -\underline{X}$$

$\Rightarrow$

$$\underline{X} - \overline{Y} \leq x - y \leq \overline{X} - \underline{Y}$$

$\Rightarrow$

$$X - Y = [\underline{X} - \overline{Y}, \overline{X} - \underline{Y}] \quad (1.15)$$

Note que

$$X - Y = X + (-Y),$$

onde

$$-Y = [-\overline{Y}, -\underline{Y}] = \{y : -y \in Y\}.$$

Perceba também que nem sempre $X - X = 0$ e será visto posteriormente que isso causará problemas na hora de derivar funcionais lineares intervalares por exemplo, que são essencialmente diferenciáveis para o caso de coeficientes constantes como será mostrado

que quando for definida a derivada intervalar, que como sabe-se, há uma subtração no numerador da fração do limite que se calcula a derivada de modo tradicional. Para evitar esse problema definiremos a seguir a diferença de Hukuhara.

Exemplo 1.3. Se $X = [-1, 0]$ e $Y = [1, 2]$

então

$$-Y = [-2, -1]$$

e

$$X - Y = X + (-Y) = [-3, -1].$$

O produto de X e Y é dado por

$$X \odot Y = \{x.y : x \in X, y \in Y\}. \quad (1.16)$$

Em termos das extremidades, o produto $X \cdot Y$ de dois intervalos X e Y é dado por

$$X \odot Y = [\min S, \max S] \quad (1.17)$$

onde $S = \{\overline{X}.\overline{Y}, \underline{X}.\underline{Y}, \overline{X}.\underline{Y}, \overline{Y}.\underline{X}\}$.

Se um dos intervalos for degenerado, tem-se

$$\alpha[\underline{X}, \overline{X}] = [\alpha\underline{X}, \alpha\overline{X}] \quad (1.18)$$

se $\alpha \leq 0$,

ou se $\alpha < 0$ então $\alpha[\underline{X}, \overline{X}] = [\alpha\overline{X}, \alpha\underline{X}]$.

Exemplo 1.4. Sejam $X = [-1, 0]$ e $Y = [1, 2]$. Então

$$S = \{-1.1, -1.2, 0.1, 0.2\} = \{-1, -2, 0\}$$

e $X \odot Y = [\min S, \max S] = [-2, 0]$. Temos também, por exemplo, $2Y = [2, 2] \odot [1, 2] = [2, 4]$.

O quociente X/Y é definido por Moore [5] como

$$X/Y = \{x/y | x \in X, y \in Y\} \quad (1.19)$$

desde que 0 não esteja em Y . Assim como acontece com os números reais, a divisão pode ser definida via multiplicação pelo inverso multiplicativo do segundo termo. Isto é, podemos calcular a equação (1.19) usando

$$X/Y = X \odot (1/Y) \quad (1.20)$$

onde

$$1/Y = \{y | 1/y \in Y\} = [1/\overline{Y}, 1/\underline{Y}] \quad (1.21)$$

novamente, assumindo que $0 \notin Y$.

Outra forma de ver um intervalo é escreve-lo em termos de sua medida e seu ponto médio da seguinte maneira:

Definição 1.8. *Qualquer intervalo X pode ser expresso como:*

$$X = m(X) + \frac{1}{2}\omega(X)[-1, 1] = m(X) + [-\frac{1}{2}\omega(X), \frac{1}{2}\omega(X)]$$

De fato, se $X = [\underline{X}, \overline{X}]$, sabe-se que $m(X) = \frac{1}{2}(\underline{X} + \overline{X})$ e $\omega(X) = \overline{X} - \underline{X}$

Sabe-se que todo intervalo da reta é um conjunto convexo, ou seja, dado quaisquer dois pontos em um intervalo, existe um segmento que os une que ainda esta inteiramente contido dentro desse mesmo intervalo, matematicamente podemos escrever:

$$X = [\underline{X}, \overline{X}] = \mu\underline{X} + (1 - \mu)\overline{X}, \mu \in [0, 1]$$

Agora como já citado antes será definido uma nova forma de calcular a diferença de dois intervalos de maneira que $X - X = 0 \forall X \in I(\mathbb{R})$ pois veja que para a definição anterior, por exemplo $[0, 1] - [0, 1] = [-1, 1] \neq 0$ que seria razoável que a diferença de duas coisas iguais seja identicamente nula, e que verá posteriormente que isso é essencial para definir derivada (que como se sabe se trata de diferenças de coisas muito próximas que era de se esperar que estivesse proximo de zero 'intervalar' essas diferenças a medida que aproximamos no limite de $h \rightarrow 0$).

Definição 1.9. (*gH-diferença de intervalos*) Tome A e B em $I(\mathbb{R})$, a *gH-diferença* entre A e B é definida como o elemento C em $I(\mathbb{R})$ tal que:

$$C = A \ominus_{gH} B = \begin{cases} A = B + C \\ \text{ou} \\ B = A - C \end{cases}$$

Ghosh[7] escreve em outras palavras, resumindo:

$$C = A \ominus_{gH} B = [\min\{\underline{a} - \underline{b}, \overline{a} - \overline{b}\}, \max\{\underline{a} - \underline{b}, \overline{a} - \overline{b}\}]$$

e a demonstração se encontra em Chalco-Cano [9]

Ghosh [7] define função paramétrica intervalar da seguinte forma:

Definição 1.10. *Seja $C_\nu^k = (C_1, C_2, \dots, C_k)^T$ (k números intervalares na forma de vetor com $k - \text{uplas}$) onde $C_j = [\underline{c}_j, \overline{c}_j]$ $j = 1, \dots, k$ uma função intervalar é uma função do tipo $F_{C_\nu^k} : \mathbb{R}^n \rightarrow I(\mathbb{R})$*

em outras palavras nada mais é do que uma função cujo os seus coeficientes são intervalos

Exemplo 1.5. *(funcional linear intervalar) é uma função intervalar no seguinte formato $L_{h_\nu^k} : \mathbb{R}^n \rightarrow I(\mathbb{R})$ com $h_\nu^k = (h_1, h_2, \dots, h_k)^T$ com $h_j = [\underline{h}_j, \overline{h}_j]$ $j = 1, \dots, k$ tal que $L_{h_\nu^k}(x) = \sum_{i=1}^n h_i \odot x_i$*

Exemplo 1.6. *Considere a seguinte função*

$F_{C_\nu^2}(x_1, x_2) = C_1 x_1 + C_2 e^{x_1}$, onde $C_1 = [1, 3]$ e $C_2 = [-2, 1]$
é uma função intervalar.

Definição 1.11. *(representação paramétrica de funções intervalares) Seja $F_{C_\nu^k} : \mathbb{R}^n \rightarrow I(\mathbb{R})$ vai ser dita uma função com parametros intervalares quando existirem k coeficientes intervalares $C_1, C_2, \dots, C_k$ denotados $C_\nu^k = (C_1, C_2, \dots, C_k)^T$ onde $C_j = [\underline{c}_j, \overline{c}_j]$ que podem ser representados:*

$$\{c(t) | c(t) = (c_1(t_1), c_2(t_2), \dots, c_k(t_k)), c_j(t_j) = \underline{c}_j + t_j(\overline{c}_j - \underline{c}_j), t = (t_1, t_2, \dots, t_k), 0 \leq t_j \leq 1, 1 \leq j \leq k\}$$

e a função pode agora ser escrita como:

$$F_{C_\nu^k} = \{f_{c(t)}(x) | f_{c(t)} : \mathbb{R}^n \rightarrow \mathbb{R}, c(t) \in C_\nu^k\}$$

Exemplo 1.7. *Considere a função anterior:*

$F_{C_\nu^2}(x_1, x_2) = C_1 x_1 + C_2 e^{x_1}$, onde $C_1 = [1, 3]$ e $C_2 = [-2, 1]$
que pode ser reescrita da seguinte forma;

$$f_{c(t)} = f_{c_1(t_1, c_2(t_2))}(x_1, x_2) = c_1(t_1)x_1 + c_2(t_2)e^{x_1} = (1 + 2t_1)x_1 + (-2 + 3t_2)e^{x_1}$$

com $0 \leq t_1, t_2 \leq 1$

com $0 \leq t_1, t_2 \leq 1$

Exemplo 1.8. *E também convenientemente pode-se escrever $F_{C_\nu^k}(x)$ da seguinte forma (desde que $f_{c(t)}$ seja continua no seu domínio $\forall t \in [0, 1]^k$)*

$$F_{C_\nu^k}(x) = [\underline{f}(x), \overline{f}(x)]$$

$$\text{onde } \underline{f}(x) = \min_{t \in [0, 1]^k} f_{c(t)}(x) \text{ and } \overline{f}(x) = \max_{t \in [0, 1]^k} f_{c(t)}(x)$$

Percebe-se que o intervalo sempre existe e é único desde que $f_{c(t)}$ seja continua no seu dominio pois como se sabe $t \in [0, 1]^k$ é compacto e $f_{c(t)}(x) = g(t)$ continua então pelo

Teorema de Weierstrass g possui máximo e mínimo em $t \in [0, 1]^k$ que coincide com o intervalo $F_{C_\nu^k}(x) = [\underline{f}(x), \overline{f}(x)]$ onde $\underline{f}(x) = \min_{t \in [0, 1]^k} f_{c(t)}(x)$ e $\overline{f}(x) = \max_{t \in [0, 1]^k} f_{c(t)}(x)$

Na definição a seguir, veremos que é possível trabalhar com n -uplas e matrizes onde cada entrada é um intervalo fechado e limitado da reta. No próximo capítulo será apresentado um Problema de Otimização Intervalar, definido no espaço vetorial intervalar n -dimensional que será definido posteriormente.

Definição 1.12. *Um vetor intervalar n -dimensional é uma n -upla ordenada onde cada coordenada é um intervalo.*

$$X = (X_1, X_2, \dots, X_n) \quad (1.22)$$

onde cada X_j é da forma $X_j = [\underline{X}_j, \overline{X}_j]$ com $j = 1, \dots, n$ Uma matriz intervalar

Definição 1.13. *A é uma matriz intervalar se cada entrada a_{ij} é um intervalo, isto é, $a_{ij} = [\underline{a}_{ij}, \overline{a}_{ij}]$.*

Definição 1.14. *o espaço vetorial intervalar é um espaço vetorial onde os elementos são vetores intervalares e é facilmente verificável que tal espaço é um espaço vetorial*

Capítulo 2

O Problema

2.1 Otimização Intervalar

Para falar de otimização intervalar pensando em usar posteriormente os métodos de newton e quase-newton precisa-se definir certas propriedades para tais funções intervalares tais como continuidade e diferenciabilidade pois só assim poderá posteriormente denominá-la função objetivo e poderá dizer se tem ou não pelo menos um ponto ótimo (no caso, eficiente).

2.1.1 Propriedades da Função Intervalar

Continuidade e diferenciabilidade são características fundamentais para classificar funções, mas para o caso de funções cuja a imagem é um intervalo não há unanimidade para a definição de continuidade, Panda [1] por exemplo definiu a continuidade da seguinte forma:

Definição 2.1. *Uma função $F_{C_\nu^k}$ é dita contínua no ponto x_0 se $f_{c(t)}$ (tal como definida na definição 1.9) é contínua para todo $t \in [0, 1]^k$, ou seja*

$$\lim_{x \rightarrow x_0} F_{C_\nu^k}(x) = F_{C_\nu^k}(x_0)$$

Ghosh também exhibe o seguinte resultado que pode facilitar o cálculo desse limite

$$\lim_{x \rightarrow x_0} F_{C_\nu^k}(x) = [\min_{t \in [0, 1]^k} \lim_{x \rightarrow x_0} f_{c(t)}(x), \max_{t \in [0, 1]^k} \lim_{x \rightarrow x_0} f_{c(t)}(x)]$$

Por outro lado Ghosh [7] define continuidade via gH -diferença pois como no seu texto justifica para certos casos de funções contínuas (quando fixado os coeficientes de tais função em números reais) deixam de ser contínuas quando os seus coeficientes são intervalos usando a definição acima mas quando ajustada para gH -diferença 'ganha' novamente a continuidade.

Definição 2.2. (*gH-continuidade*) Seja $x_0 = (x_1^0, x_2^0, \dots, x_n^0) \in D \setminus \partial D$ (interior do domínio de $F_{C_\nu^k}$) e tome $h = (h_1, \dots, h_n) \in \mathbb{R}^n$ de tal forma que $x + h \in D$. A função $F_{C_\nu^k}$ é dita *gH-continua* no ponto x_0 se existir um intervalo $E(F_{C_\nu^k}(x_0); h)$ com $\lim_{\|h\| \rightarrow 0} E(F_{C_\nu^k}(x_0); h) = 0$ tal que:

$$F_{C_\nu^k}(x_0 + h) = F_{C_\nu^k}(x_0) + E(F_{C_\nu^k}(x_0); h)$$

Moore [5] define diferenciabilidade da seguinte maneira:

Definição 2.3. (*Diferenciabilidade segundo Moore*) Uma função $F_{C_\nu^k} : D \rightarrow I(\mathbb{R}) \cup \overline{I(\mathbb{R})}$ é dita intervalar Diferenciável no ponto $x_0 \in D$ se existe o limite:

$$\lim_{h \rightarrow 0} \frac{f_{c(t)}(x_i^0 + h_i) - f_{c(t)}(x_i^0)}{\|h\|}$$

existe.

onde $F_{C_\nu^k} = \{f_{c(t)}(x) | f_{c(t)} : \mathbb{R}^n \rightarrow \mathbb{R}, c(t) \in C_\nu^k\}$ e $\overline{I(\mathbb{R})} = \{[a, b] | [b, a] \in I(\mathbb{R})\}$

Acontece que Ghosh [7] ver que essa definição implica na não diferenciabilidade de alguns funcionais lineares intervalares, por exemplo, suponha que $F_{C_\nu^k}(x_1, x_2) = [1, 2]x_1 + [1, 2]x_2$ agora aplicando na definição acima obtemos:

$$\begin{aligned} & \lim_{h \rightarrow 0} \frac{f_{c(t)}(x_i^0 + h_i) - f_{c(t)}(x_i^0)}{\|h\|} = \\ & = \lim_{h \rightarrow 0} \frac{(1 + t_1)(x_1 + h_1) + (1 + t_2)(x_2 + h_2) - (1 + t_1)x_1 + (1 + t_2)x_2}{\|h\|} = \end{aligned}$$

(lembrando que $(t_1, t_2) = t \in [0, 1]^2$)

$$= \lim_{h \rightarrow 0} \frac{[1, 2]h_1 + [1, 2]h_2}{\|h\|} =$$

Agora faça $h_2 = 0$ e $h_1 \rightarrow 0$ assim temos:

$$\lim_{h \rightarrow 0} \frac{[1, 2](h_1)}{|h_1|}$$

note que os limites laterais são:

$$\lim_{h^+ \rightarrow 0} \frac{[1, 2](h_1)}{|h_1|} = \lim_{h^+ \rightarrow 0} \frac{[1, 2](h_1)}{h_1} = [1, 2]$$

e por outro lado:

$$\lim_{h^- \rightarrow 0} \frac{[1, 2](h_1)}{|h_1|} = \lim_{h^- \rightarrow 0} \frac{[1, 2](h_1)}{-h_1} = [-2, 1]$$

logo o limite não existe e temos que $F_{C_\nu^k}(x_1, x_2) = [1, 2]x_1 + [1, 2]x_2$ não é diferenciável que seria pouco usual pois de modo geral em dimensão finita se espera que funcionais lineares sejam diferenciáveis, por esse motivo Ghosh [7] usa a diferença de hukuraha para estabelecer uma diferenciabilidade para funções cuja a imagem é um intervalo, da seguinte maneira:

Definição 2.4. (*gH-derivada parcial*) Seja $x_0 = (x_1^0, x_2^0, \dots, x_n^0)$ um ponto fixado porém arbitrário no interior do domínio D da função $F_{C_\nu^k}(x)$ e tome $h = (h_1, \dots, h_n) \in \mathbb{R}^n$ de tal forma que $x + h \in D$. Agora definimos $\phi_i(x_i) = F_{C_\nu^k}(x_1^0, \dots, x_{i-1}^0, x_i, x_{i+1}^0, \dots, x_n^0)$ dizemos que F é Hukuhara derivável parcialmente na i -ésima componente no ponto x_i^0 se o limite:

$$\lim_{h_i \rightarrow 0} \frac{\phi_i(x_i^0 + h_i) \ominus_{gH} \phi_i(x_i^0)}{h_i}$$

existe.

Denota-se a i -ésima gH - derivada parcial de $F_{C_\nu^k}$ em x_0 por $D_i F_{C_\nu^k}(x_0)$ onde $i = 1, 2, \dots, n$ Assim fazendo o i variar $i = 1, 2, \dots, n$ obtem-se o gradiente:

Definição 2.5. (*gH-gradiente*) O gH -gradiente de uma função intervalar $F_{C_\nu^k}$ no ponto $x_0 \in D$ é definido como sendo o seguinte vetor intervalar:

$$(D_1 F_{C_\nu^k}(x_0), D_2 F_{C_\nu^k}(x_0), \dots, D_n F_{C_\nu^k}(x_0))^T$$

Denota-se o gH -gradiente como $\nabla F_{C_\nu^k}(x_0)$

Agora com uma derivada mais consistente pode-se definir a diferenciabilidade de uma função cuja a imagem é um intervalo, que se chamará gH -Diferenciável

Definição 2.6. (*gH-Diferenciável*) Uma função $F_{C_\nu^k} : D \rightarrow I(\mathbb{R})$ é dita gH - Diferenciável no ponto $x_0 \in D$ se existem dois intervalos $E(F_{C_\nu^k}(x_0); h)$ e $L_{x_0} : \mathbb{R}^n \rightarrow I(\mathbb{R})$ tal que:

$$F_{C_\nu^k}(x_0 + h) \ominus_{gH} F_{C_\nu^k}(x_0) = L_{x_0}(h) \oplus \|h\| \odot E(F_{C_\nu^k}(x_0); h) \quad (2.1)$$

para $\|h\| < \delta$ e $\delta > 0$ onde $\lim_{\|h\| \rightarrow 0} E(F_{C_\nu^k}(x_0); h) = 0$

Além de L_{x_0} existir, deve ser um funcional linear intervalar.

Espera-se que as propriedades de diferenciabilidade que se tem para funções de $\mathbb{R}^n$ em $\mathbb{R}$ também valham para funções de $\mathbb{R}^n$ em $I(\mathbb{R})$. De fato, os resultados apresentados a seguir, são obtidos usando a gH -Diferença

Teorema 2.1. *Seja a função $F_{C_\nu^k}$ gH -Diferenciável em x_0 . Então L_{x_0} existe para todo $h = (h_1, \dots, h_n) \in \mathbb{R}^n$ e:*

$$L_{x_0}(h) = \sum_{i=1}^n h_i D_i F_{C_\nu^k}(x_0) \quad (2.2)$$

Demonstração. Se $h = (0, 0, \dots, 0)^t$ tem-se o caso trivial e o teorema estaria imediatamente provado, sendo assim, assume-se que $h \neq (0, 0, \dots, 0)^t$

Sendo $F_{C_\nu^k}$ diferenciável no x_0 tem-se que:

$$F_{C_\nu^k}(x_0 + h) \ominus_{gH} F_{C_\nu^k}(x_0) = L_{x_0}(h) \oplus \|h\| \odot E(F_{C_\nu^k}(x_0); h) \quad (2.3)$$

para $\|h\| > \delta$ para algum $\delta > 0$ onde o limte:

$$\lim_{h \rightarrow 0} E(F_{C_\nu^k}(x_0); h) = 0$$

da equação acima tome $h = t\nu$ assim substituindo tem-se:

$$\begin{aligned} \frac{1}{t} \odot (F_{C_\nu^k}(x_0 + t\nu) \ominus_{gH} F_{C_\nu^k}(x_0)) &= \frac{1}{t} (L_{x_0}(h) \oplus \|t\nu\| \odot E(F_{C_\nu^k}(x_0); h)) \\ \Rightarrow \lim_{t \rightarrow 0} \frac{1}{t} \odot (F_{C_\nu^k}(x_0 + t\nu) \ominus_{gH} F_{C_\nu^k}(x_0)) &= L_{x_0}(\nu) \end{aligned}$$

Em particular se $\nu = e_i$ obtem-se que:

$$\Rightarrow \lim_{t \rightarrow 0} \frac{1}{t} \odot (F_{C_\nu^k}(x_0 + te_i) \ominus_{gH} F_{C_\nu^k}(x_0)) = L_{x_0}(e_i)$$

assim perceba que o resultado acima é exatamente $D_i F_{C_\nu^k}(x_0)$ como isso pode-se escrever como:

$$L_{x_0}(h) = L_{x_0}(h_1, h_2, \dots, h_n) = L_{x_0}(h_1 e_1 + h_2 e_2 + \dots + h_n e_n)$$

agora aplicando as propriedades de L_{x_0} :

$$L_{x_0}(h_1 e_1 + h_2 e_2 + \dots + h_n e_n) = h_1 L_{x_0}(e_1) + h_2 L_{x_0}(e_2) + \dots + h_n L_{x_0}(e_n)$$

que por fim substituindo o encontrado acima tem-se:

$$\begin{aligned} h_1 L_{x_0}(e_1) + h_2 L_{x_0}(e_2) + \dots + h_n L_{x_0}(e_n) &= h_1 D_1 F_{C_\nu^k}(x_0) + h_2 D_2 F_{C_\nu^k}(x_0) + \dots + h_n D_n F_{C_\nu^k}(x_0) \\ &= \end{aligned}$$

$$L_{x_0}(h) = \sum_{i=1}^n h_i D_i F_{C_\nu^k}(x_0)$$

□

Proposição 2.1. *Seja $x_0 \in D$ e a função $F_{C_\nu^k}$ gH -Diferenciável, então a função a valores*

reais:

$$\Psi(x) = \int_{t_k=0}^1 \int_{t_{k-1}=0}^1 \dots \int_{t_1=0}^1 f_{(c_1), c_2, \dots, c_k}(x) dt_1 \dots dt_k \quad (2.4)$$

é diferenciável em x_0

Demonstração. Como a função $F_{C_\nu^k}$ é diferenciável no ponto $x_0 \in D$ então $F_{C_\nu^k}$ é continua em $x_0 \in D$.

Agora observe que cada $c_j(t_j) = \underline{c}_j + t_j(\overline{c}_j - \underline{c}_j)$ é uma função linear em $t \in [0, 1]^k$

A prova é agora seguida a partir da observação de que $f_{(c_1), c_2, \dots, c_k}$ é diferenciável em $(t_1, \dots, t_k)$ sobre o compacto $t \in [0, 1]^k$ pois é composta de funções diferenciáveis, logo sua derivada existe (podendo assim derivar dentro da integral) e é continua, consequentemente integrável.

□

De agora em diante esse trabalho será apoiado pela Ψ tendo em vista suas propriedades que serão herdadas da $F_{C_\nu^k}$.

Posteriormente será necessário o uso da matriz Hessiana para os métodos numéricos então a definição a seguir faz-se necessária:

Definição 2.7. (*gH-Hessiana*) Seja a função $F_{C_\nu^k}$ duas vezes *gH-Diferenciável* em x_0 , ou seja, para cada i , $D_i F_{C_\nu^k}$ é *gH-Diferenciável* em x_0 . A *gH-derivada* parcial de segunda ordem é calculada da seguinte maneira: $D_{ij}^2 F_{C_\nu^k} = \left\{ \frac{\partial^2 f_{c(t)}}{\partial x_i \partial x_j} | c(t) \in C_\nu^k \right\}$. Assim a *gH-Hessiana* de $F_{C_\nu^k}$ é definida como uma matriz intervalar com $n \times n$ elementos.

2.1.2 O Problema Geral

Definição 2.8. Em geral um problema de Otimização Intervalar irrestrito (IOP) é escrito da seguinte maneira:

$$\min_{x \in \mathbb{R}^n} F_{C_\nu^k}(x)$$

onde $F_{C_\nu^k} : \mathbb{R}^n \rightarrow I(\mathbb{R})$, e para aplicar métodos de solução através de algoritmos posteriores iremos pedir que $F_{C_\nu^k}$ seja duas vezes *gH-diferenciável*

Será apresentada uma relação de ordem parcial para comparação de dois intervalos e para isso será definida uma ordem parcial no conjunto $I(\mathbb{R})$ que Ghosh [7] define:

Definição 2.9. (*Comparação de elementos em $I(\mathbb{R})$*) Sejam A e B dois elementos de $I(\mathbb{R})$ então:

1. B é dito dominado por A se $a(t) \leq b(t) \forall t \in [0, 1]$ onde $a(t) = t\underline{A} + (1-t)\overline{A}$ e $b(t) = t\underline{B} + (1-t)\overline{B}$ escrevemos $A \preceq B$
2. Dizemos que $A \neq B$ se existir um $t_0 \in [0, 1]$ tal que $a(t_0) \neq b(t_0)$
3. B é dito estritamente dominado por A se $A \preceq B$ e $A \neq B$. escreve-se $A \prec B$

Agora já sabendo comparar dois intervalos, assumindo essa ordem parcial (com a definição de uma ordem parcial), se pode definir o que seria uma solução eficiente para o IOP:

Definição 2.10. (solução eficiente) Uma solução factível $\bar{x} \in \mathbb{R}^n$ é dita solução eficiente local do problema IOP se não existir algum x_ϵ numa vizinhança $N_\epsilon(\bar{x})$ tal que $F_{C_\nu^k}(x_\epsilon) \prec F_{C_\nu^k}(\bar{x})$.

A seguir veremos o principal teorema usado nesse trabalho onde será mostrado que $\Psi(x)$ e $F_{C_\nu^k}(x)$ estão interligadas de alguma forma, no que diz respeito a minimizadores e soluções eficientes.

Teorema 2.2. (Caracterização de soluções eficientes) Seja $F_{C_\nu^k}$ uma função intervalar. E seja $\bar{x}$ um minimizador local de $\Psi(x) = \int_{t \in [0,1]^k} f_{c(t)}(x) dt$ então $\bar{x}$ é uma solução eficiente local do problema $\min_{x \in \mathbb{R}^n} F_{C_\nu^k}(x)$.

Demonstração. Suponha por absurdo que $\bar{x}$ seja minimizador de Ψ mas não seja uma solução eficiente local do problema IOP, assim deve existir algum x_δ em $N_\delta(\bar{x})$ (uma vizinhança de $\bar{x}$) para algum δ tal que $F_{C_\nu^k}(x_\delta) \prec F_{C_\nu^k}(\bar{x})$ portanto $f_{c(t)}(x_\delta) < f_{c(t)}(\bar{x}) \forall t \in [0, 1]^k$ consequentemente

$$\Psi(x_\delta) = \int_{t \in [0,1]^k} f_{c(t)}(x_\delta) dt < \int_{t \in [0,1]^k} f_{c(t)}(\bar{x}) dt$$

que é claramente absurdo posto que por hipótese $\bar{x}$ é minimizador de Ψ □

Perceba que o teorema anterior não garante a volta, ou seja, se $\bar{x}$ é solução eficiente não necessariamente ele é um minimizador. Para essa questão será proposto uma nova abordagem em uma seção futura onde o problema é atacado sem explorar a Ψ visto que assim poderia deixar escapar alguma solução eficiente dado que todo minimizador é eficiente mas nem todo ponto eficiente é minimizador.

Métodos Numéricos

Método de Newton

Ghosh [7] usa as duas versões do método de Newton para a busca de um ponto eficiente e a primeira a ser testada é a iteração $x^{m+1} = x^m - [\nabla^2 \Psi(x^m)]^{-1} \nabla \Psi(x^m)$ mas para diminuir o numero de passos iterativos considera-se um λ_m (peso) em cada passo onde $\lambda_m = \operatorname{argmin}\{\Psi(x^0 + \lambda_m x^m)\}$ assim as novas iterações ficam $x^{m+1} = x^m - \lambda_m [\nabla^2 \Psi(x^m)]^{-1} \nabla \Psi(x^m)$ mas, inicialmente considera-se o método clássico.

Como se sabe o método de newton adaptado a otimização resolve problemas irrestritos de funções definidas em $\mathbb{R}^n$ a valores reais. Para encontrar pontos eficientes de $F_{C_v^k}$ recorreremos a sua $\Psi(x) = \int_{t_k \in [0,1]^k} f_{c(t)}(x) dt$ a qual é diferenciável duas vezes (caso $F_{C_v^k}$ seja) e seus pontos minimizadores são exatamente as soluções eficientes de $F_{C_v^k}$ (contudo vale ressaltar que se x^{**} é solução eficiente não garante que é minimizador de Ψ mas a reciproca é verdadeira)

Considera-se $\Psi(x) = \int_{t_k \in [0,1]^k} f_{c(t)}(x) dt$ com $F_{C_v^k}$ duas vezes gH - Diferenciável, logo $\Psi(x)$ é também duas vezes diferenciável. Assim faz-se o método de newton clássico:

1. defina um ϵ para critério de parada e defina um "start"(x^0) .
2. se $\| \nabla \Psi(x^m) \| \leq \epsilon$ pare.
3. caso contrário $x^{m+1} = x^m - [\nabla^2 \Psi(x^m)]^{-1} \nabla \Psi(x^m)$
4. volte ao passo 2

Agora o método melhorado com os pesos λ_m

Considera-se $\Psi(x) = \int_{t_k \in [0,1]^k} f_{c(t)}(x) dt$ com $F_{C_v^k}$ duas vezes gH - Diferenciável, logo $\Psi(x)$ é também duas vezes diferenciável. assim pode adaptar o método de quase-newton da seguinte maneira:

1. defina um ϵ para critério de parada e defina um "start"(x^0) .
2. se $\| \nabla \Psi(x^m) \| \leq \epsilon$ pare.
3. caso contrário $x^{m+1} = x^m - \lambda_m [\nabla^2 \Psi(x^m)]^{-1} \nabla \Psi(x^m)$
4. volte ao passo 2

Onde

$$\lambda_m = \operatorname{arg} \min_{\lambda_m \leq 0} \Psi(x^m + \lambda_m \nabla \Psi(x^m))$$

.

Vale ressaltar que se a função Ψ for do tipo quadrática, ou seja; $\Psi(x) = x^t A x + b^t x + c$ qualquer um dos métodos acima converge em uma iteração se A for semidefinida positiva

(prova em Ghosh [7])

Ghosh [7] resolveu dois exemplos usando cada um dos algoritmos e os resultados foram os mesmos, entretanto quando usado o peso λ_m a quantidade de passos diminuiu bastante. O primeiro exemplo foi o seguinte

$$\min F_{C_p^5}(x_1, x_2) = [1, 2]x_1 + [-1, 2]x_2 + [1, 3]x_1^2 + [0, 1]x_1x_2 + [2, 3]x_2^2, x_1 \in \mathbb{R}, x_2 \in \mathbb{R}.$$

A função intervalar quadrática $F_{C_p^5}$ pode ser escrita em termos da combinação convexa dos extremos dos intervalos tal como foi feito no capítulo anterior da seguinte forma:

$$f_{c(t)} = (1 + t_1)x_1 + (-1 + 3t_2)x_2 + (1 + 2t_3)x_1^2 + t_4x_1x_2 + (2 + t_5)x_2^2$$

onde

$$t = (t_1, \dots, t_5) \in [0, 1]^5$$

Agora para encontrar a função auxiliar $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ com $\Psi \in C^\infty$ basta integrar $f_{c(t)}$ em relação a t no "cubo" $t = (t_1, \dots, t_5) \in [0, 1]^k$ da seguinte forma:

$$\begin{aligned} \Psi(x_1, x_2) &= \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 f_{c(t)} dt_1 dt_2 dt_3 dt_4 dt_5 \\ &= \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 (1+t_1)x_1 + (-1+3t_2)x_2 + (1+2t_3)x_1^2 + t_4x_1x_2 + (2+t_5)x_2^2 dt_1 dt_2 dt_3 dt_4 dt_5 \end{aligned}$$

dessa forma integrando $f_{c(t)}$ obtemos $\Psi(x_1, x_2) = \frac{3}{2}x_1 + \frac{1}{2}x_2 + 2x_1^2 + \frac{1}{2}x_1x_2 + \frac{5}{2}x_2^2$

Agora aplicando o algoritmo 1 acima na função usando um nível de significancia $\epsilon = 10^{-4}$ e um "start" a partir do ponto $(x_0, y_0) = (3, 2)$ na primeira iteração o algoritmo já converge para o ponto $x^* = (x_1, y_1) = (-0.367, -0.063)$ que é uma solução eficiente do IOP acima considerado.

2.1.3 Método Quase-Newton (versão BFGS)

Essa proposta não foi explorada por Ghosh [7], entretanto é preciso reconsiderar que o método de newton assume, entre outros, o custo da Matriz Hessiana invertida a cada iteração sendo assim tornando-se inviável para problemas com dimensões superiores, computacionalmente falando.

Uma forma de "escapar" desse problema (de tornando-se inviável para problemas com dimensões superiores) é um método, muito popular devido a seu bom desempenho numérico, que é o método de Broyden, Fletcher, Goldfarb, Shanno (BFGS) (ver Friedlander [2])

Sendo assim assume-se as mesmas condições a priori, como objetivo ainda é encontrar pontos eficientes de $F_{C_\nu^k}$. Então pra isso considera-se a sua $\Psi(x) = \int_{t_k \in [0,1]^k} f_{c(t)}(x) dt$ a qual é diferenciável duas vezes (caso $F_{C_\nu^k}$ seja) e seus pontos minimizadores são exatamente as soluções eficientes de $F_{C_\nu^k}$

Considera-se $\Psi(x) = \int_{t_k \in [0,1]^k} f_{c(t)}(x) dt$ com $F_{C_\nu^k}$ duas vezes gH - Diferenciável, logo $\Psi(x)$ é também duas vezes diferenciável. assim façamos o método de quase-newton:

1. defina um ϵ para critério de parada e defina um "start"(x^0) .
2. se $\| \nabla \Psi(x^m) \| \leq \epsilon$ pare.
3. Calcule H_m usando a atualização BFGS
4. Calcule $x^{m+1} = x^m - \lambda_k H_m \nabla \Psi(x^m)$
5. volte ao passo 2

Onde H_m é uma matriz da mesma dimensão da Hessiana, encontrada pela seguinte expressão:

1. $H_0^{BFGS} = I$
2. $H_{m+1}^{BFGS} = H_m + \frac{1+q_m^t H_m q_m}{q_m^t p_m} \frac{p_m p_m^t}{p_m^t q_m} - \frac{p_m q_m^t H_m + H_m q_m p_m^t}{q_m^t p_m}$

onde $p_m = x_{m+1} - x_m$ e $q_m = \nabla \Psi(x_{m+1}) - \nabla \Psi(x_m)$ e $\lambda_m = \arg \min_{\lambda_m \leq 0} \Psi(x^m + \lambda_m \nabla \Psi(x^m))$. Próximo teorema será visto que esse método é consistente no que diz respeito a sua convergencia.

Teorema 2.3. *Seja $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ com $\Psi \in C^3$ tal que existe x^* minimizador local de Ψ com $\nabla^2 \Psi(x^*)$ definida positiva. Existem $\delta > 0$ e $\epsilon > 0$ tais que se $x^0 \in \mathbb{B}(x^*, \delta)$ e $\|H_0 - \nabla^2 \Psi(x^*)\| < \epsilon$ as sequencias $\{x^m\}$ e $\{H_m\}$ gerada pelo método BFGS usando $\lambda_m = 1 \forall m \in \mathbb{N}$ no passo 2, verifica-se:*

1. H_m é definida positiva $\forall m \in \mathbb{N}$
2. $\lim x_m = x^*$
3. a ordem de convergência é pelo menos super linear.

Demonstração. Ver Dennis e Schnabel [8]

□

Capítulo 3

Algumas simulações

3.1 Testando Método de Newton

Exemplo 1 (newton)

Ghosh [7] resolveu dois exemplos usando cada um dos algoritmos e os resultados foram os mesmos, entretanto quando usado o peso λ_m a quantidade de passos diminuiu bastante. O primeiro exemplo foi o seguinte $\min F_{C^5_\nu}(x_1, x_2) = [1, 2]x_1 + [-1, 2]x_2 + [1, 3]x_1^2 + [0, 1]x_1x_2 + [2, 3]x_2^2, x_1 \in \mathbb{R}, x_2 \in \mathbb{R}$.

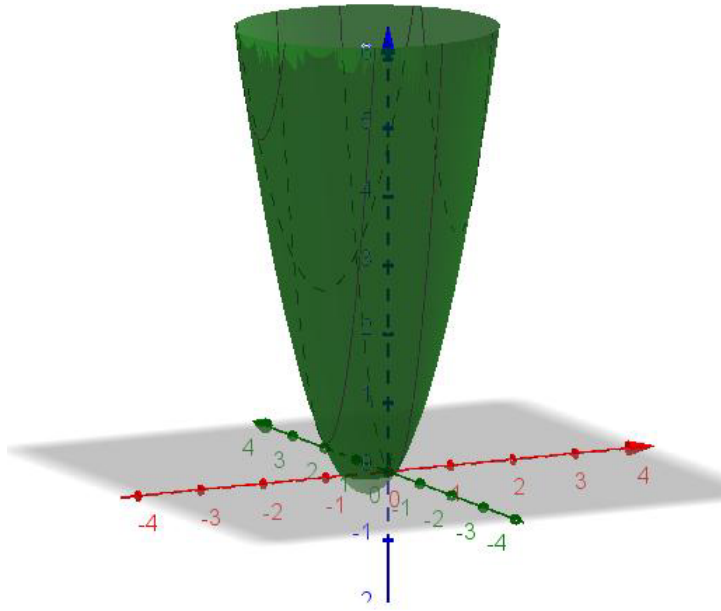
A função intervalar quadrática $F_{C^5_\nu}$ obviamente que pode ser escrita em termos da combinação convexa dos extremos dos intervalos tal como foi feito no capítulo anterior dessa forma:

$$f_{c(t)} = (1 + t_1)x_1 + (-1 + 3t_2)x_2 + (1 + 2t_3)x_1^2 + t_4x_1x_2 + (2 + t_5)x_2^2$$

onde $t = (t_1, \dots, t_5) \in [0, 1]^k$ Agora para encontrar a função auxiliar afim de que tenha-se uma função $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ e $\Psi \in C^\infty$ basta integrar $f_{c(t)}$ em relação a t no "cubo" $t = (t_1, \dots, t_5) \in [0, 1]^k$ dessa maneira:

$$\int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 (1 + t_1)x_1 + (-1 + 3t_2)x_2 + (1 + 2t_3)x_1^2 + t_4x_1x_2 + (2 + t_5)x_2^2 dt_1 dt_2 dt_3 dt_4 dt_5$$

dessa forma integrando $f_{c(t)}$ obtemos $\Psi(x_1, x_2) = \frac{3}{2}x_1 + \frac{1}{2}x_2 + 2x_1^2 + \frac{1}{2}x_1x_2 + \frac{5}{2}x_2^2$



Agora aplicando o algoritmo 1 na função acima usando um nível de significância $\epsilon = 10^{-4}$ e um "start" a partir do ponto $(x_0, y_0) = (3, 2)$ na primeira iteração o algoritmo já converge para o ponto $x^* = (x_1, y_1) = (-0.367, -0.063)$ que é uma solução eficiente do IOP acima considerado. A seguir vemos a tabela de desempenho do algoritmo quando aplicado a esse mesmo exemplo:

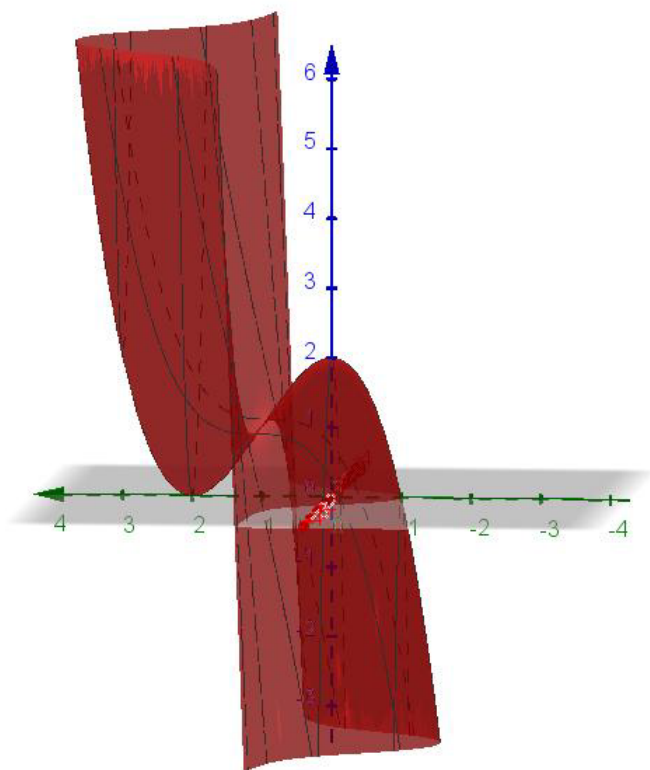
m	(x_1^m, x_2^m)	$F_{C_\nu^5}(x_1^m, x_2^m)$	$\nabla\Phi(x_1^m, x_2^m)$	$\ \nabla\Phi(x_1^m, x_2^m)\ $
0	(3,2)	[18,54]	$(14.500, 12.000)^t$	18.822
1	$(-0.367, -0.063)$	$[-0.718, 0.136]$	$10^{-15} * (-0.666, 0.444)^t$	$8.006 * 10^{-16}$

Exemplo 2 (newton)

O segundo exemplo trata de uma quadrática em $\mathbb{R}^2$ intervalar no seguinte formato: $\min F_{C_p^5}(x_1, x_2) = [-3, 0]x_1^2 + [0, 1]x_2^3 + [-2, -1]x_2^2 + [1, 2]x_1^2x_2 + [1, 3]$, $x_1 \in \mathbb{R}, x_2 \in \mathbb{R}$. e assim como no exemplo anterior tem-se que primeiramente escrever $F_{C_p^5}$ em termos das combinações convexas dos extremos dos respectivos intervalos: $f_{c(t)}(x_1, x_2) = (-3 + 3t_1)x_1^2 + t_2x_2^3 + (-2 + t_3)x_2^2 + (1 + t_4)x_1^2x_2 + (1 + 2t_5)$, $x_1 \in \mathbb{R}, x_2 \in \mathbb{R}$.

dessa forma, assim como no exemplo anterior, integrando $f_{c(t)}$ obtemos

$$\Psi(x_1, x_2) = -\frac{3}{2}x_1^2 + \frac{1}{2}x_2^3 + 2 - \frac{3}{2}x_2^2 + \frac{3}{2}x_1^2x_2$$



Agora calculando a sequência $\{x^m\}$ para encontrar o ponto eficiente $\bar{x}$ usará-se a seguinte sequência:

$$x^{m+1} = x^m - [\nabla^2 \Psi(x^m)]^{-1} \nabla \Psi(x^m)$$

onde:

$$\nabla \Psi(x^m) = \begin{pmatrix} \frac{\partial \Psi(x^n)}{\partial x_1} \\ \frac{\partial \Psi(x^n)}{\partial x_2} \end{pmatrix}$$

e

$$\nabla^2 \Psi(x^m) = \begin{pmatrix} \frac{\partial^2 \Psi(x^n)}{\partial x_1^2} & \frac{\partial^2 \Psi(x^n)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 \Psi(x^n)}{\partial x_1 \partial x_2} & \frac{\partial^2 \Psi(x^n)}{\partial x_2^2} \end{pmatrix}$$

Como percebe-se, será feito o uso do primeiro algoritmo com um nível de significância $\epsilon = 10^{-4}$ ou seja, quando $\|\nabla \Psi(x_1^m, x_2^m)\|$ for menor que $\epsilon = 10^{-4}$ o algoritmo para nesse ponto (x_1^m, x_2^m) (que será a solução eficiente do IOP) e usando um "start" $(x_1^m, x_2^m) = (-2, 4)$ (que é de onde começará as iterações) será feita a busca por $\bar{x}$. A tabela abaixo mostra que depois de 5 iterações foi encontrado $\bar{x} = (\bar{x}_1, \bar{x}_2) = (0, 2)$ como solução eficiente do IOP. (dado que $(0, 2)$ é minimizador de Ψ)

m	(x_1^m, x_2^m)	$FC_p^5(x_1^m, x_2^m)$	$\nabla \Psi(x_1^m, x_2^m)$	$\ \nabla \Psi(x_1^m, x_2^m)\ $
0	(-2,4)	[27,83]	$(-18, 18)^t$	25.456
1	$(-0.8, -2.8)$	$[-14.808, 20.696]$	$(4.32, -4.32)^t$	6.11
2	$(-0.246, -2.246)$	$[-9.136, 9.559]$	$(9.2, -9.2)^t$	1.301
3	$(-0.041, -2.041)$	$[-7.329, 7.340]$	$(0.127, -0.127)^t$	0.179
4	$(-0.002, -2.002)$	$[-7.012, 7.012]$	$(0.004, -0.004)^t$	0.006
5	$(0, 2)$	$[-7, 7]$	$10^{-4*}(0.069, , -0.069,)^t$	$0.983 * 10^{-5}$

Agora calculando a sequência $\{x^m\}$ de outra maneira para encontrar o ponto eficiente $\bar{x}$ usará-se a seguinte sequência:

$$x^{m+1} = x^m - \lambda_m [\nabla^2 \Psi(x^m)]^{-1} \nabla \Psi(x^m)$$

onde:

$$\lambda_m = \arg \min_{\lambda_m \leq 0} \Psi(x^m + \lambda_m d^m)$$

e

$$d^m = -[\nabla^2 \Psi(x^m)]^{-1} \nabla \Psi(x^m)$$

e

$$\nabla \Psi(x^m) = \begin{pmatrix} -3x_1 - 3x_1x_2 \\ -1.5x_2 + 1.5x_1^2 \end{pmatrix}$$

e

$$\nabla^2 \Psi(x^m) = \begin{pmatrix} -3 - 3x_2 & -3x_1 \\ -3x_1 & -1.5 \end{pmatrix}$$

Como percebe-se, será feito o uso do segundo algoritmo com o mesmo nível de significancia anterior $\epsilon = 10^{-4}$ ou seja, quando $\|\nabla\Psi(x_1^m, x_2^m)\|$ for menor que $\epsilon = 10^{-4}$ o algoritmo para nesse ponto (x_1^m, x_2^m) (que será a solução eficiente do IOP) e usando o mesmo "start" anterior $(x_1^m, x_2^m) = (-2, 4)$ (que é de onde começará as iterações) será feita a busca por $\bar{x}$. A tabela abaixo mostra que depois de 1 iteração foi encontrado, como era de se esperar, o mesmo ponto $\bar{x} = (\bar{x}_1, \bar{x}_2) = (0, 2)$ como solução eficiente do IOP. (dado que $(0, 2)$ é minimizador de Ψ)

m	(x_1^m, x_2^m)	$F_{C_v^5}(x_1^m, x_2^m)$	$\nabla\Psi(x_1^m, x_2^m)$	$\ \nabla\Psi(x_1^m, x_2^m)\ $
0	(-2,4)	[27,83]	$(-18, 18)^t$	25.456
1	(0, 2)	$[-7, 7]$	$10^{-4*}(0.069, , -0.069,)^t$	$0.983 * 10^{-5}$

3.2 Testando Método Quase-Newton

A título de comparação será aplicado o Método Quase-Newton para os mesmos exemplos que Ghosh [7] aplicou na seção anterior para a função quadrática e cúbica intervalares $\min F_{C^5_\nu}(x_1, x_2) = [1, 2]x_1 + [-1, 2]x_2 + [1, 3]x_1^2 + [0, 1]x_1x_2 + [2, 3]x_2^2, x_1 \in \mathbb{R}, x_2 \in \mathbb{R}$. e $G_{C^5_\nu}(x_1, x_2) = [-3, 0]x_1^2 + [0, 1]x_2^3 + [-2, -1]x_2^2 + [1, 2]x_1^2x_2 + [1, 3], x_1 \in \mathbb{R}, x_2 \in \mathbb{R}$.

Exemplo 1 (quase-newton)

Como já citado, Ghosh [7] não resolveu nenhum dos dois exemplos usando quase-newton e os resultados foram os mesmos, entretanto quando usado o peso λ_m a quantidade de passos diminuiu bastante. O primeiro exemplo foi o seguinte $\min F_{C^5_\nu}(x_1, x_2) = [1, 2]x_1 + [-1, 2]x_2 + [1, 3]x_1^2 + [0, 1]x_1x_2 + [2, 3]x_2^2, x_1 \in \mathbb{R}, x_2 \in \mathbb{R}$.

A função intervalar quadrática $F_{C^5_\nu}$ também será escrita em termos da combinação convexa dos extremos dos intervalos tal como foi feito no capítulo anterior dessa forma:

$$f_{c(t)} = (1 + t_1)x_1 + (-1 + 3t_2)x_2 + (1 + 2t_3)x_1^2 + t_4x_1x_2 + (2 + t_5)x_2^2$$

onde $t = (t_1, \dots, t_5) \in [0, 1]^k$ Agora para encontrar a função auxiliar afim de que tenha-se uma função do tipo $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ e $\Psi \in C^\infty$ basta integrar $f_{c(t)}$ em relação a t no "cubo" $t = (t_1, \dots, t_5) \in [0, 1]^k$ do mesmo modo tem-se:

$$= \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 (1+t_1)x_1 + (-1+3t_2)x_2 + (1+2t_3)x_1^2 + t_4x_1x_2 + (2+t_5)x_2^2 dt_1 dt_2 dt_3 dt_4 dt_5$$

dessa forma integrando $f_{c(t)}$ obtemos $\Psi(x_1, x_2) = \frac{3}{2}x_1 + \frac{1}{2}x_2 + 2x_1^2 + \frac{1}{2}x_1x_2 + \frac{5}{2}x_2^2$

Agora aplicando o algoritmo quase newton (versão BFGS) usando um nível de significancia $\epsilon = 10^{-4}$ e um "start" a partir do ponto $(x_0, y_0) = (3, 2)$ o algoritmo converge para o ponto $x^* = (x_1, y_1) = (-0.367, -0.063)$ depois de 5 iterações e já sabe-se de atemão que é uma solução eficiente do IOP acima considerado. A seguir vemos a tabela de desempenho do algoritmo quando aplicado a esse mesmo exemplo:

m	(x_1^m, x_2^m)	$F_{C^5_\nu}(x_1^m, x_2^m)$	$\nabla\Phi(x_1^m, x_2^m)$	$\ \nabla\Phi(x_1^m, x_2^m)\ $
0	[3,2]	[18,54]	[14.500, 12.000] ^t	18.822
1	[0.04, -0.45]	[-0.718, 0.136]	[1.43, -1.73] ^t	2.24
2	[-0.31, -0.028]	[18,54]	[0.25, 0.20] ^t	0.32
3	[-0.3601, -0.067]	[18,54]	[0.02, -0.03] ^t	0.038
4	[-0.3661, -0.062]	[18,54]	0.004, 0.0034] ^t	0.0054
5	[-0.367, -0.063]	[-0.718, 0.136]	10^{-5} [7.14, 5.91] ^t	10^{-5} 9.27

Exemplo 2 (quase-newton)

O segundo exemplo trata-se do mesmo polinomio bivariado de grau 3 intervalar no me formato: $\min G_{C_v^5}(x_1, x_2) = [-3, 0]x_1^2 + [0, 1]x_2^3 + [-2, -1]x_2^2 + [1, 2]x_1^2x_2 + [1, 3], x_1 \in \mathbb{R}, x_2 \in \mathbb{R}$. e assim como no exemplo anterior tem-se que primeiramente escrever $G_{C_v^5}$ em termos das combinações convexo dos extremos dos respectivos intervalos: $f_{c(t)}(x_1, x_2) = (-3 + 3t_1)x_1^2 + t_2x_2^3 + (-2 + t_3)x_2^2 + (1 + t_4)x_1^2x_2 + (1 + 2t_5), x_1 \in \mathbb{R}, x_2 \in \mathbb{R}$.

dessa forma, assim como no exemplo anterior, integrando $f_{c(t)}$ obtemos

$$\Psi(x_1, x_2) = -\frac{3}{2}x_1^2 + \frac{1}{2}x_2^3 + 2 - \frac{3}{2}x_2^2 + \frac{3}{2}x_1^2x_2$$

Agora calculando a sequência $\{x^m\}$ de outra maneira para encontrar o ponto eficiente $\bar{x}$ usará-se a seguinte sequência:

$$x^{m+1} = x^m - \lambda_m H_m \nabla \Psi(x^m)$$

onde:

$$\lambda_m = \arg \min_{\lambda_m \leq 0} \Psi(x^m + \lambda_m d^m)$$

e

$$d^m = -[\nabla^2 \Psi(x^m)]^{-1} \nabla \Psi(x^m)$$

e

$$\nabla \Psi(x^m) = \begin{pmatrix} \frac{\partial \Psi(x^m)}{\partial x_1} \\ \frac{\partial \Psi(x^m)}{\partial x_2} \end{pmatrix}$$

e

$$H_{m+1}^{BFGS} = H_m + \frac{1 + q_m^t H_m q_m}{q_m^t p_m} \frac{p_m p_m^t}{p_m^t q_m} - \frac{p_k q_m^t H_m + H_m q_m p_m^t}{q_m^t p_m}$$

onde $H_0 = I_n$ e $p_k = x_{k+1} - x_k$ e $q_k = \nabla \Psi(x_{k+1}) - \nabla \Psi(x_k)$ e $\lambda_m = \arg \min_{\lambda_m \leq 0} \Psi(x^m + \lambda_m d^m)$

Como percebe-se, será feito o uso do segundo algoritmo com o mesmo nível de significancia de todos anteriores $\epsilon = 10^{-4}$ ou seja, quando $\|\nabla \Psi(x_1^m, x_2^m)\|$ for menor que $\epsilon = 10^{-4}$ o algoritmo para nesse ponto (x_1^m, x_2^m) (que espera-se que seja o aproximadamente a mesma solução encontrada anteriormente pois será a solução eficiente do IOP) e usando o mesmo "start" anterior $(x_1^m, x_2^m) = (-2, 4)$ (que é de onde começará as iterações) será feita a busca por $\bar{x}$. A tabela abaixo mostra que depois de 1 iteração foi encontrado, como era de se esperar, o mesmo ponto $\bar{x} = (\bar{x}_1, \bar{x}_2) = (0, 2)$ como solução eficiente do IOP. (dado que $(0, 2)$ é minimizador de Ψ)

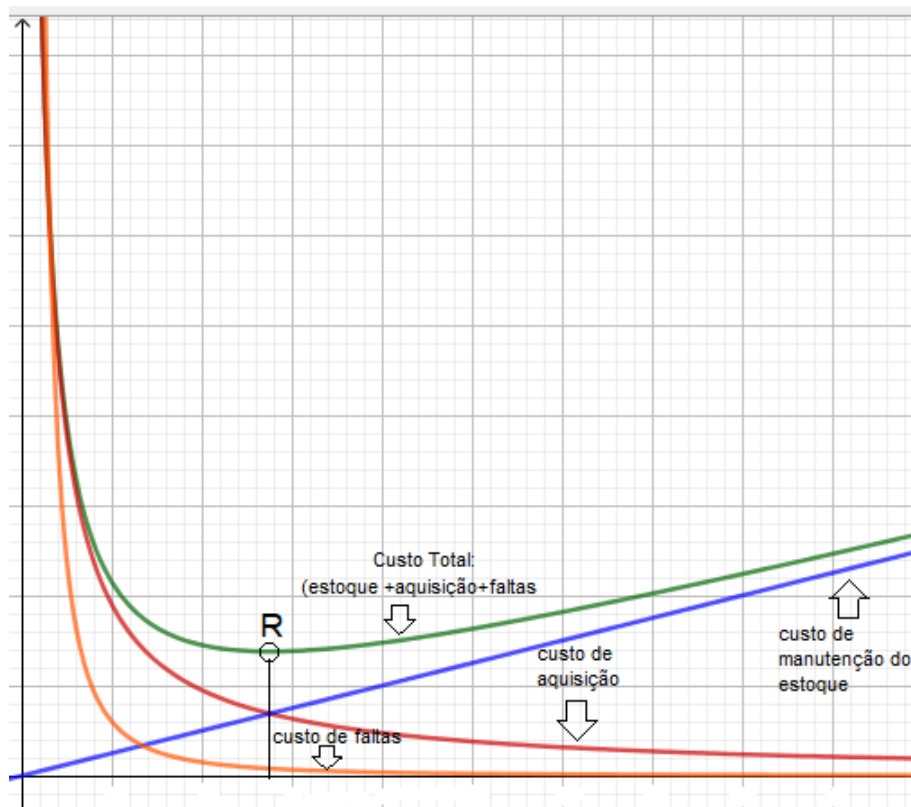
m	(x_1^m, x_2^m)	$F_{C^5_\nu}(x_1^m, x_2^m)$	$\nabla\Psi(x_1^m, x_2^m)$	$\ \nabla\Psi(x_1^m, x_2^m)\ $
0	(-2,4)	[27,83]	$(-18, 18)^t$	25.456
1	(0, 2)	$[-7, 7]$	$10^{-4*}(0.069, , -0.069,)^t$	$0.983 * 10^{-5}$

Exemplo 3 (Problema de Estoque)

Há muitas variáveis a se considerar para a modelagem de um problema de estoque, mas nesse caso será considerado o custo de manutenção, custo de aquisição e custo de faltas, e a função custo total é dada pela seguinte expressão:

$$C(x) = \frac{aK}{x} + \frac{xh}{2}$$

e terá o seguinte grafico:



onde a é a taxa de consumo, K é o custo de reabastecimento unitário, h é custo unitário de manutenção do estoque e x obviamente é a quantidade de unidades no estoque, e $C(x)$ é o custo associado a essas unidades no estoque. Claramente trata-se de um exemplo bem simplificado dos casos gerais para esse tipo de problema mas esse serve para ilustrar como é razoável pensar em problemas intervalares pois note que " a " (a taxa de consumo médio) dificilmente será constante e não há como prever a demanda exata, entretanto pode-se pensar que a esta compreendida em um intervalo, então assim é possível flexibilizar a ideia de obter uma resposta como um número fixo pensando dessa forma. Agora afim de encontrar o x^* tal que $C(x^*) \leq C(x) \forall x > 0$, suponha por um momento que a seja uma constante afim de calcular x^* , então olhando para função C nota-se que se trata de uma função polinomial que por sua vez é diferenciável infinitas vezes para $x > 0$, assim derivando $C(x)$ e igualando a zero tem-se que $x^* = (\frac{2aK}{h})^{\frac{1}{2}}$.

Agora voltando a raciocinar "intervalarmente" deve agora definir como calcular raiz quadrada de um intervalo, muito simples, dado que já se sabe multiplicar intervalos, sabe-se que se r é a raiz de a , então $r.r = a$, mas $r = [\underline{r}, \bar{r}]$ e $a = [\underline{a}, \bar{a}]$, dessa forma, $a^{\frac{1}{2}} = [\underline{a}^{\frac{1}{2}}, \bar{a}^{\frac{1}{2}}]$ desde que $\underline{a} > 0$ e $\bar{a} > 0$ dessa forma tem-se que o estoque ótimo é o intervalo dado por: $x^* = [(\frac{2K}{h})^{\frac{1}{2}}\underline{a}^{\frac{1}{2}}, (\frac{2K}{h})^{\frac{1}{2}}\bar{a}^{\frac{1}{2}}]$ pensando dessa forma ainda não sabe-se qual deve ser o valor do estoque, apenas que ele está compreendido nesse intervalo x^* , agora se pensar usando a função auxiliar Ψ (ou seja, integrando a $C(x)$ ao longo do intervalo $a = [\underline{a}, \bar{a}] = \underline{a}(1-t) + \bar{a}t$ onde $t \in [0, 1]$) tem-se que:

$$\begin{aligned}\Psi(x) &= \int_0^1 C_t(x) dt = \int_0^1 \frac{aK}{x} + \frac{xh}{2} dt = \\ &= \int_0^1 \frac{[\underline{a}, \bar{a}]K}{x} + \frac{xh}{2} dt = \\ &= \int_0^1 \frac{((1-t)\underline{a} + t\bar{a})K}{x} + \frac{xh}{2} dt = \\ &= \frac{K}{x} \frac{(\underline{a} + \bar{a})}{2} + \frac{xh}{2}\end{aligned}$$

agora fazendo $\Psi'(x) = 0$ obtem-se que $x^* = (\frac{(\underline{a} + \bar{a})K}{h})^{\frac{1}{2}}$ que se assemelha bastante com a forma original com a distinção de que " a " é um intervalo mas é como se escolhesse um representante desse intervalo, o seu ponto central, já definido no primeiro capítulo, e adotasse como se esse fosse o próprio coeficiente " a ".

Agora para analisar como seria achar esse mesmo x^* via método quase-newton, serão atribuídos valores a essas constantes e um intervalo para " a ", e assim será visto que x^* coincide com o encontrado analiticamente acima.

É claro que para a função

$$\Psi(x) = \frac{K}{x} \frac{\underline{a} + \bar{a}}{2} + \frac{xh}{2}$$

A implementação é trivial e desnecessária para esse caso tendo em vista que se trata de uma função univariáda e diferenciável como visto, mas suponha que se trate de um problema mais complexo de várias variáveis nesses casos faz-se necessário a implementação de algoritmos que resolvam de maneira não-manual e nesses casos seria necessário o método acima apresentados, pois sabe-se que em problemas gerais existem várias demandas ($(a_1, a_2, \dots, a_n)$ vetor intervalar) tais como suas respectivas quantidades $(x_1, x_2, \dots, x_n)$ que combinadas gerariam um custo associado $C(x)$ ou seja, trata-se de uma função $\mathbb{R}^n \rightarrow \mathbb{R}$ coerciva em $\mathbb{R}^{n+}$ pois se as quantidades $(x_1, x_2, \dots, x_n)$ forem infinitamente grandes, o custo para o estoque dessas quantidades também será infinitamente grande, e se essas quantidades forem nenhuma o custo também é infinitamente grande pois se não há estoque, não há lucro. Para casos mais gerais como citado fica em aberto para modelagem em trabalhos

futuros

5

Capítulo 4

Considerações Finais

Este capítulo destina-se a análise dos resultados obtidos a partir dos pequenos testes feitos nos capítulos anteriores, na busca de pontos críticos e minimizadores da função objetivo F_{C^k} . Em tudo, sempre foi encontrado tais minimizadores, em poucas iterações tanto com o método de newton quanto o método quase-newton, e em um dos casos o método quase-newton converge em menos iterações, que foi o caso do segundo exemplo, mostrando assim como em alguns casos pode se sobressair a frente do método de newton classico, além de que, em si próprio, o método quase-newton já exige menos do computador, pois não inverte matriz como já citado anteriormente, assim pode-se dizer que tem o melhor "custo-benefício" comparando aos dois.

4.1 A Primeira Simulação

Como visto esse primeiro exemplo foi uma função quadrática bivariada que como sabe-se para esse caso o método newton converge em duas iterações para o ponto $(-0.367, -0.063)$, entretanto nesse exemplo em específico o quase-newton precisou de 5 iterações para encontrar o mesmo minimizador que foi também o ponto $(-0.367, -0.063)$ lembrando que ambos partem do mesmo "start" que é o ponto $(0,2)$ e foi usado também o mesmo critério de para com $\epsilon = 0.0001$. Vale ressaltar também que, apesar de precisar apenas uma iteração, o método de newton precisou inverter uma matriz 2×2 que lhe somam 6 iterações internas (4 cofatores, 1 determinante e a multiplicação) dentro dessa iteração, sem falar que o número de iterações para esse método aumenta mais que exponencialmente a medida que a ordem da matriz aumenta.

Vale ressaltar que como o exemplo é relativamente pequeno pode-se considerar um empate no desempenho para esse caso pois seria imperceptível a diferença no tempo de convergência para esse caso, mas o que se tem de vantagem nisso é agora ter pelo menos duas saídas para esse tipo de problema que hora ou outra pode ou não convergir em ambos algoritmos.

4.2 A Segunda Simulação

Aqui nesse caso pode-se perceber que em certos casos há a possibilidade do método quase-newton precisar até de menos iterações. Apesar de como já dito, para esses casos pequenos se considera um empate em termos de desempenho mas é curioso ver que se pode escolher ambas opções de resolução que chega (de maneiras diferente) ao mesmo resultado partindo do mesmo "start" $(-2, 4)$ com mesmo nível de significancia $\epsilon = 0.0001$ ambos convergem para o ponto $(0, 2)$ que se trata de um mínimo local (de fato um minimizador) e vale ressaltar que a $F_{C_\nu^k}$ pode ter outros pontos eficientes que a Ψ não captou pois obviamente que há perdas de informação quando integrando a $F_{C_\nu^k}$ ao longo de todos os intervalos de incertezas, para essas questões (quem seria os outros pontos eficientes) será sugerido na próxima seção uma sugestão de como olhar o problema de outra forma sem passar pela integral de $F_{C_\nu^k}$ (ou seja sem olhar para Ψ) pensando na possibilidade de encontrar outros pontos eficientes (via definição de eficiente dita nessa dissertação) mas vale ressaltar também que há outras formas de comparar dois intervalos via uma transformação linear também explorada por Silveira [3] onde se transporta via uma transformação linear ϕ um vetor intervalar em $I(\mathbb{R}^n)$ a $\mathbb{R}^{2n}$ ambiente onde esta-se habituado a comparar parcialmente vetores.

4.3 Uma Nova Abordagem

Uma maneira totalmente diferente de abordar esse problema é encara-lo como um problema de otimização restrita ao invés de irrestrita afim de que possa encontrar, quem sabe, os pontos eficientes que eventualmente poderiam não ser minimizadores dado que o teorema que garante que todos os minimizadores são pontos eficientes não vale a volta, ou seja, pode ser que existam pontos eficientes de $F_{C_\nu^k}$ que não são minimizadores de Ψ . A forma que é feita é partir do mesmo problema de otimização intervalar

$$\min_{x \in \mathbb{R}^n} F_{C_\nu^k}(x)$$

onde $F_{C_\nu^k} : \mathbb{R}^n \rightarrow I(\mathbb{R})$, e novamente será pedido que $F_{C_\nu^k}$ seja duas vezes gH -diferenciável. A grande mudança é observar que $F_{C_\nu^k}$ pode ser escrita da seguinte forma

$$F_{C_\nu^k}(x) = \{f_{c(t)}(x) | f_{c(t)} : \mathbb{R}^n \rightarrow \mathbb{R}, c(t) \in C_\nu^k\} = G(t, x)|_{t \in [0,1]^k}$$

onde $G(t, x) = f_{c*(t)}(x)$, e $c*(t) = (c_1(t_1), c_2(t_2), \dots, c_k(t_k))$ em que $c_j(t_j) = \underline{c}_j + t_j(\overline{c}_j - \underline{c}_j)$, $t = (t_1, t_2, \dots, t_k) \in \mathbb{R}^k$

Agora o problema geral de otimização intervalar irrestrito pode ser escrito da seguinte

forma:

$$\min_{t \in [0,1]^k} \min_{x \in \mathbb{R}^n} G(t, x)$$

pode também escrever da forma classica

$$\min G(t, x)$$

$$s.a : t \in [0, 1]^k, x \in \mathbb{R}^n$$

Olhando agora para $G(t, x)$, perceba que se trata de uma função $G : \mathbb{R}^{k+n} \rightarrow \mathbb{R}$ também duplamente diferenciável pelos teoremas vistos anteriormente e é evidente que para esse tipo de problema há a programação não-linear com diversas ferramentas para resolver tal problema mas a questão que fica é se tais minimizadores do problema restrito são pontos eficientes do problema original e se nesse formato garante encontrar todos (ou pelo menos mais do que usando a Ψ) os pontos eficientes. Essas perguntas não serão respondidas nesse trabalho mas perceba que é pertinente e ficam, a principio, para trabalhos futuros.

Capítulo 5

Conclusão

Apesar de parecer insignificante em termos de tempo para encontrar a mesma solução via método quase-newton, deve se pensar em casos gerais onde há centenas de variáveis e milhares constantes nas componentes da matriz hessiana que dificultariam e muito sua inversão, em alguns casos pode ser impossível, e nesses casos que o método quase-newton pode se sobressair pois se trata de uma rotina que não inverte matrizes, mas um outro problema é a obtenção de soluções eficientes, pois para cada "start" o algoritmo acha uma (ou nenhuma) solução eficiente para o problema inicial e disso temos dois problemas, primeiro é que como saber se achou todos os minimizadores de Ψ mesmo supondo que tenha achado todos eles, como saber se achou todos os minimizadores?

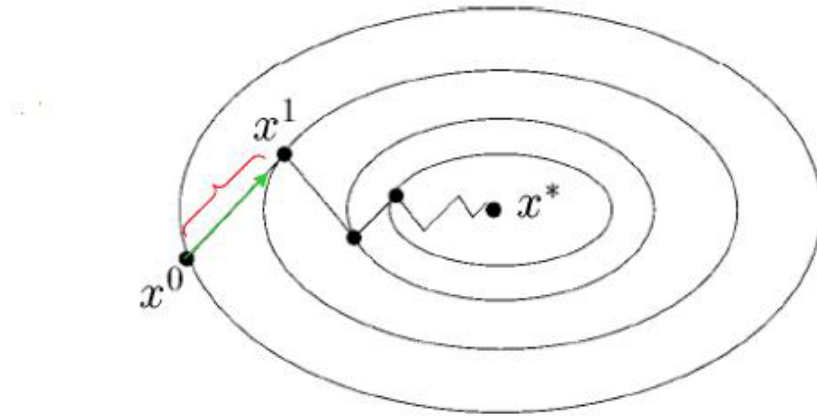
A mais preocupante é a segunda pergunta pois não é garantido que todos os pontos eficientes são minimizadores, para isso foi sugerido a ultima seção antes dessa conclusao (nova abordagem) foi uma das buscas para esse tipo de questionamento, apesar de haver outras saídas, a sugerida difunde com a programação não-linear que tem diversas ferramentas (como as condições KKT) para encontrar os minimizadores.

Em termos de comparações, para esses exemplos testados em dimensões baixas (2 ou 3) pode-se considerar um empate em desempenho de ambos algoritmos testados, dado que a diferença no tempo de convergência de 5 para 2 iterações é infinitesimal (imperceptível). Para testar o desempenho de fato, precisaria de um exemplo que exigisse mais de ambos algoritmo (um exemplo em dimensão mais que 10 por exemplo) entretanto só o fato de ter uma opção para resolver de duas formas o mesmo problema já é uma grande vantagem, tendo em vista que o método de newton em alguns casos diverge para problemas onde o "start" está distante do ponto minimizador, diferente do método do gradiente que é bom para "start" distante mas problemático quando o "start" esta perto do ponto minimizador

Abaixo um exemplo de como o método do gradiente converge:

A partir dai entra a grande vantagem do método quase-newton que ele começa como o método do gradiente e acaba como o método de newton pois veja que o método é:

1. defina um ϵ para critério de parada e defina um "start"(x^0) .



2. se $\| \nabla \Psi(x^m) \| \leq \epsilon$ pare.
3. Calcule H_m usando a atualização BFGS
4. Calcule $x^{m+1} = x^m - \lambda_k H_m \nabla \Psi(x^m)$
5. volte ao passo 2

Onde H_m é uma matriz da mesma dimensão da Hessiana, encontrada pela seguinte expressão:

1. $H_0^{BFGS} = I$
2. $H_{m+1}^{BFGS} = H_m + \frac{1+q_m^t H_m q_m}{q_m^t p_m} \frac{p_m p_m^t}{p_m^t q_m} - \frac{p_m q_m^t H_m + H_m q_m p_m^t}{q_m^t p_m}$

onde $p_m = x_{m+1} - x_m$ e $q_m = \nabla \Psi(x_{m+1}) - \nabla \Psi(x_m)$ ai note que na primeira iteração o método quase-newton trabalha de modo idêntico ao gradiente dado que $H_0 \nabla \Psi(x^0) = \nabla \Psi(x^0)$ pois $H_0 = I$

já a partir da segunda iteração ele aproxima com a matriz

$$H_1^{BFGS} = H_0 + \frac{1 + q_0^t H_0 q_0}{q_0^t p_0} \frac{p_0 p_0^t}{p_0^t q_0} - \frac{p_0 q_0^t H_0 + H_0 q_0 p_0^t}{q_0^t p_0}$$

que é uma matriz que aos poucos se aproximará da inversa hessiana, sem precisar calcular e inverter a tal matriz hessiana. Assim para certos casos pode ser uma ótima opção pois veja que quando longe ele se sobressai ao método de newton e quando perto ele se sobressai ao método de newton pois a matriz H_{m+1}^{BFGS} vai se aproximando da inversa da hessiana que na verdade é a matriz de newton que trabalha bem quando perto do ponto minimizador.

Capítulo 6

Apêndice

6.1 Implementação dos Algoritmos

Método Quase-Newton

O método Quase-Newton foi implementado no Matlab (a2016) da seguinte maneira:

```
disp('Would you like to clear your workspace memory and command screen?');
z=input('To do so enter 1, any other number to continue:');
if z==1
    clc;clear
end
choice=input('To do a maximization enter 1, for minimization enter 2:');a=2;
while a==2
    if choice~=1&&choice~=2
        disp('Wrong input!');a=2;
        choice=input('To do a maximization enter 1, for minimization enter 2:');
    else
        a=1;
    end
end
a=2;mark=0;flag=0;count=0;syms b;u=[];warning off all
try
    while a==2
        n=floor(input('Enter the dimension of function:'));
        if n<1
            disp('Kindly enter a positive integer');a=2;
        else
            a=1;
        end
    end
catch
    disp('Kindly enter a positive integer');
end
a=2;
for i=1:n
    syms (['x' num2str(i)]);
end
while a==2
    try
        f=input('Enter the function in terms of variables x_i(i.e. x1,x2,etc.):');a=1;
```

```

        catch
            disp('Kindly recheck the index of variables and format of expression!');a=2;
        end
    end
    if choice==1
        f=-f;
    end
    x0=(input('Enter the initial approximation row vector of variables:'))';a=2;
    while a==2
        if length(x0)~=n
            disp('The dimension of initial approximation is incorrect!');a=2;
            x0=(input('Enter the initial approximation as row vector of variables:'))';
        else
            a=1;
        end
    end
    try
        eps=abs(input('Enter the error tolerance:'));
    catch
        disp('Kindly enter a real number!');
    end
    g=f;
    try
        for i=1:n
            h(i)=diff(g,['x' num2str(i)]);
            for j=1:n
                k(i,j)=diff(h(i),['x' num2str(j)]);
            end
        end
    catch
        disp('Your optimization problem can not be solved');
        return
    end
    disp('Hessian matrix of the function=')
    if choice==1

```

```

        disp(-k)
    elseif choice==2
        disp(k)
    end
    for i=1:n
        for j=1:n
            if i==j
                d=det(-k(1:i,1:j));
                if choice==1
                    d=-d;
                end
                SOL=solve(d);
                if str2num(char(d))<=0
                    mark=mark+1;u=[u i];
                elseif isempty(SOL)==0
                    for m=1:length(SOL)
                        if isreal(SOL(m))==1||isa(SOL(m),'sym')
                            mark=mark+1;
                            if (length(find(u==i))==0)
                                u=[u i];
                            end
                        end
                    end
                end
            end
        end
    end
end
end
if mark>0
    if choice==1
        fprintf('\n\nThe %gth principal minor of Hessian is not negative at all real x!\n',u);
        fprintf('So the function is not concave globally and hence the global maximization is not guaranteed!\n');
    elseif choice==2
        fprintf('\n\nThe %gth principal minor of Hessian is not positive at all real x!\n',u);
        fprintf('So the function is not convex globally and hence the global minimization is not guaranteed!\n');
    end
end

```

```

else
    if choice==1
        fprintf('\nAll the principal minors of Hessian are negative so the function is concave globally!\n');
        disp('Hence a global maximization is possible!');
    elseif choice==2
        fprintf('\nAll the principal minors of Hessian are positive so the function is convex globally!\n');
        disp('Hence a global minimization is possible!');
    end
end
X=x0;
if mark>0
    S=eye(n);
else
    S=k;
    for i=1:n
        S=(subs(S,['x' num2str(i)],X(i)));
    end
end
%S=input('Enter a positive definite matrix ');
if n==2
    A=[];B=[];C=f;
end
while flag~=1
    count=count+1;steplength=0;
    grad=h';fprintf('\n-----%gth Iteration-----\n\n\n',count)
    if n==2
        A=[A X(1)];B=[B X(2)];
    end
    for i=1:n
        grad=subs(grad,['x' num2str(i)],X(i));
    end
    disp('Present point:');disp(X);
    disp('Gradient=');
    if choice==1
        disp(-grad)
    elseif choice==2

```

```

        disp(grad)
    end
    if max(abs(grad))>eps
        fprintf('\n\nThe error tolerance you provided has not been achieved yet\n');
        flag=input('To terminate enter 1, any other number to continue:');
    end
    if flag==1
        hes=k;
        for i=1:n
            hes=subs(hes,['x' num2str(i)],X(i));
        end
        fprintf('\n\n.....Result.....\n\n')
        if length(find(eig(hes)>0))==length(eig(hes))
            disp('At the present point the function is convex so it may be a local minimum!');
        elseif length(find(eig(hes)<0))==length(eig(hes))
            disp('At the present point the function is concave so it may be a local maximum!');
        else
            disp('The present point is not a local extremum!');
        end
        disp('Presently the hessian matrix is:');
        if choice==1
            disp(-hes)
        elseif choice==2
            disp(hes)
        end
        fprintf('The optimum point at %g error is:\n',max(abs(grad)));
        disp(X);
        for i=1:n
            f=subs(f,['x' num2str(i)],X(i));
        end
        fprintf('\n\nAnd the function value here is:\n')
        if choice==1
            disp(-f)
        elseif choice==2
            disp(f)
        end
    end
end

```

id#

```

        fprintf('The total number of iterations performed=%g\n',count);
    end
    if max(abs(grad))<=eps
        fprintf('\n\nThe error tolerance you provided has been achieved.\n');
        hes=k;
        for i=1:n
            hes=subs(hes,['x' num2str(i)],X(i));
        end
        fprintf('\n\n.....Result.....\n\n')
        if length(find(eig(hes)>0))==length(eig(hes))
            disp('Also at the present point the function is convex so it may be a local minimum!');
        elseif length(find(eig(hes)<0))==length(eig(hes))
            disp('Also at the present point the function is concave so it may be a local maximum!');
        else
            disp('However the present point is not a local extremum point!');
        end
        disp('Presently the hessian matrix is:');
        if choice==1
            disp(-hes)
        elseif choice==2
            disp(hes)
        end
        fprintf('The optimum point at %g error is:\n',max(abs(grad)));
        disp(X);
        for i=1:n
            f=subs(f,['x' num2str(i)],X(i));
        end
        fprintf('\n\nAnd the function value here is:\n')
        if choice==1
            disp(-f)
        elseif choice==2
            disp(f)
        end
        fprintf('The total number of iterations performed=%g\n',count);flag=1;
    elseif max(abs(grad))>eps && flag~=1
        fun=f;dir=-S*grad;
    end
end

```

```

for i=1:n
    fun=subs(fun,['x' num2str(i)],X(i)+b*dir(i));
end
diff(fun,b);
d=solve(diff(fun,b));
if isempty(d)==1
    steplength=1;
else
    t=double(d);
    dd=diff(diff(fun,b));
    for i=1:length(t)
        if isreal(t(i))==1
            if subs(dd,'b',t(i))>0
                steplength=t(i);break
            end
        end
    end
    if steplength==0
        for i=1:length(t)
            if isreal(t(i))==1
                if t(i)>0
                    steplength=t(i);break
                end
            end
        end
    end
    if steplength==0
        steplength=1;
    end
end
func=f;
for i=1:n
    func=subs(func,['x' num2str(i)],X(i));
end
disp('Functional value at present=');
if choice==1

```

```

        disp(-funct)
    elseif choice==2
        disp(funct)
    end
    disp('Step size taken=');disp(steplength);;
    X=X+steplength*dir;p=steplength*dir;
    grad_new=h';
    for i=1:n
        grad_new=subs(grad_new,['x' num2str(i)],X(i));
    end
    q=grad_new-grad;
    S=S-(S*q*q'*S)/(q'*S*q)+(p*p')/(p'*q);
    if count>100
        disp('You already have performed 100 iterations and it seems that no extremum of the function exists!');
        flag=input('It is recommended that you terminate the procedure, to do so enter 1, any other number to continue:');
    end
end
end
try
    if n==2&&length(A)>=2
        [a,b]=meshgrid(A,B);C=subs(C,{x1,x2},{a,b});
        view([-50,30]);axis tight; hold on
        surf(a,b,C,'facecolor','green','edgecolor','b','facelighting','gouraud')
        view([-50,30]);axis tight;shading interp;
        plot(A,B,'-mo',...
            'LineWidth',2,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor',[.49 1 .63],...
            'MarkerSize',12);
    end
catch
    disp('Unable to generate plot of function!');
end
end
disp('Functional value at present=');
if choice==1

```

Está totalmente disponível no matlab chamando o comando *trainbfg(.)*.

Método de Newton

O método de Newton foi implementado no Matlab (a2016) da seguinte maneira:

```
clc
clear
format long

% Function Definition (Enter your Function here):
syms X Y;
f = X - Y + 2*X^2 + 2*X*Y + Y^2;

% Initial Guess (Choose Initial Guesses):
x(1) = 1;
y(1) = -5;
e = 10^(-8); % Convergence Criteria
i = 1; % Iteration Counter

% Gradient and Hessian Computation:
df_dx = diff(f, X);
df_dy = diff(f, Y);
J = [subs(df_dx,[X,Y], [x(1),y(1)]) subs(df_dy, [X,Y], [x(1),y(1)])]; % Gradient
ddf_ddx = diff(df_dx,X);
ddf_ddy = diff(df_dy,Y);
ddf_dxdy = diff(df_dx,Y);
ddf_ddx_1 = subs(ddf_ddx, [X,Y], [x(1),y(1)]);
ddf_ddy_1 = subs(ddf_ddy, [X,Y], [x(1),y(1)]);
ddf_dxdy_1 = subs(ddf_dxdy, [X,Y], [x(1),y(1)]);
H = [ddf_ddx_1, ddf_dxdy_1; ddf_dxdy_1, ddf_ddy_1]; % Hessian
S = inv(H); % Search Direction
```

```

% Optimization Condition:
while norm(J) > e
    I = [x(i),y(i)]';
    x(i+1) = I(1)-S(1,:)*J';
    y(i+1) = I(2)-S(2,:)*J';
    i = i+1;
    J = [subs(df_dx,[X,Y], [x(i),y(i)]) subs(df_dy, [X,Y], [x(i),y(i)])]; % Updated Jacobian
    ddf_ddx_1 = subs(ddf_ddx, [X,Y], [x(i),y(i)]);
    ddf_ddy_1 = subs(ddf_ddy, [X,Y], [x(i),y(i)]);
    ddf_dxdy_1 = subs(ddf_dxdy, [X,Y], [x(i),y(i)]);
    H = [ddf_ddx_1, ddf_dxdy_1; ddf_dxdy_1, ddf_ddy_1]; % Updated Hessian
    S = inv(H); % New Search Direction
end

% Result Table:
Iter = 1:i;
X_coordinate = x';
Y_coordinate = y';
Iterations = Iter';
T = table(Iterations,X_coordinate,Y_coordinate);

% Plots:
fcontour(f, 'Fill', 'On');
hold on;
plot(x,y,'*-r');
grid on;

% Output:
fprintf('Initial Objective Function Value: %d\n\n',subs(f,[X,Y], [x(1),y(1)]));
if (norm(J) < e)
    fprintf('Minimum succesfully obtained...\n\n');
end
fprintf('Number of Iterations for Convergence: %d\n\n', i);
fprintf('Point of Minima: [%d,%d]\n\n', x(i), y(i));
fprintf('Objective Function Minimum Value after Optimization: %f\n\n', subs(f,[X,Y], [x(i),y(i)]));
disp(T)

```

É de domínio publico para quem tem matlab sobre o comando *newton(.)*.

Referências Bibliográficas

- [1] G.PANDA and A. K. Bhurjee. *Efficient solution of interval optimization problem. Mathematical Methods of Operations Research*, 2012.
- [2] ANA FRIEDLANDER , *ELEMENTOS DE PROGRAMACÃO NÃO-LINEAR* , 1994.
- [3] T. P. da Silveira and S.D.B Bitar *Estudos sobre Modelos de Otimização Matemática utilizando Funções com Parâmetros Intervalares*, Manaus, 2017.
- [4] R. E. Moore: *Interval analysis*, Prentice-Hall, Englewood Cliffs, 1966.
- [5] MOORE, L. R. B. Kearfott, and M. J. Cloud *Introduction to interval analysis. Philadelphia: SIAM*, 2009.
- [6] R. E. Moore *Methods and applications of interval analysis.*, Philadelphia: SIAM, 1979.
- [7] GHOSH, D. *Newton method to obtain efficient solutions of the optimization problems with interval-valued objective functions*, J. Appl. Math. Comput. India, 2017.
- [8] DENNIS, J. E and SCHNABEL, R. B. *Numerical methods for unconstrained optimization and nonlinear equations Englewood Cliffs*, Prentice Hall, 1983.
- [9] Chalco-Cano, Y., Rufian-Lizana, A., Roman-Flores, H., Jimenez-Gamero,M.D *Calculus for intervalvalued functions using generalized Hukuhara derivative and applications*, 219, 49–67, 2013.
- [10] YANO, K. and NAGANO, T. *Einstein spaces admitting a one-parameter group of conformal transformation*, Ann. of Math., 69 (1959), 451-461.