UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

Bruno Stevanato Trevisan 168170
Matheus Araujo Souza 184145
Rebeca Lie Yatsuzuka Silva 223944

Projeto 1: Minimização Irrestrita

Campinas
23/11/2020

Sumário

1 Introdução

Neste projeto realizamos as implementações dos métodos Gradiente, Newton Globalizado e Secante Globalizado, a fim de verificar seus como se comportam ao aplica-los nas seguintes funções: Quadrática, Styblinsky-tang, Rosenbrook, Rastring. A analise foi feita comparando o desempenho dos métodos para diferentes valores da precisão ε e dos pontos iniciais x^0. [1]

2 Desenvolvimento dos Métodos

Os métodos desenvolvidos neste projeto fazem parte dos métodos clássicos de descida, ou seja todos eles tem em comum o fato de tomarem uma direção do passo oposta ao gradiente e utilizar a busca linear como forma de determinar (t_k), para resolver um problema do tipo:

$$\text{Minimizar } f(x),\, x \in \mathbb{R}^n \tag{1}$$

2.1 Gradiente (GR)

O método do gradiente, que também conhecido por método de máxima descida, é utilizado para resolução de problemas de minimização irrestrita como (1), no qual se utiliza da busca linear na direção $\boldsymbol{d^k}$ contraria ao gradiente $\boldsymbol{-\nabla f(x^k)}$ (figura 1).

Para minimizar a função $\boldsymbol{f}$, partimos de um ponto inicial x_k e, a cada iteração k, escolhemos $\boldsymbol{d^k = -\nabla f(x^k)}$ pois como $\boldsymbol{f}$ aumenta na direção $\boldsymbol{\nabla f(x^k)}$ e como queremos minimizar $\boldsymbol{f}$ temos:

$$d^k = -\nabla f(x^k) \tag{2}$$

Escolhida a direção $\boldsymbol{d^k}$, temos que definir o tamanho do passo $\boldsymbol{t_k}$ que daremos na direção $\boldsymbol{d^k}$, a partir de $\boldsymbol{x^k}$ buscando minimizar a função (3), com $\boldsymbol{t}$ sujeito a $\boldsymbol{t} \geq 0$:

$$f(x^k + td^k) \tag{3}$$

com $\boldsymbol{t_k}$ definido calcularemos por fim

$$x^{k+1} = x^k + t_k d^k \tag{4}$$

logo podemos escrever seu algorítimo como:

Algoritmo 1: Métodos do Gradiente

Sejam $\alpha \in (0, 1)$, $\gamma \in (0, 1)$, $\varepsilon > 0$, $M \in \mathbb{N}$ e $x^0 \in \mathbb{R}^n$

Tomando $k = 0$

enquanto $(||\nabla f(x^k)|| \geq \varepsilon)$ *e* $(k < M)$ **faça**

  $d^k = -\nabla f(x^k)$.

  $t_k = 1$ **enquanto** $f(x^k + t_k d^k) > f(x^k) + \alpha t_k \nabla f(x^k)^T d^d$ **faça** $t_k \leftarrow \gamma t_k$;

  $x^{k+1} = x^k + t_k d^k$ e $k \leftarrow k + 1$

fim

Para a implementação utilizamos as sugestões de valores dos parâmetros iniciais, dados no projeto. (falar das dificuldades se hover, e como ele funciona (com variaveis simbólicas , como decidimos valor de epsilon, e os valores de M))

2.2 Newton Globalizado (NG)

O método de Newton Globalizado é utilizado para resolução de problemas de minimização irrestrita como (1), no qual a partir da direção $\boldsymbol{\nabla^2 f(x^k) d^k = -\nabla f(x^k)}$, dado que $\boldsymbol{\nabla^2 f(x^k)}$ existe e é definida positiva.

A ideia do método de Newton é minimizar a cada iteração a aproximação quadrática em torno do ponto atual x_k, uma das peculiaridades desse método se da ao fato de que a hermitiana de $\boldsymbol{f(x^k)}$ deve ser definida positiva para que $\boldsymbol{d^k}$ seja uma direção de descida

$$[\nabla^2 f(x^k) + \mu I] d^k = -\nabla f(x^k) \tag{5}$$

O preço da convergência rápida do método de Newton é a sobrecarga necessária para calcular a matriz Hessiana, e para resolver o sistema de equações.

Algoritmo 2: Métodos de Newton Globalizado

Sejam $\sigma > 0$, $\alpha \in (0, 1)$, $\gamma \in (0, 1)$, $\theta \in (0{,}1)$, $\varepsilon > 0$, $M \in \mathbb{N}$ e $x^0 \in \mathbb{R}^n$

Tomando $k = 0$

enquanto $(||\nabla f(x^k)|| \geq \varepsilon)$ *e* $(k < M)$ **faça**

- $\mu = 0$
- resolver $[\nabla^2 f(x^k) + \mu I]d^k = -\nabla f(x^k)$
- **se** *Não deu certo ou* $\nabla f(x^k)^T d^k > -\theta ||\nabla f(x^k)|| \cdot ||d^k||$ **então**
  - $\mu \leftarrow \max\{2\mu, \beta\}$
  - retorne à resolução do sistema linear.
- **se** $||d^k|| < \sigma ||\nabla f(x^k)||$ **então** $d^k \leftarrow \sigma \frac{||\nabla f(x^k)||}{||d^k||} d^k$;
- $t_k = 1$. **enquanto** $f(x^k + t_k d^k) > f(x^k) + \alpha t_k \nabla f(x^k)^T d^k$ **faça** $t_k \leftarrow \gamma t_k$;
- $x^{k+1} = x^k + t_k d^k$ e $k \leftarrow k + 1$

fim

(Peculiaridades do método para implementação)

2.3 Secante Globalizado (DFP)

O método DFP é classificado como método quase-Newton. Pois toma $\boldsymbol{d^k} = -\boldsymbol{H_k \nabla f(x^k)}$, onde $\boldsymbol{H_k} \in \mathbb{R}^n$ é uma matriz simétrica, de forma que se $\boldsymbol{H_k}$ for definida positiva, $\boldsymbol{d^k}$ é uma direção de descida.

Em problemas quadráticos, ele gera as direções do método do gradiente conjugado ao mesmo tempo que constrói a inversa da Hessiana. A cada passo a inversa da

Hessiana é aproximada pela soma de duas matrizes simétricas de posto 1.

Algoritmo 3: Métodos Secante Globalizado (DFP)

Sejam $\sigma > 0$, $\alpha \in (0, 1)$, $\gamma \in (0, 1)$, $\theta \in (0,1)$, $\varepsilon > 0$, $M \in \mathbb{N}$ e $H_0 \in \mathbb{R}^{n\times n}$

$H_0 = H_0^T$ e definida positiva, e $x^0 \in \mathbb{R}^n$

Faça $k = 0$

enquanto $(||\nabla f(x^k)|| \geq \varepsilon)$ *e* $(k < M)$ **faça**

Faça $d^k = -H_k \nabla f(x^k)$

se $\nabla f(x^k)^T d^k > -\theta ||\nabla f(x^k)|| \cdot ||d^k||$ **então** $d^k = -H_k \nabla f(x^k)$ e $H_k = I$;

se $||d^k|| < \sigma ||\nabla f(x^k)||$ **então** $d^k \leftarrow \sigma \frac{||\nabla f(x^k)||}{||d^k||} d^k$;

Faça $t_k = 1$. **enquanto** $f(x^k + t_k d^k) > f(x^k) + \alpha t_k \nabla f(x^k)^T d^k$ **faça** $t_k \leftarrow \gamma t_k$;

Faça $x^{k+1} = x^k + t_k d^k$, $p^k = x^{k+1} - x^k$, $q^k = \nabla f(x^{k+1}) - \nabla f(x^k)$ e $r^k = H_k q^k$

se $(p^k)^T q^k > 0$ **então** $H_{k+1} = H_k + \frac{p^k (p^k)^T}{(p^k)^T q^k} - \frac{r^k (r^k)^T}{(r^k)^T q^k}$ **senão** $H_{k+1} = H_k$;

Faça $k \leftarrow k + 1$

fim

3 Implementação

Para a criação do código usamos os passos fornecidos pelos algoritmos já mencionado, aqui vamos focar na parte mais técnica e sobre alguns comandos da sua criação e o objetivo deles. Como pode ser visto nas primeiras linhas no código, iniciamos implementado as variáveis simbólicas foi que já começamos a declaração de quem são, logo após definimos quais são as funções que vamos usar para a aplicação do método, usamos as funções gradient e hessian que vão calcular com base na conversão de variáveis simbólicas o vetor gradiente e a matriz hessiana, definimos logo depois qual vai ser o método de descida escolhido, podemos partir para as explicações mais específicas dentro das funções, primeiro comando a ser utilizado foi o subs na definição do matlab temos que, subs(x) vai retornar uma cópia do valor que é obtido com a substituição numérica nos valores simbólicos, em outras palavras vamos trazer do simbólico para o numérico. Um segundo comando muito importante utilizado foi o vpa(), nele definimos um intervalo mínimo de casas decimais que deve ser usada nos cálculos, um dos problemas que enfrentamos em apenas usar o subs() foi que ele não convertia valores racionais, deixava tudo na forma de fração então a utilização do comando vpa() veio para contornar esse problema. No método de newton usamos o comando chol() com esse comando estamos realizando o decomposição de cholesky, optamos por esse método por se tratar do mais eficiente e recomendado para esses casos, com isso concluímos nossos detalhamentos de comandos mais específicos.

4 Resultados e Análise

4.1 Quadrática

4.1.1 Uma variável

Temos um comportamento muito interessante nesse caso, a velocidade de convergência é extremamente rápida em todos os métodos de descida, porem a precisão do método de newton acaba sendo menor do que as outras, todos convergem em apenas uma única iteração.

Gradiente					
ponto inicial x0	Vk(numero de iterações)	Vx	VF(valor da função)	VG(valor do Gradiente)	tempo [s]
[5]	1	0.0	0.0	0.0	1.47
[-48]	1	0.0	0.0	0.0	0.88
[200]	1	0.0	0.0	0.0	0.78

Newton					
ponto inicial x0	Vk(numero de iterações)	Vx	VF(valor da função)	VG(valor do Gradiente)	tempo [s]
[5]	1	1,69E-12	2,85E-24	3,38E-12	2.02
[-48]	1	-1,62E-11	2,63E-22	-3,24E-11	1.35
[200]	1	6,75E-11	4,56E-21	1,35E-10	0.95

DFP					
ponto inicial x0	Vk(numero de iterações)	Vx	VF(valor da função)	VG(valor do Gradiente)	tempo [s]
[5]	1	0.0	0.0	0.0	1.67
[-48]	1	0.0	0.0	0.0	1.09
[200]	1	0.0	0.0	0.0	1.08

4.1.2 Duas variáveis

Novamente temos um tempo muito bom de convergência, mas podemos ver que a precisão do método de descida de DFP ficou pior com relação ao teste de uma variável, o método do Gradiente acabou ficando o mais preciso porem de todos o de Newton manteve apenas uma iteração.

Gradiente					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[5;8]	2	(0;0)	0.0	(0;0)	2.571211 seconds.
[-10;-20]	2	(0;0)	0.0	(0;0)	1.768042 seconds.
[-78;1]	2	(0;0)	0.0	(0;0)	1.719801 seconds.

Newton					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[5;8]	1	(1.6885e-12; 0)	2,85E-24	(3.3771e-12 ;0)	1.611424 seconds.
[-10;-20]	1	(-3.3771e-12; 0)	1,14E-23	(-6.7541e-12; 0)	1.181963 seconds.
[-78;1]	1	(-2.6341e-11; 0)	6,94E-22	(-5.2682e-11; 0)	1.137645 seconds.

DFP					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[5;8]	6	(1.21e-7; -3.5047e-9)	1,47E-14	(2.4201e-7; -1.4019e-8)	5.056762 seconds.
[-10;-20]	6	(-2.4926e-8; 3.9194e-10)	6,22E-16	(-4.9852e-8; 1.5678e-9)	4.764068 seconds.
[-78;1]	4	(1.4621e-10; -5.6322e-12)	2,14E-20	(2.9242e-10; -2.2529e-11)	3.515018 seconds.

4.1.3 Três variáveis

Para esse teste tanto o método do Gradiente quanto o DFP se saíram inferiores ao método de newton em todos os quesitos número de iterações, tempo de convergência, valor do Gradiente e valor da variável x.

Bem, podemos concluir que o método de Newton acabou se saindo superior nos testes realizados, ele acabou mantendo uma velocidade rápida de convergência e permaneceu sempre com uma única iteração.

Gradiente					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[10;-2;50]	29	(1.8626e-8; 0; -9.3132e-8)	2,64E-14	(3.7253e-8; 0; -5.5879e-7)	22.282538 seconds.
[1;0;1]	23	(1.1921e-7; 0; -1.1921e-7)	5,68E-14	(2.3842e-7; 0; -7.1526e-7)	17.549668 seconds.
[100;-100;400]	32	(2.3283e-8; 0; 9.3132e-8)	2,66E-14	(4.6566e-8; 0; 5.5879e-7)	23.385696 seconds.

Newton					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[10;-2;50]	1	(3.3771e-12; 0; 1.4304e-10)	6,14E-20	(6.7541e-12; 0; 8.5821e-10)	1.337654 seconds.
[1;0;1]	1	(3.3771e-13; 0; 2.8607e-12)	2,47E-23	(6.7541e-13; 0; 1.7164e-11)	1.018960 seconds.
[100;-100;400]	1	(3.3771e-11; 0; 1.1443e-9)	3,93E-18	(6.7541e-11; 0; 6.8657e-9)	1.057829 seconds.

DFP					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[10;-2;50]	8	(-3.0364e-9; -5.7973e-10; -1.4147e-9)	1,59E-17	(-6.0729e-9; -2.3189e-9; -8.4885e-9)	7.438407 seconds.
[1;0;1]	7	(8.3782e-10; 0; -6.2931e-10)	1,89E-18	-(1.6756e-9; 0; -3.7758e-9)	5.877162 seconds.
[100;-100;400]	10	(6.2284e-10; -7.3318e-12; -1.6425e-9)	8,48E-18	(1.2457e-9; -2.9327e-11; -9.8551e-9)	7.480197 seconds.

4.2 Rosenbrook

4.2.1 Duas variável

Para Rosenbrook pudemos observar desde o inicio que o método mais demorado tanto em questão de tempo quanto numero de iterações foi o gradiente, portanto fim de explorar mais pontos optamos por observar apenas as convergência para um mínimo local. Entretanto enquanto o Gradiente demorava muito para converter, o método de newton foi o que acabou tendo o melhor desempenho, como pode ser observado a baixo.

Newton					
ponto inicial x0	Vk	Vx	VF	VG	tempo
[1; -1]	1	[1; 1]	7,11E-30	[1,06E-13; -5,33E-14	4,65282
[2; 2]	13	[1; 1.0001]	2,93E-05	[0.0010742; -0.00048884]	2,942

Newton					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[1; -1]	1	[1; 1]	7,11E-30	[1,06E-13; -5,33E-14	4,65
[2; 2]	13	[1; 1.0001]	2,93E-05	[0.0010742; -0.00048884]	2,94

DFP					
ponto inicial x0	Vk	Vx1	VF	VG	tempo
[1; -1]	500	[0,97743; 0.95432]	0,00054839	[0.15598; -0.10319]	155,35
[2; 2]	500	[-0,63359; 0,36259]	2,8195	[-13.11; -7.7672]	236,17

4.2.2 Três variáveis

Temos que o mesmo ocorre para três variáveis, novamente o método do Gradiente se saiu o mais lento de todos, mas desta vez o DFP teve seu rendimento bem prejudicado, se assimilando ao do gradiente.

Gradiente					
ponto inicial x0	Vk(numero de iterações)	Vx1	VF(valor da função)	VG(valor do Gradiente)	tempo
[2;2;2]	1500	[1.0602; 1.1241; 1.2644]	0,019083	[0.083869; -0.053461; 0.14187]	1341.80
[1;2;1]	1500	[0.98022; 0.96096; 0.92306]	0,0019314	[-0.087127; 0.092077; -0.075913]	1252.72

Newton					
ponto inicial x0	Vk(numero de iterações)	Vx1	VF(valor da função)	VG(valor do Gradiente)	tempo
[2;2;2]	17	[1,0; 1,0; 1,0]	1,53E-14	[1,1944e-6; 4,199e-6; -2.3811e-6]	
[1;2;1]	22	[1,0; 1,0; 1,0]	6,71E-14	[2.1905e-6; 7.7204e-6; -4.2527e-6]	4,24

DFP					
ponto inicial x0	Vk(numero de iterações)	Vx1	VF(valor da função)	VG(valor do Gradiente)	tempo [s]
[2;2;2]	1376	[0.99997; 0.99994; 0.99988]	4,33E-05	[0.00010602; 0.00045934; -0.00032732]	505.00
[1;2;1]	1500	[0.86563; 0.73843; 0.518]	0,17275	[3.4985; 5.3598; -5.4568]	538.46

4.3 Styblinsky–Tang

4.3.1 Uma variável

Nessa parte inicial da análise temos que a maioria dos pontos convergiram para o resultado ótimo, começamos em pontos bem próximos e aplicamos o procedimento de que os mesmo pontos testados para um método deveria ser aplicados para os demais, na realidade essa é a forma padrão de testes, para esses teste temos que nos pontos escolhidos o Gradiente foi o mais lento de todos, porém nele todos convergiram para o ponto ótimo, já no método do DFP mesmo sendo o segundo mais rápido, teve um dos testes que acabaram convergindo para um mínimo local.

Gradiente					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
-4	9	-29.035	-78.332	7,25E-07	8.858067
-8	14	-29.035	-78.332	6,35E-07	14.715744
-11	14	-29.035	-78,332	7,04E-07	16,045403
20	13	-2,9035	-78,332	7,23E-07	15,197545
50	13	-2,9035	-78,332	5,65E-07	14,311247

Newton					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
-4	5	-2,9035	-78,332	-5,9651e-12	1.761224
-8	7	-2,9035	-78.332	-2,6442e-11	2.374812
-11	7	-2,9035	-78.332	-1,93E-09	2.274232
20	8	-2,904	-78.332	-4,48E-10	2.468857
50	8	-2,9035	-78.332	-6,28E-13	3.325637

DFP					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
-4	8	-2,9035	-78,3320	1,40E-08	4.622058
-8	9	-2,9035	-50,0590	-1,25E-08	5.266210
-11	7	2,7468	-50,059	4,87E-08	4.639716
20	11	-2,9035	-78,332	-1,65E-09	6.660908
50	11	-2,9035	-78,3320	-1,19E-09	7.485850 .

4.3.2 Duas variáveis

Temos que o mesmo ocorre para duas variáveis, novamente o método do Gradiente se saiu o mais lento de todos, mas tanto o Newton quanto o DFP não convergiram para a solução ótima e sim para mínimos locais.

Gradiente					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[-4;-4]	10	[-2.9035; -2.9035]	-156,66	[-5.8506e-8; -5.8506e-8]	13.930516
[-8;-8]	14	[-2.9035; -2.9035]	-156,66	[6.3533e-7; 6.3533e-7]	18.044610
[-11;-11]	14	[-2.9035; -2.9035]	-156,66	[7.0366e-7; 7.0366e-7]	17.838776
[20;20]	14	[-2.9035; -2.9035]	-156,66	[-5.8344e-8; -5.8344e-8]	19.717899
[20;-8]	13	[-2.9035; -2.9035]	-156,66	[7.2279e-7; 1.4567e-7]	20.206696
[-3;-2.5]	8	[-2.9035; -2.9035]	-156,66	[-3.5694e-8; 7.2881e-7]	9.870995

Newton					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[-4;-4]	5	[-2.9035; -2.9035]	-156,66	[-5.9651e-12; -5.9651e-12]	2.552961
[-8;-8]	7	[-2.9035; -2.9035]	-156,66	[-2.6442e-11; -2.6442e-11]	3.207678
[-11;-11]	7	[-2.9035; -2.9035]	-156,66	[-1.9296e-9; -1.9296e-9]	3.284838
[20;20]	8	[-2.9035; -2.9035]	-156,66	[-4.4778e-10; -4.4778e-10]	3.592583
[20;-8]	8	[-2.9035; 2.7468]	-128,39	[-2.6058e-8; 0]	3.895719
[-3;-2.5]	4	[-2.9035; -2.9035]	-156,66	[-7.3445e-20; -1.6663e-8]	2.370714

DFP					
ponto inicial x0	Vk	Vx	VF	VG	tempo
[-4;-4]	8	[-2.9035 ;-2.9035]	-156,66	[1.3997e-8; 1.3997e-8]	6.456128
[-8;-8]	9	[2.7468; 2.7468]	100,12	[-1.2471e-8; -1.2471e-8]	6.724789
[-11;-11]	7	[2.7468; 2.7468]	-100,12	[4.8735e-8; 4.8735e-8]	6.279565
[20;20]	11	[-2.9035; -2.9035]	-156,66	[-1.6536e-9; -1.6536e-9]	8.975024
[20;-8]	18	[-2.9035; -2.9035]	-156,66	[-3.6623e-9; -5.069e-8]	13.814034
[-3;-2.5]	7	[-2.9035; -2.9035]	-156,66	[-7.4717e-8; -5.7364e-7]	6.267862

4.3.3 Três variáveis

Para esses testes de 3 variáveis buscamos variar bastante os pontos,também em uma das execuções colocamos três pontos iguais pensando que eles iriam convergir mais rápido no caso do x1=x2=x3=-8, mesmo assim ele acabou não sendo a execução dos métodos. No caso teve um teste que saiu bem acima da média nos tempos que foi para o método de descida do DFP, para o ponto [2;4;8], se compararmos ponto a ponto e olhando para cada método,com exceção desse ponto comentado, o do Gradiente acabou demorando mais, mas diferente dos casos passados nenhuma das execuções dele convergiu para o ótimo. Se olharmos para o quesito conversão no ponto ótimo o newton acabou se saindo superior nos testes para 3 variáveis. Novamente podemos ver que em todos os casos nos convergimos para mínimos locais em algumas execuções, nesse teste para três variáveis isso se intensificou ainda mais, como temos uma convergência global é bom que isso ocorra, só mostra a eficiência do nosso método.

Gradiente					
ponto inicial x0	Vk	Vx1	VF	VG	tempo [s]
[-4;-3;-5]	13	[-2.9035; -2.9035; 2.7468]	-206,72	[6.902e-7; 2.2641e-8; 1.9945e-8]	21,27
[-8;-8;-8]	15	[-2.9035; -2.9035; -2.9035]	-235	[-5.1284e-8; -5.1284e-8; -5.1284e-8]	21,94
[2;4;8]	10	[2.7468; 2.7468; -2.9035]	-178,45	[6.1545e-8; 9.1502e-9; 4.8378e-7]	16,31
[-3;-3;2]	10	[-2.9035; -2.9035; 2.7468]	-206,72	[-4.7045e-8; -4.7045e-8; 5.3637e-8]	16,00

Newton					
ponto inicial x0	Vk	Vx1	VF	VG	tempo [s]
[-4;-3;-5]	6	[-2.9035 -2.9035 -2.9035]	-235,0	[-2.5915e-25 -2.351e-38 -1.6535e-14]	3,22
[-8;-8;-8]	7	[-2.9035; -2.9035; -2.9035]	-235	[-2.6442e-11 -2.6442e-11 -2.6442e-11]	4,03
[2;4;8]	7	[2.7468 2.7468 2.7468]	-150,18	[2.4963e-26 0 1.2516e-9]	4,53
[-3;-3;2]	4	[-2.9035 -2.9035 2.7468]	-206,72	[-2.0329e-11 -2.0329e-11 5.8521e-9]	2,81

DFP					
ponto inicial x0	Vk	Vx1	VF	VG	tempo [s]
[-4;-3;-5]	20	[-2.9035 -2.9035 2.7468]	-206,72	[-5.9942e-7 -1.7255e-7 2.21e-7]	18,73
[-8;-8;-8]	9	[2.7468 2.7468 2.7468]	-150,18	[-1.2471e-8 -1.2471e-8 -1.2471e-8]	8,28
[2;4;8]	180	[2.7468 -2.9035 -2.9035]	-206,72	[-9.4948e-11 5.9415e-8 5.5658e-8]	127,14
[-3;-3;2]	7	[-2.9035 -2.9035 2.7468]	-206,72	[2.7437e-8 2.7437e-8 -1.0652e-8]	7,45

4.3.4 Quatro variáveis

Analisando de forma geral temos que o DFP acabou sendo o mais ineficiente, nenhum dos pontos em que testamos convergiu para a solução ótima e ainda os testes mais demorados estão presentes nele. O método de descida mais eficiente para esses testes foram o de Newton e o do Gradiente, o melhor foi o de newton sendo ele o que teve duas conversões para o ponto ótimo e seu tempo foi o mais rápido de todos o que acabou surpreendendo, na realidade ao fazer as simulações pensava que esse seria o método de descida mais demorado, fechamos aqui nossas simulações, não precisamos variar muito o valor do epsilon, podemos ver que o valor do gradiente que chegamos em todos os casos é muito pequeno, nossas execuções sempre convergiram para todos os casos, mesmo que em alguns mesmo variando os pontos caímos novamente no mesmo mínimo local, que na realidade foi outro fato surpreendente.

Gradiente					
ponto inicial x0	Vk	Vx1	VF	VG	tempo [s]
[-3;-3;2;-4]	9	(-2,9035; -2,9035; 2,7468; -2,9035)	-285,06	(2,4959e-8; 2,4959e-8; -5,0868e-9; 7,248e-7)	16,50769
[-7;-7;-8;-3]	17	(-2,9035; -2,9035; -2,9035; -2,9035)	-313,33	(-4,9764e-8 -4,9764e-8; -2,7912e-8; -5,6881e-8)	28,973379
[-3;-3;-5;-2]	13	(-2,9035; -2,9035; 2,7468; -2,9035)	-285,06	(2,2641e-8; 2,2641e-8; 1,9945e-8; 7,276e-7)	22,995454
[-6;-6;-5;-3]	14	(-2,9035; -2,9035; 2,7468; -2,9035)	-285,06	(5,6958e-7; 5,6958e-7; 7,5889e-8; -1,2827e-9)	24,425708

Newton					
ponto inicial x0	Vk	Vx1	VF	VG	tempo [s]
[-3;-3;2;-4]	6	(-2,9035; -2,9035; 2,7468; -2,9035)	-285,06	(-2,351e-38 -2,351e-38 1,611e-12 -2,5915e-25)	4,532532
[-7;-7;-8;-3]	7	(-2,9035; -2,9035; -2,9035; -2,9035)	-313,33	(-6,79e-15 -6,79e-15 -2,6442e-11 -2,351e-38)	4,273982
[-3;-3;-5;-2]	6	(-2,9035; -2,9035; -2,9035; -2,9035)	-313,33	(-2,351e-38 -2,351e-38 -1,6535e-14 -1,1133e-9)	4,586978
[-6;-6;-5;-3]	6	(-2,9035; -2,9035; -2,9035; -2,9035)	-313,33	(-1,4714e-9 -1,4714e-9 -1,6535e-14 -2,351e-38)	3,723049

DFP					
ponto inicial x0	Vk	Vx1	VF	VG	tempo [s]
[-3;-3;2;-4]	11	(-2,9035; -2,9035; 2,7468; -2,9035)	-285,06	(4,9172e-8 4,9172e-8 -7,1389e-9 -6,2577e-9)	12,337805
[-7;-7;-8;-3]	13	(2,7468 2,7468 2,7468 -2,9035)	-228,51	(7,8469e-8 7,8469e-8 -7,0974e-7 2,2015e-7)	13,189085
[-3;-3;-5;-2]	207	(-2,9035 -2,9035 2,7468 -2,9035)	-285,06	(6,9672e-9 2,7303e-8 4,0953e-8 -4,4534e-8)	154,869699
[-6;-6;-5;-3]	313	(2,7468; 2,7468; -2,9035; -2,9035)	-256,78	(-3,5412e-8; -5,328e-7; 7,0469e-7; -2,3627e-7)	235,225675

4.4 Rastrigin

4.4.1 Uma variável

Temos um comportamento muito interessante nesse caso, a velocidade de convergência é extremamente rápida em todos os métodos de descida, porem a precisão do método de newton acaba sendo menor do que as outras, além de ser o único método que não convergiu em apenas uma iteração,

Gradiente					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[3]	1	9,419E-19	-10	-3,74E-16	0.847467 seconds.
[9]	1	-1,11E-18	-10	-4,4095e-16	0.824599 seconds.
[-22]	1	1,25E-18	-10	4,95E-16	0.838471 seconds.

Newton					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[3]	2	2,98	-1,05	6,10E-08	1.004391 seconds.
[9]	3	8,95	70,59	1,46E-10	1.165206 seconds.
[-22]	4	-21,88	471,44	5,31E-10	1,488246 seconds.

DFP					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[3]	1	-9,42E-19	-10	-3,74E-16	1.259402 seconds.
[9]	1	-1,11E-18	-10	-4,41E-16	1.035487 seconds.
[-22]	1	1,25E-18	-1,00E+01	4,95E-16	1.073453 seconds.

4.4.2 Duas variáveis

Para duas variaveis encontramos o mesmo comportamento apresentado para uma variável.

Gradiente					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[4;5]	1	(-6.9118e-19; -4.3368e-19)	-20	(-2.7425e-16; -1.7208e-16)	1.085538 seconds.
[-6;-4]	1	(1.8838e-18; 6.9118e-19)	-20	(7.4746e-16; 2.7425e-16)	1.116331 seconds.
[8;-8]	1	(-1.3824e-18; 1.3824e-18)	-20	(-5.485e-16; 5.485e-16)	1.133838 seconds.

Newton					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[4;5]	3	(3.9798; 4.9747)	20,79	(-9.9254e-11; -1.3957e-10)	1.751698 seconds.
[-6;-4]	3	(-5.9696; -3.9798)	31,74	(-8.7827e-11; 9.9254e-11)	1.677938 seconds.
[8;-8]	3	(7.9592; -7.9592)	1,07E+02	(-7.8094e-11; 7.8094e-11)	1.686419 seconds.

DFP					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[4;5]	1	(-6.9118e-19; -4.3368e-19)	-20	(-2.7425e-16; -1.7208e-16)	1.472706 seconds.
[-6;-4]	1	(1.8838e-18; 6.9118e-19)	-20,00	(7.4746e-16; 2.7425e-16)	1.366526 seconds.
[8;-8]	1	(-1.3824e-18; 1.3824e-18)	-20,00	(-5.485e-16; 5.485e-16)	1.428168 seconds.

4.4.3 Três variáveis

Novamente encontramos o mesmo comportamento apresentado anteriormente. Bem, podemos concluir que o método de Newton acabou se saindo inferior nos testes realizados, embora acabou mantendo uma velocidade rápida de convergência, ele foi o único método que não chegou convergiu em apenas uma iteração.

Gradiente					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[1;1;1]	1	(-1.7279e-19; -1.7279e-19; -1.7279e-19)	-30	(-6.8562e-17; -6.8562e-17; -6.8562e-17)	1.087279 seconds.
[-4;-8;2]	1	(6.9118e-19; 1.3824e-18; -3.4559e-19)	-30	(2.7425e-16; 5.485e-16; -1.3712e-16)	1.162042 seconds.
[4;5;20]	1	(-6.9118e-19; -4.3368e-19; -1.7347e-18)	-30	(-2.7425e-16; -1.7208e-16; -6.8831e-16)	1.316854 seconds.

Newton					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[1;1;1]	2	(0.99496; 0.99496; 0.99496)	-2,70E+01	(1.1572e-11; 1.1572e-11; 1.1572e-11)	2.866332 seconds.
[-4;-8;2]	3	(-3.9798; -7.9592; 1.9899)	5,36E+01	(9.9254e-11; 7.8094e-11; -5.5e-11)	2.144949 seconds.
[4;5;20]	4	3.9798; 4.9747; 19.891)	4,09E+02	(-9.9254e-11; -1.3957e-10; -3.2829e-10)	3.107022 seconds.

DFP					
ponto inicial x0	Vk	Vx	VF	VG	tempo [s]
[1;1;1]	1	(-1.7279e-19; -1.7279e-19; -1.7279e-19)	-30	(-6.8562e-17; -6.8562e-17; -6.8562e-17)	1.482098 seconds.
[-4;-8;2]	1	(6.9118e-19; 1.3824e-18; -3.4559e-19)	-3,00E+01	(2.7425e-16; 5.485e-16; -1.3712e-16)	1.518194 seconds.
[4;5;20]	1	(-6.9118e-19; -4.3368e-19; -1.7347e-18)	-3,00E+01	(-2.7425e-16; -1.7208e-16; -6.8831e-16)	1.937925 seconds.

5 Conclusão

Em suma podemos observar que independentemente do método aplicado a função em si vai ditar muito a convergência, sendo para alguns métodos indiferente a proximidade entre x_0 com o ponto ótimo, já para outro o mesmo pode ser dito sobre o numero de variáveis, etc. Foi possível perceber também que nossa convergência dos métodos foi global, todas as execuções foram para algum ponto de mínimo, isso só reforça a eficiência dos métodos, e do nosso código em si, os pontos foram escolhidos de maneira para estressar ao máximo os nossos algoritmos e conseguimos analisar situações bem interessantes a partir disso.

6 Anexos

6.1 figuras

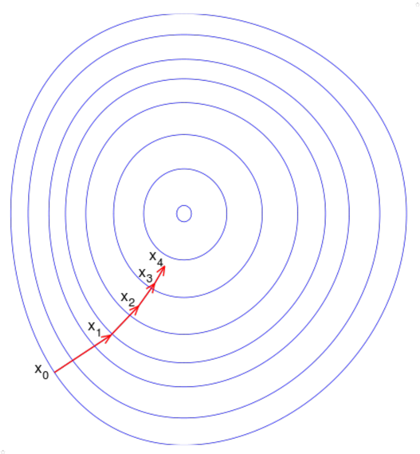

Figura 1: a

Referências

[1] Ana Friedlander. Elementos de programaçao nao-linear. 2012.