INF 1608 - Análise Numérica
Gradiente Descendente

Gabriel Vasconcellos, Eduardo Maksoud

12/12/2021

1 Introdução

O tema do trabalho é de otimização, seu objetivo é implementar e comparar duas variações do método do Gradiente Descendente (ou método do Máximo Declive), usado para encontrar o mínimo de funções com n variáveis.

A primeira variação do método define o passo por uma constante e a segunda emprega o método da Interpolação Parabólica Sucessiva (IPS) para determiná-lo.

Deseja-se comparar também os dois critérios para a definição de uma nova estimativa que podem ser adotados no método IPS, que são a substituição pela estimativa menos recente ou pela pior estimativa.

Para rodar o programa, basta utilizar a sequência de comandos abaixo:

```
    gcc -Wall descgrad.c main.c matriz.c -o myprog -lm
    ./myprog <numero de variaveis> <lista de valores para as variaveis>

Exemplo : ./myprog 2 1 2
```

2 Desenvolvimento

O módulo principal do trabalho é o “descgrad.h” e em seu desenvolvimento foram utilizadas funções do módulo “matriz.h” do primeiro laboratório da disciplina. O protótipo das duas funções do módulo “descgrad.h” está descrito abaixo:

A função abaixo recebe como parâmetros a dimensão do problema n, a função f de múltiplas variáveis desejada, um vetor v que representa a estimativa inicial e um valor a que representa o tamanho fixo do passo.

```c
    int descgrad_constante(int n, double (*f)(double *v), double *v,
                           double a);
```

A função abaixo recebe como parâmetros a dimensão do problema n, as estimativas r, s e t utilizadas no método de IPS, o critério c escolhido para o IPS, a função f de múltiplas variáveis desejada e um vetor v que representa a estimativa inicial.

```c
    int descgrad_ips(int n, int r, int s, int t, int c,
                     double (*f)(double *v), double *v);
```

O retorno das duas funções acima corresponde ao número de iterações necessárias para que se atinja a convergência. Nesta implementação, define-se que o resultado atingiu a convergência quando o gradiente se encontra abaixo de uma certa tolerância (um valor muito próximo de zero).

Considere o pseudo-código abaixo para entender o funcionamento dos algoritmos:

```
    while(verifica_convergencia(gradiente,tol))
        gradiente = calcula_gradiente(n,f,v,h);
        //definir o passo a com o metodo IPS ou usar uma constante

        //calcular a nova estimativa v
        for i = 0,1,2,3,...,n
            v[i] -= a * gradiente[i]
```

Como pode ser observado, no começo do algoritmo calcula-se o gradiente de forma numérica, e para isto é preciso preencher um vetor com as derivadas parciais em relação a todas as variáveis da função f.

Considerando que f_1 e f_2 são as taxas de variação da função f em relação a variável de interesse x, foi possível obter a derivada parcial a respeito de x através da fórmula abaixo:

$$\frac{\partial f}{\partial x} = \frac{f_1 - f_2}{2 \cdot h} \tag{1}$$

Aplicando esta fórmula a todas as variáveis da função f, monta-se o vetor do gradiente, como mostra a figura abaixo:

$$\Delta f = \begin{bmatrix} \frac{\partial f}{\partial x_0} \\ \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \tag{2}$$

A partir deste ponto, a implementação do algoritmo com passo constante é trivial, pois para calcular a próxima estimativa só é preciso conhecer o tamanho do passo a que foi fornecido como parâmetro e seguir na direção contrária do gradiente até que este se aproxime de zero, momento em que já seria possível considerar a que a resposta atingiu a convergência.

Entretanto, se desejamos obter o valor do passo visando minimizar a função $f(x)$ na direção oposta a do gradiente, precisamos utilizar o método IPS. O método recebe três estimativas: r, s e t, que são usadas para criar uma parábola que passa pelos pontos $f(r)$, $f(s)$ e $f(t)$. A nova estimativa do IPS será dada pelo ponto mínimo desta parábola, e pode ser calculada de acordo com a imagem abaixo:

$$u = \frac{r+s}{2} + \frac{(f(s) - f(r)) \cdot (t-r) \cdot (t-s)}{2 \cdot [(s-r) \cdot (f(t) - f(s)) - (f(s) - f(t)) \cdot (t-s)]} \tag{3}$$

Como dito anteriormente, dependendo do critério escolhido para o método IPS, a nova estimativa poderá substituir a pior estimativa ou a menos recente. No primeiro caso, a definição da pior estimativa é dada pela fórmula abaixo:

$$\textit{pior estimativa} = \max(\max(f(r), f(s)), f(t)) \tag{4}$$

3 Resultados e Análise

Para o trabalho foram utilizados valores de tolerância para o gradiente de $4.5 \cdot 10^{-2}$ e passo de 10^{-2} e testadas as seguintes funções:

$$f : f(x, y) = x^4 + y^4 + 2 \cdot x^2 \cdot y^2 + 6 \cdot x \cdot y - 4 \cdot x - 4 \cdot y + 1 \tag{5}$$

$$Rosenbrock2D : f(x, y) = 100 \cdot (y - x^2)^2 + (x - 1)^2 \tag{6}$$

$$Rosenbrock3D : f(x, y, z) = 100 \cdot (y - x^2)^2 + (x - 1)^2 + 100 \cdot (z - y^2)^2 + (y - 1)^2 \tag{7}$$

3.1 Tabelas com resultados da função f

Valores dos mínimos encontrados			
Estimativa \ Método	Passo constante	MIPS I	MIPS II
1, 2	-0.465, 1.13	-0.466, 1.13	-0.465, 1.13
2, 4	-0.465, 1.13	-0.465, 1.13	-0.465, 1.13
3, 10	-0.465, 1.13	-0.466, 1.13	-0.466, 1.13

Número de iterações para a convergência			
Estimativa \ Método	Passo constante	MIPS I	MIPS II
1, 2	25	6	6
2, 4	46	4	4
3, 10	102	5	5

Como pode ser observado, estimativas piores (mais afastadas do ponto mínimo da função) provocam um aumento no número de iterações do método com passo constante, enquanto que o número de iterações para o método do IPS foi significativamente menor e permaneceu o mesmo para as variações MIPS I e MIPS II.

3.2 Tabelas com resultados da função Rosenbrock 2D

Valores dos mínimos encontrados			
Estimativa \ Método	Passo constante	MIPS I	MIPS II
1, 2	1.04,1.07	1.04, 1.08	1.05,1.1
2, 4	1.04,1.07	1.01,1.03	1.03,1.06
3, 10	1.04,1.08	1.04,1.07	1.05,1.1

Número de iterações para a convergência			
Estimativa \ Método	Passo constante	MIPS I	MIPS II
1, 2	6291	185	331
2, 4	12995	421	505
3, 10	41133	1215	1757

Novamente o passo constante se mostrou ineficiente se comparado com a definição do passo através do IPS. Neste caso já é possível notar que o MIPS I foi mais eficiente do que o MIPS II, já que o número de iterações dele foi significativamente menor.

3.3 Tabelas com resultados da função Rosenbrock 3D

Valores dos mínimos encontrados			
Estimativa \ Método	Passo constante	MIPS I	MIPS II
1, 2, 3	1.01, 1.02, 1.05	0.966, 0.933, 0.871	1.01, 1.03, 1.06
2, 4, 6	1.01, 1.02, 1.05	0.965, 0.931, 0.866	0.967, 0.936, 0.875
3, 10, 15	1.01, 1.02, 1.05	0.967, 0.935, 0.875	1.02, 1.03, 1.06

Número de iterações para a convergência			
Estimativa \ Método	Passo constante	MIPS I	MIPS II
1, 2, 3	9614	2248	1295
2, 4, 6	22325	842	12168
3, 10, 15	82289	748	26978

Nota-se que a precisão do MIPS I neste experimento foi a pior entre os demais métodos. Ainda assim, nas últimas estimativas, o MIPS I chegou no resultado em um número muito menor de iterações. Outra questão interessante foi observar que o método do MIPS I precisou de um número menor de iterações na estimativa (3, 10, 15) que seria a mais afastada do ponto de mínimo da função. Além disso, o método do MIPS II partindo da estimativa (3, 10, 15) encontrou uma resposta mais precisa do que quando usou a estimativa (2, 4, 6).

3.4 Conclusão

Em suma, pode-se observar que conforme aumenta a dimensão, número de variáveis da função, o resultado da MIPS I e MIPS II são menos precisos, enquanto o passo constante mantém uma alta precisão.

Outro ponto importante é o aumento no número de iterações conforme aumenta a dimensão para o passo constante e, principalmente, para o MIPS II, que em casos mais afastados no mínimo global como a estimativa (3, 10) para Rosenbrock 2D teve seu mínimo calculado em 1757 iterações enquanto a estimativa (3, 10, 15) da Rosenbrock 3D teve 26978 iterações. Em contrapartida, o MIPS I não teve seu número de iterações aumentado, pois o seu gradiente alcançava valores muito pequenos, resultando no fim precoce do método.