MÔNICA DÂMARIS DE SOUZA ZANARDINI

Métodos Numéricos em Programação Não-Linear

CURITIBA

2002

Sumário

Capítulo 1

Método de Newton

Seja o problema de resolver o sistema de equações não-lineares $F(x) = 0$, onde

$$F(x) = \begin{pmatrix} f_1(x) & f_2(x) & \cdots & f_m(x) \end{pmatrix}^t$$

com $F : \mathbb{R}^n \to \mathbb{R}^m$.

No qual, a matriz jacobiana da função F é denotado por:

$$J(x) = \begin{pmatrix} \nabla f_1(x) & \nabla f_2(x) & \cdots & \nabla f_m(x) \end{pmatrix}^t .$$

Um dos processos para resolver $F(x) = 0$ é o método de Newton, desenvolvido por Simpson no século XVIII. Esse método é baseado na resolução aproximada de problemas.

O k-ésimo problema é dado pela aproximação de $F(x)$ pela série de Taylor em torno do ponto x_k:

$$F(x) \approx L_k(x) = F(x_k) + J(x_k)(x - x_k).$$

O ponto seguinte x_{k+1} é a solução de:

$$L_k(x) = F(x_k) + J(x_k)(x - x_k) = 0.$$

Se $J(x_k)$ é não singular temos que $L_k(x) = 0$ tem uma única solução, portanto o método de Newton em cada interação k consiste em resolver o seguinte sistema linear :

$$J(x_k)S_k = -F(x_k)$$

$$x_{k+1} = x_k + S_k.$$

Dessa forma o processo gera uma sequência de pontos $\{x_k\}$ que converge para a solução do problema.

O algoritmo para esse problema é dado por:

Dado $x_0 \in \mathbb{R}^n$.

Faça k=0.

Repita

Resolva $J(x_k)S_k = -F(x_k)$.

$x_{k+1} = x_k + S_k$.

$k = k + 1$.

Até a convergência.

1.1 Método de Newton com Busca Linear

Encontrar a solução de um sistema de equações não-lineares da forma $F(x) = 0$ é o mesmo que minimizar a seguinte função:

$$f(x) = \frac{1}{2}\|F(x)\|^2 = \frac{1}{2}\sum_{i=1}^{m} f_i(x)^2.$$

Nesse caso,

$$\nabla f(x) = J(x)^t F(x)$$

$$\nabla^2 f(x) = J(x)^t J(x) + \sum_{i=1}^{m} f_i(x)\nabla^2 f_i(x).$$

Com $F : \mathbb{R}^n \to \mathbb{R}^m$, $J(x) = F'(x) \in \mathbb{R}^{m\times n}$.

Considere o problema de minimização sem restrições da seguinte maneira:

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

Os métodos para resolver esse problema são iterativos, portanto dado um ponto x_k da iteração k, devemos encontrar x_{k+1} tal que $f(x_{k+1}) < f(x_k)$. Para garantir o decrescimento faremos $x_{k+1} = x_k + td_k$, onde $d_k \in \mathbb{R}^n$ é chamada de direção de descida, tal que, para $\epsilon > 0$ e $t \in (0, \epsilon]$ temos

$$f(x_k + td_k) < f(x_k).$$

As direções que formam um ângulo maior do que 90^0 com o gradiente da função em x_k são direções de descida.

Para obter t_k tal que $f(x_k+t_kd_k) < f(x_k)$ podemos utilizar a condição a seguir.

Condição de Armijo

Sejam $x_k, d_k \in \mathbb{R}^n$ tais que $\nabla f(x_k) \neq 0$, $\nabla f(x_k)^t d_k < 0$ e $\alpha \in (0,1)$. Existe $\epsilon = \epsilon(\alpha) > 0$ tal que

$$f(x_k + td_k) \leq f(x_k) + \alpha t \nabla f(x_k)^t d_k$$

para todo $t \in (0, \epsilon]$.

Essa condição evita passos grandes. Porém, passos muito pequenos são inevitáveis, simplismente porque passos grandes não permitem um decréscimo adequado, mas é importante que passos grandes sejam tentados. Por isso, o primeiro passo sempre será $t_k = 1$ e diminuir o passo sem exageros até que a condição de Armijo seja satisfeita. De certa forma, esse mecanismo não inibe passos curtos, pois o próprio d_k pode ser muito pequeno, isso motiva a introdução da 'condição β' definida por:

$$\|d_k\| \geq \beta \|\nabla f(x_k)\|,$$

com $\beta > 0$.

Não é interessante quando o ângulo de d_k e $\nabla f(x_k)$ tenda a 90^0, essa situação poderá provocar convergência a um ponto não estacionário, para inibir essa eventualidade temos a 'condição do ângulo'

$$\nabla f(x_k)^t d_k \leq -\theta \|\nabla f(x_k)\| \|d_k\|,$$

com $\theta \in (0,1)$.

Logo, o algoritmo de Newton com busca linear é:

Dados $x_0 \in \mathbb{R}^n$, $\alpha \in (0,1)$, $\beta > 0$, $\theta \in (0,1)$.

k=0.

Repita

Se $\nabla^2 f(x_k) > 0$.

Resolva $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$.

Senão

Defina $B_k = \nabla^2 f(x_k) + \mu I$, $\mu > 0$ tal que $B_k > 0$.

Resolva $B_k d_k = -\nabla f(x_k)$.

Se $\nabla f(x_k)^t d_k > -\theta \|\nabla f(x_k)\| \|d_k\|$.

Faça $\mu = \max\{2\mu, 10\}$.

Calcule $B_k = \nabla^2 f(x_k) + \mu I$.

Resolva $B_k d_k = -\nabla f(x_k)$.

Se $\|d_k\| < \beta \|\nabla f(x_k)\|$ faça $d_k = \beta \frac{\|\nabla f(x_k)\|}{\|d_k\|} d_k$.

Obter t_k de modo a satisfazer $f(x_k + t_k d_k) \leq f(x_k) + \alpha t_k \nabla f(x_k)^t d_k$.

$x_{k+1} = x_k + t_k d_k$.

$k = k + 1$.

Até a convergência.

Capítulo 2

Método Secante

O método Secante obedece a filosofia dos métodos quase-Newton e a maioria desses métodos tem como objetivo obter resultados satisfatórios com esforço computacional menor do que o método de Newton, fazendo uma aproximação da matriz Jacobiana.

Os métodos quase-Newton são baseados no seguinte formato:

$$B_k S_k = -F(x_k)$$

$$x_{k+1} = x_k + s_k.$$

O método secante é obtido através da generalização do problema unidimensional de mesmo nome. Pensando dessa maneira temos que na k-ésima iteração $F(x)$ é aproximada por:

$$F(x) \approx L_k(x) = F(x_k) + B_k(x - x_k).$$

Escrevendo para a iteração k+1:

$$F(x) \approx L_{k+1}(x) = F(x_{k+1}) + B_{k+1}(x - x_{k+1}).$$

A idéia é impor que a função linear $L_{k+1}(x)$ interpole x_k e x_{k+1}. Logo as seguintes igualdades devem ser satisfeitas $L_{k+1}(x_k) = F(x_k)$ e $L_{k+1}(x_{k+1}) = F(x_{k+1})$. Vemos que a condição $L_{k+1}(x_{k+1}) = F(x_{k+1})$ é automaticamente satisfeita. Quanto a condição $L_{k+1}(x_k) = F(x_k)$ temos que é equivalente à:

$$F(x_k) = F(x_{k+1}) + B_{k+1}(x_k - x_{k+1})$$

ou

$$B_{k+1}S_k = y_k,$$

onde $y_k = F(x_{k+1}) - F(x_k)$.

Esse sistema é linear onde a matriz B_{k+1} é a incógnita. Diferentes escolhas de B_{k+1} definem diferentes métodos secantes. Veremos aqui o método de Broyden, onde dada uma matriz inicial B_0 temos que na iteração k a matriz B_{k+1} é construida da seguinte maneira:

$$B_{k+1} = B_k + \frac{(y_k - B_k S_k)(S_k)^t}{(S_k)^t(S_k)}$$

onde $S_k = x_{k+1} - x_k$.

Para obter o próximo valor de x resolvemos o sistema $B_{k+1}S_{k+1} = -F(x_{k+1})$, logo $x = S_{k+1} + x_{k+1}$.

O método de Broyden é o mais utilizado da família dos métodos secantes.

Algoritmo:

Dados $x_0 \in \mathbb{R}^n$, $B_0 \in R^{n \times n}$.

k=0.

Repita

Resolva $B_k S_k = -F(x_k)$.

$x_{k+1} = x_k + S_k$.

$y_k = F(x_{k+1}) - F(x_k)$.

$B_{k+1} = B_k + \frac{(y_k - B_k S_k)(S_k)^t}{(S_k)^t (S_k)}$.

$k = k + 1$.

Até a convergência

Capítulo 3

Regiões de Confiança

Considere o problema genérico de otimização:

$$\text{Minimizar } f(x),$$

onde x pertence a um conjunto arbitrário Ω de $\mathbb{R}^n$. A idéia é, a cada iteração k, construir uma aproximação quadrática em torno de x_k:

$$f(x_k + h) \approx \psi(h) = f(x_k) + \nabla f(x_k)^t h + \frac{1}{2} h^t B_k h$$

onde $B_k \in \mathbb{R}^{n \times n}$ é simétrica.

Em um intervalo pequeno em torno de x_k podemos confiar no modelo, ou seja, no conjunto $\{x \in \Omega | \|h\| \leq \Delta\}$, no qual $\Delta > 0$.

A cada iteração k do algoritmo temos um subproblema que consiste em minimizar uma função quadrática com restrição de desigualdade $\psi(h) = f(x_k) + \nabla f(x_k)^t h + \frac{1}{2} h^t B_k h$ s.a. $\|h\| \leq \Delta$ e para a próxima iteração, se $f(x_k + h) < f(x_k)$

então fazemos $x_{k+1} = x_k + h_k$, onde h_k é a solução do subproblema, caso contrário $x_{k+1} = x_k$. A atualização de Δ depende de $\rho = \frac{f(x_k)-f(x_{k+1})}{\psi(0)-\psi(h_k)}$, se $\rho < 0.25$ o modelo está ruim nesse intervalo então diminuimos o valor de Δ, quando $\rho > 0.75$ o modelo está muito bom, Δ poderá ser aumentado e se $0.25 \leq \rho \leq 0.75$ o modelo está razoável e Δ não é atualizado.

Algoritmo para o método de Newton com regiões de confiança:

Dados $x_0 \in \mathbb{R}^n$, $\Delta_{min} > 0$, $\Delta_{max} > 0$, $\Delta_0 \in (\Delta_{min}, \Delta_{max})$

k=0.

Repita

$h_k = argmin\{f(x_k)+\nabla f(x_k)^t h+\frac{1}{2}h^t\nabla^2 f(x_k)h \quad \text{s.a.} \quad \|h\| \leq \Delta_k\}$.

Faça $\rho_k = \frac{f(x_k)-f(x_{k+1})}{\psi(0)-\psi(h_k)}$.

Se $\rho_k < 0.25$ então $\Delta_{k+1} = max\{\frac{1}{4}\|h_k\|, \Delta_{min}\}$.

Se $\rho_k > 0.75$ e $\|h_k\| = \Delta_k$ então $\Delta_{k+1} = min\{2\Delta_k, \Delta_{max}\}$.

Se '$0.25 \leq \rho_k \leq 0.75$ ' ou '$\rho_k > 0.75$ e $\|h_k\| < \Delta_k$' então $\Delta_{k+1} = \Delta_k$.

Se $\rho_k \leq 0$ então $x_{k+1} = x_k$ senão $x_{k+1} = x_k + h_k$.

Até a convergência

Para resolver o subproblema quadrático com restrição utilizamos o seguinte teorema:

TEOREMA 3.1 *O vetor h_k é uma solução global do problema de região de confiança se, e somente se, para $\lambda \geq 0$ as seguintes condições se verificam:*

$$(\nabla^2 f(x_k) + \lambda I)h_k = -\nabla f(x_k)$$

$$\lambda(\|h_k\| - \Delta) = 0$$

$$\nabla^2 f(x_k) + \lambda I \geq 0$$

Algoritmo para o subproblema quadrático:

Dados $\lambda_0 = 0$, $\Delta > 0$.

i=0.

Resolva $\nabla^2 f(x_k)h_i = -\nabla f(x_k)$.

Se $\|h_i\| \leq \Delta$ e $\nabla^2 f(x_k) \geq 0$ então $h_k = h_i$ é a solução do problema.

Se $\|h_i\| > \Delta$

$i = i + 1$.

$\lambda_i = 10^{-3}$.

Repita

Faça a fatoração de Cholesky: $R^t R = \nabla^2 f(x_k) + \lambda_i I$.

Resolva $R^t s = -\nabla f(x_k)$ e $Rh_i = s$.

$\lambda_{i+1} = \lambda_i + \left(\frac{\|h_i\|}{\|s\|}\right)^2 \left(\frac{\|h_i\| - \Delta}{\Delta}\right)$.

i=i+1.

Enquanto $\|h_i\| \neq \Delta$.

$h_k = h_i$.

Capítulo 4

Penalização

A penalização é um procedimento que consiste em converter problemas complexos em outros cuja resolução é conhecida. A penalização acrescenta um termo na função objetivo de maneira que a restrição seja eliminada.

Na penalização interna a função objetivo é modificada agregando um termo funcional que tende a infinito quando o ponto se aproxima da fronteira. Os métodos de penalização interna são conhecidos por métodos de barreira.

Na penalização externa acrescenta-se um termo na função objetivo cujo custo aumenta com a violação das restrições.

4.1 Métodos de Barreira

Consideraremos os problemas de otimização da seguinte forma:

$$\text{Minimizar } f(x)$$

$$\text{s.a. } c(x) \geq 0$$

$$x \in \mathcal{D}$$

onde $\mathcal{D}$ é um subconjunto do $\mathbb{R}^n$, $c : \mathbb{R}^n \to \mathbb{R}^m$ e $\Omega = \{x \in \mathcal{D} | c(x) \geq 0\}$.

Existem dois exemplos clássicos de funções barreiras para esse problema: a função inversa

$$P(x) = \sum_{i=1}^{m} \frac{1}{c_i(x)}$$

e a função barreira logarítmica

$$P(x) = -\sum_{i=1}^{m} \log(c_i(x)).$$

Acrescentando a função barreira à função objetivo do problema de otimização temos:

$$\text{Minimizar } f(x) + tP(x)$$

$$\text{s.a. } x \in \Omega^0,$$

onde $t > 0$ e decresce a cada iteração.

Método básico de penalização interna:

Dados $t_1 > 0$, $x_0 \in \Omega^0$.

k=1;

Repita

Encontre $x_k = x(t_k)$ solução global de

$$\text{Minimizar } f(x) + t_k P(x)$$

$$\text{s.a. } x \in \Omega^0.$$

Escolher t_{k+1} tal que $0 < t_{k+1} < t_k$.

k=k+1.

Até a convergencia.

4.2 Penalização Externa

Seja o seguinte problema de otimização:

$$\text{Minimizar } f(x)$$

$$\text{s.a. } h(x) = 0,$$

onde $h : \mathbb{R}^n \to \mathbb{R}^m$ e $\Omega = \{x \in \mathbb{R}^n | h(x) = 0\}$.

Veremos alguns exemplos de funções de penalização para esse problema:

$$P(x) = \|h(x)\|_2^2$$

$$P(x) = \|h(x)\|_2$$

$$P(x) = \|h(x)\|_1$$

Podemos reescrever o problema de otimização com restrições de igualdade como:

$$\text{Minimizar } f(x) + \rho P(x), \quad x \in \mathbb{R}^n.$$

O parâmetro ρ aumenta a cada iteração.

Sistematizando as idéias temos o seguinte algoritmo de penalização externa:

Dados $\rho_1 \geq 0$, $x_0 \in \mathbb{R}^n$.

k=1;

Repita

Encontre $x_k = x(\rho_k)$ solução global de

$$\text{Minimizar } f(x) + \rho_k P(x).$$

Escolher ρ_{k+1} tal que $\rho_{k+1} > \rho_k$.

k=k+1.

Até a convergencia.

Considere o seguinte problema com restrições de desigualdade

$$\text{Minimizar } f(x)$$

$$\text{s.a. } c(x) \geq 0,$$

onde $c : \mathbb{R}^n \to \mathbb{R}^p$ e $\Omega = \{x \in \mathbb{R}^n | c(x) \geq 0\}$.

Nesse caso podemos usar a função penalização descrita abaixo:

$$P(x) = \sum_{i=1}^{p} (\min \{0, c_i(x)\})^2.$$

Capítulo 5

Lagrangeano Aumentado

Consideraremos o problema de minimização com restrições de igualdade

$$\text{Minimizar } f(x) \quad \text{(PE)}$$

$$\text{s.a. } h(x) = 0,$$

onde $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$.

As condições de otimalidade de primeira ordem para esse problema são dadas pelo sistema não-linear

$$\nabla f(x) + h'(x)^t y = 0$$

$$h(x) = 0.$$

Definindo a função Lagrangeana de maneira usual $\mathrm{l}(x, y) = f(x) + h(x)^t y$ e fazendo $\nabla \mathrm{l}(x, y) = 0$ obtemos as condições de otimalidade descritas acima, portanto, um problema equivalente para (PE) é dado por:

$$\text{Minimizar } f(x) + y^t h(x)$$

$$\text{s.a. } h(x) = 0.$$

Aplicando a penalização quadrática

$$\text{Minimizar } f(x) + y^t h(x) + \frac{\rho}{2} h(x)^t h(x),$$

resolvendo esse problema, obtemos

$$\nabla f(x) + h'(x)^t y + \rho h'(x)^t h(x) = 0$$

ou

$$\nabla f(x) + h'(x)^t (y + \rho h(x)) = 0.$$

Esse raciocínio sugere o seguinte algoritmo para o problema de minimização com restrições de igualdade (PE):

Dados $x_0 \in \mathbb{R}^n$, $\rho_1 > 0$ e $y_1 \in \mathbb{R}^m$.

k=1.

Repita

x_k recebe o minimizador de $f(x) + y_k^t h(x) + \frac{\rho_k}{2} h(x)^t h(x)$ tomando x_{k-1} como ponto inicial.

Se $\|h(x_k)\| > 0.1\|h(x_{k-1})\|$ então $\rho_k = 10\rho_k$.

Faça $y_{k+1} = y_k + \rho_k h(x_k)$.

$\rho_{k+1} = \rho_k$.

$k = k + 1$.

Até a convergencia.

Em cada passo do método é garantido, pelo processo de minimização que $\nabla f(x_k) + h'(x_k)^t(y_k + \rho h(x_k)) = 0$.

Capítulo 6

Conclusão

O método de Newton para resolver um sistema de equações não-lineares utiliza a matriz Jacobiana , enquanto o método secante faz uma aproximação dessa matriz, cada método aqui tratado possue suas características próprias abrangendo diferentes problemas de otimização.

Referências Bibliográficas

[1] Martinez, José Mário. Santos, Sandra Augusta. **20º Colóquio de Matemática - Métodos Computacionais de Otimização** . Rio de Janeiro, IMPA, 1995.

[2] Nocedal, Jorge. Wright, Stephen J. **Numerical Optimization**. New York: Springer-Verlag, 1999.