Formulário

Equação iterativa		Convergência	
secante	Newton	superlinear	quadrática
$x_{k+1} = x_k - \frac{(x_k - x_{k-1}) f(x_k)}{f(x_k) - f(x_{k-1})}$	$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$	$\lim\limits_{k \to +\infty} \frac{\lvert x^* - x_{k+1}\rvert}{\lvert x^* - x_k\rvert^{1.618}} = L$	$\lim\limits_{k \to +\infty} \frac{\lvert x^* - x_{k+1}\rvert}{\lvert x^* - x_k\rvert^{2}} = L$

Convergência: $\frac{f''(\xi)}{2f'(\eta)}$. Onde $\xi, \eta \in I$.   CP: $\frac{|x_{k+1} - x_k|}{|x_{k+1}|} \le \epsilon_1$ e $|f(x_{k+1})| \le \epsilon_2$

Equação iterativa		CP
Jacobi	Gauss-Seidel	
$Dx^{(k+1)} = (L+U)x^{(k)} + b$	$(D-L)x^{(k+1)} = Ux^{(k)} + b$	$\frac{\lVert x^{(k+1)} - x^{(k)} \rVert}{\lVert x^{(k+1)} \rVert} \le \epsilon$
$x^{(k+1)} = C_J x^{(k)} + M^{-1} b$	$x^{(k+1)} = C_{GS} x^{(k)} + M^{-1} b$	
$M = D$, $DC_J = (L+U)$	$M = (D-L)$, $MC_{GS} = U$	

Jacobiano	Equação iterativa	CP
$\begin{pmatrix} \frac{\partial f_1(x_1,x_2,\ldots,x_n)}{\partial x_1} & \ldots & \frac{\partial f_1(x_1,x_2,\ldots,x_n)}{\partial x_n} \\ \ldots & & \ldots \\ \frac{\partial f_n(x_1,x_2,\ldots,x_n)}{\partial x_1} & \ldots & \frac{\partial f_n(x_1,x_2,\ldots,x_n)}{\partial x_n} \end{pmatrix}$	$x^{(k+1)} = x^{(k)} + \Delta_x$ $J(x^{(k)})\Delta_x = -f(x^{(k)})$	$\frac{\lVert \Delta_x \rVert}{\lVert x^{(k+1)} \rVert} \le \epsilon_1$ $\lVert f(x^{(k+1)}) \rVert \le \epsilon_2$

Polinómio interpolador de Newton:

$$\boxed{p_n(x) = f_0 + (x - x_0)[x_0, x_1] + \cdots + (x - x_0)\cdots(x - x_{n-1})[x_0, \ldots, x_n]}$$

Erro de truncatura: $\boxed{R_n(x) = (x - x_0)(x - x_1)\cdots(x - x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!}, \text{ com } \xi \in [x_0, x_n]}$

Expressão do segmento i da *spline* cúbica

$$\begin{aligned} s_3^i(x) = {} & \frac{M_{i-1}}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{M_i}{6(x_i - x_{i-1})}(x - x_{i-1})^3 + \left[\frac{f_{i-1}}{(x_i - x_{i-1})} - \frac{M_{i-1}(x_i - x_{i-1})}{6}\right](x_i - x) \\ & + \left[\frac{f_i}{(x_i - x_{i-1})} - \frac{M_i(x_i - x_{i-1})}{6}\right](x - x_{i-1}) \text{ para } i = 1, 2, \ldots, n \end{aligned}$$

Expressão para os pontos interiores (nó i)

$$\begin{aligned} (x_i - x_{i-1})M_{i-1} + 2(x_{i+1} - x_{i-1})M_i + (x_{i+1} - x_i)M_{i+1} = {} & \frac{6}{(x_{i+1} - x_i)}(f_{i+1} - f_i) \\ & - \frac{6}{(x_i - x_{i-1})}(f_i - f_{i-1}) \end{aligned}$$

Spline Natural	*Spline* Completa
$M_0 = 0$ $M_n = 0$	$2(x_1 - x_0)M_0 + (x_1 - x_0)M_1 = \frac{6}{(x_1 - x_0)}(f_1 - f_0) - 6f_0'$ $2(x_n - x_{n-1})M_n + (x_n - x_{n-1})M_{n-1} = 6f_n' - \frac{6}{(x_n - x_{n-1})}(f_n - f_{n-1})$

Erro de truncatura *spline* cúbica $\boxed{|f(x) - s_3(x)| \le \frac{5}{384}h^4 M_4}$ $\boxed{\left|f'(x) - s_3'(x)\right| \le \frac{1}{24}h^3 M_4}$ com $\max_{\xi \in [x_0, x_n]} |f^{(iv)}(\xi)| \le M_4 \qquad h = \max_{0 \le i \le n-1}(x_{i+1} - x_i)$

Modelo linear

$$\begin{pmatrix} \sum_{i=1}^m \phi_1^2(x_i) & \ldots & \sum_{i=1}^m \phi_1(x_i)\phi_n(x_i) \\ \ldots & \ldots & \ldots \\ \sum_{i=1}^m \phi_n(x_i)\phi_1(x_i) & \ldots & \sum_{i=1}^m \phi_n^2(x_i) \end{pmatrix} \begin{pmatrix} c_1 \\ \ldots \\ c_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m f_i \phi_1(x_i) \\ \ldots \\ \sum_{i=1}^m f_i \phi_n(x_i) \end{pmatrix}$$

Resíduos dos mínimos quadrados: $\boxed{\sum_{i=1}^m (f_i - M(x_i))^2}$

Polinómios ortogonais: $\boxed{P_{i+1} = A_i(x - B_i)P_i(x) - C_iP_{i-1}(x)}$, $\boxed{P_0(x) = 1}$, $\boxed{P_{-1}(x) = 0}$

$\boxed{A_i = 1}$ $\boxed{B_i = \frac{\langle xP_i(x), P_i(x)\rangle}{\langle P_i(x), P_i(x)\rangle}}$ $\boxed{C_0 = 0 \text{ e } C_i = \frac{\langle P_i(x), P_i(x)\rangle}{\langle P_{i-1}(x), P_{i-1}(x)\rangle}}$

Coeficientes do modelo polinomial $\boxed{c_j = \frac{\sum_{i=1}^m f_iP_j(x_i)}{\sum_{i=1}^m P_j(x_i)^2}}$.

Método de Nelder-Mead - Notação

Simplex $< X_1, \ldots, X_{n+1} >$ ordenado por ordem crescente dos valores de f		
centroide	$\bar{x} = \frac{1}{n}\sum_{i=1}^n X_i$	
reflectido	$x_r = (1+\alpha)\bar{x} - \alpha X_{n+1}$	$\alpha = 1$
expandido	$x_e = \gamma x_r + (1-\gamma)\bar{x}$	$\gamma = 2$
contraído interior	$x_c = \beta X_{n+1} + (1-\beta)\bar{x}$	$\beta = \frac{1}{2}$
contraído exterior	$\hat{x}_c = \beta x_r + (1-\beta)\bar{x}$	$\beta = \frac{1}{2}$
encolher simplex	$x_i = \frac{X_i + X_1}{2}$	
critério de paragem	$\frac{1}{\Delta}\max_{2\le i\le n+1}\|X_i - X_1\| \le \epsilon$	$\Delta = \max(1, \|X_1\|)$

Método de Nelder-Mead - Algoritmo

Dado o Simplex ordenado fazer:

1. Se $f(x_r) < f(X_n)$ então
   (a) Se $f(x_r) \ge f(X_1)$ aceitar reflectido e saltar para o passo 3.
   (b) Se $f(x_r) < f(X_1)$ então expandir simplex:
      i. Se $f(x_e) < f(X_1)$ então aceitar expandido e saltar para o passo 3.
      ii. Se $f(x_e) \ge f(X_1)$ então aceitar reflectido e saltar para o passo 3.
2. Se $f(x_r) \ge f(X_n)$ então contrair simplex:
   (a) Se $f(x_r) \ge f(X_{n+1})$ contrair para o interior.
      i. Se $f(x_c) < f(X_n)$ aceita-se o contraído para o interior e saltar para o passo 3.
      ii. Se $f(x_c) \ge f(X_n)$ encolhe-se o simplex e saltar para o passo 3.
   (b) Se $f(x_r) < f(X_{n+1})$ contrair para o exterior.
      i. Se $f(\hat{x}_c) < f(X_n)$ aceita-se o contraído para o exterior e saltar para o passo 3.
      ii. Se $f(\hat{x}_c) \ge f(X_n)$ encolhe-se o simplex e saltar para o passo 3.
3. Se o critério de paragem não for verificado iniciar nova iteração.

Redução significativa (critério de Armijo): $\boxed{f(x^{(k)}) - f(x^{(k)} + \omega^j\alpha d^{(k)}) \ge -\mu_1\omega^j\alpha g(x^{(k)})^Td^{(k)}}$.

Divisões sucessivas de α por dois consiste em usar $\omega = 0.5$, $\alpha = 1$ e determinar o menor j inteiro ($j = 0, 1, 2, 3 \ldots$) que provoca uma redução significativa.

Direcção de descida: $\boxed{g(x^{(k)})^Td^{(k)} \le \eta\|g(x^{(k)})\|\|d^{(k)}\|}$.

Quase ortogonal: $\boxed{|g(x^{(k)})^Td^{(k)}| \le \eta\|g(x^{(k)})\|\|d^{(k)}\|}$.

Critério de paragem: $\boxed{\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k+1)}\|} \le \epsilon_1 \text{ e } \frac{|f^{(k+1)} - f^{(k)}|}{|f^{(k+1)}|} \le \epsilon_2 \text{ e } \|g^{(k+1)}\| \le \epsilon_3}$

Método de Descida Máxima - Algoritmo

Em cada iteração fazer:

1. Calcular $d^{(k)} = -g(x^{(k)})$.
2. Usando divisões sucessivas de α por dois determinar $\alpha^{(k)}$.
3. Fazer $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$.
4. Se o critério de paragem não for verificado iniciar nova iteração.

Método de segurança de Newton - Algoritmo

Em cada iteração fazer:

1. Resolver o sistema linear $G(x^{(k)})d^{(k)} = -g(x^{(k)})$. Se o sistema não tiver solução única fazer $d^{(k)} = -g^{(k)}$ e ir para o passo 4.
2. Se a direcção for *quase* ortogonal ao gradiente fazer $d^{(k)} = -g^{(k)}$ e ir para o passo 4.
3. Se a direcção não for de descida fazer $d^{(k)} = -d^{(k)}$.
4. Usando divisões sucessivas de α por dois determinar $\alpha^{(k)}$.
5. Fazer $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$.
6. Se o critério de paragem não for verificado iniciar nova iteração.

Método quasi-Newton

Aproximação à inversa da Hessiana BFGS: $\boxed{H^{(0)} = I}$

$$\boxed{H^{(k+1)} = \left(I - \frac{s^{(k)} y^{(k)T}}{s^{(k)T} y^{(k)}}\right) H^{(k)} \left(I - \frac{y^{(k)} s^{(k)T}}{y^{(k)T} s^{(k)}}\right) + \frac{s^{(k)} s^{(k)T}}{s^{(k)T} y^{(k)}}}$$

$$\boxed{s^{(k)} = \alpha^{(k)} d^{(k)} = x^{(k+1)} - x^{(k)} \text{ e } y^{(k)} = g^{(k+1)} - g^{(k)}}$$

Método quasi-Newton - Algoritmo

Em cada iteração fazer:

1. Calcular $d^{(k)} = -H^{(k)} g(x^{(k)})$.
2. Usando divisões sucessivas de α por dois determinar $\alpha^{(k)}$.
3. Fazer $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$.
4. Se o critério de paragem não for verificado então calcular $H^{(k+1)}$ e iniciar nova iteração.
