Explicação numérica: mínimos quadrados para aproximar uma função quadrática

Deseja-se criar uma função quadrática $f(\alpha) = \theta_1 + \theta_2\alpha + \theta_3\alpha^2$ que melhor aproxime uma série de pontos $(x_i, y_i) \in \mathbb{R}^2$.

A solução é similar à do Exemplo 2. Pode-se pensar em um sistema

$$\theta_1 + \theta_2 x_1 + \theta_3 x_1^2 = y_1$$
$$\dots$$
$$\theta_1 + \theta_2 x_i + \theta_3 x_i^2 = y_i$$

correspondente a

$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_i & x_i^2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_i \end{bmatrix}.$$

Obteve-se então A e b para $||A\theta - b||^2$,

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_i & x_i^2 \end{bmatrix}, b = \begin{bmatrix} y_1 \\ \vdots \\ y_i \end{bmatrix}, \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

com o vetor residual r, $r_i = \theta_1 + \theta_2 x_i + \theta_3 x_i^2 - y_i$.

Uma implementação prática deste caso pode ser vista em “parabola.py”.