Introdução à Computação em Física

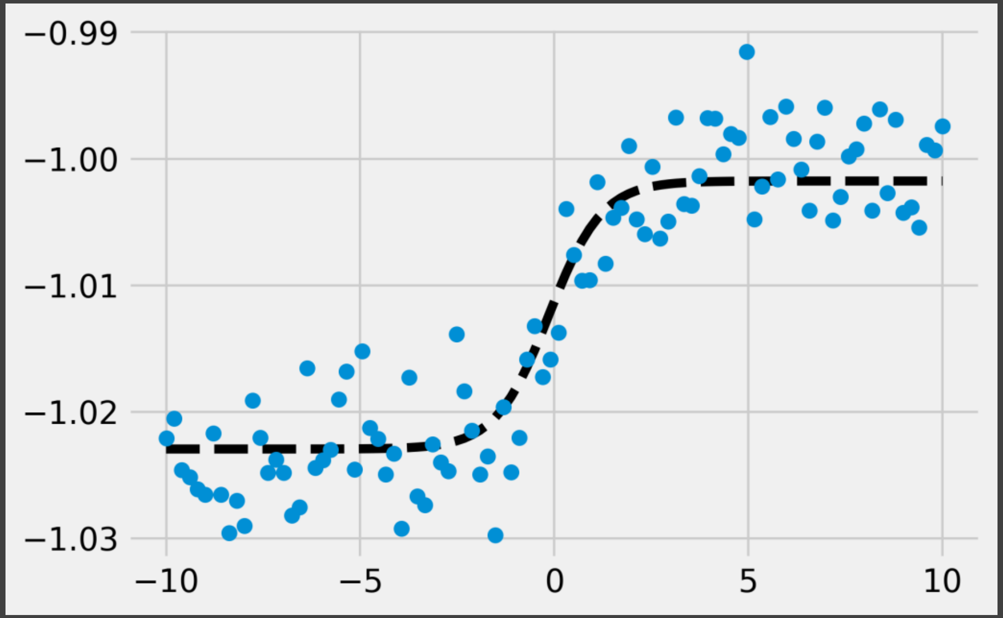
MÍNIMOS QUADRADOS

PROF. WALBER

Refs.:
Cálculo numérico, aspectos teóricos e computacionais (2nd edição), M. A. G. Ruggiero, V. L. da Rocha Lopes
Computational Physics, R. Landau, M. J. Paez, C. C. Bordelanu

Ajustes de curvas

Experimentos em laboratório geram um conjunto de dados que requerem uma análise gráfica em muitos casos. A obtenção de uma função matemática que represente (ajuste) os dados obtidos permite um melhor estudo do fenômeno em questão.



Mínimos quadrados

Seja um conjunto de pontos obtidos de um experimento:

$$(x_k, f(x_k)), k = 1, 2, ..., m$$

E sejam n funções (m > n)

$$g_1(x), g_2(x), \ldots, g_n(x)$$

A ideia é encontrar os coeficientes α_1, α_2, α_3, ..., α_n tais que a função abaixo se aproxime ao máximo de f(x)

$$\varphi(x) = \alpha_1 g_1(x) + \alpha_2 g_2(x) + \cdots + \alpha_n g_n(x)$$

Mínimos quadrados

O método dos mínimos quadrados consiste em determinar os α's de tal forma que a soma do quadrado dos desvios

$$d_k = f(x_k) - \varphi(x_k)$$

Seja mínima. Ou seja, queremos minimizar

$$\sum_{k=1}^{m} d_k^2 = \sum_{k=1}^{m} (f(x_k) - \varphi(x_k))^2$$

Seja a função F:

$$F(\alpha_1, \alpha_2, \ldots, \alpha_n) = \sum_{k=1}^{m} (f(x_k) - \varphi(x_k))^2$$

Mínimos quadrados

Temos então que minimizar

$$F(\alpha_1, \alpha_2, \ldots, \alpha_n) = \sum_{k=1}^{m} (f(x_k) - \alpha_1 g_1(x_k) - \alpha_2 g_2(x_k) - \cdots - \alpha_n g_n(x_k))^2$$

Podemos fazer isso através da primeira derivada:

$$\left.\frac{\partial F}{\partial \alpha_j}\right|_{(\alpha_1, \alpha_2, \ldots, \alpha_n)} = 0$$

j=1, 2, ..., n

Mínimos quadrados

Assim sendo,

$$\left.\frac{\partial F}{\partial \alpha_j}\right|_{(\alpha_1,\alpha_2,\ldots,\alpha_n)} = 2\sum_{k=1}^{m}[f(x_k) - \alpha_1 g_1(x_k) - \alpha_2 g_2(x_k) - \cdots - \alpha_n g_n(x_k)][-g_j(x_k)]$$

Logo, temos que

$$\sum_{k=1}^{m}[f(x_k) - \alpha_1 g_1(x_k) - \alpha_2 g_2(x_k) - \cdots - \alpha_n g_n(x_k)][g_j(x_k)] = 0$$

j=1, 2, ..., n

Temos por fim, uma equação para cada valor de j.

Mínimos quadrados

Dessas equações temos o seguinte sistema linear,

$$\begin{cases} [\sum_{k=1}^{m} g_1(x_k)g_1(x_k)]\alpha_1 + \cdots + [\sum_{k=1}^{m} g_n(x_k)g_1(x_k)]\alpha_n = \sum_{k=1}^{m} f(x_k)g_1(x_k) \\ [\sum_{k=1}^{m} g_1(x_k)g_2(x_k)]\alpha_1 + \cdots + [\sum_{k=1}^{m} g_n(x_k)g_2(x_k)]\alpha_n = \sum_{k=1}^{m} f(x_k)g_2(x_k) \\ \cdots \\ [\sum_{k=1}^{m} g_n(x_k)g_1(x_k)]\alpha_1 + \cdots + [\sum_{k=1}^{m} g_n(x_k)g_n(x_k)]\alpha_n = \sum_{k=1}^{m} f(x_k)g_n(x_k) \end{cases}$$

Sistema linear com n equações e n incógnitas: $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$

Mínimos quadrados

Sistema linear pode ser escrito na forma matricial Aα = b,

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

$$b = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{pmatrix}$$

$$b_i = \sum_{k=1}^{m} f(x_k) g_i(x_k)$$

$$a_{ij} = \sum_{k=1}^{m} g_j(x_k) g_i(x_k)$$

Sobre solução dos sistemas lineares

- Sistemas lineares podem ser resolvidos pelos métodos vistos nas aulas anteriores. Podemos, por exemplo, utilizar nossa implementação do método de eliminação de Gauss como um módulo externo.

Ex.:

```
from eliminGauss import *
```

- Além disso, podemos utilizar a parte de álgebra linear da biblioteca SciPy ou Numpy. Possuem funções que nesse sentido são similares.

Sobre solução dos sistemas lineares

- Vale ressaltar que as opções usadas no python são *wrappers* das rotinas da Linear Algebra Package (LAPACK). Veremos mais à frente no curso mais sobre a LAPACK (http://www.netlib.org/lapack/).

Ex.:

```
import numpy as np

### Matriz dos coeficientes
A = np.array([[-7.0, 3.0, 2.0],
     [1.0, 2.0, -1.0],
     [1.0, 1.0, -2.0]])

### vetor constante
b = np.array([-2.0, 2.0, 0.0])


### solucao

x = np.linalg.solve(A,b)

print 'Solucao x = ', x
```

- Baseado na rotina dgesv:

DGESV computes the solution to a real system of linear equations
   A * X = B

The LU decomposition with partial pivoting and row interchanges is
 used to factor A as   A = P * L * U,
 where P is a permutation matrix, L is unit lower triangular, and U is
 upper triangular. The factored form of A is then used to solve the
 system of equations A * X = B.

Sobre solução dos sistemas lineares

- De maneira similar, via SciPy

Ex.:

```python
import numpy as np
from scipy import linalg

### Matriz dos coeficientes
A = np.array([[-7.0, 3.0, 2.0],
     [1.0, 2.0, -1.0],
     [1.0, 1.0, -2.0]])

### vetor constante
b = np.array([-2.0, 2.0, 0.0])


### solucao usando SciPy

x = linalg.solve(A,b)

print 'Solucao x = ', x
```

- Baseado também nas rotinas LAPACK:

The generic, symmetric, Hermitian and positive definite solutions are obtained via calling ?GESV, ?SYSV, ?HESV, and ?POSV routines of LAPACK respectively.

Sobre solução dos sistemas lineares

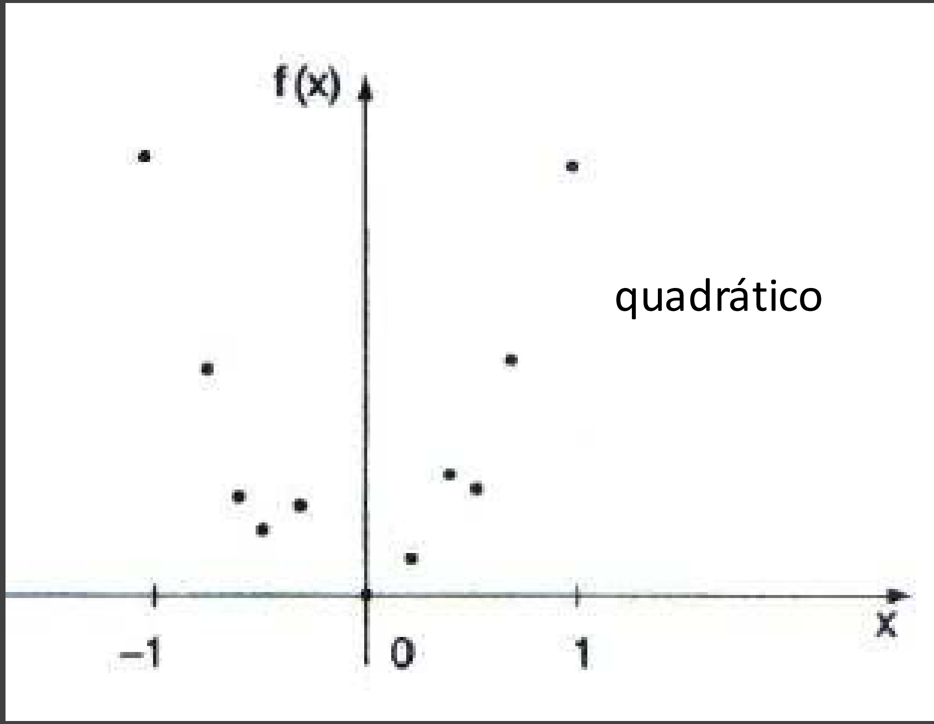
- Exemplo: Obtendo a matriz LU via SciPy e solução do sistema linear

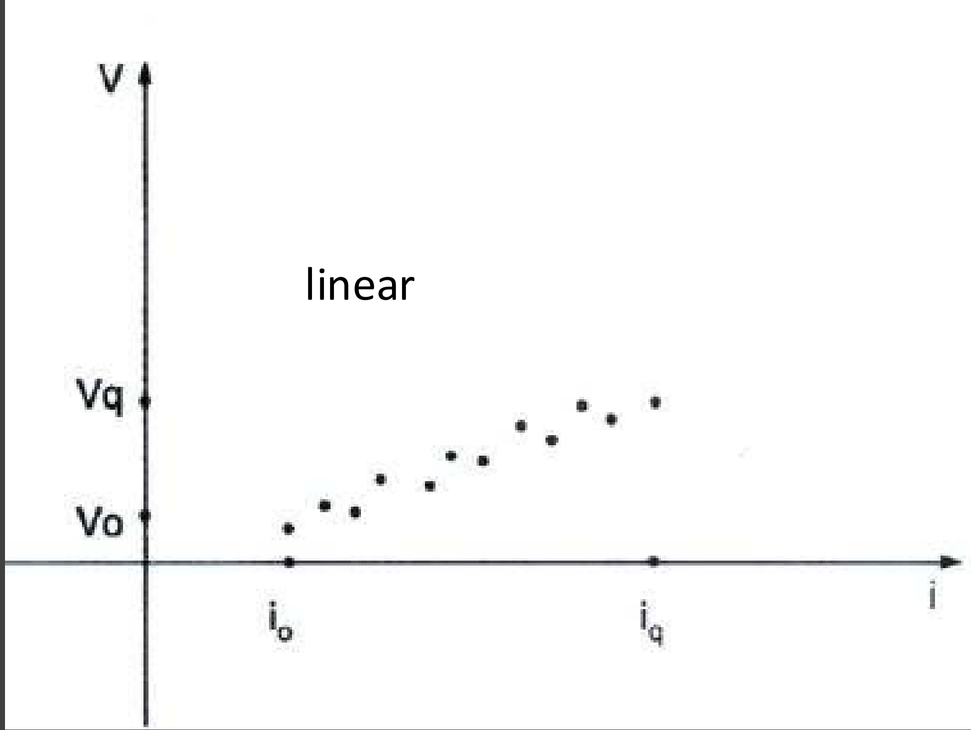
```python
import numpy as np
from scipy import linalg

### Matriz dos coeficientes
A = np.array([[-7.0, 3.0, 2.0],
     [1.0, 2.0, -1.0],
     [1.0, 1.0, -2.0]])

### vetor dos coeficientes
b = np.array([-2.0, 2.0, 0.0])

### Obtendo o produto das matrizes LU, e o vetor associado ao pivotamento
lu, piv = linalg.lu_factor(A, overwrite_a=False, check_finite=True)

print ' LU  = ', lu

print ' piv = ', piv

### Solucao do sistema linear
x = linalg.lu_solve((lu,piv),b)

print ' Solucao  x = ', x
```

lu(N, N) ndarray
Matrix containing U in its upper triangle, and L in its lower triangle. The unit diagonal elements of L are not stored.
piv(N,) ndarray
Pivot indices representing the permutation matrix P: row i of matrix was interchanged with row piv[i].

Retomando : Mínimos quadrados

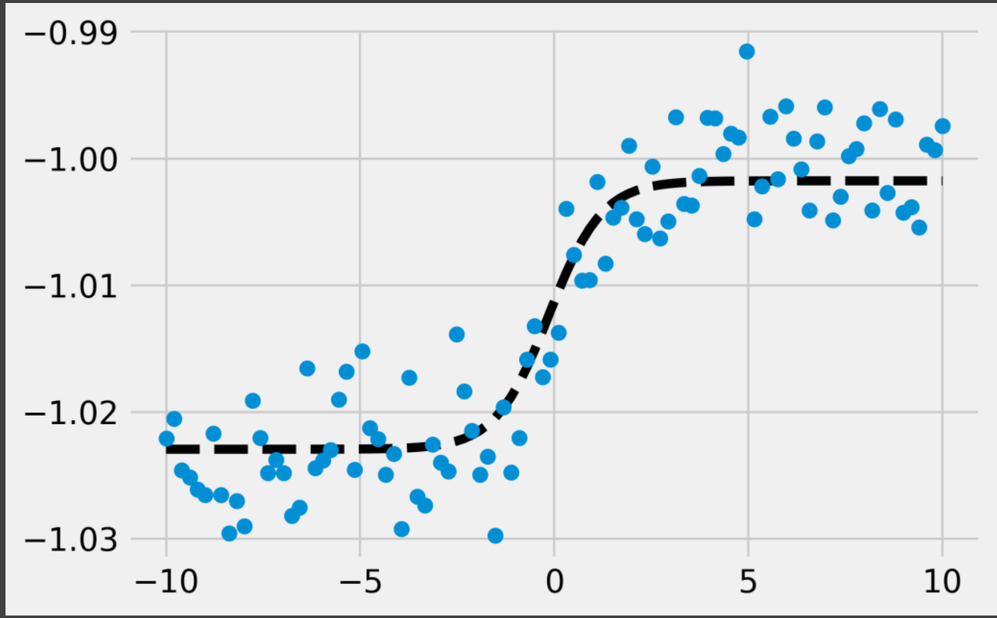
Escolha das funções dependem do embasamento teórico do experimento, ou podem ser estimadas do ponto de vista gráfico:





Mínimos quadrados

Medida da qualidade do ajuste da curva pode ser obtido pelo cálculo de d:



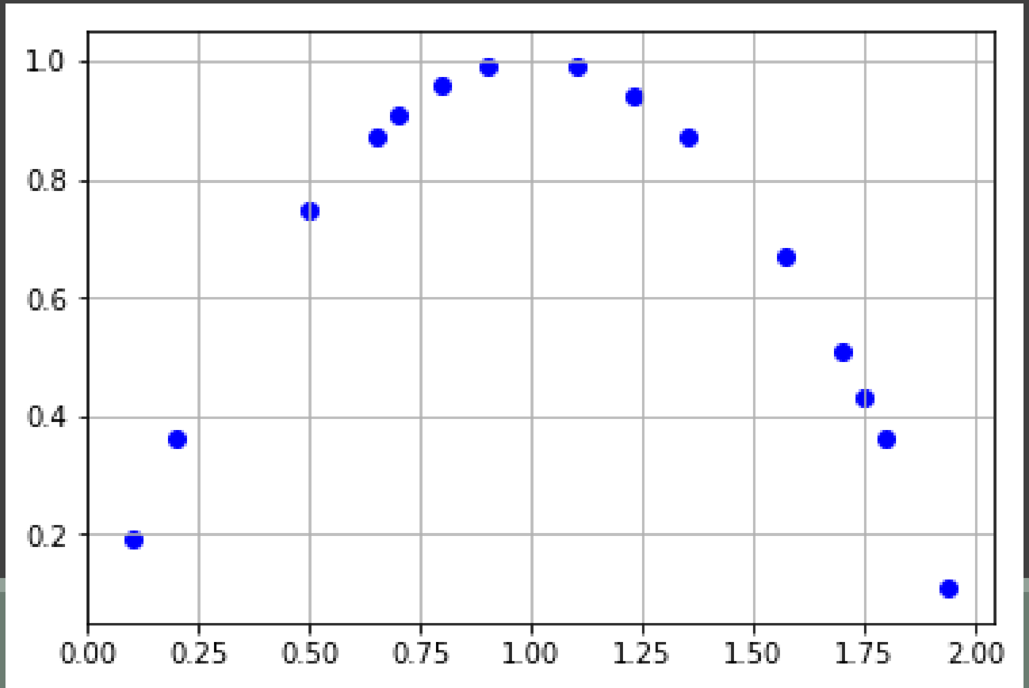
$$d = \sum_{k=1}^{m} [f(x_k) - \varphi(x_k)]^2$$

Pode-se dizer que em nosso contexto, quanto menor for o valor de d, melhor o ajuste.

Implementação (atividade prática)

1. Vamos considerar que temos os seguintes pontos experimentais:

x	0.10	0.20	0.50	0.65	0.70	0.80	0.90	1.10	1.23	1.35	1.57	1.70	1.75	1.80	1.94
f(x)	0.19	0.36	0.75	0.87	0.91	0.96	0.99	0.99	0.94	0.87	0.67	0.51	0.43	0.36	0.11



Iremos nesse caso considerar:

$$g_1(x) = 1, g_2(x) = x, g_3(x) = x^2$$

Implementação (atividade prática)

1. Vamos considerar que temos os seguintes pontos experimentais:

x	0.10	0.20	0.50	0.65	0.70	0.80	0.90	1.10	1.23	1.35	1.57	1.70	1.75	1.80	1.94
f(x)	0.19	0.36	0.75	0.87	0.91	0.96	0.99	0.99	0.94	0.87	0.67	0.51	0.43	0.36	0.11

$$g_1(x) = 1, g_2(x) = x, g_3(x) = x^2$$

$$a_{11} = \sum_{k=1}^{m} 1.0$$

$$a_{12} = \sum_{k=1}^{m} x_k$$

$$a_{13} = \sum_{k=1}^{m} x_k^2$$

$$a_{22} = \sum_{k=1}^{m} x_k^2$$

$$a_{23} = \sum_{k=1}^{m} x_k^3$$

$$a_{33} = \sum_{k=1}^{m} x_k^4$$

Implementação (atividade prática)

2. Sejam os mesmos pontos anteriores:

x	0.10	0.20	0.50	0.65	0.70	0.80	0.90	1.10	1.23	1.35	1.57	1.70	1.75	1.80	1.94
f(x)	0.19	0.36	0.75	0.87	0.91	0.96	0.99	0.99	0.94	0.87	0.67	0.51	0.43	0.36	0.11

Considere agora as seguintes funções:

$$g_1(x) = 1, g_2(x) = sen\left(\frac{\pi x}{2}\right)$$

Qual ajuste foi o melhor? o anterior ou utilizando as funções acima ?

Mínimos quadrados - Caso contínuo

Mínimos quadrados-Caso contínuo

Agora vamos considerar o caso em que dada uma função f(x) contínua em [a,b] e escolhidas as funções :

$$g_1(x), g_2(x), \ldots, g_n(x)$$

Todas contínuas em [a,b], queremos determinar os coeficientes α_1, α_2, α_3, ..., α_n de modo que

$$\varphi(x) = \alpha_1 g_1(x) + \alpha_2 g_2(x) + \cdots + \alpha_n g_n(x)$$

se aproxime ao máximo de f(x) no intervalo [a,b]

Mínimos quadrados – Caso contínuo

De maneira análoga ao visto anteriormente podemos achar os coeficientes minimizando a função F,

$$\begin{cases} [\int_a^b g_1(x)g_1(x)dx]\alpha_1 + \cdots + [\int_a^b g_n(x)g_1(x)dx]\alpha_n = \int_a^b f(x)g_1(x)dx \\ [\int_a^b g_1(x)g_2(x)dx]\alpha_1 + \cdots + [\int_a^b g_n(x)g_2(x)dx]\alpha_n = \int_a^b f(x)g_2(x)dx \\ \cdots \\ [\int_a^b g_n(x)g_1(x)dx]\alpha_1 + \cdots + [\int_a^b g_n(x)g_n(x)dx]\alpha_n = \int_a^b f(x)g_n(x)dx \end{cases}$$

Sistema linear com n equações e n incógnitas: α_1, α_2, α_3, ..., α_n

Mínimos quadrados – Caso contínuo

De maneira bastante similar, temos agora o sistema linear Aα = b, onde

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

$$b = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{pmatrix}$$

$$\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_n \end{pmatrix}$$

$$a_{ij} = \int_a^b g_j(x) g_i(x) dx$$

$$b_i = \int_a^b f(x) g_i(x) dx$$

Exemplo:

Seja a função $f(x) = 4x^3$. Vamos aproximar f(x) por um polinômio de primeiro grau, no intervalo [0,1]:

$$g_1(x) = 1, g_2(x) = x$$

$$a_{11} = \int_0^1 dx = 1$$

$$a_{12} = \int_0^1 x dx = 0.5 = a_{21}$$

$$a_{22} = \int_0^1 x^2 dx = 1/3$$

$$b_1 = \int_0^1 4x^3 dx = 1$$

$$b_2 = \int_0^1 4x^4 dx = 4/5$$

$$\Rightarrow \begin{cases} \alpha_1 + 0.5\alpha_2 = 1 \\ 0.5\alpha_1 + (1/3)\alpha_2 = 4/5 \end{cases}$$

Solução:
$\alpha 1 = -4/5$
$\alpha 2 = 18/5$

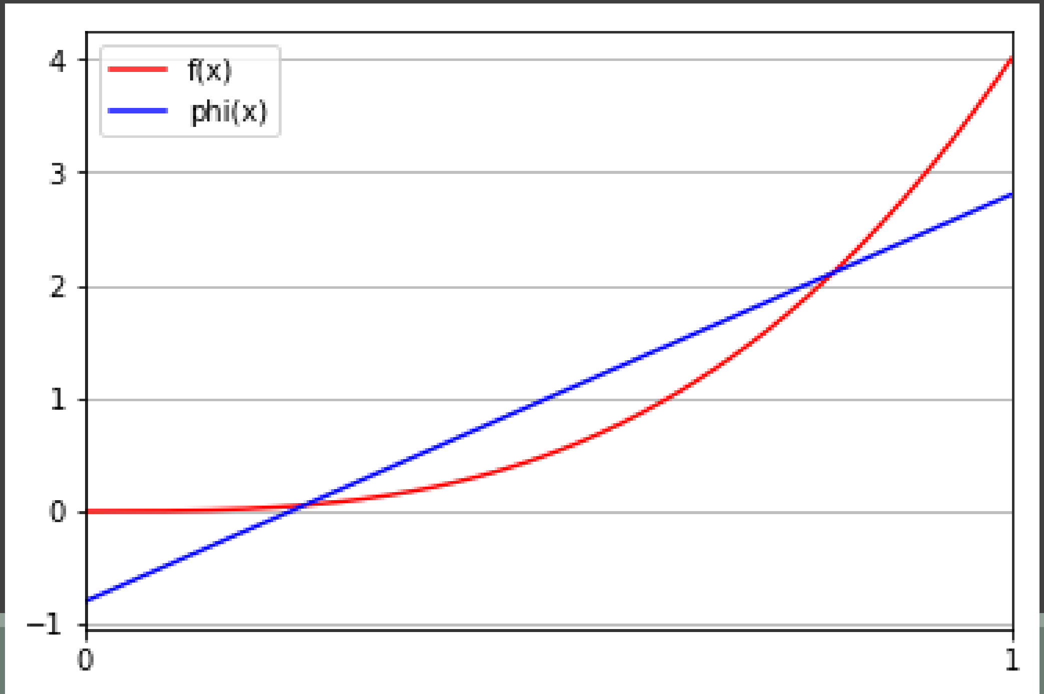
Exemplo:

Seja a função $f(x) = 4x^3$. Vamos aproximar f(x) por um polinômio de primeiro grau, no intervalo [0,1]:

$$g_1(x) = 1, g_2(x) = x$$

Solução:
$\alpha_1 = -4/5$
$\alpha_2 = 18/5$

$$\varphi(x) = (18/5)x - 4/5$$



Mínimos quadrados - Caso não linear

Mínimos quadrados-Caso não linear

Em alguns casos, a família de funções escolhidas pode não ser linear nos parâmetros, por exemplo:

$$\varphi(x) = \alpha_1 e^{\alpha_2 x}$$

Para aplicar o método dos mínimos quadrados temos que linearizar a função acima:

$$y = \alpha_1 e^{\alpha_2 x} \Rightarrow z = ln(y) = ln(\alpha_1) + \alpha_2 x = \beta_1 + \beta_2 x$$

Problema então consiste em ajustar os dados de z por uma reta, considerando

$$g_1(x) = 1, g_2(x) = x \Rightarrow$$

Obtidos os parametros usaremos os mesmos para calcular os parâmetros originais.

Mínimos quadrados-Caso não linear

Temos então que:

$$\varphi(x) = \alpha_1 e^{\alpha_2 x} \Rightarrow$$

$$y = \alpha_1 e^{\alpha_2 x}$$

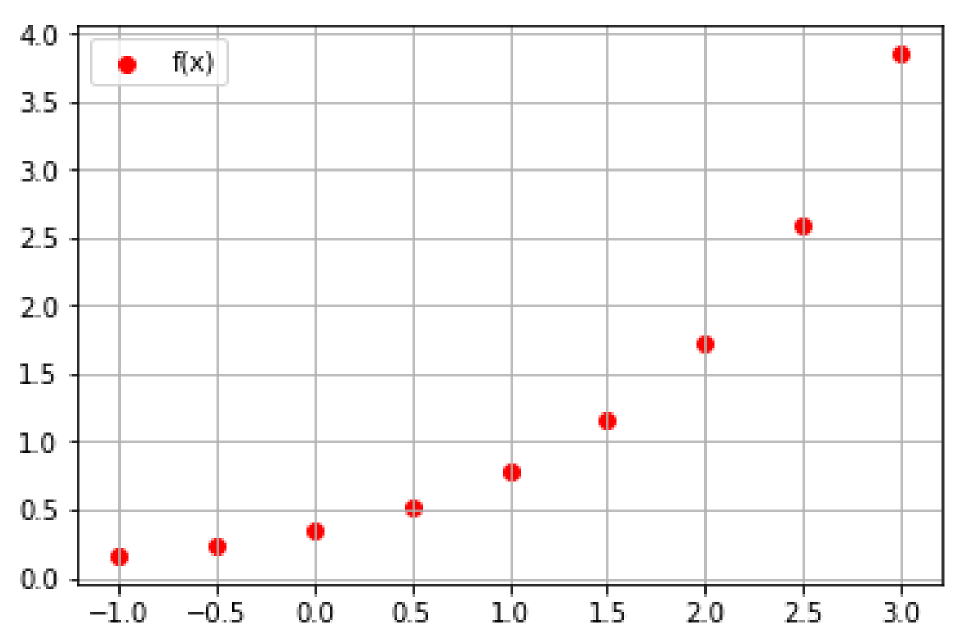
$$z = ln(y) = ln(\alpha_1) + \alpha_2 x = \beta_1 + \beta_2 x$$

Vale ressaltar que os parâmetros obtidos não são ótimos dentro do critério dos quadrados mínimos, pois estamos ajustando o problema linearizado por quadrados mínimos , não o problema original.

Mínimos quadrados-Caso não linear (atividade prática)

Exemplo:

x	-1.0	-0.5	0.0	0.5	1.0	1.5	2.0	2.5	3.0
f(x)	0.157	0.234	0.350	0.522	0.778	1.162	1.733	2.586	3.858



$$z = ln(y) = ln(\alpha_1) + \alpha_2 x = \beta_1 + \beta_2 x$$

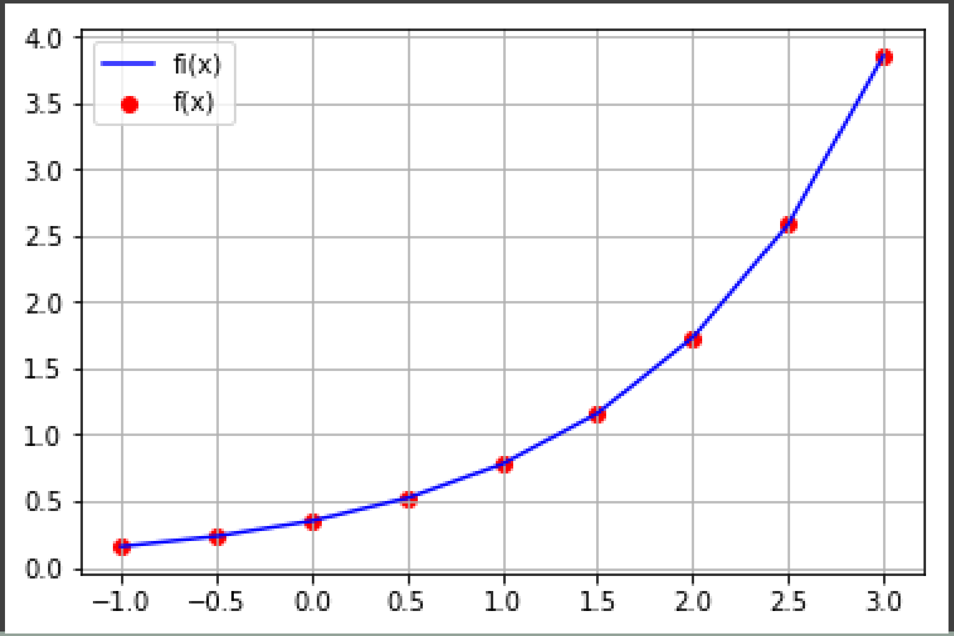
$$g_1(x) = 1, g_2(x) = x$$

Quais os valores de β_1 e β_2 ?

Mínimos quadrados-Caso não linear (atividade prática)

Exemplo:

x	-1.0	-0.5	0.0	0.5	1.0	1.5	2.0	2.5	3.0
f(x)	0.157	0.234	0.350	0.522	0.778	1.162	1.733	2.586	3.858



Dos valores de β_1 e β_2, obtemos α_1 e α_2.
Logo, obtemos a função de ajuste