

Documentació de la llibreria `funcions_labs.py`

Facultat de Física - Universitat de Barcelona

9 de juliol de 2025

1 Introducció

Aquesta llibreria conté funcions de suport per a pràctiques de laboratori en assignatures de física. Inclou eines per a la lectura de dades, ajustos de corbes, representació gràfica i models matemàtics habituals.

2 Funcions principals

2.1 `llegeix_dades`

Llegeix un fitxer de dades delimitat per tabuladors, comes o altres separadors.

```
llegeix_dades(path, separador='\t', comentari='#', skip=0)
```

Retorna: Array o tupla d'arrays amb les columnes de dades.

2.2 `format_resultat`

Formata un valor amb incertesa en notació científica en format LaTeX.

```
format_resultat(valor, incertesa)
```

2.3 `ajusta_dades`

Fa un ajust als punts de dades proporcionats segons un model donat. Està pensada per ser cridada des de la funció `mostra_dades`, encara que també es pot usar de manera independent.

```
ajusta_dades(x, y, funcio_model, incert_y=None, x_min=None, x_max=None,
             exclude=None, v_i=None)
```

Retorna: Diccionari amb paràmetres ajustats, errors, χ^2 , RMSE, R^2 i màscara de punts usats.

2.4 `mostra_dades`

Representa gràficament les dades (amb error) i opcionalment hi superposa un ajust.

```
mostra_dades(x, y, funcio_model=None, incert_x=None, incert_y=None,
             log_x=False, log_y=False, titol=None,
             xlabel='x', ylabel='y', forma='o',
             x_min=None, x_max=None, exclude=None,
             mida_figura=(4.5, 4), label_dades="Dades",
             valors_inicials=None, ponderacio=True)
```

3 Models matemàtics disponibles

Tots aquests models accepten un argument opcional `option="Text"` que retorna una cadena LaTeX del model:

- `model_lineal(x, a, b)`: $y = a_0x + a_1$
- `model_exponencial(x, a, b)`: $y = a_0e^{a_1x}$
- `model_potencial(x, a, b)`: $y = a_0x^{a_1}$
- `model_polinomi(x, *coeficients)`: $y = a_0 + a_1x + a_2x^2 + \dots$
- `model_gaussià(x, A, x0, sigma)`: $y = a_0e^{-\frac{(x-a_1)^2}{2a_2^2}}$
- `model_gaussià_base(x, A, x0, sigma, a, b)`: Gaussiana + fons lineal
- `model_lorentziana(x, A, x0, gamma)`: Lorentziana
- `model_sinus(x, a0, a1, a2)`: $y = a_0 \sin(a_1x + a_2)$

4 Exemples d'ús

Exemple: ajust lineal

```
from funcions_labs import mostra_dades, model_lineal

mostra_dades(x, y, funcio_model=model_lineal,
              incert_y=dy, xlabel=r'$t_{\square}(s)$', ylabel=r'$d_{\square}(m)$')
```

Exemple: ajust exponencial amb exclusió de punts

```
mostra_dades(x, y, funcio_model=model_exponencial,
              incert_y=dy, exclude=[0, -1])
```

5 Descripció detallada dels paràmetres de `mostra_dades`

La funció `mostra_dades` permet representar gràficament un conjunt de dades experimentals i, si es desitja, superposar-hi un ajust.

```
mostra_dades(x, y, funcio_model=None, incert_x=None, incert_y=None,
             log_x=False, log_y=False, titol=None,
             xlabel='x', ylabel='y', forma='o',
             x_min=None, x_max=None, exclude=None,
             mida_figura=(4.5, 4), label_dades="Dades",
             valors_inicials=None, ponderacio=True)
```

`x, y` Arrays de dades experimentals.

- Exemple: `x = np.array([1, 2, 3]), y = np.array([2.1, 3.9, 6.2])`

`funcio_model` Funció matemàtica d'ajust. Si no s'indica, només es mostren les dades.

- Exemple: `funcio_model=model_lineal`

`incert_x`, `incert_y` Errors associats a les mesures. Si són `None`, no es representen barres d'error.

- Exemple: `incert_y = np.array([0.1, 0.1, 0.2])`

`log_x`, `log_y` Booleans que activen escales logarítmiques.

- Exemple: `log_y=True` per a gràfiques semilogarítmiques.

`titol` Títol de la gràfica.

- Exemple: `titol="Ajust lineal"`

`xlabel`, `ylabel` Etiquetes dels eixos.

- Exemple: `xlabel="Temps (s)"` i `ylabel="Posicio (m)"`

`forma` Format dels punts ('o' cercle, 's' quadrat, etc.)

- Exemple: `forma='s'` per quadrats

`x_min`, `x_max` Valors mínim i màxim de l'eix X a tenir en compte per a l'ajust.

- Exemple: `x_min=1.5`, `x_max=4.0`

`exclude` Índexs o valors de x a excloure de l'ajust.

- Exemple: `exclude=[0]` per ometre el primer punt.

`mida_figura` Tuple que defineix la mida de la figura en polzades.

- Exemple: `mida_figura=(6, 5)`

`label_dades` Etiqueta per a la llegenda dels punts experimentals.

- Exemple: `label_dades="Mesura directa"`

`valors_inicials` Llista amb els valors inicials dels paràmetres d'ajust (passat a `curve_fit`).

- Exemple: `valors_inicials=[1, 0.5]` per iniciar amb una pendent 1 i ordenada 0.5

`ponderacio` Valor Booleà que defineix si l'ajust s'ha de fer pesant cada punt amb l'invers del quadrat de la seva incertesa o no. En darrer aquest cas es fa regressió simple, que és equivalent a no posar la incertesa de les y's.

- Exemple: `valors_inicials=False` per fer regressió simple

Exemple complet:

```
mostra_dades(x, y, funcio_model=model_lineal, incert_y=dy,
             xlabel='t(s)', ylabel='x(m)', titol='Moviment_rectilini',
             x_min=1, x_max=5, exclude=[2], forma='s',
             mida_figura=(6, 4), label_dades="Dades_filtrades",
             ponderacio=False)
```

Aquest codi mostra només els punts entre 1 i 5, excloent el tercer, i hi ajusta una recta.

6 Exemples extrets del notebook

Aquest apartat mostra exemples pràctics d'ús de la llibreria `funcions_labs.py`, extrets del notebook `Exemples_f_labs.ipynb`.

1. Expressar magnituds amb incertesa

```
from funcions_labs import format_resultat, dM

dM(format_resultat(234.234,1.275))
dM(format_resultat(434.3445e11,137e8))
```

Resultats mostrats en el notebook:

$$234.2 \pm 1.3 \quad (43434 \pm 14) \times 10^9$$

2. Ajust lineal de dades experimentals

```
x = np.array([1, 2, 3, 4, 5])
y = np.array([1.1, 2.0, 2.9, 4.2, 5.1])
dy = np.array([0.2, 0.2, 0.2, 0.2, 0.2])

mostra_dades(x, y, funcio_model=model_lineal, incert_y=dy,
              xlabel='x', ylabel='y', titol='Ajust lineal')
```

$$y = a_0x + a_1$$

$$a_0 = 1.08 \pm 0.06, a_1 = 1.0 \pm 0.2, \chi^2_\nu = 0.30, \text{RMSE} = 0.0849, R^2 = 0.997$$

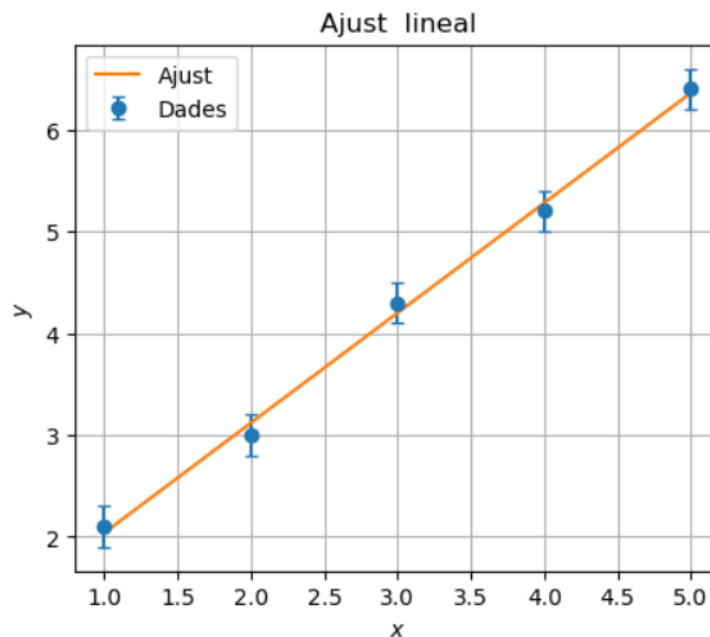


Figura 1: Gràfica de dades amb error i ajust lineal. Inclou la sortida simbòlica amb els paràmetres ajustats.

3. Ajust amb exclusió de punts i rang limitat

```
mostra_dades(x, y, incert_y=dy, funcio_model=model_lineal,
              titol='Ajust lineal eliminant algun punt i seleccionant el  

              ↳ rang',
              xlabel='$Temps$(s)', ylabel='$Distància$(m)',
              x_max=7, exclude=[9])
```

$$y = a_0x + a_1$$

$$a_0 = 3.6 \pm 0.5, a_1 = 9 \pm 2, \chi^2_\nu = 0.11, \text{RMSE} = 1.24, R^2 = 0.98$$

Ajust lineal eliminant algun punt i seleccionant el rang

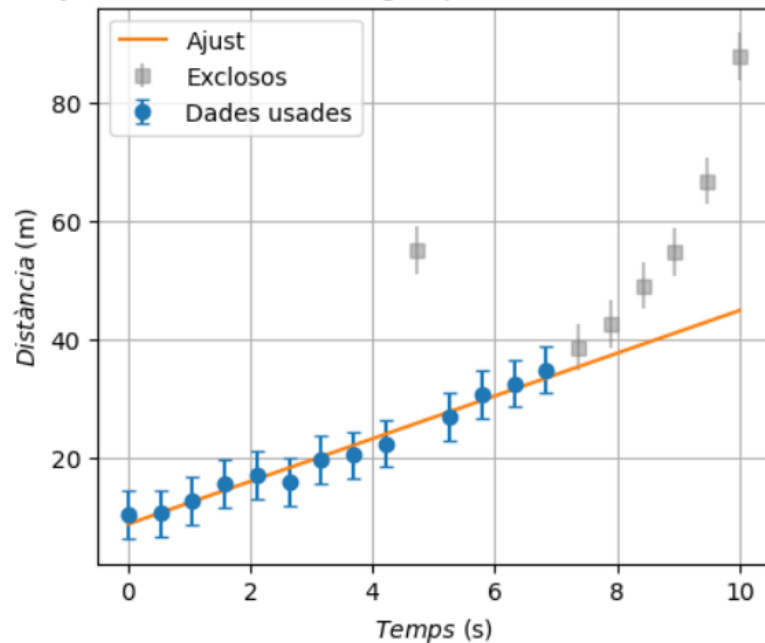


Figura 2: Ajust lineal fins a $x = 7$, amb exclusió del nové punt. Els punts exclosos es mostren en gris.

4. Ajust exponencial

```
x = np . linspace (0 , 2 , 6)
y = np . array ([1.05 , 1.55 , 2.75 , 4.1 , 7.2 , 10.9])
dy = 0.1*y
mostra_dades (x , y , funcio_model = model_exponencial ,
               incert_y = dy, titol = 'Ajust_exponencial',
               xlabel = r'$t$ (s)' , ylabel = r'$N$', log_y=True)
```

5. Ajust sinusoidal

```
x = np . linspace (0 , 2* np . pi , 50)
y = 3 * np.sin(2.5 * x + 0.5) + np.random.normal (0, 0.3, len(x))
mostra_dades (x , y , funcio_model = model_sinus ,
               titol = 'Ajust_sinusoidal', valors_inicials =[2 ,3, 0])
```

$$y = a_0 e^{a_1 x}$$

$$a_0 = 1.02 \pm 0.07, \quad a_1 = 1.19 \pm 0.06, \quad \chi^2_\nu = 0.24, \quad \text{RMSE} = 0.176, \quad R^2 = 0.997$$

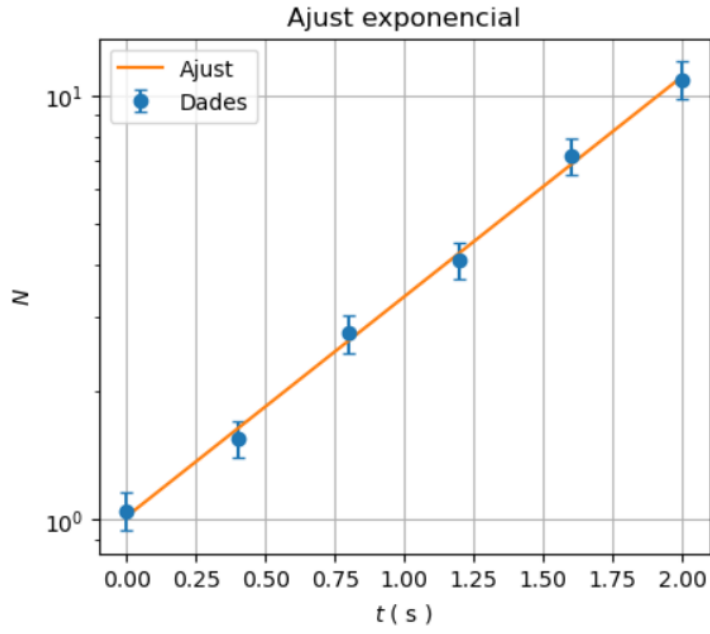


Figura 3: Ajust a una funció exponencial $y = a_0 e^{a_1 x}$ amb incerteses emprant escala logarítmica en l'eix vertical.

$$y = a_0 \sin(a_1 x + a_2)$$

$$a_0 = 3.1 \pm 0.2, \quad a_1 = 2.49 \pm 0.03, \quad a_2 = 0.47 \pm 0.13, \quad \chi^2_\nu = 0.09, \quad \text{RMSE} = 0.297, \quad R^2 = 0.98$$

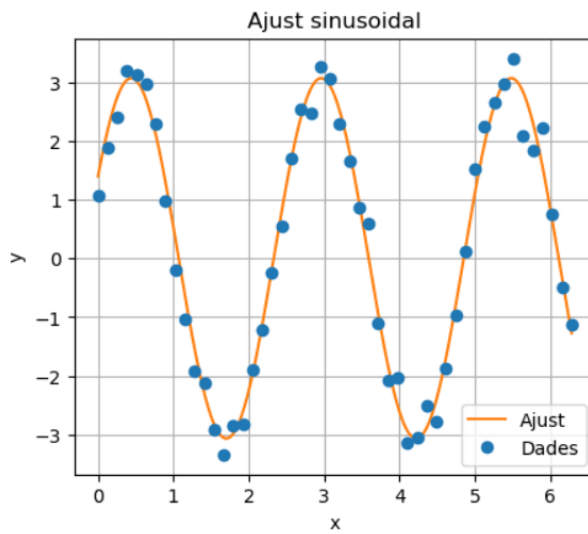


Figura 4: Ajust sinusoidal proporcionant una estimació inicial dels paràmetres a ajustar.