



Università degli Studi di Pisa
Laboratorio di Fisica 1

Corso di Laurea Triennale in Fisica

Relazione di laboratorio

Equazione della Catenaria

25/09/2021

Gruppo di lavoro A-1-3-2:

Marco Ricci m.ricci66@studenti.unipi.it
Matricola 641084

Martino Barbieri m.barbieri20@studenti.unipi.it
Matricola 640522

Docente:

Prof. Luca Baldini

Anno Accademico 2021-2022

Indice

1	Introduzione	2
1.1	Obiettivo	2
1.2	Materiale utilizzato	2
2	Raccolta dati	3
2.1	Catena	3
2.2	Fotografia	3
3	Analisi dati	4
3.1	Metodi di analisi	4
3.2	Incertezze	4
3.3	Risultati dell'analisi	4
4	Conclusioni	9
A	Appendice: Derivazione dell'equazione della Catenaria	10
B	Appendice: Codici Python utilizzati	11

1 Introduzione

La catenaria è il nome di una funzione matematica. Questa curva piana iperbolica è anche la forma assunta da una fune **inestensibile**, **omogenea** e **flessibile**, vincolata ai due estremi e lasciata pendere sotto la sola azione del suo stesso peso.

$$y = C(x, a, x_0, y_0) = y_0 + a \cosh\left(\frac{x - x_0}{a}\right) = y_0 + \frac{a}{2} \left[\exp\left(\frac{x - x_0}{a}\right) + \exp\left(-\frac{x - x_0}{a}\right) \right] \quad (1)$$

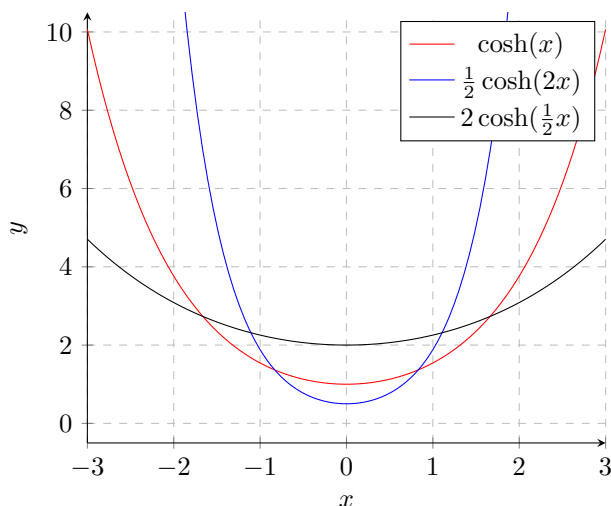


Figura 1: Alcuni grafici di catenarie.

Lasciamo nell'Appendice A la derivazione analitica della catenaria.

1.1 Obiettivo

Il fine di questa esperienza è di verificare sperimentalmente se il profilo assunto da una catena reale sotto azione della sola forza di gravità possa essere compatibile con la previsione teorica ottenuta tramite il calcolo differenziale. Quindi vogliamo comprendere in quali condizioni la coincidenza tra modello e realtà è maggiore ed infine ipotizzare le ragioni per cui vi sono delle discrepanze.

1.2 Materiale utilizzato

1. Catena a maglie di diametro 3.0 mm;
2. macchina fotografica modello *Canon Rp*, con obiettivo *RF 35 mm f1.8 MACRO IS STM*;
3. filo a piombo;
4. righello come unità di misura.

2 Raccolta dati

In questa sezione descriveremo brevemente gli accorgimenti che abbiamo utilizzato per ridurre potenziali errori sistematici e raccogliere misure in una situazione il più possibile prossima a quella ideale, dove abbiamo ricavato analiticamente la forma che ci aspettiamo la corda assuma.

Sul sito <https://unilab.mbarbieri.it> rendiamo disponibile tutto il materiale raccolto ed elaborato. Lasciamo nell'Appendice B il codice Python utilizzato per l'analisi delle immagini e per l'estrapolazione dei dati.

2.1 Catena

Cerchiamo di scegliere una fune che sia il più possibile prossima alla definizione di fune omogenea, flessibile e non estensibile. Per tale ragione optiamo per una catena con maglie molto piccole rispetto alla lunghezza della catena appesa, (3.0 ± 0.5) mm su (2 ± 1) m (come controllo ripetiamo l'esperienza con uno spago, un filo da cucito e una catena per biciclette, ottenendo risultati analoghi).

Verifichiamo inoltre che l'allungamento sia trascurabile; infatti, usando strumenti aventi sensibilità 1 mm, non siamo in grado di misurare una variazione nella lunghezza della catena quando ad essa è appeso un corpo di un peso (100 ± 1) gp, superiore a quello della catena di un ordine di grandezza.

Verifichiamo inoltre che la catena non sia appoggiata al piano di riferimento (nel nostro caso una lavagna) perché ciò porterebbe ad una variazione dalle ipotesi iniziali: introdurrebbe cioè forze di attrito statico tra i due corpi.

2.2 Fotografia

Per scattare al meglio la fotografia, il nostro obiettivo è quello di fare in modo che il piano del sensore della macchina fotografica sia parallelo al piano immaginario su cui giace la catena. Inoltre, vogliamo che il sensore sia centrato il più possibile e che quindi la retta normale al piano del sensore e passante per il suo centro intersechi l'asse di simmetria della catenaria.

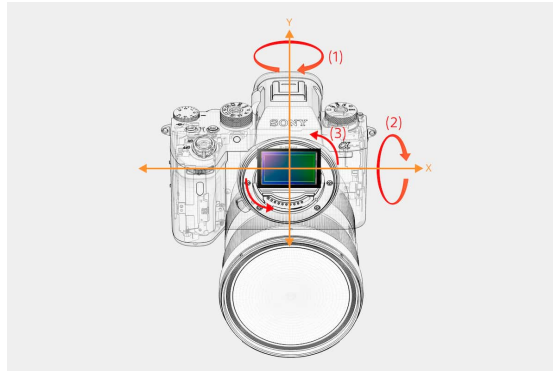


Figura 2: Rotazioni possibili del sensore

Rotazioni del sensore intorno all'asse x (Fig. 2) determinerebbero deformazioni dell'immagine finale. Fissando due fili a piombo, di egual misura, nei medesimi punti di ancoraggio degli estremi della catena, essi risultano non paralleli se fotografati senza correzioni di tale rotazione.

Rotazioni del sensore intorno all'asse y (Fig. 2) invece porterebbero a deformazioni ottiche di altra natura. I fili a piombo, nell'immagine, dovrebbero risultare di lunghezze differenti,

sottolineiamo "dovrebbero" perché tali modificazioni nelle lunghezze sono di tale entità da non poter essere misurate nell'immagine.

Rotazioni del sensore intorno all'asse z (Fig. 2) porterebbero ad una semplice rotazione dell'immagine senza subire deformazioni. Ciò non significa che tale rotazione sia innocua, in quanto la catena lo script Python che trova il *best fit* presuppone che l'asse verticale dell'immagine sia parallela al campo di gravità locale.

La prevenzione di queste rotazioni potrebbe essere fatta con una livella a bolla (per evitare rotazioni intorno a z) e con i due fili a piombo (per evitare rotazioni intorno a x e y), dall'altra parte siamo provvisti di una strumentazione di maggior precisione e che quindi adoperiamo. La macchina fotografica *Canon Rp* è dotata di un sensore interno capace di segnalare a schermo eventuali rotazioni intorno all'asse x ed z (con una risoluzione migliore di un grado).

Verifichiamo che il sensore della macchina fotografica sia ben calibrato, comparando la lettura del sensore con quella ottenuta tramite una app specifica per telefono. Questo confronto non dimostra definitivamente che non ci siano problemi nell'inclinazione della macchina fotografica; infatti non abbiamo ancora la certezza che il sensore di rotazione ed il sensore fotografico siano allineati all'interno del corpo della macchina. Così inquadrano un'immagine di riferimento e verifichiamo che quello che ci viene mostrato sullo schermo della macchina fotografica sia esattamente la fotografia finale. In questo modo confermiamo che le indicazioni sull'inclinazioni della macchina siano attendibili per il sensore.

3 Analisi dati

3.1 Metodi di analisi

Per il rilevamento delle misure è stato utilizzato un programma scritto da noi in Python. Inizialmente un file di immagine viene tradotto in una matrice di triplette di interi a 8 bit, i quali rappresentano le componenti RGB che definiscono il colore del pixel. Questa matrice viene analizzata e i pixel il cui colore è prossimo a quello della catena vengono contrassegnati. Viene poi effettuata la regressione sulle coordinate dei pixel contrassegnati.

3.2 Incertezze

Ciascun pixel contrassegnato è parte della catena. Siccome, per la modalità operativa con cui raccogliamo i dati, l'incertezza sulla coordinata orizzontale della foto è la stessa che abbiamo sulla coordinata verticale, le ipotesi richieste dal metodo dei minimi quadrati ordinari non sono soddisfatte. Eseguiamo dunque la regressione per minimi quadrati totali.

Per stimare l'incertezza, possiamo approssimare la catena nell'intorno di ogni punto come un segmento di spessore fissato. La normale alla curva in un punto interseca un numero N di pixel che corrisponde alla larghezza, in pixel, della catena. Il programma non distingue i pixel sul bordo tra la catena e lo sfondo e i pixel allineati col centro della catena stessa. Dunque, gli N pixel contrassegnati hanno eguale probabilità di contenere il misurando. La probabilità omogenea prevede che l'incertezza statistica sia dunque $\sigma = \frac{N}{\sqrt{12}}$ px.

3.3 Risultati dell'analisi

Le possibili ragioni per cui l'immagine della catena ottenuta non è una perfetta catenaria sono di 2 diverse nature: intrinseche o estrinseche. Le prime riconducibili al fatto che la catena non è esattamente una fune omogenea, flessibile e non estensibile e le seconde legate alla modalità

seguita per prendere le misure, che comprendono incertezze dovute alla strumentazione utilizzata e a errori casuali di natura umana.

Scattando foto di perlustrazione da diverse posizioni notiamo che nel grafico dei residui ci sono due diversi andamenti che paiono indipendenti. Uno simmetrico e uno antisimmetrico. Dunque comparando il grafico dei residui di fotografie in cui cerchiamo di evitare rotazioni intorno all'asse y ed altre in cui invece volutamente esasperiamo questa rotazione notiamo una forte correlazione tra l'errore antisimmetrico e l'angolo di cui è deviata la fotocamera; possiamo quindi attribuire questo tipo di errore proprio alla rotazione attorno a y . Per cercare di abbattere nelle misure l'errore antisimmetrico modifichiamo il nostro modello aggiungendo ai parametri del *fit* una grandezza che corregge la rotazione (per piccoli angoli).

Scattando fotografie da diverse distanze tentiamo di individuare, studiando l'intervallo di confidenza delle misure, quali siano gli effetti generati dalle distorsioni ottiche. Le incertezze dovute a cause ottiche derivano dal fatto che l'approssimazione lineare parassiale stima solo al primo ordine in funzione dell'angolo tra il raggio ottico di interesse e l'asse parassiale il comportamento dei raggi ottici. Ci aspettiamo dunque che l'errore decresca al crescere della distanza tra la macchina fotografica e la catena.

Tabella 1: Intervallo di confidenza $1 - \beta$ della misura fotografando la catenaria con un angolo θ massimo tra un raggio che arriva dalla catenaria e l'asse parassiale¹.

θ rad	$1 - \beta$
$0.427 \pm 5\%$	41.2%
$0.160 \pm 5\%$	51.3%
$0.0777 \pm 5\%$	47.0%

Nella prima misura, l'altezza della catenaria, un basso intervallo di confidenza e un andamento riconoscibile dei residui indicano che gli errori grandangolari sono rilevanti.

Pure nelle altre due misure si riconosce lo stesso andamento dei residui, con circa la stessa ampiezza, nonostante θ diminuisca di parecchio e $1 - \beta$ aumenti in modo considerevole. Questo suggerisce in tali misure non sia influente l'errore dovuto ad aberrazioni ottiche.

Nell'ultima misura le incertezze nelle singole misure aumentano di molto a causa della vicinanza tra la sensibilità dello strumento (1 px) e l'incertezza statistica σ .

¹Abbiamo stimato quest'angolo come l'arcotangente del rapporto tra metà della larghezza della catenaria e la distanza tra essa e la macchina fotografica

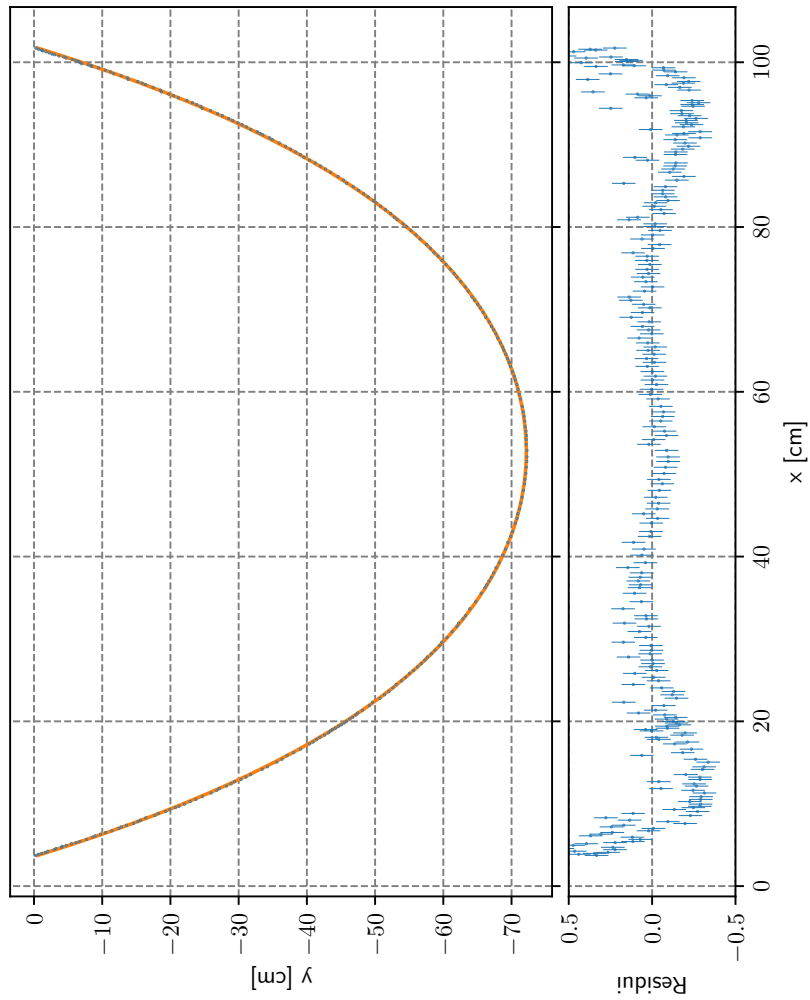


Figura 3: Catenaria da vicino

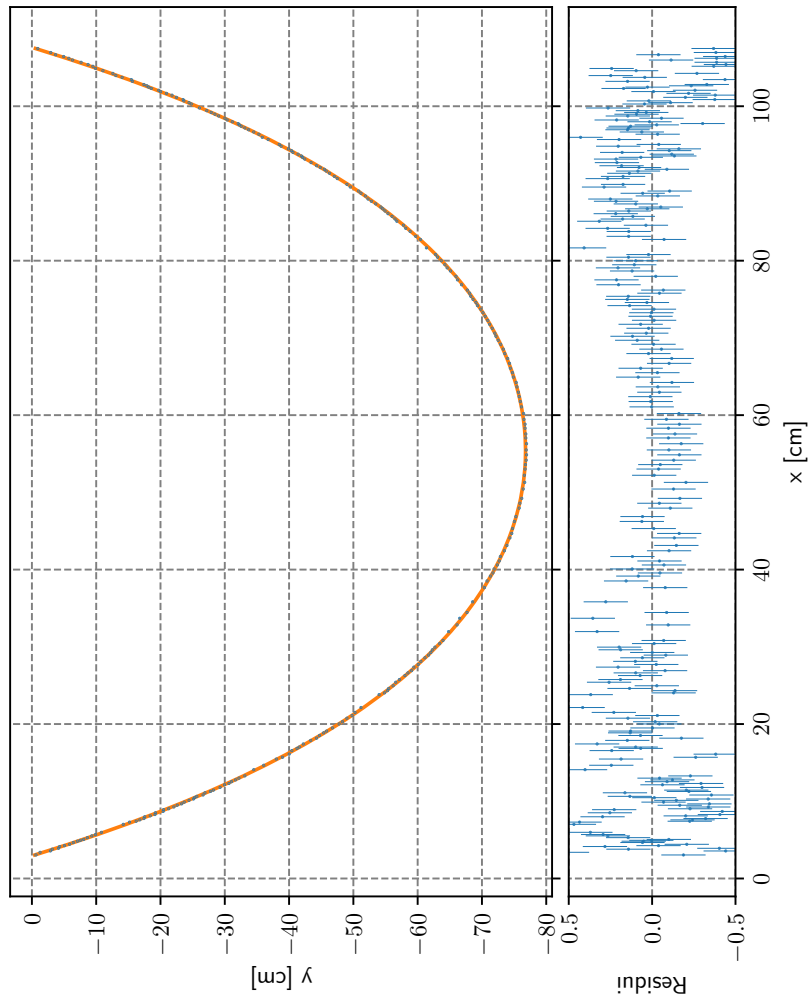


Figura 4: Catenaria da media distanza

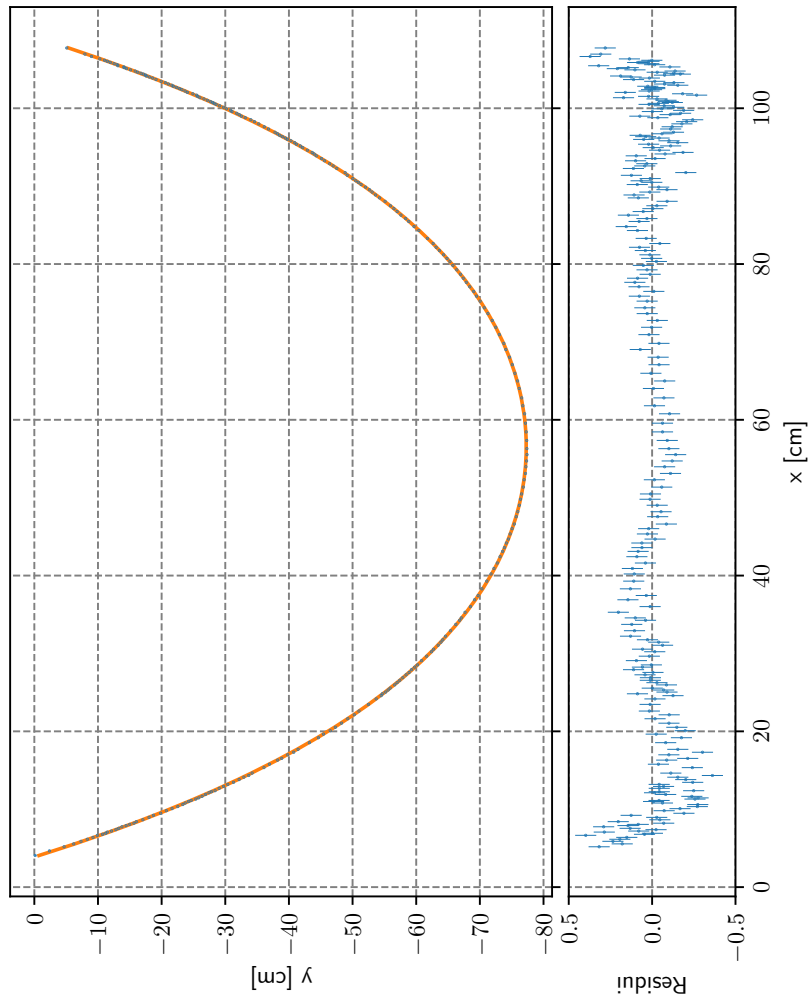


Figura 5: Catenaria da maggiore distanza

4 Conclusioni

Abbiamo verificato che la catenaria segue approssimativamente la forma prevista analiticamente.

Le differenze tra la forma aspettata e la forma reale seguita dalla catena sono nell'ordine di una parte su mille rispetto alle dimensioni caratteristiche della forma della catena.

Si manifesta un errore sistematico che viene reso graficamente da una curva dei residui simmetrica rispetto all'asse verticale. Le possibili cause di questo errore sono le differenze tra la struttura della catena e un filo ideale di spessore nullo, perfettamente flessibile e continuo. Esse non sono dovute ad alterazioni ottiche.

A Appendice: Derivazione dell'equazione della Catenaria

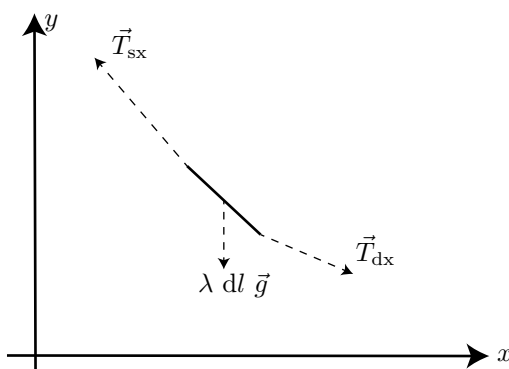
Si consideri una fune continua, perfettamente flessibile e omogenea, di densità di massa lineare λ , fissata in due punti P_1, P_2 . Si vuole ricavare la forma della corda tra P_1, P_2 .

Per il principio di irrigidimento, su ogni parte infinitesima dl della corda, le risultanti delle forze e dei momenti devono essere nulle. Le forze agenti sul pezzo di corda sono:

- le tensioni ai suoi estremi;
- la forza di gravità $\lambda \cdot dl \cdot \vec{g}$.

I momenti sono nulli per l'ipotesi di perfetta flessibilità.

La forma della corda giace su un piano (si può dimostrare agevolmente con il Principio dei Lavori Virtuali) parallelo a \vec{g} e passante per P_1, P_2 . Sia $\hat{y} = -\frac{\vec{g}}{\|\vec{g}\|}$ e \hat{x} un versore giacente nel piano e perpendicolare a \hat{y} .



La forza peso non esercita componente orizzontale, per cui le componenti orizzontali delle tensioni devono eguagliarsi; ciò vale in ogni parte della corda, quindi la componente orizzontale della tensione è costante e la chiamiamo T_x .

La differenza delle componenti orizzontali deve essere eguale e opposta alla forza peso.

$$dT_y = \frac{dT_y}{dx} dx = \lambda g dl = \lambda g \sqrt{dx^2 + dy^2} = \lambda g \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx$$

La tensione della corda è localmente parallela alla corda in ogni punto. Quindi

$$\frac{dy}{dx} = \frac{T_y}{T_x} \Rightarrow dT_y = T_x \frac{d^2 y}{dx^2} dx$$

$$T_x \frac{d^2 y}{dx^2} dx = \lambda g \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx$$

Questa ODE si risolve col metodo della separazione delle variabili. Le costanti di integrazione sono ricavabili dalle condizioni al bordo oppure dalla lunghezza della catena.

$$\int \frac{d\frac{dy}{dx}}{\sqrt{1 + \left(\frac{dy}{dx}\right)^2}} = \int \frac{\lambda g}{T_x} dx$$

$$\text{settsinh} \frac{dy}{dx} = \frac{\lambda g}{T_x} (x - x_0)$$

$$y = \int \frac{dy}{dx} dx = \int \sinh \frac{\lambda g}{T_x} (x - x_0) dx$$

$$y = y_0 + \frac{T_x}{\lambda g} \cosh \frac{\lambda g}{T_x} (x - x_0)$$

□

B Appendice: Codici Python utilizzati

Alcuni parametri, come *lim*, *scala* e *col* vanno modificati per ogni immagine per ottenere risultati ottimali.

```

1 import numpy as np
2 import matplotlib
3 from matplotlib import pyplot as plt
4 from scipy import odr
5 import math
6 from PIL import Image
7 from IPython.display import display
8 import sys
9
10 # PROGRESS BAR
11 def printProgressBar (iteration, total, prefix = '', suffix = '', decimals = 0,
12     length = 100, fill = '#', printEnd = "\r"):
13     percent = ("{0:." + str(decimals) + "f}").format(100 * (iteration / float(
14         total)))
15     filledLength = int(length * iteration // total)
16     bar = fill * filledLength + '-' * (length - filledLength)
17     print(f'\r{prefix} |{bar}| {percent}% {suffix}', end = printEnd)
18     # Print New Line on Complete
19     if iteration == total:
20         print()
21
22 # DEFINIZIONI CATENARIA PER FIT
23 def catenaryTheta(B, x):
24     a = B[0]
25     c = B[1]
26     x0 = B[2]
27     theta = B[3]
28     return (c + a * np.cosh((x - x0) / a)) * (1 + x * theta / 180 * 3.1415926)
29
30 if len(sys.argv) < 2:
31     print('Inserire il nome del file del file di input: ')
32     img = input()
33 else:
34     # NOME DEL FILE DA ESAMINARE
35     img = sys.argv[1]
36
37 img = Image.open(img)

```

```

36
37
38 # OTTIENI DALL'OGGETTO img LA MATRICE CHE CONTIENE I PIXEL
39 pin = img.load()
40
41 # CREA IMMAGINE DI OUTPUT
42 imgout = Image.new('RGB', (img.size[0],img.size[1]), "white")
43 pout = imgout.load()
44
45 # UPPER BOUND PER SOMMA COLORI CATENA IN RGB
46 lim = 30
47 # COLORE MEDIO DELLA CATENA
48 col = [140,100,60]
49
50 x = np.array([])
51 y = np.array([])
52
53 outStr = ""
54
55 with open('out.txt','w') as f:
56     for i in range(0,img.size[0], 1):
57         # SCANSIONA LA MATRICE DEL FILE DI INPUT PIXEL PER PIXEL
58         # SALVA I PIXEL IN out.txt PER BACKUP E IN x,y
59         printProgressBar(i, img.size[0]-1, prefix = 'Calcolo:', suffix = '',
60             length = 40)
61         for j in range(0,img.size[1],1):
62             if (pin[i,j][0]-col[0])**2+(pin[i,j][1]-col[1])**2+(pin[i,j][2]-col
63                 [2])**2<lim**2:
64                 pout[i,j] = (0,0,0)
65                 # f.write("%d %d\n" % (i, j) )
66                 x=np.append(x,i)
67                 y=np.append(y,-j)
68             else:
69                 pout[i,j] = (255,255,255)
70
71 # CREA IMMAGINE DI ANTEPRIMA DI OUTPUT (UTILE PER DEBUG/SETTAGGIO DI lim)
72 imgout.save("out.png")
73
74 sigma = 12/(12.**0.5) # SIGMA SULLA MISURAZIONE = ALTEZZA BARRE DI ERRORE
75 bar_l = 0.3 # LARGHEZZA BARRE DI ERRORE
76 mar_size = 0.3 # LARGHEZZA PALLINO
77
78 # FIT CON QUADRATI TOTALI (ODR)
79 poprt, boh, pcov = odr.odr(catenaryTheta, [1500., -3500., 1500., 0.0], y, x)
80 a_hat, c_hat, x0_hat, theta_hat = poprt
81 sigma_a, sigma_c, sigma_x0, sigma_theta = np.sqrt(pcov.diagonal())
82
83 print('a %f\n\ttsa %f\nc %f\n\tsc %f\n\ttx0 %f\n\ttsx0 %f\n\tnth %f\n\tsth %f\n' % (a_hat,
84     sigma_a, c_hat, sigma_c, x0_hat, sigma_x0, theta_hat, sigma_theta))
85
86 # CALCOLO DELL'INTERVALLO DI CONFIDENZA
87 # RIDUZIONE DEL NUMERO DI PUNTI DA STAMPARE
88 ok = 0
89 tot = 0
90 N = 250 # NUMERO DI PUNTI
91 mod = round(len(x)/N)
92 xx = np.array([])
93 yy = np.array([])
94 for i in range(0, len(x)):
95     if (catenaryTheta([a_hat, c_hat, x0_hat, theta_hat], x[i]) - y[i])**2 <
96         sigma**2:
97         ok = ok+1

```

```

94     tot = tot+1
95     if i%mod == 0:
96         xx = np.append(xx, x[i])
97         yy = np.append(yy, y[i])
98
99     print('(1-beta)=%f\n' % (1.*ok/tot))
100
101     # GRAFICO
102     plt.rcParams["text.usetex"] = True
103
104     scala = 105.35/5145
105
106     fig = plt.figure('Fit e residui')
107     fig.add_axes((0.15, 0.32, 0.8, 0.65))
108     plt.errorbar(scala*xx, scala*yy, xerr=scala*sigma, yerr=scala*sigma, fmt='o',
109                 markersize=mar_size, linewidth=bar_l)
110     plt.plot(scala*xx, scala*catenaryTheta([a_hat, c_hat, x0_hat, theta_hat], xx))
111     plt.grid(which='both', ls='dashed', color='gray')
112     plt.ylabel(r'y [cm]')
113     # RESIDUI
114     fig.add_axes((0.15, 0.1, 0.8, 0.2))
115     res = yy - catenaryTheta([a_hat, c_hat, x0_hat, theta_hat], xx)
116     plt.errorbar(scala*xx, scala*res, yerr=scala*sigma, xerr=scala*sigma, fmt='o',
117                 markersize=mar_size, linewidth=bar_l)
118     plt.grid(which='both', ls='dashed', color='gray')
119     plt.xlabel(r'x [cm]')
120     plt.ylabel(r'Residui')
121     plt.ylim(-0.5, 0.5)
122
123     # FINE
124     plt.savefig('catenaria.pdf')
125     plt.show()

```

Codice 1: Estrapola i dati dall'immagine