

Relazione di Laboratorio

Corso: Sperimentazioni di Fisica 1 – LT
Anno accademico: 2013 – 2014
Docente: dott. C. Sada

Gruppo di Lavoro: Riccardo Milocco – 1070552 – riccardo.milocco@studenti.unipd.it
Andrea de Marco – 1069510 – andrea.demarco.2@studenti.unipd.it

Data di consegna della relazione: martedì, 25 marzo 2014

L'ESTENSIMETRO

Obbiettivi delle esperienze:

Durante le due sessioni di laboratorio per questo esperimento, ci siamo preoccupati di raccogliere sufficienti dati per determinare:

- la risposta elastica lineare, ovvero il valore di K , la dipendenza lineare del modulo dell'allungamento di un corpo elastico dal modulo della forza applicata;
- il modulo di Young, che lega la deformazione e la tensione (longitudinali) subite dal materiale.

(A.d.M)

Apparato strumentale:

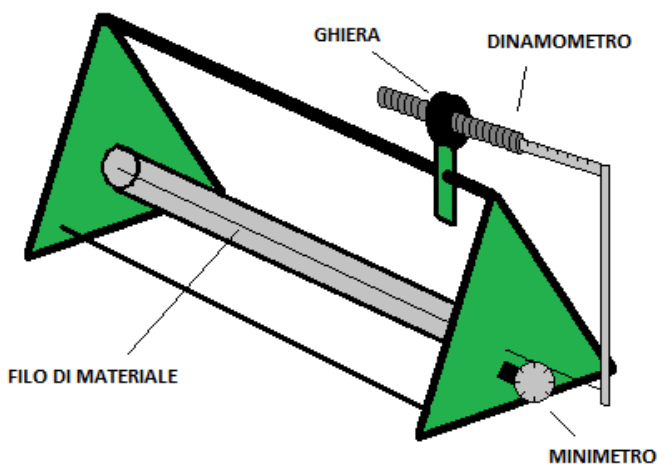


Figura 1 – Schema dell'apparato strumentale - Estensimetro

Estensimetro: dispositivo per la misura della dilatazione elastica di un corpo (in questo caso un filo metallico) soggetto a una forza sollecitante. È costituito da:

- un **dinamometro**: strumento atto alla misurazione delle forze, provvisto di una scala graduata con una sensibilità di 100 grammi;
- un **minimetro** a orologio, o comparatore: misura l'allungamento del materiale soggetto a una forza. È costituito da una punta mobile passante attraverso una serie d'ingranaggi: questo sistema permette di tarare lo strumento. **La lettura è permessa da un quadrante simile a un orologio, la cui scala consente di apprezzare 10 micron;**

- una **ghiera**: cilindro metallico che permette di regolare la forza applicata al materiale;

Principio di funzionamento: ruotando la ghiera, mediante una leva metallica verticale, si esercita una trazione sul filo, che subisce un allungamento o un accorciamento a seconda del verso di rotazione.

Il dinamometro indica quale forza viene applicata al materiale, tenendo presente che la forza rappresentata è pari a un quarto della forza applicata. Una volta applicata la forza, il minimetro esprime il valore dell'allungamento o accorciamento del materiale.

(A.d.M)

Metodologia di misura

Per prima cosa ci si è preoccupati di ridurre al minimo gli errori sistematici che potevano presentarsi durante il corso dell'esperimento. In particolare, si è cercato fissare la posizione degli ingranaggi del minimetro, estraendo e reinserendo la punta mobile cercando di assecondare lo strumento (senza forzature).

In questo modo è stato così possibile trovare una situazione in cui i giochi meccanici all'interno del dispositivo fossero trascurabili.

Inoltre data l'elevata sensibilità dello strumento, si è prestata notevole attenzione ad ogni vibrazione esterna che avrebbe potuto portare uno squilibrio meccanico all'interno dello strumento, e quindi ad un'errata valutazione dell'allungamento del filo.

Secondariamente abbiamo deciso di porre l'indicatore del dinamometro a 200 g sulla scala graduata: per ottenere un allungamento bisogna essere certi che il filo sia teso.

Eliminando l'errore di parallasse, si è deciso di valutare il valore assunto dal filo a seconda di quale frazione della scala si riuscisse ad apprezzare, attribuendo a tale frazione il valore di 1/10 (ossia 1 micron).

Ridotti al minimo gli errori sistematici, si è passati all'effettiva misurazione degli allungamenti: dal valore di riferimento ($F = 200\text{g}$), si è aumentato di 100g alla volta la forza agente sul filo, registrando tutti gli allungamenti fino a 1000g. Successivamente è stata portata la forza a 1100g e di nuovo a 1000g per registrare la fase di accorciamento, fino a 200g.

La stessa procedura di rilevazione dei dati è stata utilizzata per ogni estensimetro.

(R.M.)

Metodologia di rielaborazione dati

Materiale: Acciaio

Innanzitutto abbiamo stabilito che l'errore relativo a ciascuna forza applicata fosse trascurabile rispetto agli errori relativi agli allungamenti: siccome la rotazione della ghiera comporta degli spostamenti millimetrici dell'indicatore rispetto alla scala siamo abbastanza sicuri di non allontanarci più di 0,5 millimetri dal valore della forza espresso sulla scala graduata.

Fatto ciò abbiamo costruito dei grafici per evidenziare la dipendenza ($F, \Delta l$) in allungamento e in accorciamento, a confronto con le rispettive rette interpolanti.

Il calcolo della retta interpolante $y = a + bx$ ci permette di ricavare il valore di K , infatti tale valore corrisponde all'inclinazione della retta interpolante stessa, ovvero il parametro b .

Conoscendo dunque il valore di K è stato in seguito calcolato il modulo di Young (E), essendo essi legati dalla relazione:

$$K = \frac{l_0}{S} \cdot \frac{1}{E}$$

dove l_0 è la lunghezza a riposo del filo e S la sua sezione.

Quindi sono stati calcolati gli errori a posteriori mediante la formula di propagazione dell'errore. Data la funzione $E(l_0, S, K)$, l'errore associato è:

$$\sigma_E^2 \approx \left(\frac{\partial E}{\partial l_0}\right)^2 \sigma_{l_0}^2 + \left(\frac{\partial E}{\partial S}\right)^2 \sigma_S^2 + \left(\frac{\partial E}{\partial K}\right)^2 \sigma_K^2$$

$$\sigma_E^2 \approx \left(\frac{1}{SK}\right)^2 \sigma_{l_0}^2 + \left(-\frac{l_0}{KS^2}\right)^2 \sigma_S^2 + \left(-\frac{l_0}{SK^2}\right)^2 \sigma_K^2$$

L'errore sulla sezione è stato calcolato a posteriori, con il medesimo criterio:

$$\sigma_s^2 \approx \left(\frac{\partial S}{\partial d} \right)^2 \sigma_d^2 \approx \left(\frac{\pi d}{2} \right)^2 \sigma_d^2$$

In particolare l'errore associato a K coincide con l'errore calcolato per il parametro b .

In seguito abbiamo calcolato una media pesata dei valori dei moduli di Young ottenuti per l'acciaio assumendo come pesi i reciproci delle relative varianze.

Quindi è stata calcolata la compatibilità tra la media pesata dei moduli di Young e ogni singolo valore del modulo di Young ottenuto per ogni estensimetro.

Infine abbiamo deciso di costruire due grafici per mettere in evidenza le proprietà elastiche (o meno) del materiale: a parità di sezione, è stato costruito un grafico (l_0, K) ; viceversa a parità di lunghezza, è stato costruito un grafico (S, K) .

Materiale: Tungsteno

L'approccio a questo materiale è stato in parte differente, poiché disponiamo di soli due campioni di misure, uno in allungamento e in uno accorciamento, per lo stesso estensimetro.

Come in precedenza è stata disposta in grafico la dipendenza $(F, \Delta l)$ in allungamento e in accorciamento a confronto con la corrispondente retta interpolante $y = a + bx$, verificando che le due rette (quella relativa all'allungamento e quella relativa all'accorciamento) fossero parallele.

Inoltre, come in precedenza, il parametro b corrisponde al valore ricercato di K .

A partire dai valori così ottenuti di K , uno per l'allungamento e uno per l'accorciamento, sono stati calcolati i rispettivi valori dei moduli di Young.

Tuttavia in questo caso abbiamo preferito il calcolo della media aritmetica associandole l'errore della media, metodo ritenuto più adeguato.

(A.d.M)

Risultati sperimentali - Acciaio

l_0 : lunghezza del filo a riposo

d : diametro del filo

S : sezione del filo

a, b : parametri della retta interpolante $y = a + bx$

K : costante elastica del materiale

E : modulo di Young del materiale

$\langle E \rangle$: media pesata dei moduli di Young

\bar{E} : valore medio dei moduli di Young

σ_{l_0} : errore relativo a l_0

σ_d : errore relativo a d

σ_S : errore relativo a S

σ_a, σ_b : errori relativi ai due parametri a e b

σ_E : errore della stima del modulo di Young

$\langle \sigma_E \rangle$: errore della media pesata dei moduli di Young

$\sigma_{\bar{E}}$: errore della media dei moduli di Young

(unità di misura)

Per semplicità di lettura verranno riportate le misure e gli errori relativi a l_0, d, S solamente nella tabella relativa all'elaborazione dei dati in allungamento.

Materiale: ACCIAIO - allungamento						
Estens.	7	10	11	13	15	18
l_0 (m)	0,950	0,950	0,950	0,900	0,700	0,400
σ_{l_0} (m)	0,002	0,002	0,002	0,002	0,002	0,002
d (m)	330×10^{-6}	406×10^{-6}	432×10^{-6}	279×10^{-6}	279×10^{-6}	279×10^{-6}
σ_d (m)	$3,3 \times 10^{-6}$	$4,1 \times 10^{-6}$	$4,3 \times 10^{-6}$	$2,8 \times 10^{-6}$	$2,8 \times 10^{-6}$	$2,8 \times 10^{-6}$
S (m ²)	$8,6 \times 10^{-8}$	$1,3 \times 10^{-7}$	$1,5 \times 10^{-7}$	$6,1 \times 10^{-8}$	$6,1 \times 10^{-8}$	$6,1 \times 10^{-8}$
σ_S (m ²)	$0,2 \times 10^{-8}$	$0,03 \times 10^{-7}$	$0,03 \times 10^{-7}$	$0,1 \times 10^{-8}$	$0,1 \times 10^{-8}$	$0,1 \times 10^{-8}$
a (m)	$7,2 \times 10^{-6}$	$-1,8 \times 10^{-6}$	$0,7 \times 10^{-6}$	$-15,6 \times 10^{-6}$	$-15,6 \times 10^{-6}$	$-4,2 \times 10^{-6}$
σ_a (m)	$2,5 \times 10^{-6}$	$1,2 \times 10^{-6}$	$1,8 \times 10^{-6}$	$5,6 \times 10^{-6}$	$5,0 \times 10^{-6}$	$2,1 \times 10^{-6}$
b (m/N)	$55,0 \times 10^{-6}$	$35,0 \times 10^{-6}$	$31,0 \times 10^{-6}$	$70,0 \times 10^{-6}$	$55,0 \times 10^{-6}$	$33,0 \times 10^{-6}$
K (m/N)	$55,0 \times 10^{-6}$	$35,0 \times 10^{-6}$	$31,0 \times 10^{-6}$	$70,0 \times 10^{-6}$	$55,0 \times 10^{-6}$	$33,0 \times 10^{-6}$
$\sigma_{b,K}$ (m/N)	$0,1 \times 10^{-6}$	$0,1 \times 10^{-6}$	$0,1 \times 10^{-6}$	$0,3 \times 10^{-6}$	$0,3 \times 10^{-6}$	$0,1 \times 10^{-6}$
E (N/m ²)	$20,2 \times 10^{10}$	$21,0 \times 10^{10}$	$21,0 \times 10^{10}$	$21,1 \times 10^{10}$	$20,9 \times 10^{10}$	$20,1 \times 10^{10}$
σ_E (N/m ²)	$1,0 \times 10^{10}$	$1,1 \times 10^{10}$	$1,1 \times 10^{10}$	$1,1 \times 10^{10}$	$1,4 \times 10^{10}$	$2,3 \times 10^{10}$

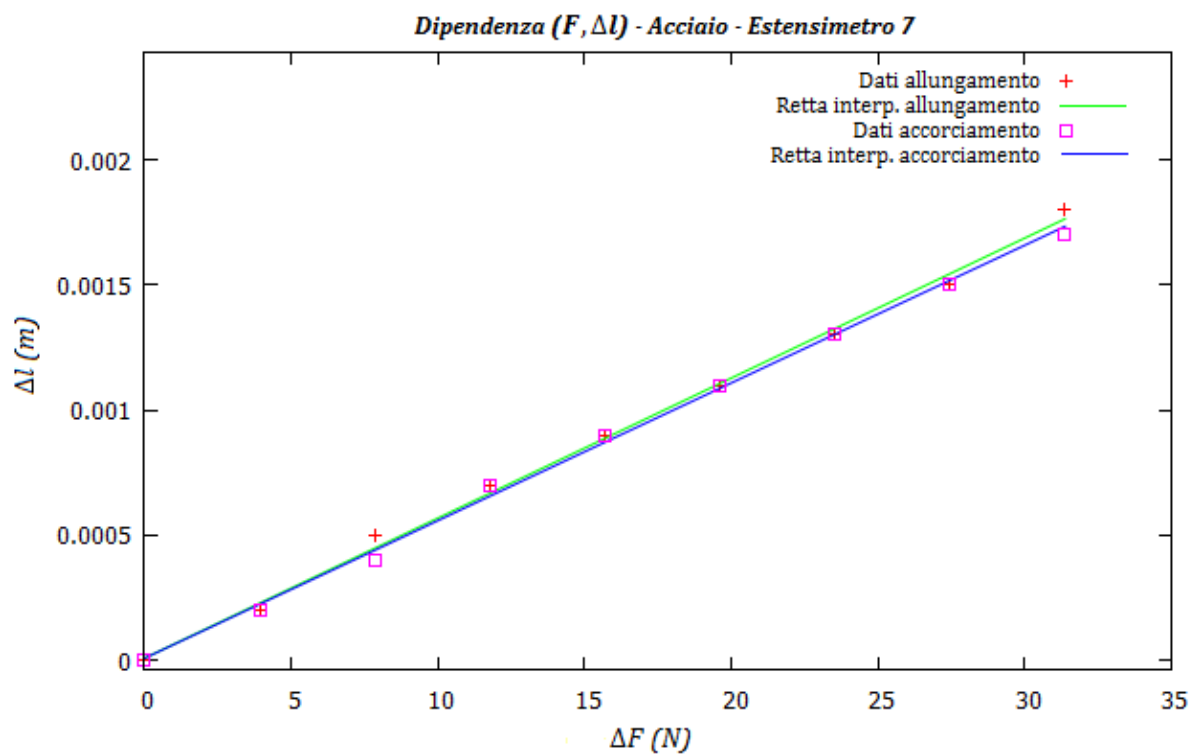
Materiale: ACCIAIO - accorciamento						
Estens.	7	10	11	13	15	18
$a \text{ (m)}$	$7,7 \times 10^{-6}$	$-9,0 \times 10^{-6}$	$7,7 \times 10^{-6}$	$8,2 \times 10^{-5}$	$-4,0 \times 10^{-5}$	$-1,9 \times 10^{-5}$
$\sigma_a \text{ (m)}$	$2,5 \times 10^{-6}$	$0,9 \times 10^{-6}$	$2,2 \times 10^{-6}$	$0,7 \times 10^{-5}$	$0,3 \times 10^{-5}$	$0,4 \times 10^{-5}$
$b \text{ (m/N)}$	$56,0 \times 10^{-6}$	$36,0 \times 10^{-6}$	$31,0 \times 10^{-6}$	$69,0 \times 10^{-6}$	$55,0 \times 10^{-6}$	$33,0 \times 10^{-6}$
$K \text{ (m/N)}$	$56,0 \times 10^{-6}$	$36,0 \times 10^{-6}$	$31,0 \times 10^{-6}$	$69,0 \times 10^{-6}$	$55,0 \times 10^{-6}$	$33,0 \times 10^{-6}$
$\sigma_{b,K} \text{ (m/N)}$	$0,2 \times 10^{-6}$	$0,1 \times 10^{-6}$	$0,1 \times 10^{-6}$	$0,4 \times 10^{-6}$	$0,2 \times 10^{-6}$	$0,2 \times 10^{-6}$
$E \text{ (N/m}^2\text{)}$	$19,8 \times 10^{10}$	$20,6 \times 10^{10}$	$20,7 \times 10^{10}$	$21,4 \times 10^{10}$	$20,7 \times 10^{10}$	$19,6 \times 10^{10}$
$\sigma_E \text{ (N/m}^2\text{)}$	$1,0 \times 10^{10}$	$1,1 \times 10^{10}$	$1,1 \times 10^{10}$	$1,1 \times 10^{10}$	$1,4 \times 10^{10}$	$2,2 \times 10^{10}$

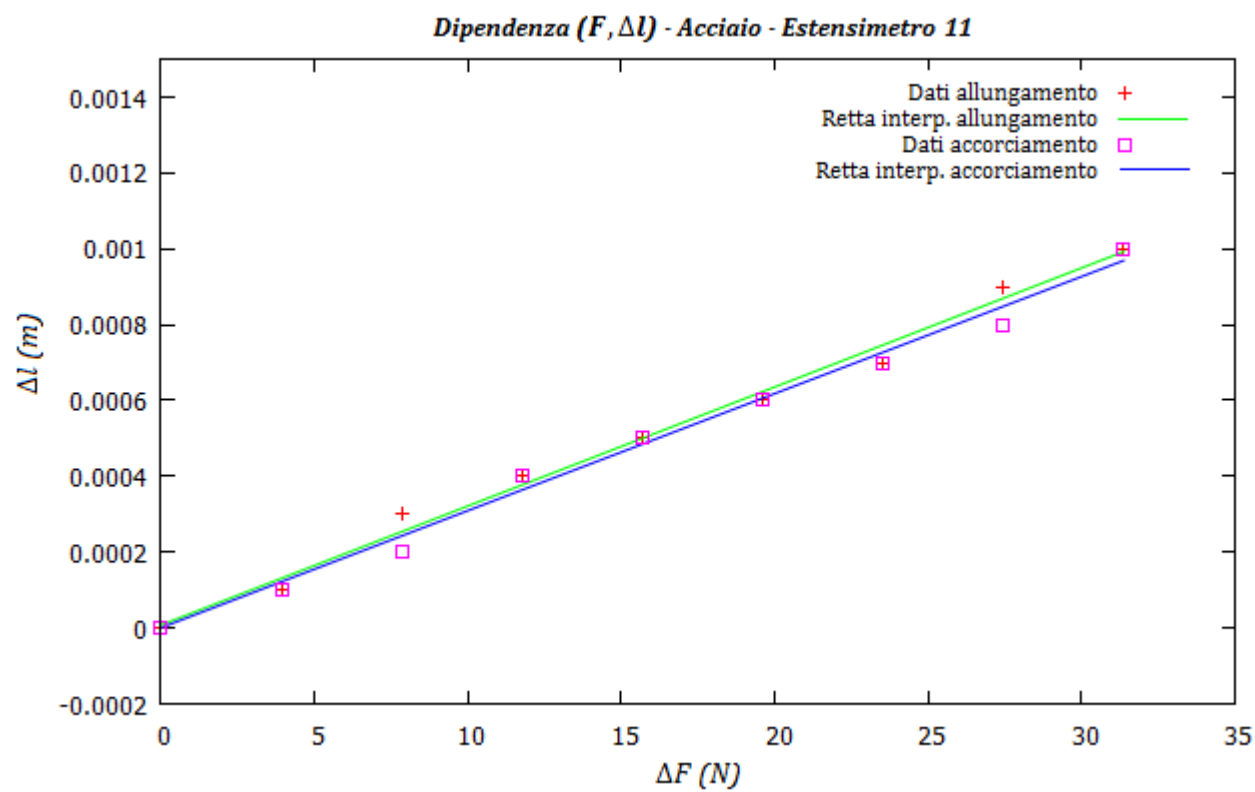
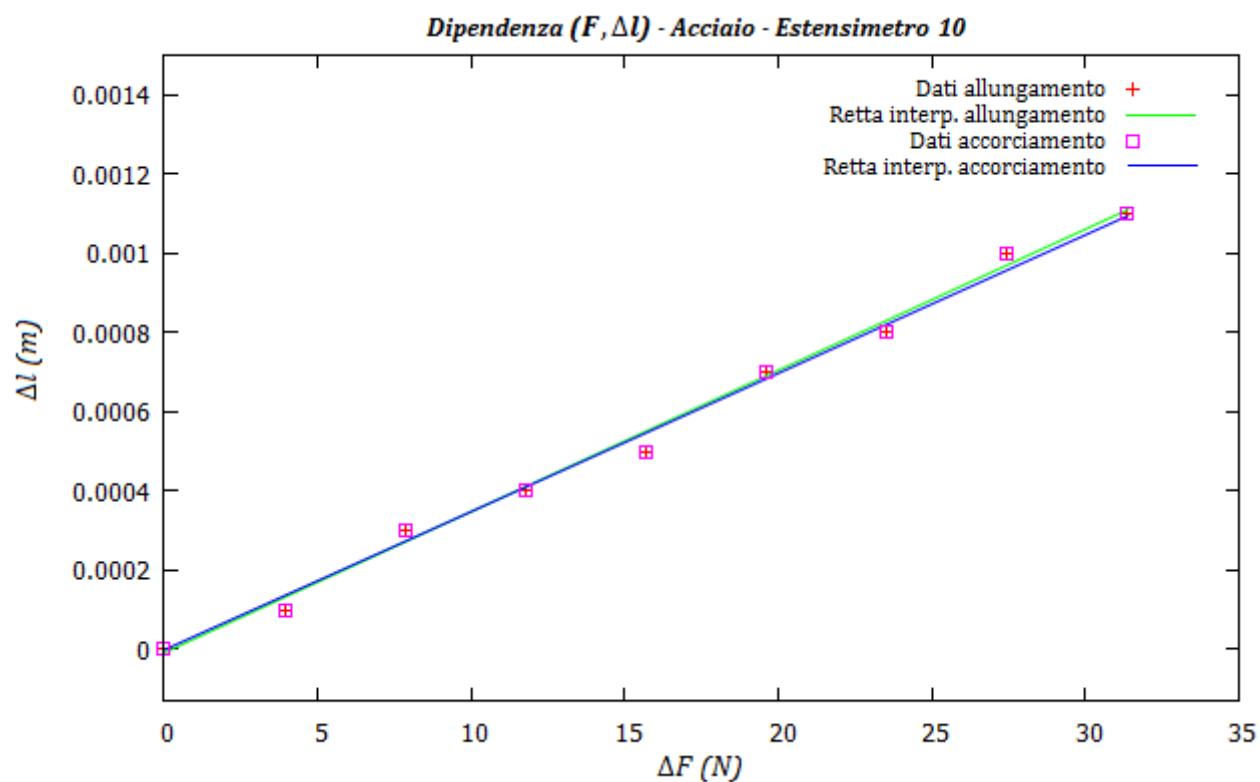
COMPATIBILITÀ DELLE RISPOSTE ELASTICHE					
Estens.	Stima $K \text{ (m/N)}$		Stima $\bar{K} \text{ (m/N)}$	Compatibilità	
	Allungamento	Accorciamento		Allungamento	Accorciamento
7	$(5,5 \pm 0,01) \times 10^{-5}$	$(5,6 \pm 0,02) \times 10^{-5}$	$(5,6 \pm 0,05) \times 10^{-5}$	1 - <i>OTTIMA</i>	0,9 - <i>OTTIMA</i>
10	$(3,5 \pm 0,01) \times 10^{-5}$	$(3,6 \pm 0,01) \times 10^{-5}$	$(3,5 \pm 0,04) \times 10^{-5}$	1 - <i>OTTIMA</i>	1 - <i>OTTIMA</i>
11	$(3,1 \pm 0,01) \times 10^{-5}$	$(3,1 \pm 0,01) \times 10^{-5}$	$(3,1 \pm 0,03) \times 10^{-5}$	0,9 - <i>OTTIMA</i>	0,9 - <i>OTTIMA</i>
13	$(7,0 \pm 0,03) \times 10^{-5}$	$(6,9 \pm 0,04) \times 10^{-5}$	$(6,9 \pm 0,04) \times 10^{-5}$	0,8 - <i>OTTIMA</i>	0,7 - <i>OTTIMA</i>
15	$(5,5 \pm 0,03) \times 10^{-5}$	$(5,5 \pm 0,02) \times 10^{-5}$	$(5,5 \pm 0,03) \times 10^{-5}$	0,7 - <i>OTTIMA</i>	0,8 - <i>OTTIMA</i>
18	$(3,3 \pm 0,01) \times 10^{-5}$	$(3,3 \pm 0,02) \times 10^{-5}$	$(3,3 \pm 0,05) \times 10^{-5}$	1 - <i>OTTIMA</i>	0,9 - <i>OTTIMA</i>

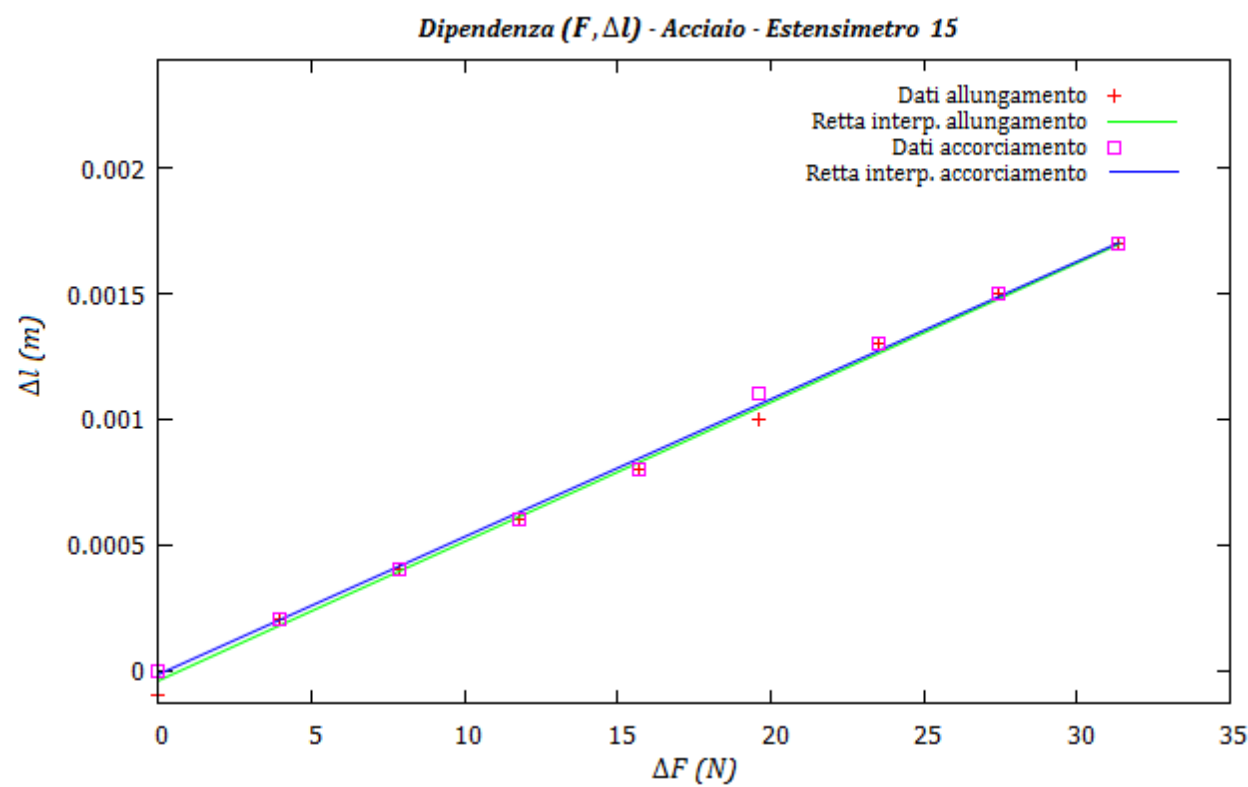
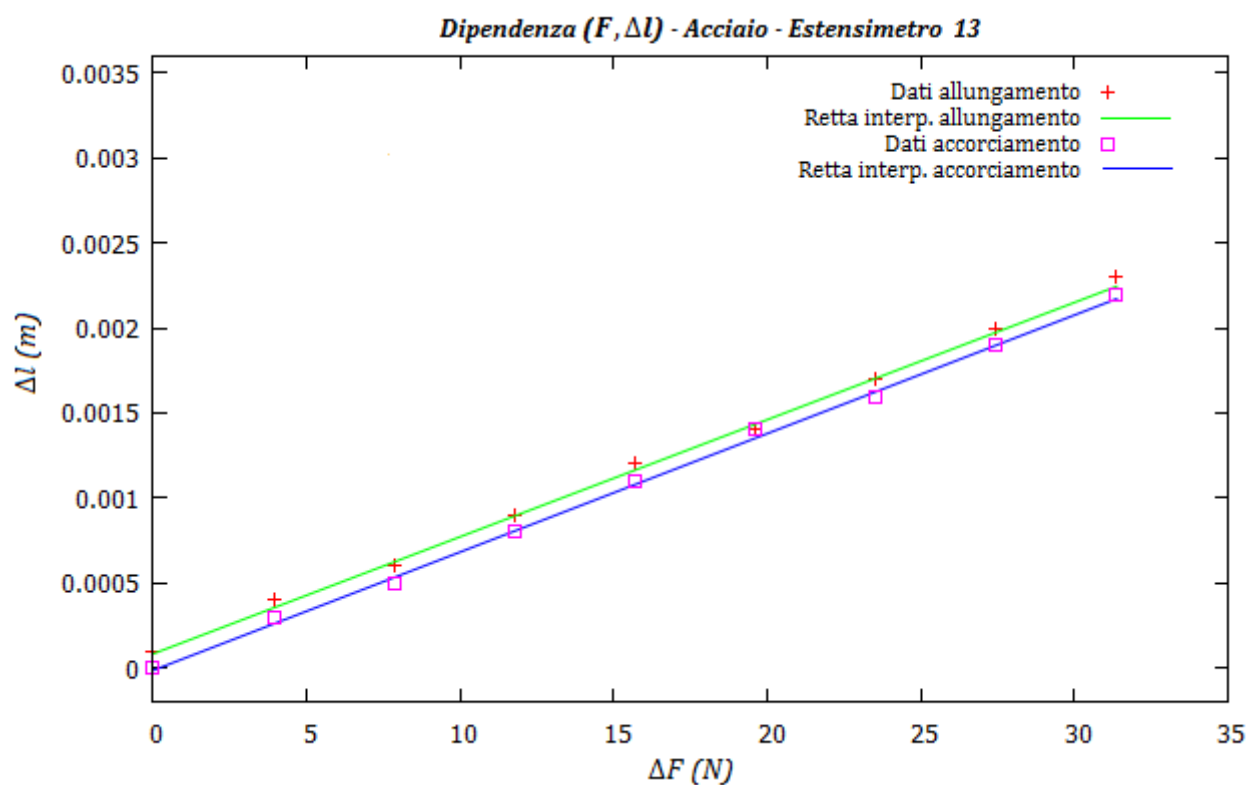
Media pesata e relativo errore moduli Young - Acciaio	
$\langle E \rangle \text{ (N/m}^2\text{)}$	$20,6 \times 10^{10}$
$\langle \sigma_E \rangle \text{ (N/m}^2\text{)}$	$0,4 \times 10^{10}$

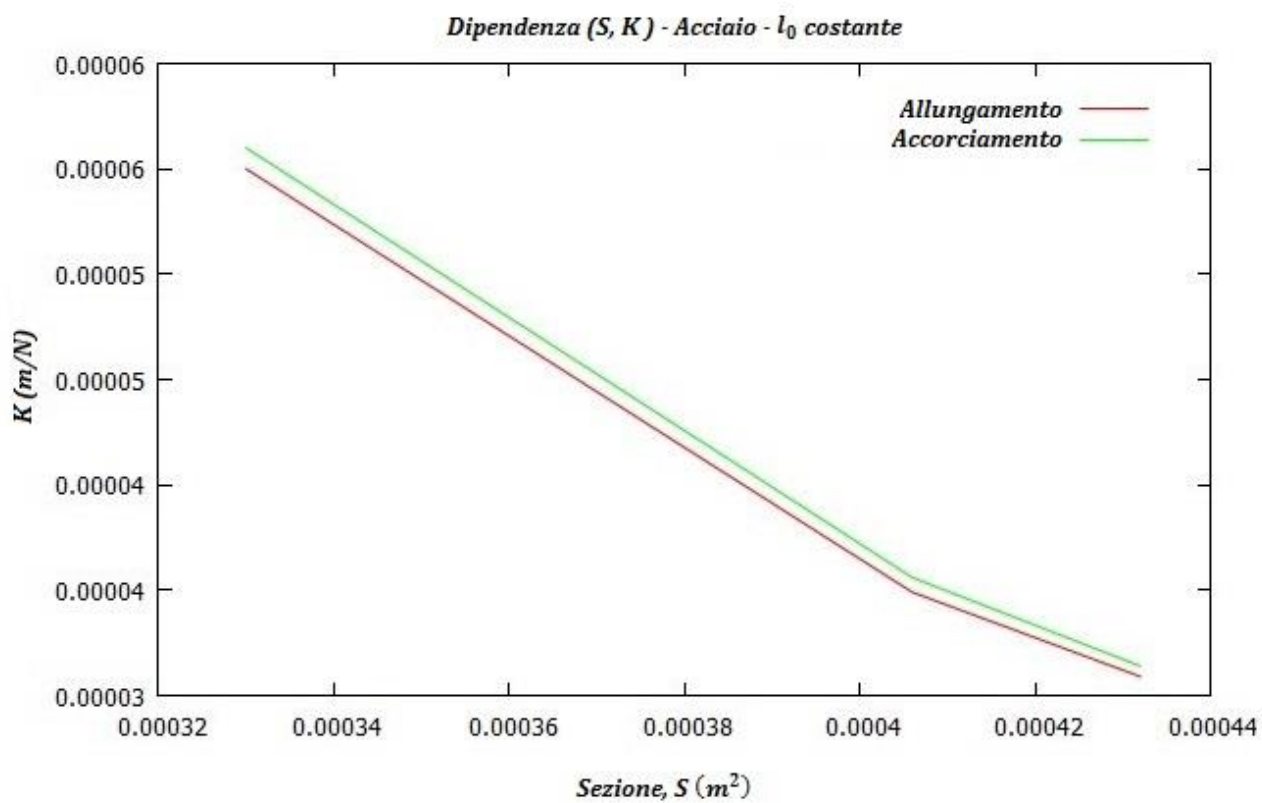
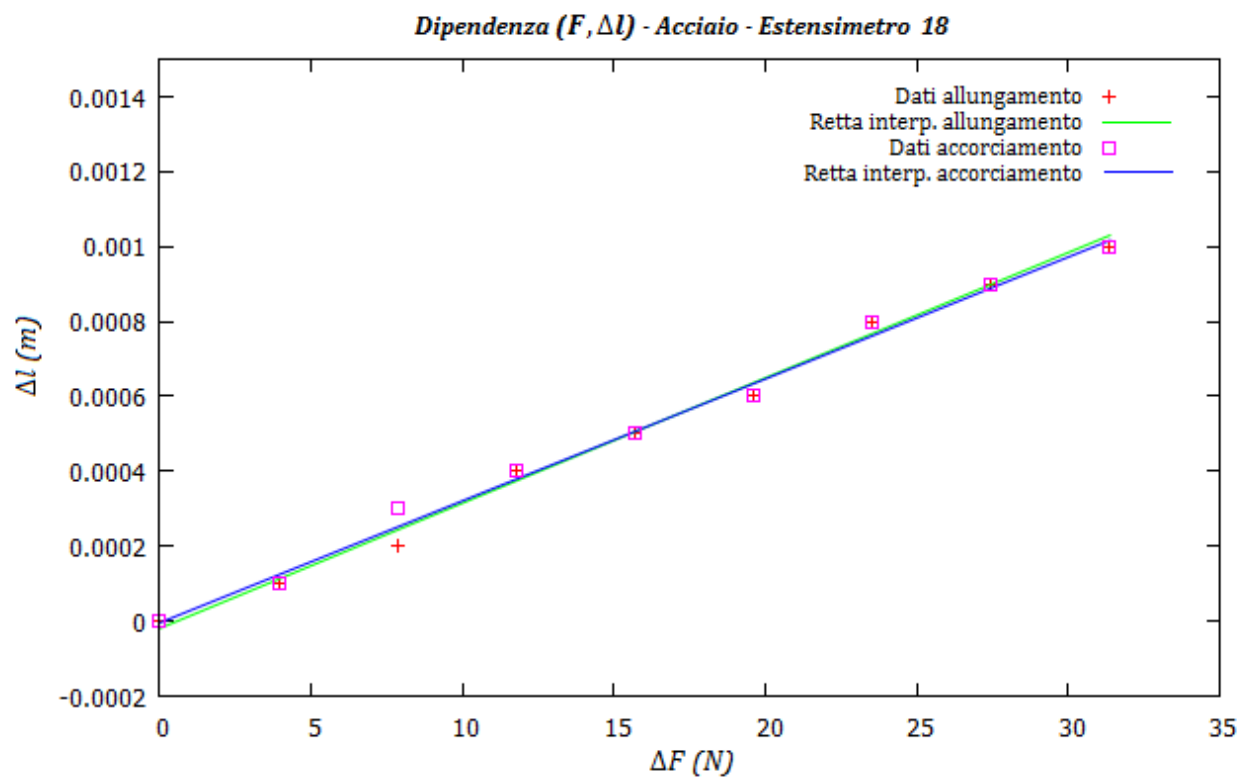
COMPATIBILITÀ DEI MODULI DI YOUNG

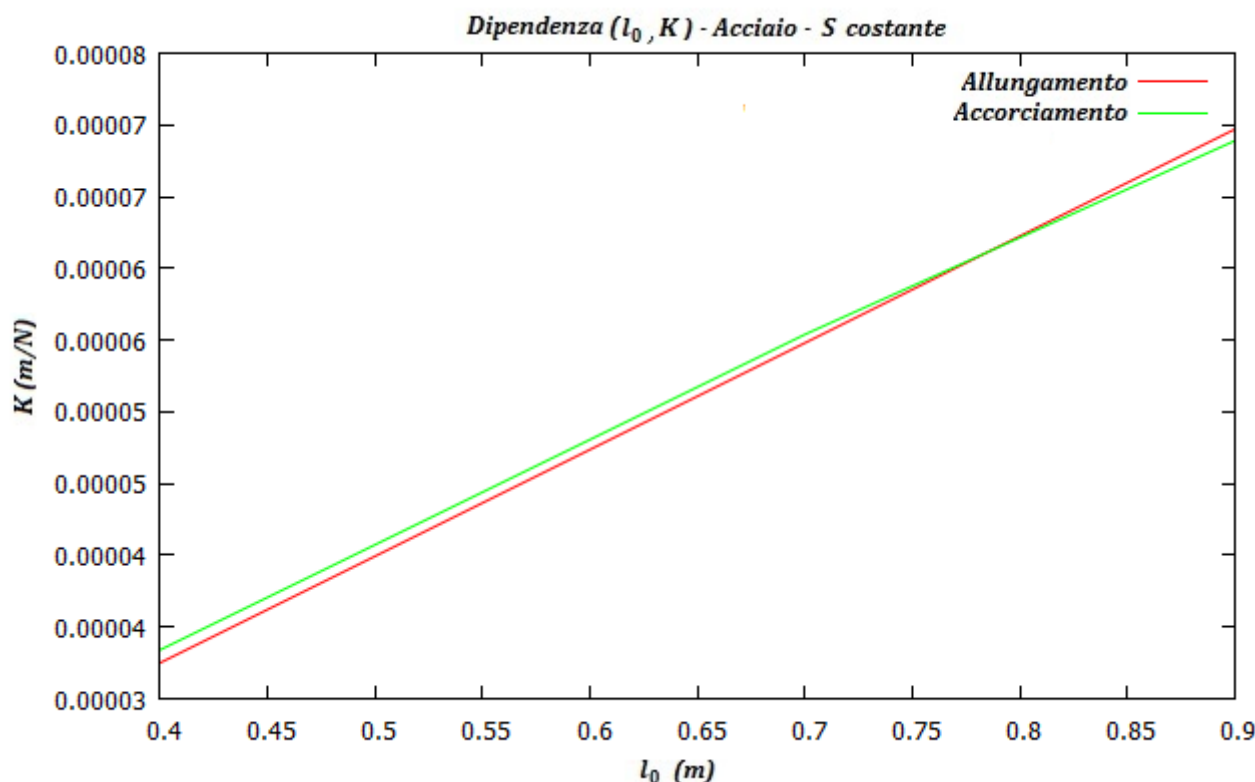
Estens.	Stima E (N/m^2)		Stima $\langle E \rangle$ (N/m^2)	Compatibilità	
	Allungamento	Accorciamento		Allungamento	Accorciamento
7	$(20,2 \pm 1,0) \times 10^{10}$	$(19,8 \pm 1,0) \times 10^{10}$	$(20,6 \pm 0,4) \times 10^{10}$	0,4 - <i>OTTIMA</i>	0,7 - <i>OTTIMA</i>
10	$(21,0 \pm 1,1) \times 10^{10}$	$(20,6 \pm 1,1) \times 10^{10}$	$(20,6 \pm 0,4) \times 10^{10}$	0,3 - <i>OTTIMA</i>	0,04 - <i>OTTIMA</i>
11	$(21,0 \pm 1,1) \times 10^{10}$	$(20,7 \pm 1,1) \times 10^{10}$	$(20,6 \pm 0,4) \times 10^{10}$	0,3 - <i>OTTIMA</i>	0,02 - <i>OTTIMA</i>
13	$(21,1 \pm 1,1) \times 10^{10}$	$(21,4 \pm 1,1) \times 10^{10}$	$(20,6 \pm 0,4) \times 10^{10}$	0,4 - <i>OTTIMA</i>	0,5 - <i>OTTIMA</i>
15	$(20,9 \pm 1,4) \times 10^{10}$	$(20,7 \pm 1,4) \times 10^{10}$	$(20,6 \pm 0,4) \times 10^{10}$	0,2 - <i>OTTIMA</i>	0,03 - <i>OTTIMA</i>
18	$(20,1 \pm 2,3) \times 10^{10}$	$(19,6 \pm 2,2) \times 10^{10}$	$(20,6 \pm 0,4) \times 10^{10}$	0,2 - <i>OTTIMA</i>	0,5 - <i>OTTIMA</i>











(A.d.M – Grafici: R.M.)

Discussione dei risultati sperimentali - Acciaio

L'interpolazione dei dati e i grafici ottenuti, hanno messo in evidenza la risposta elastica del materiale, infatti: **se le rette in allungamento e in accorciamento per lo stesso estensimetro sono parallele, allora hanno la medesima risposta elastica lineare, altrimenti significa che sono stati commessi degli errori sistematici, essendo l'acciaio un materiale elastico.**

Come si può notare dai grafici e dai dati ottenuti, in **nessun** caso le rette sono risultate parallele (nel caso relativo all'estensimetro 11, i valori sono stati arrotondati, dunque non sono eguali). Ciò è facilmente spiegabile pensando al fatto che il **minimetro è uno strumento molto sensibile** agli stimoli esterni come ad esempio piccole vibrazioni. Queste possono portare a una modifica della posizione di equilibrio degli ingranaggi dello strumento, comportando un cambio di posizione dell'indicatore sulla scala.

Anche i dati ottenuti in fase di accorciamento risentono di questi errori: un esempio è il valore a 200 g, che in un solo caso è coinciso col valore dell'accorciamento $0,0 \pm 1 \mu m$ (si ricorda essere il valore di partenza delle misurazioni), per l'estensimetro numero 7.

Osservando i grafici, notiamo che le migliori distribuzioni di dati rispetto alle rette interpolanti sono quelle relative agli estensimetri 13 e 15. Infatti seguono un andamento rettilineo complessivamente coincidente con la retta di interpolazione.

Tuttavia anche negli altri casi i dati si distribuiscono lungo la retta interpolante, anche se in maniera meno regolare. Bisogna inoltre tener presente che la raccolta dati è stata influenzata da una deformazione di alcuni **fili metallici, poiché rimasti in tensione per una settimana**. È un esempio l'estensimetro numero **13**, che infatti presenta la maggiore differenza tra i parametri a , ovvero le intercette con l'asse y ; il valore di accorciamento dello stesso a 200g è pari a $10,3 \times 10 \mu m$.

Nonostante ciò abbiamo una conferma della dipendenza lineare prevista dalla teoria tra la forza applicata e l'allungamento subito dal materiale.

Si può riscontrare anche dagli ultimi due grafici presentati, i quali sono stati utili per comprendere quanto pesassero la sezione e la lunghezza del filo sulla stima di K.

Essendo K dipendente sia dalla lunghezza del filo che dalla sezione, ci si aspetta che al variare della sola lunghezza, K vari proporzionalmente, ovvero:

$$\frac{K}{l_0} = \frac{1}{S} \cdot \frac{1}{E}$$

Mantenendo invece costante la lunghezza a riposo, la relazione tra K e S è di proporzionalità inversa, ovvero:

$$KS = l_0 \cdot \frac{1}{E}$$

Come si può notare dai grafici, tale rapporto sarebbe stato più evidente con un numero molto maggiore di 3 estensimetri (per questo esperimento ne abbiamo tre di pari lunghezza a riposo e tre di pari sezione): avremmo ottenuto una migliore approssimazione dei grafici delle funzioni.

Comunque, a parità di lunghezza a riposo abbiamo ottenuto una curva monotona decrescente, come ci aspettavamo (con un numero maggiore di estensimetri tale curva si sarebbe potuta approssimare a un'iperbole equilatera avente asintoti coincidenti con gli assi cartesiani, ovvero $x \cdot y = \text{costante}$)

A parità di sezione abbiamo riscontrato un cambio di inclinazione, più evidente per i dati in accorciamento.



Aspettandoci un andamento lineare della funzione (in senso analitico), e riscontrando invece un lieve aumento dell'inclinazione giungiamo alla conclusione che i risultati ottenuti sono approssimativamente

corretti, ma non rispettano la proporzione tra lunghezze ed elasticità.

Ad esempio, con i risultati ottenuti, considerando la proporzione:

$$\frac{K_{0,700}}{l_{0,700}} = \frac{K_{0,400}}{l_x}$$



troviamo che il valore l_x che ci aspettiamo essere 0,400 m, è invece pari a 0,420 m.

Infine il confronto tra i moduli di Young ricavati dalla stima di K e la media pesata degli stessi, ha prodotto dei risultati molto soddisfacenti, riscontrando in ogni caso compatibilità ottima.

In particolare le migliori sono quelle relative alla fase di accorciamento degli estensimetri 10, 11, 15 (rispettivamente 0,04; 0,02; 0,03).

Risultati sperimentali - Tungsteno

l_0 : lunghezza del filo a riposo

d : diametro del filo

S : sezione del filo

a, b : parametri della retta interpolante $y = a + bx$

K : costante elastica del materiale

E : modulo di Young del materiale

\bar{E} : valore medio dei moduli di Young

σ_{l_0} : errore relativo a l_0

σ_d : errore relativo a d

σ_S : errore relativo a S

σ_a, σ_b : errori relativi ai due parametri a e b

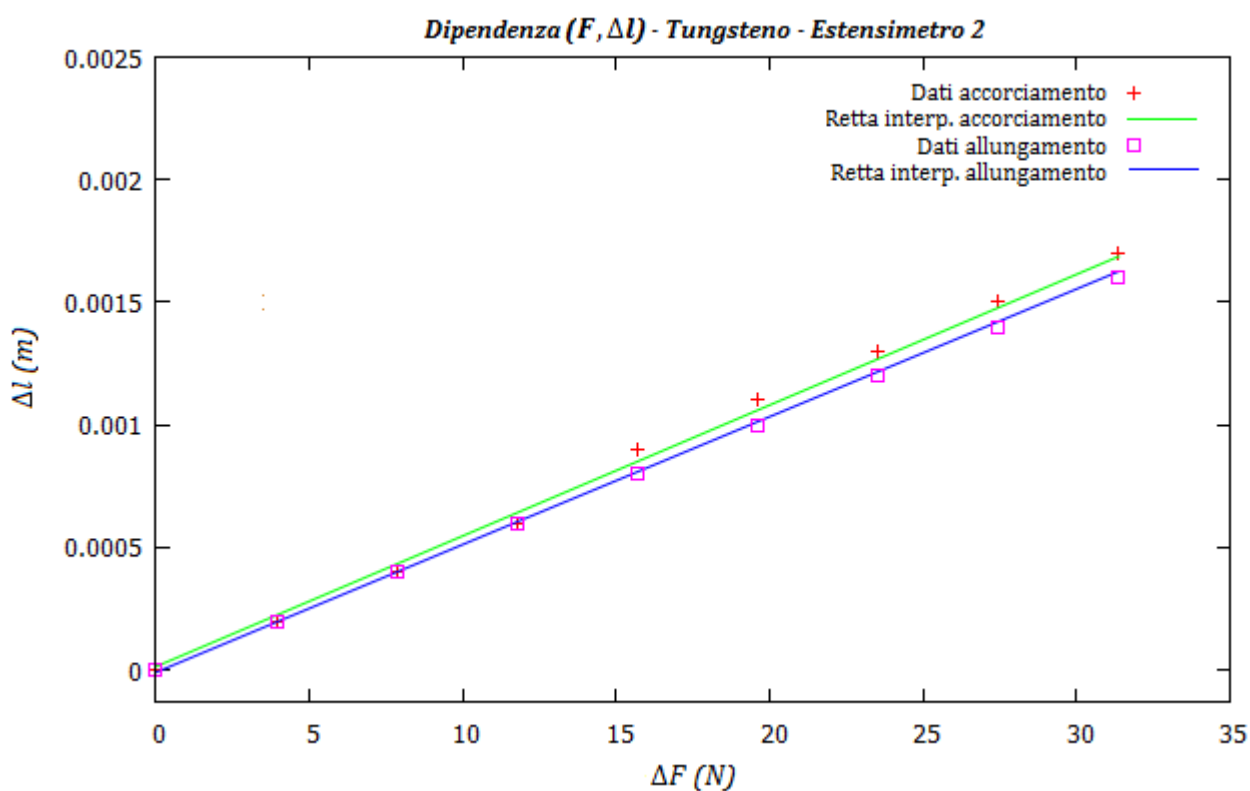
σ_E : errore della stima del modulo di Young

$\sigma_{\bar{E}}$: errore della media dei moduli di Young
(unità di misura)

	Materiale: TUNGSTENO	
Estensimetro 2	Allungamento	Accorciamento
l_0 (m)	1,0	
σ_{l_0} (m)	0,002	
d (m)	$25,0 \times 10^{-5}$	
σ_d (m)	$0,5 \times 10^{-5}$	
S (m ²)	$4,9 \times 10^{-8}$	
σ_S (m ²)	$0,2 \times 10^{-8}$	
a (m)	$-9,4 \times 10^{-6}$	$1,4 \times 10^{-5}$
σ_a (m)	$3,7 \times 10^{-6}$	$0,2 \times 10^{-5}$
b (m/N)	$5,2 \times 10^{-5}$	$5,3 \times 10^{-5}$
K (m/N)	$5,2 \times 10^{-5}$	$5,3 \times 10^{-5}$
$\sigma_{b,K}$ (m/N)	$0,02 \times 10^{-5}$	$0,01 \times 10^{-5}$
E (N/m ²)	$39,1 \times 10^{10}$	$38,2 \times 10^{10}$
σ_E (N/m ²)	$2,4 \times 10^{10}$	$2,3 \times 10^{10}$
\bar{K} (m/N)	$5,3 \times 10^{-5}$	
$\sigma_{\bar{K}}$ (m/N)	$0,06 \times 10^{-5}$	
\bar{E} (N/m ²)	$38,6 \times 10^{10}$	
$\sigma_{\bar{E}}$ (N/m ²)	$0,03 \times 10^{10}$	

COMPATIBILITÀ DELLE RISPOSTE ELASTICHE					
Estens.	Stima K (m/N)		Stima \bar{K} (m/N)	Compatibilità	
	Allungamento	Accorciamento		Allungamento	Accorciamento
2	$(5,2 \pm 0,02) \times 10^{-5}$	$(5,3 \pm 0,01) \times 10^{-5}$	$(5,3 \pm 0,06) \times 10^{-5}$	0,9 - <i>OTTIMA</i>	1,0 - <i>OTTIMA</i>

COMPATIBILITÀ DEI MODULI DI YOUNG					
Estens.	Stima E (N/m ²)		Stima \bar{E} (N/m ²)	Compatibilità	
	Allungamento	Accorciamento		Allungamento	Accorciamento
2	$(39,2 \pm 2,4) \times 10^{10}$	$(38,2 \pm 2,3) \times 10^{10}$	$(38,6 \pm 0,03) \times 10^{10}$	0,2 - <i>OTTIMA</i>	0,2 - <i>OTTIMA</i>



(A.d.M – Grafici: R.M.)

Discussione dei risultati sperimentali - Tungsteno

Il grafico mette in evidenza la buona distribuzione dei dati in fase di allungamento lungo la retta interpolante; in accorciamento invece i dati si distribuiscono in parte con ordinata maggiore del punto appartenente alla retta e avente la stessa ascissa, in parte con ordinata minore.

Inoltre le due rette non sono parallele: quella relativa alla fase di accorciamento ha una pendenza maggiore.

Ciò è spiegabile attribuendo la causa d'errore a vibrazioni esterne, che hanno influito sull'equilibrio degli ingranaggi del comparatore.

Tuttavia la linearità degli allungamenti ci dimostra che questo materiale è elastico: il coefficiente di ciascuna retta interpolante, come anticipato in precedenza, è pari alla risposta elastica lineare K .

Tramite il valore di K , abbiamo quindi calcolato il valore di ciascun modulo di Young con il relativo errore. In seguito è stata calcolata la loro media aritmetica e l'errore associato, confrontando infine ciascun valore col valor medio tramite la compatibilità.

Si è riscontrata una compatibilità ottima in entrambi i casi.

(A.d.M)

Conclusioni

L'esperienza condotta ci ha permesso di stabilire quale fosse il metodo migliore per verificare l'elasticità meccanica di un metallo a sezione circolare.

Nel caso dell'acciaio, la stima di K è stata possibile interpolando i dati raccolti con una retta il cui coefficiente è proprio la risposta elastica del materiale. La sua stima è stata decisiva per poter in seguito determinare il modulo di Young E per il materiale.

Inoltre la dipendenza (l_0, K) ha evidenziato il fatto che la risposta elastica è proporzionale alla lunghezza del filo di materiale, a parità di sezione; viceversa la dipendenza (S, K) mostra che aumentando la sezione del filo diminuisce la risposta elastica del materiale.

Pertanto si è dimostrato sperimentalmente che la risposta elastica K di un materiale è strettamente dipendente dalla sua struttura geometrica.

In particolare l'estensimetro numero 13 è quello in cui si ha riscontrato una migliore compatibilità con il valore medio della risposta elastica.

La media pesata dei moduli di Young stimati per ogni estensimetro è stata utile per definire un unico valore di E che tenesse conto degli errori commessi e che servisse a confrontare i singoli risultati rispetto ad un riferimento. In tutti i casi abbiamo ottenuto una compatibilità ottima, tuttavia per quanto riguarda la distribuzione dei dati risulta migliore il campione relativo all'estensimetro 13.

Ciò significa che anche se la compatibilità non sia quella ottimale, i dati raccolti sono stati affetti da un minor numero di errori sistematici, contrariamente alle aspettative: il filo di tale estensimetro aveva subito una deformazione poiché rimasto in tensione per una settimana.

Nel caso del tungsteno è stato più complesso fare delle valutazioni poiché disponevamo di un esiguo numero di dati.

Nonostante la distribuzione dei dati sia in entrambi i casi pressoché lineare, le risposte elastiche ottenute mediante l'interpolazione dei dati sono differenti, anche se in minima parte. Le rette interpolanti dunque non sono parallele, ma i dati si distribuiscono (specie in allungamento) lungo le stesse. Sarebbe stato più interessante confrontare i dati raccolti per questo estensimetro con i quelli raccolti da un secondo estensimetro con le due caratteristiche (sezione e lunghezza a riposo) in comune, per capire come la composizione chimica del materiale sottoposto alla trazione influisca sull'elasticità.

Si è ottenuto un valore medio della risposta elastica pari a $(5,3 \pm 0,06) \times 10^{-5} \text{ m/N}$ che ci è servito per stabilire quale valore di K stimato in precedenza fosse migliore. In particolare il valore di K stimato a partire dai dati raccolti in allungamento è risultato essere quello più compatibile con la media aritmetica.

Per quanto riguarda il modulo di Young si è confrontato ciascun valore stimato a partire dai parametri K con la loro media aritmetica mediante compatibilità. In entrambi i casi la compatibilità è risultata ottima, ma guardando la distribuzione dei dati, notiamo che i dati in accorciamento sono affetti da un errore maggiore: si discostano infatti dalla retta interpolante in maniera più evidente rispetto all'allungamento.

Pertanto riteniamo che la migliore stima del modulo di Young sia quella ottenuta in fase di allungamento, ovvero $(39,2 \pm 2,4) \times 10^{10} \text{ N/m}^2$.

(R.M e A.d.M)

Appendici

Estensimetro Acciaio		200g	300g	400g	500g	600g	700g	800g	900g	1000g
7	Allungamento ($\times 10 \mu\text{m}$)	0,0	22,6	44,2	65,6	87,0	109,2	130,3	151,8	173
	Accorciamento($\times 10 \mu\text{m}$)	0,0	23,1	45,1	66,1	88,5	111,9	132,8	154	175,9
10	Allungamento ($\times 10 \mu\text{m}$)	0,0	13,4	27,0	40,8	54,9	68,5	82,0	95,5	109,5
	Accorciamento($\times 10 \mu\text{m}$)	-0,9	13,3	27,0	40,9	54,9	68,9	83,2	97,1	110,9
11	Allungamento ($\times 10 \mu\text{m}$)	0,0	12,1	24,8	36,4	48,3	60,1	72,7	84,9	97,1
	Accorciamento($\times 10 \mu\text{m}$)	0,3	13,1	26,0	37,8	50,1	61,9	74,9	86,8	99,2
13	Allungamento ($\times 10 \mu\text{m}$)	0,0	25,5	51,9	80,0	107,2	135,6	162,2	190,8	217,1
	Accorciamento($\times 10 \mu\text{m}$)	10,3	35,0	60,9	88,5	116,0	142,4	170,9	198,1	225,0
15	Allungamento ($\times 10 \mu\text{m}$)	0,0	18,9	40,9	62,5	84,0	105,8	127,9	148,9	170,4
	Accorciamento($\times 10 \mu\text{m}$)	-5,0	18,0	40,0	61,5	82,8	104,7	126,1	147,9	169,5
18	Allungamento ($\times 10 \mu\text{m}$)	0,0	12,0	25,1	37,9	50,8	62,8	76,0	89,0	102,0
	Accorciamento($\times 10 \mu\text{m}$)	-3,0	11,9	24,9	37,5	50,9	63,0	76,5	90,0	103,0
Estens. Tungsteno										
2	Allungamento ($\times 10 \mu\text{m}$)	0,0	19,0	39,6	60,2	81,0	101,1	121,2	141,8	163,5
	Accorciamento($\times 10 \mu\text{m}$)	1,5	22	43,2	64,2	85,1	106,1	127,2	147,2	168,9

Programmi in C++:

- "computoke.cxx"
- "mediapesata.cxx"
- "mediatung.cxx"
- "mediakacc.cxx"

(R.M.)

**** Codice del programma "computoke.cxx" ****

```
#include <iostream>
#include <fstream>
#include <cmath>
#include <iomanip>
#include <string>

double sommatoria( double *, double *, int );

int main()
{
    using namespace std;
    string yesorno;

    cout << "\n\n ***** Esperimento Estensimetro *****" << endl << endl;

do
{
    char reply_mat, reply_numest;

    cout << "Quale materiale ti interessa analizzare?" << endl
         << "\na) Acciaio" << endl
         << "b) Tungsteno" << endl;

    cin >> reply_mat;

    double l0, diametro, err_lung, errdiametro, e_teorico;

    ifstream write;
    if( reply_mat == 'a' )
    {
        cout << "\nDi quale strumento vuoi analizzare i dati?" << endl
             << "\na) Estensimetro n' 7 - L = 950 +- 2 mm | S = 0.330 mm +- 1%" <<
endl
             << "b) Estensimetro n' 10 - L = 950 +- 2 mm | S = 0.406 mm +- 1%" << endl
             << "c) Estensimetro n' 11 - L = 950 +- 2 mm | S = 0.432 mm +- 1%" << endl
             << "d) Estensimetro n' 13 - L = 900 +- 2 mm | S = 0.279 mm +- 1%" << endl
             << "e) Estensimetro n' 15 - L = 700 +- 2 mm | S = 0.279 mm +- 1%" << endl
             << "f) Estensimetro n' 18 - L = 400 +- 2 mm | S = 0.279 mm +- 1%" << endl
<< endl;

        cin >> reply_numest;

        if( reply_numest == 'a' )
        {
            write.open("estensimetro7acc.txt");
            l0 = 0.950;
            diametro = 0.000330;
            err_lung = 0.002;
            errdiametro = diametro / 100;
        }
        else if( reply_numest == 'b' )
        {
            write.open("estensimetro10acc.txt");
```



```

        l0 = 0.950;
        diametro = 0.000406;
        err_lung = 0.002;
        errdiametro = diametro / 100;
    }
    else if( reply_numest == 'c' )
    {
        write.open("estensimetro11acc.txt");
        l0 = 0.950;
        diametro = 0.000432;
        err_lung = 0.002;
        errdiametro = diametro / 100;
    }
    else if( reply_numest == 'd' )
    {
        write.open("estensimetro13acc.txt");
        l0 = 0.900;
        diametro = 0.000279;
        err_lung = 0.002;
        errdiametro = diametro / 100;
    }
    else if( reply_numest == 'e' )
    {
        write.open("estensimetro15acc.txt");
        l0 = 0.700;
        diametro = 0.000279;
        err_lung = 0.002;
        errdiametro = diametro / 100;
    }
    else if( reply_numest == 'f' )
    {
        write.open("estensimetro18acc.txt");
        l0 = 0.400;
        diametro = 0.000279;
        err_lung = 0.002;
        errdiametro = diametro / 100;
    }
}
else if( reply_mat == 'b' )
{
    cout << "\n *** Estensimetro n' 2 - Tungsteno - 1000 +- 2 mm | 0,250 +- 0,005 mm
***" << endl;
    write.open("estensimetro2tung.txt");
    l0 = 1.0;
    diametro = 0.000250;
    err_lung = 0.002;
    errdiametro = 0.000005;
}

const int totaledati = 18;
double allungamenti[totaledati];

for( int i = 0; i < totaledati; i++ )

```

```

        write >> allungamenti[i];

write.close();

double forze[9];
int k = 9;
cout << "\n Forze    Allungamenti    Accorciamenti" << endl << endl;

for ( int i = 0; i < 9; i++ )
{
    forze[i] = 0.1 * 4 * i * 9.806;
    allungamenti[i] = allungamenti[i] * pow( 10, -5 );
    allungamenti[k] = allungamenti[k] * pow( 10, -5 );
    cout << setiosflags( ios::fixed | ios::showpoint ) << setprecision(1) << setw(5) <<
forze[i]
        << setprecision(6) << setw(20) << allungamenti[i] << setw(20) <<
setprecision(6) << allungamenti[k] << endl;
    k++;
}

double quad_forze[9];
for ( int i = 0; i < 9; i++ )
    quad_forze[i] = forze[i] * forze[i];

double sommaquadforze = sommatoria( quad_forze, quad_forze + 9, 9 );

double sommaforze = sommatoria( forze, forze + 9, 9 );

double sommallungamenti, sommallungperforza, delta, addendo_1a, addendo_2a, a,
addendo_1b, addendo_2b, b;

double allungperforza[9];

delta = 9 * sommaquadforze - pow ( sommaforze, 2 );

string prosric;

do{

string allacc;

    cout << "\n\nVuoi calcolare la retta interpolante dell'allungamento o dell'accorciamento?"
<< endl;
    cin >> allacc;

double errore_y, sigma_a, sigma_b, E, sigma_E;

double quad_diff_ret_y[9];

if( allacc == "allungamento" )
{
    sommallungamenti = sommatoria( allungamenti, allungamenti + 9, 9 );

```

```

        for( int i = 0; i < 9; i++ )
            allungperforza[i] = allungamenti[i] * forze[i];
    }
    else if( allacc == "accorciamento" )
    {
        sommallungamenti = sommatoria( allungamenti + 9, allungamenti + 18, 9 );

        for( int i = 0; i < 9; i++ )
            allungperforza[i] = allungamenti[( i + 9 )] * forze[i];
    }

    sommallungperforza = sommatoria( allungperforza, allungperforza + 9, 9 );

    addendo_1a = sommaquadforze * sommallungamenti;
    addendo_2a = sommaforze * sommallungperforza;

    a = ( 1.0 / delta ) * ( addendo_1a - addendo_2a );

    addendo_1b = 9 * sommallungperforza;
    addendo_2b = sommaforze * sommallungamenti;

    b = ( 1.0 / delta ) * ( addendo_1b - addendo_2b );

    int p;
    if( allacc == "allungamento" )
        p = 0;
    else if( allacc == "accorciamento" )
        p = 9;

    for ( int i = 0; i < 9; i++ )
        quad_diff_ret_y[i] = pow ( ( a + b * forze[i] - allungamenti[( i + p )] ), 2 );

    errore_y = sqrt ( sommatoria ( quad_diff_ret_y, quad_diff_ret_y + 9, 9 ) / ( 9 - 2 ) );

    sigma_a = errore_y * sqrt ( sommatoria ( quad_forze, quad_forze + 9, 9 ) / delta );

    sigma_b = errore_y * sqrt ( 9.0 / delta );

    double sezione = M_PI * pow( diametro, 2 ) / 4;

    E = l0 / ( sezione * b );

    double add_1_sE = pow( sezione, 2 ) * pow( b, 2 ) * err_lung;

    double sigmasezione = M_PI * diametro * errdiametro / 2;

    double add_2_sE = pow( b, 2 ) * pow( l0, 2 ) * pow( sigmasezione, 2 );

    double add_3_sE = pow( sezione, 2 ) * pow( l0, 2 ) * pow( sigma_b, 2 );

    sigma_E = sqrt( add_1_sE + add_2_sE + add_3_sE ) / ( pow( b, 2 ) * pow( sezione, 2 ) );

    cout << setprecision(0) << "\n\ndelta: " << delta << endl << setprecision(7) << "a: " << a
    << endl

```

```

        << "b: " << b << endl << "\nSigma di a e': " << sigma_a << endl << "Sigma di b e': "
<< sigma_b << endl
        << "L'errore della y: " << errore_y << endl
        << setprecision(10) << "Sezione: " << sezione << endl
        << "L'errore della sezione: " << sigmasezione << endl
        << setprecision(0) << "\n\nE: " << E << " +- " << sigma_E << endl
        << setprecision(7) << "K: " << b << " +- " << sigma_b << endl;

```

```

ofstream fout, gout, hout;

```

```

if( reply_mat == 'a' )

```

```

{

```

```

    if( reply_numest == 'a' )

```

```

    {

```

```

        if( allacc == "allungamento" )

```

```

        {

```

```

            fout.open( "dati_allungamento_7acc.txt" );

```

```

            gout.open( "retta_allungamento_7acc.txt" );

```

```

            hout.open( "e_allungamento_7acc.txt" );

```

```

        }

```

```

        else if( allacc == "accorciamento" )

```

```

        {

```

```

            fout.open( "dati_accorciamento_7acc.txt" );

```

```

            gout.open( "retta_accorciamento_7acc.txt" );

```

```

            hout.open( "e_accorciamento_7acc.txt" );

```

```

        }

```

```

    }

```

```

    else if( reply_numest == 'b' )

```

```

    {

```

```

        if( allacc == "allungamento" )

```

```

        {

```

```

            fout.open( "dati_allungamento_10acc.txt" );

```

```

            gout.open( "retta_allungamento_10acc.txt" );

```

```

            hout.open( "e_allungamento_10acc.txt" );

```

```

        }

```

```

        else if( allacc == "accorciamento" )

```

```

        {

```

```

            fout.open( "dati_accorciamento_10acc.txt" );

```

```

            gout.open( "retta_accorciamento_10acc.txt" );

```

```

            hout.open( "e_accorciamento_10acc.txt" );

```

```

        }

```

```

    }

```

```

    else if( reply_numest == 'c' )

```

```

    {

```

```

        if( allacc == "allungamento" )

```

```

        {

```

```

            fout.open( "dati_allungamento_11acc.txt" );

```

```

            gout.open( "retta_allungamento_11acc.txt" );

```

```

            hout.open( "e_allungamento_11acc.txt" );

```

```

        }

```

```

        else if( allacc == "accorciamento" )

```

```

        {

```

```

            fout.open( "dati_accorciamento_11acc.txt" );

```

```

            gout.open( "retta_accorciamento_11acc.txt" );

```

```

            hout.open( "e_accorciamento_11acc.txt" );

```

```

        }

```

```

}
else if( reply_numest == 'd' )
{
    if( allacc == "allungamento" )
    {
        fout.open( "dati_allungamento_13acc.txt" );
        gout.open( "retta_allungamento_13acc.txt" );
        hout.open( "e_allungamento_13acc.txt" );
    }
    else if( allacc == "accorciamento" )
    {
        fout.open( "dati_accorciamento_13acc.txt" );
        gout.open( "retta_accorciamento_13acc.txt" );
        hout.open( "e_accorciamento_13acc.txt" );
    }
}
else if( reply_numest == 'e' )
{
    if( allacc == "allungamento" )
    {
        fout.open( "dati_allungamento_15acc.txt" );
        gout.open( "retta_allungamento_15acc.txt" );
        hout.open( "e_allungamento_15acc.txt" );
    }
    else if( allacc == "accorciamento" )
    {
        fout.open( "dati_accorciamento_15acc.txt" );
        gout.open( "retta_accorciamento_15acc.txt" );
        hout.open( "e_accorciamento_15acc.txt" );
    }
}
else if( reply_numest == 'f' )
{
    if( allacc == "allungamento" )
    {
        fout.open( "dati_allungamento_18acc.txt" );
        gout.open( "retta_allungamento_18acc.txt" );
        hout.open( "e_allungamento_18acc.txt" );
    }
    else if( allacc == "accorciamento" )
    {
        fout.open( "dati_accorciamento_18acc.txt" );
        gout.open( "retta_accorciamento_18acc.txt" );
        hout.open( "e_accorciamento_18acc.txt" );
    }
}
}

else if( reply_mat == 'b' )
{
    if( allacc == "allungamento" )
    {
        fout.open( "dati_allungamento_2tung.txt" );
        gout.open( "retta_allungamento_2tung.txt" );
        hout.open( "etungall.txt" );

```

```

    }
    else if( allacc == "accorciamento" )
    {
        fout.open( "dati_accorciamento_2tung.txt" );
        gout.open( "retta_accorciamento_2tung.txt" );
        hout.open( "etungacc.txt" );
    }
}

int j;

if( allacc == "allungamento" ) j = 0;
else if( allacc == "accorciamento" ) j = 9;

for ( int i = 0; i < 9; i++ )
    fout << setiosflags( ios::fixed | ios::showpoint ) << setprecision(2) << forze[i] << "
"
    << setprecision(4) << allungamenti[( i + j )] << endl;

fout.close();

double retta_interpolante[9];

for ( int i = 0; i < 9; i++ )
{
    retta_interpolante[i] = a + b * forze[i];
    gout << setiosflags( ios::fixed | ios::showpoint ) << setprecision(3) << forze[i] << "
" << retta_interpolante[i] << endl;
}

gout.close();

hout << E << " " << sigma_E << endl;

hout.close();

cout << "\n\nSe vuoi ricalcolare allungamento e/o accorciamento inserisci 'ricomincia'" <<
endl << "Altrimenti inserisci una qualsiasi stringa" << endl;
cin >> prosric;

} while( prosric == "ricomincia" );

cout << "\n\nVuoi rieseguire il programma? ";
cin >> yesorno;

do
{
    if( yesorno != "si" && yesorno != "no" )
    {
        cout << "\n*** Risposta non valida *** \nVuoi rieseguire il programma? ";
        cin >> yesorno;
    }
} while( yesorno != "si" && yesorno != "no" );
}

```

```

while( yesorno == "si" );

    return 0;
}

double sommatoria( double *inizio, double *fine, int totalemisure )
{
    double sum = 0;
    double *ptr;

    for( ptr = inizio; ptr != fine; ptr++ )
        sum = sum + *ptr;

    return sum;
}

```

**** Codice del programma “mediapesata.cxx” ****

```

#include <iostream>
#include <fstream>
#include <cmath>
#include <iomanip>
#include <string>

double sommatoria( double *, double *, int );

int main()
{
    using namespace std;
    string yesorno;

    do
    {
        cout << "\n\nCalcoliamo il valore di E per l'acciaio tramite media pesata" << endl;

        //prendo i valori dal documento di testo che ho precedentemente creato con a, sigma di a, b e
        sigma di b di seguito.

        double e[12];
        double sigma_e[12];

        ifstream fin;
        fin.open("tutti_e.txt");

        for ( int i = 0; i < 12; i++ )
            fin >> e[i];

        fin.close();

        fin.open("tutti_sigma_e.txt");

        for ( int i = 0; i < 12; i++ )
            fin >> sigma_e[i];
    }
    while( yesorno != "no" );

    return 0;
}

```

```

fin.close();

double peso_e[12];
for( int i = 0; i < 12; i++ )
    peso_e[i] = 1 / ( sigma_e[i] * sigma_e[i] );

double media_pesata_e;
double err_medpes_e;

cout << endl;

double e_i_peso_e[12];

for ( int i = 0; i < 12; i++ )
    e_i_peso_e[i] = e[i] * peso_e[i];

media_pesata_e = sommatoria( e_i_peso_e, e_i_peso_e + 12, 12 ) / sommatoria( peso_e, peso_e
+ 12, 12 );

err_medpes_e = sqrt( 1.0 / sommatoria( peso_e, peso_e + 12, 12 ) );

cout << "\nMedia pesata di E: " << media_pesata_e << endl
    << "Errore: " << err_medpes_e << endl;

cout << "\nLa compatibilita' tra i valori di E stimati la loro media pesata e':" << endl << endl
    << " ALLUNGAMENTO" << endl
    << " Estensimetro    Valore di Compatibilita'          Tipo di Compatibilita'" <<
endl << endl;

int numest[] = { 7, 10, 11, 13, 15, 18 };
int k = 0;

for ( int i = 0; i < 12; i++ )
{
    double compatibilita = 0;

    compatibilita = fabs( e[i] - media_pesata_e ) / sqrt( pow( sigma_e[i], 2 ) + pow(
err_medpes_e, 2 ) );

    if( k == 6 )
    {
        cout << "\n ACCORCIAMENTO " << endl
            << " Estensimetro    Valore di Compatibilita'          Tipo di
Compatibilita'" << endl << endl;
        k = 0;
    }

    cout << setw(8) << numest[k] << setw(25) << setprecision(1) << compatibilita <<
setw(37);

    if( compatibilita < 1 ) cout << "OTTIMA!" << endl;
    else if( 1 <= compatibilita && compatibilita < 2 ) cout << "BUONA" << endl;
    else if( 2 <= compatibilita && compatibilita < 3 ) cout << "pessima" << endl;

```



```

        else if( 3 < compatibilita ) cout << "Non c'e' compatibilita'" << endl;

        k++;
    }

    cout << "\n\nVuoi rieseguire il programma? ";
    cin >> yesorno;

    do
    {
        if( yesorno != "si" && yesorno != "no" )
        {
            cout << "*** Risposta non valida *** \nVuoi rieseguire il programma? ";
            cin >> yesorno;
        }
    } while( yesorno != "si" && yesorno != "no" );
}
while( yesorno == "si" );

return 0;
}

double sommatoria( double *inizio, double *fine, int totalemisure )
{
    double sum = 0;
    double *ptr;

    for( ptr = inizio; ptr != fine; ptr++ )
        sum = sum + *ptr;

    return sum;
}

```

**** Codice del programma "mediatung.cxx" ****

```

#include <iostream>
#include <fstream>
#include <cmath>
#include <iomanip>

double media( double *, double *, int );
double sommatoriadiff( double *, double *, int );
double erroreqmedio( double *, double *, int );
double erroremedia( double *, double *, int );
double sommatoria( double *, double *, int );

int main()
{
    using namespace std;

    ifstream fin, gin;

```

```

fin.open( "etung.txt" );
gin.open( "ktung.txt" );

double e[2];
double k[2];

for( int i = 0; i < 2; i++ )
{
    fin >> e[i];
    gin >> k[i];
}

fin.close();
gin.close();

fin.open( "sigmaetung.txt" );
gin.open( "sigmaktung.txt" );

double sigmae[2];
double sigmak[2];

for( int i = 0; i < 2; i++ )
{
    fin >> sigmae[i];
    gin >> sigmak[i];
}

fin.close();
gin.close();

double meane = media( e, e + 2, 2 );
double meank = media( k, k + 2, 2 );

double errmeane = erroremedia( e, e + 2, 2 );
double errmeank = erroremedia( k, k + 2, 2 );

cout << "\n\n ** MEDIA di E - TUNGSTENO **" << endl << endl
    << "E medio: " << meane << endl
    << "errore: " << errmeane << endl
    << "\nK medio: " << meank << endl
    << "errore: " << errmeank << endl;

cout << "\n\n*** COMPATIBILITA' E ***" << endl;
for ( int i= 0; i < 2; i++ )
{
    double compatibilita = 0;

    compatibilita = fabs( e[i] - meane )/ sqrt( pow( sigmae[i], 2 ) + pow( errmeane, 2 ) );

    if( i == 0 ) cout << "\n ALLUNGAMENTO" << endl;
    else if( i == 1 ) cout << "\n ACCORCIAMENTO" << endl;

    cout << "Valore di Compatibilita'      Tipo di Compatibilita'" << endl << endl;
}

```

```

        cout << setw(12) << setprecision(1) << compatibilita << setw(37);

        if( compatibilita < 1 ) cout << "OTTIMA!" << endl;
        else if( 1 <= compatibilita && compatibilita < 2 ) cout << "BUONA" << endl;
        else if( 2 <= compatibilita && compatibilita < 3 ) cout << "pessima" << endl;
        else if( 3 < compatibilita ) cout << "Non c'e' compatibilita'" << endl;

    }

    cout << "\n\n*** COMPATIBILITA' K ***" << endl;
    for ( int i= 0; i < 2; i++ )
    {
        double compatibilita = 0;

        compatibilita = fabs( k[i] - meank )/ sqrt( pow( sigmak[i], 2 ) + pow( errmeank, 2 ) );

        if( i == 0 ) cout << "\n ALLUNGAMENTO" << endl;
        else if( i == 1 ) cout << "\n ACCORCIAMENTO" << endl;

        cout << "Valore di Compatibilita'      Tipo di Compatibilita'" << endl << endl;

        cout << setw(12) << setprecision(1) << compatibilita << setw(37);

        if( compatibilita < 1 ) cout << "OTTIMA!" << endl;
        else if( 1 <= compatibilita && compatibilita < 2 ) cout << "BUONA" << endl;
        else if( 2 <= compatibilita && compatibilita < 3 ) cout << "pessima" << endl;
        else if( 3 < compatibilita ) cout << "Non c'e' compatibilita'" << endl;

    }

    return 0;
}

double sommatoria( double *inizio, double *fine, int totalemisure )
{
    double sum = 0;
    double *ptr;

    for( ptr = inizio; ptr != fine; ptr++ )
        sum = sum + *ptr;

    return sum;
}

double media( double *inizio, double *fine, int totalemisure )
{
    double sum = 0;
    double *ptr;

    sum = sommatoria( inizio, fine, totalemisure );

    return ( sum / totalemisure );
}

```

```

double sommatoriadiff( double *inizio, double *fine, int totalemisura )
{
    double *ptr;
    double sum = 0;
    double mean = media( inizio, fine, totalemisura );

    for( ptr = inizio; ptr != fine; ptr++ )
        sum = sum + pow( ( *ptr - mean ), 2 );

    return sum;
}

double erroreqmedio( double *inizio, double *fine, int totalemisura )
{
    return sqrt( sommatoriadiff( inizio, fine, totalemisura ) / ( totalemisura - 1 ) );
}

double erroremedia( double *inizio, double *fine, int totalemisura )
{
    return erroreqmedio( inizio, fine, totalemisura ) / sqrt( totalemisura );
}

```

**** Codice del programma "mediakacc.cxx" ****

```

#include <iostream>
#include <fstream>
#include <cmath>
#include <iomanip>

double media( double *, double *, int );
double sommatoriadiff( double *, double *, int );
double erroreqmedio( double *, double *, int );
double erroremedia( double *, double *, int );
double sommatoria( double *, double *, int );

int main()
{
    using namespace std;

    char reply_numest;

    cout << "\nDi quale strumento vuoi calcolare K medio, relativo errore e compatibilità?" <<
endl
        << "\na) Estensimetro n' 7 - L = 950 +- 2 mm | S = 0.330 mm +- 1%" <<
endl
        << "b) Estensimetro n' 10 - L = 950 +- 2 mm | S = 0.406 mm +- 1%" << endl
        << "c) Estensimetro n' 11 - L = 950 +- 2 mm | S = 0.432 mm +- 1%" << endl
        << "d) Estensimetro n' 13 - L = 900 +- 2 mm | S = 0.279 mm +- 1%" << endl
        << "e) Estensimetro n' 15 - L = 700 +- 2 mm | S = 0.279 mm +- 1%" << endl
        << "f) Estensimetro n' 18 - L = 400 +- 2 mm | S = 0.279 mm +- 1%" << endl
    << endl;
}

```

```

cin >> reply_numest;

double ktutti[12];

ifstream gin( "kacc.txt" );

for( int i = 0; i < 12; i++ )
    gin >> ktutti[i];

gin.close();

double sigmaktutti[12];

gin.open( "sigmakacc.txt" );

for( int i = 0; i < 12; i++ )
    gin >> sigmaktutti[i];

gin.close();

double kparziali[2];
double sigmakparziali[2];

if( reply_numest == 'a' )
{
    for( int i = 0; i < 2; i++ )
    {
        kparziali[i] = ktutti[i];
        sigmakparziali[i] = sigmaktutti[i];
    }
}
else if( reply_numest == 'b' )
{
    for( int i = 0; i < 2; i++ )
    {
        kparziali[i] = ktutti[( i + 2 )];
        sigmakparziali[i] = sigmaktutti[( i + 2 )];
    }
}
else if( reply_numest == 'c' )
{
    for( int i = 0; i < 2; i++ )
    {
        kparziali[i] = ktutti[( i + 4 )];
        sigmakparziali[i] = sigmaktutti[( i + 4 )];
    }
}
else if( reply_numest == 'd' )
{
    for( int i = 0; i < 2; i++ )
    {
        kparziali[i] = ktutti[( i + 6 )];
        sigmakparziali[i] = sigmaktutti[( i + 6 )];
    }
}

```

```

else if( reply_numest == 'e' )
{
    for( int i = 0; i < 2; i++ )
    {
        kparziali[i] = ktutti[( i + 8 )];
        sigmakparziali[i] = sigmaktutti[( i + 8 )];
    }
}
else if( reply_numest == 'f' )
{
    for( int i = 0; i < 2; i++ )
    {
        kparziali[i] = ktutti[( i + 10 )];
        sigmakparziali[i] = sigmaktutti[( i + 10 )];
    }
}

for( int i = 0; i < 2; i++ )
    cout << kparziali[i] << "    " << sigmakparziali[i] << endl;

double meank = media( kparziali, kparziali + 2, 2 );
double errmean = erroremedia( kparziali, kparziali + 2, 2 );

cout << "\n\n ** MEDIA di K - ACCIAIO **" << endl << endl
    << "K medio: " << meank << endl
    << "errore: " << errmean << endl;

cout << "\n\n*** COMPATIBILITA' K ***" << endl;
for ( int i = 0; i < 2; i++ )
{
    double compatibilita = 0;

    compatibilita = fabs( kparziali[i] - meank )/ sqrt( pow( sigmakparziali[i], 2 ) + pow(
errmean, 2 ) );

    if( i == 0 ) cout << "\n ALLUNGAMENTO" << endl;
    else if( i == 1 ) cout << "\n ACCORCIAMENTO" << endl;

    cout << "Valore di Compatibilita'      Tipo di Compatibilita'" << endl << endl;

    cout << setw(12) << setprecision(1) << compatibilita << setw(37);

    if( compatibilita < 1 ) cout << "OTTIMA!" << endl;
    else if( 1 <= compatibilita && compatibilita < 2 ) cout << "BUONA" << endl;
    else if( 2 <= compatibilita && compatibilita < 3 ) cout << "pessima" << endl;
    else if( 3 < compatibilita ) cout << "Non c'e' compatibilita'" << endl;
}

return 0;
}

```

```

double sommatoria( double *inizio, double *fine, int totalemisura )
{
    double sum = 0;
    double *ptr;

    for( ptr = inizio; ptr != fine; ptr++ )
        sum = sum + *ptr;

    return sum;
}

double media( double *inizio, double *fine, int totalemisura )
{
    double sum = 0;
    double *ptr;

    sum = sommatoria( inizio, fine, totalemisura );

    return ( sum / totalemisura );
}

double sommatoriadiff( double *inizio, double *fine, int totalemisura )
{
    double *ptr;
    double sum = 0;
    double mean = media( inizio, fine, totalemisura );

    for( ptr = inizio; ptr != fine; ptr++ )
        sum = sum + pow( ( *ptr - mean ), 2 );

    return sum;
}

double erroreqmedio( double *inizio, double *fine, int totalemisura )
{
    return sqrt( sommatoriadiff( inizio, fine, totalemisura ) / ( totalemisura - 1 ) );
}

double erroremedia( double *inizio, double *fine, int totalemisura )
{
    return erroreqmedio( inizio, fine, totalemisura ) / sqrt( totalemisura );
}

```