



UNIVERSITÀ DEGLI STUDI DI CATANIA

DIPARTIMENTO DI FISICA E ASTRONOMIA “ETTORE MAJORANA”

CORSO DI LAUREA IN FISICA

Relazioni di Laboratorio di Fisica 3

QUATTRO RELAZIONI
BREVI

Anno Accademico 2024 – 2025

Indice

Indice	i
Sommario	ii
1 Implementazione Numerica della Formula di Bethe–Bloch	1
1.1 Il modello	1
1.2 La simulazione	2
2 Misura di Temperature con Arduino	5
2.1 L'esperimento	5
2.2 Dati	7
2.3 Conclusioni	9
3 Misura di resistenze con un multimetro digitale	10
3.1 Il multimetro	10
3.2 Resistori	11
3.3 Altre misure di resistenze	13
4 Accettanza Geometrica di un Rivelatore	16
4.1 Il sistema fisico proposto	16
4.2 Codice e generazione dei punti	18
4.3 Esempi e immagini	20
A Codice per Arduino	24
Bibliografia	25

Sommario

In QUESTO documento sono raccolte le quattro relazioni brevi da svolgere durante il corso annuale di *Laboratorio di Fisica 3* del Corso di Laurea in *Fisica* presso l’Università degli Studi di Catania.

Le esperienze sono esposte nei quattro capitoli seguenti:

1. *Implementazione numerica della formula di Bethe–Bloch.* Attraverso un codice in C che implementa numericamente la formula di Bethe–Bloch ho simulato il passaggio di una particella α a 5 MeV attraverso un sottile foglio di alluminio, realizzando un grafico che rappresenta l’energia della particella e la quantità di energia ceduta in funzione della distanza percorsa dentro il materiale.
2. *Misura di temperature con Arduino.* Attraverso l’uso di un microcontrollore Arduino, un sensore di temperatura e un semplice codice ho misurato la variazione di temperatura di una stanza in seguito all’accensione del riscaldamento. Nella relazione analizzo qualitativamente i dati raccolti ed estraprolo una possibile funzione che ne modelli l’andamento.
3. *Misura di resistenze con un multmetro digitale.* Utilizzando un multmetro digitale ho effettuato la misura dei resistori forniti dal kit del multmetro, verificandone la distribuzione statistica. A partire dai risultati di questo studio ho confrontato le misure di alcune delle resistenze collegate in parallelo con i valori previsti teoricamente. Infine ho trovato la resistività di un anello d’argento sfruttando di nuovo una misura di resistenza e considerazioni geometriche.
4. *Accettanza geometrica di un rivelatore.* Con un altro codice in C ho applicato il metodo Montecarlo per valutare numericamente l’accettanza geometrica di un rivelatore a forma di dischetto in presenza di una sorgente di radiazioni isotropa ed estesa. Dai dati simulati ho realizzato delle immagini rappresentative del sistema e un istogramma che mostri la distribuzione della radiazione sul sensore.

Riporto inoltre i collegamenti alle mie due repository su GitHub dove è possibile consultare e scaricare i codici sorgente in C qualora si desiderasse utilizzarli.

1. <https://github.com/ImAure/bethe-bloch-simulation>
4. <https://github.com/ImAure/geometric-acceptance>

Il codice realizzato per l'esperienza con Arduino, essendo molto più breve, è invece esposto in appendice §A.

1 Implementazione Numerica della Formula di Bethe–Bloch

1.1 Il modello

Come descritto dal Particle Data Group [4], la formula di Bethe–Bloch è un modello sperimentale che descrive la perdita di energia di particelle cariche pesanti di media energia—come protoni e particelle α —nella materia:

$$\left\langle -\frac{dE}{dx} \right\rangle = K z^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \log \frac{2m_e c^2 \beta^2 \gamma^2 W_{\max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right], \quad (1.1)$$

dove β e γ sono le usuali quantità relativistiche mentre il resto dei simboli sono esplicitati in Tab. 1.1. La perdita di energia media data dalla (1.1) è misuarta in $\text{MeV g}^{-1} \text{cm}^2$ ma può essere portata in MeV cm^{-1} moltiplicando entrambi i membri per la densità volumica ρ del bersaglio misurata in g cm^{-3} . La quantità W_{\max} è la massima energia che una particella carica può cedere a un elettrone e si

Simbolo	Definizione	Valore o unità di misura
$m_e c^2$	massa a riposo dell'elettrone $\times c^2$	0.510 998 950 00(15) MeV
r_e	raggio classico dell'elettrone $e^2 / 4\pi\epsilon_0 m_e c^2$	2.817 940 322 7(19) fm
N_A	numero di Avogadro	6.022 140 857(74) $\times 10^{23} \text{ mol}^{-1}$
ρ	densità	g cm^{-3}
x	massa per unità di area	g cm^{-2}
M	massa della particella incidente	$\text{MeV } c^{-2}$
E	energia della particella incidente $\gamma M c^2$	MeV
W_{\max}	massima energia trasferibile per collisioni	MeV
z	numero di carica della particella incidente	
Z	numero atomico del bersaglio	
A	numero di massa atomica del bersaglio	
K	$4\pi N_A r_e^2 m_e c^2$	$0.307 075 \text{ MeV mol}^{-1} \text{ cm}^2$
I	energia media di eccitazione	eV
$\delta(\beta\gamma)$	correzione di ionizzazione	

Tabella 1.1: Notazione e unità di misura per la formula di Bethe–Bloch. Si tratta di un riassunto della tabella del PDG [4].

Tabella 1.2: caption!!

esprime come

$$W_{\max} = \frac{2m_e c^2 \beta^2 \gamma^2}{1 + \gamma m_e/M + (m_e/M)^2}. \quad (1.2)$$

La (1.1) e la (1.2) sono valide nell'approssimazione $0.1 \lesssim \beta\gamma \lesssim 1000$ poiché al limite inferiore la velocità del proiettile diventa confrontabile con la “velocità” degli elettroni atomici mentre al limite superiore gli effetti radiativi non sono più trascurabili.

1.2 La simulazione

Ho scelto di simulare la perdita di energia di una particella α a 5 MeV attraverso un foglio di alluminio di spessore¹ 0.022 mm. Per realizzare la simulazione ho scritto un codice in C che implementa la (1.1) in modo numerico.

1.2.1 Descrizione del codice e dei calcoli

L'utente che lancia il programma sceglie lo spessore del materiale bersaglio e il numero di step in cui suddividere il calcolo. Il programma quindi chiede all'utente di selezionare un proiettile tra α , protoni o muoni e un materiale tra alluminio e rame. I dati come le masse delle particelle, le densità dei materiali e i valori di I , sono definiti come costanti all'inizio del programma. Non li riporto in questo elaborato per brevità—la totalità del codice, costanti incluse, è consultabile sul mio profilo GitHub.²

Assumiamo che in ciascun intervallo di spessore³ ds tutte le quantità variabili di nostro interesse siano costanti—velocità, energia, perdita di energia *et cetera*. Dalla teoria della relatività ristretta scriviamo per l' n -esimo intervallo

$$\begin{aligned} E_n &= T_n + Mc^2, \\ E_n &= \gamma_n Mc^2, \end{aligned}$$

dove M è la massa a riposo della particella incidente, T_n la sua energia cinetica ed E_n la sua energia totale. Note queste ultime due quantità, è possibile calcolare il fattore di Lorentz γ_n e β_n^2 come

$$\gamma_n = \frac{T_n + Mc^2}{Mc^2}, \quad \beta_n^2 = 1 - \frac{1}{\gamma_n^2}.$$

¹Dopo aver provato con uno spessore di 0.015 mm e aver constatato che la curva risultava tagliata—la particella non cedeva tutta l'energia all'alluminio—ho scelto di aumentare di poco lo spessore per il gusto di un grafico più completo.

²Repository: <https://github.com/ImAure/bethe-bloch-simulation>

³Ricordo che in questo caso la x non indica una distanza.

Noti γ_n e β_n , è possibile calcolare la perdita di energia media per unità di lunghezza attraverso la (1.1) moltiplicata per ρ . Possiamo in particolare calcolare l'energia cinetica con cui la particella α entra nello strato successivo, T_{n+1} come

$$T_{n+1} = T_n + dT_n = T_n + \rho \left\langle -\frac{dE}{dx} \right\rangle_n ds ,$$

essendo naturalmente $dT_n \leq 0$. Il calcolo di $\rho |\langle -dE / dx \rangle_n|$ è svolto dalla funzione `bethe()`, con l'ausilio della funzione `wmax()` che in particolare calcola il termine W_{\max} :

```

1 double wmax(double beta2, double gamma, double gamma2, double m, double
2   m2) {
3   return (2.0 * E_M * beta2 * gamma2) / (1.0 + (2.0 * gamma * m) + m2
4   );
5 }
6 double bethe(double z_inc, double z_tar, double a_tar, double beta2,
7   double gamma2, double w_max, double i2) {
8   return (((K * z_inc * z_inc * z_tar) / (a_tar * beta2)) * (0.5 *
9     log((2.0 * E_M * beta2 * gamma2 * w_max) / (i2 * 1.0e-12)) - beta2))
10  ;
11 }
```

Il calcolo viene quindi ripetuto a partire dalla nuova energia T_{n+1} per ottenere la perdita di energia attraverso il successivo strato. Per ogni reiterazione il codice stampa su un file la distanza totale percorsa, l'energia cinetica della particella e la quantità $|dT_n| = \rho |\langle -dE / dx \rangle_n|$.

1.2.2 Risultati della simulazione

Osserviamo adesso i dati che si ottengono inserendo nel programma un numero arbitrariamente alto di step pari a 500. In Fig. 1.1 sono riportati in due grafici i valori dell'energia cinetica della particella α e del potere d'arresto del materiale al variare della distanza percorsa.

Dal primo grafico risulta evidente la decrescita dell'energia della particella. Si nota che l'energia cinetica non si annulla del tutto, nonostante si avvicini ragionevolmente a 0 MeV: questo può essere dovuto a imprecisioni nel codice come il modo in cui vengono gestiti valori di T negativi e valori di dT che farebbero aumentare l'energia.

Nel secondo grafico invece è riportato l'andamento del potere d'arresto, detto comunemente *curva di Bragg*. L'energia depositata dalla particella α è inversamente proporzionale al quadrato della velocità, per questo subito prima del totale arresto si osserva il massimo deposito di energia nel tratto di grafico detto *picco di Bragg*.

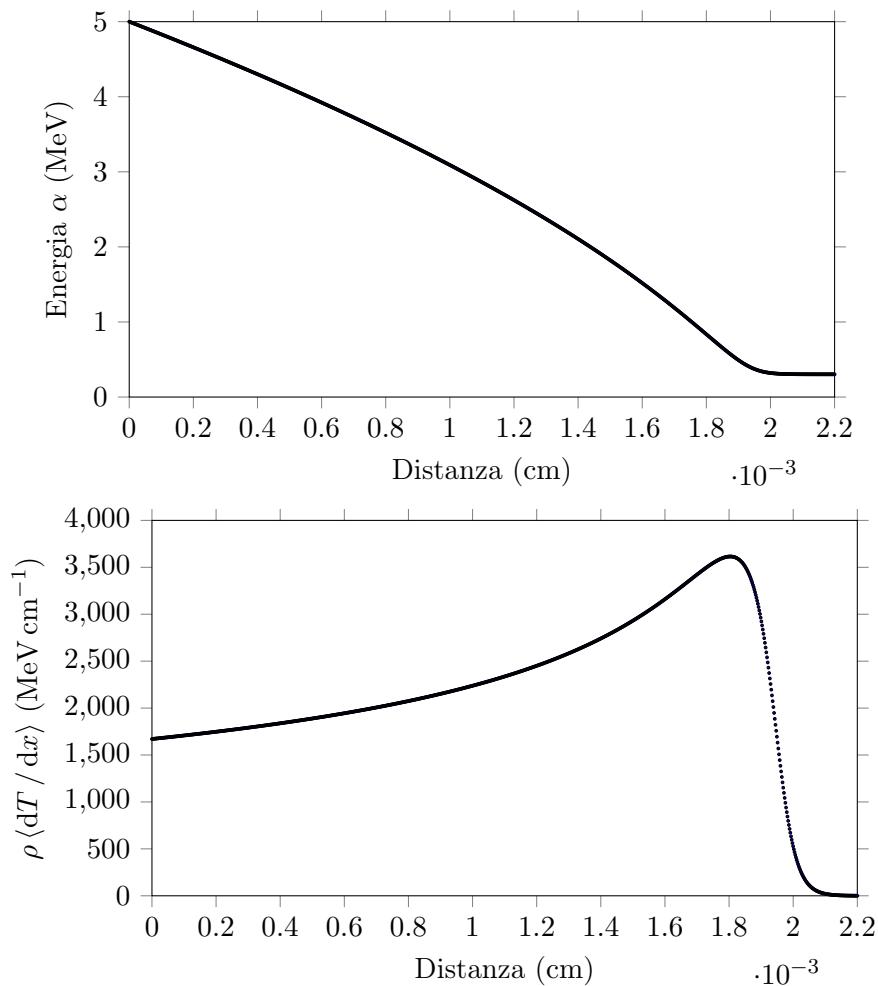


Figura 1.1: In alto l'energia della particella α che diminuisce man mano che la particella penetra l'alluminio. In basso il potere d'arresto con l'evidente picco di Bragg.

2 Misura di Temperature con Arduino

Tra le esperienze svolte con Arduino Uno riporto in particolare la misura della variazione della temperatura della mia stanza da letto in seguito all'accensione del riscaldamento in casa.

2.1 L'esperimento

L'obiettivo dell'esperienza è quello di valutare qualitativamente l'andamento della temperatura della stanza per fare una stima di quanto velocemente si riscaldi e a quale temperatura tenda asintoticamente.

2.1.1 Preparazione della stanza

Per massimizzare l'escursione termica ho effettuato la misura durante una sera invernale avendo preventivamente aperto le finestre per abbassare la temperatura della stanza.

Per migliorare la circolazione dell'aria ed evitare un eccessivo gradiente di temperatura—il radiatore caldo si trova in un angolo della stanza mentre i vetri freddi della finestra si trovano dal lato opposto—ho acceso dei ventilatori: uno a soffitto per limitare la raccolta dell'aria calda in alto e un più piccolo ventilatore da tavolo per allontanare l'aria calda dal radiatore e facilitare il riscaldamento dell'aria fredda.

Infine, per isolare il più possibile il sistema, ho chiuso le tende sulla finestra per ridurre la dispersione di calore attraverso il vetro freddo e mantenuto la porta chiusa per non disperdere calore nel resto della casa.

2.1.2 Strumenti utilizzati

Gli strumenti utilizzati per la presa dei dati sono:

- Una microcontrollore Arduino Uno con un sensore di temperatura TMP36;
- Un computer per compilare ed eseguire il codice sulla scheda Arduino e prelevare i dati.

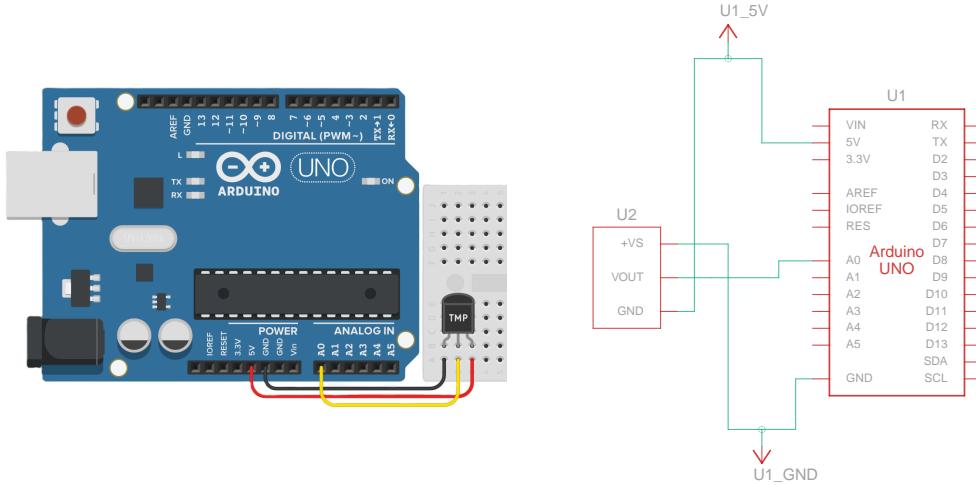


Figura 2.1: A sinistra una rappresentazione digitale del circuito realizzato per l'esperimento. A destra lo schema del circuito. Entrambe le illustrazioni sono state realizzate con Tinkercad®.

Il sensore TMP36 è un sensore di temperatura a semiconduttore pensato per operare in un range di temperature che va da -40°C a $+125^{\circ}\text{C}$. Esso presenta tre pin: $+vs$, $vout$ e gnd . Il primo e l'ultimo servono per l'alimentazione che deve essere compresa tra 2.7 V e 5.5 V con una corrente inferiore ai $50\ \mu\text{A}$, che garantisce un surriscaldamento per effetto Joule trascurabile. Il secondo pin invece sostituisce una differenza di potenziale rispetto al gnd proporzionale alla temperatura misurata. La sensibilità del sensore fornita dal costruttore è di $\pm 1^{\circ}\text{C}$ e il suo fattore di scala è di $10\text{ mV}^{\circ}\text{C}^{-1}$ [1].

La scheda Arduino attraverso i pin analogici accetta in input delle differenze di potenziale che vanno da 0 V a 5 V che vengono convertite in un segnale digitale che assume valori discreti da 0 a 1023.

2.1.3 Circuito e codice

Il circuito realizzato per l'esperimento è quello rappresentato in Fig. 2.1. L'alimentazione al sensore è fornita tramite i pin 5v e gnd mentre il segnale in uscita dal sensore viene letto dal pin A0 della scheda.

Per effettuare le misure ho usato il codice riportato in appendice A. A intervalli di 30 s la lettura discreta di tensione data dal sensore¹ e la converte in un numero decimale tra 0 V e 5 V attraverso la formula

$$\frac{(\text{float})\text{analogRead(SENSOR_PIN)}}{\text{MAX_READ}} * \text{MAX_V},$$

¹Come detto prima si tratta di un valore tra 0 e 1023

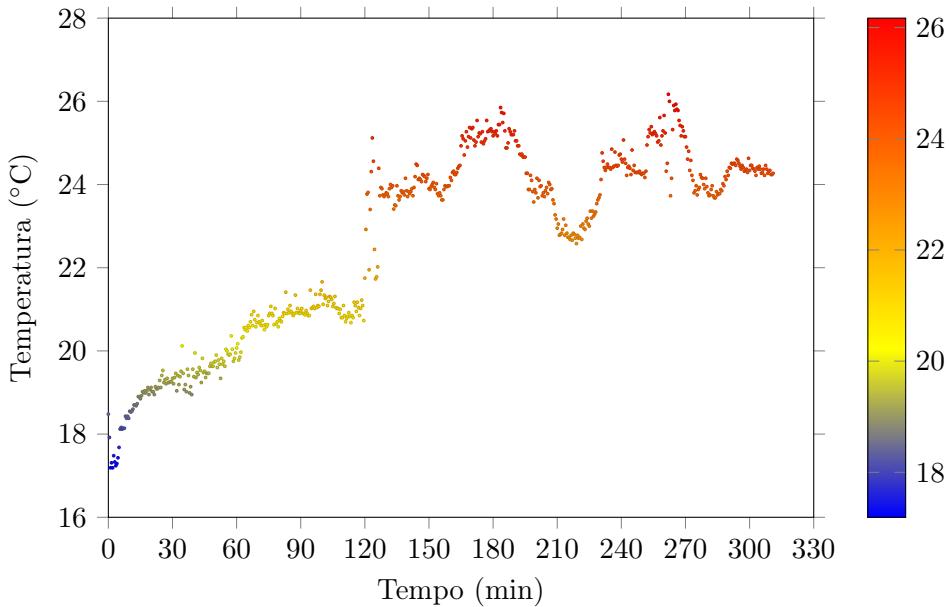


Figura 2.2: Andamento della temperatura nel tempo

essendo $\text{MAX_READ} = 1023$ e $\text{MAX_V} = 5 \text{ V}$. Sapendo che a una tensione di 0 V corrisponde una temperatura di 0.5°C e a 4.5 V corrispondono 100°C , la conversione della lettura in gradi Celsius è data da

$$\left[\frac{(\text{float})\text{analogRead}(\text{SENSOR_PIN})}{\text{MAX_READ}} * \text{MAX_V} - A \right] * B, \quad (2.1)$$

dove $A = 0.5 \text{ V}$ e $B = 100^\circ\text{C} \text{ V}^{-1}$ sono i fattori di scala.

La conversione dei valori discreti in temperatura è eseguita dal codice tra le righe 20 e 23 applicando la (2.1). Vengono eseguite $N = 20$ misure consecutive di cui è contestualmente calcolata la media che viene a sua volta stampata a schermo. Infine il codice attende il tempo mancante per raggiungere i 30 s dall'inizio del loop.

2.2 Dati

Attraverso il codice riportato al punto §2.1.3 ho misurato la temperatura della stanza ogni 30 s per circa cinque ore e mezza; in Fig. 2.2 sono riportati tutti i dati raccolti, avendo convertito il tempo in minuti.

2.2.1 Prime considerazioni

Osservando il grafico si nota una evidente crescita di temperatura che, dopo una crescita regolare, oscilla fino a stabilizzarsi poco sopra i 24°C .

I primi punti a temperatura più elevata possono essere dovuti a un precedente contatto con le mani e un riscaldamento del sensore che è poi tornato a temperatura

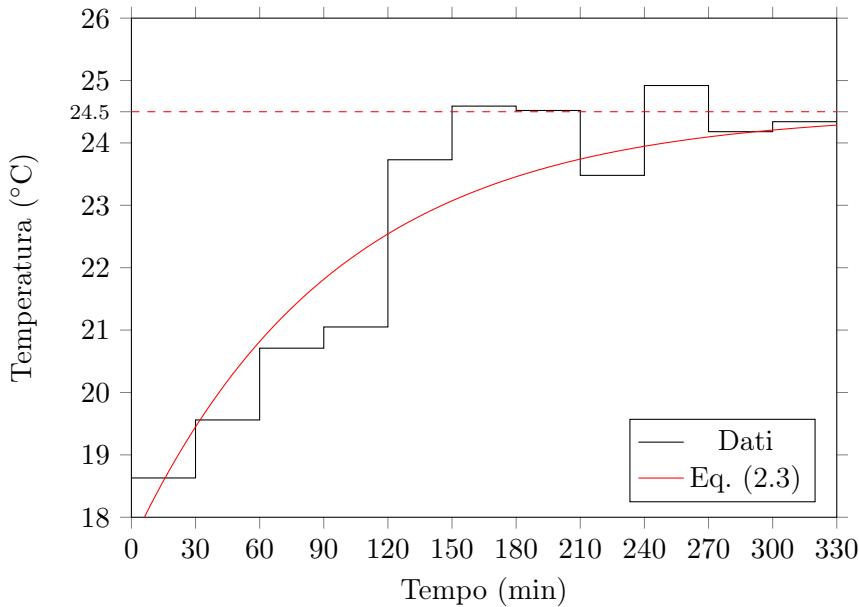


Figura 2.3: Temperature mediate in intervalli di 30 min.

ambiente. L'ampia oscillazione dei dati intorno ai 130 min deve essere dovuta a un movimento del sensore, che ho dovuto spostare di qualche centimetro. È quindi possibile che anche le successive oscillazioni siano dovute alla nuova posizione del sensore in un punto con un flusso d'aria più dinamico.

2.2.2 Breve analisi dei dati

Per visualizzare il macro-andamento della temperatura tamponando il rumore, ho deciso di suddividere le misure in intervalli temporali di 30 min e riportare nel grafico in Fig. 2.3 la temperatura media per ciascun intervallo.

Anche in questo caso si notano l'andamento crescente della temperatura e il salto, ma le oscillazioni intorno alla temperatura finale di circa 24 °C risultano smorzate. Si osserva inoltre che la temperatura si stabilizza intorno a questo valore 150 min—oppure 2 h 30 min—dopo l'accensione del riscaldamento.

Svolgiamo adesso un fit dei dati di carattere prettamente qualitativo. Dal momento che il radiatore a regime avrà una temperatura T^* fissata e tenendo conto della dispersione del calore verso l'esterno, è ragionevole assumere che la temperatura della stanza $T(t)$ tenda asintoticamente a un valore finito $T_f \leq T^*$ per $t \rightarrow +\infty$. Una funzione crescente che ha un comportamento simile è

$$T(t) = T_f \left[1 - \exp\left(-\frac{t-t_0}{\tau}\right) \right], \quad (2.2)$$

per qualche valore di T_f , t_0 e τ . Supponiamo arbitrariamente che la temperatura

limite² sia $T_f = 24.5^\circ\text{C}$; per quanto detto prima supponiamo inoltre che il tempo caratteristico³ sia $\tau = 95 \text{ min} \approx 63\%$ di 150 min. Imponendo che sia $T(0) = 18.63^\circ\text{C}$ troviamo

$$t_0 = \tau \log \left[1 - \frac{T(0)}{T_f} \right] \approx -135 \text{ min}.$$

Infine per centrare meglio la funzione rispetto agli intervalli di 30 min sommiamo a t_0 un valore di 15 min, pari a metà dell'intervallo di tempo. La funzione finale che si ottiene sostituendo questi valori nella (2.2) e si trova rappresentata in Fig. 2.3 è

$$T(t) = 24.5 \left[1 - \exp \left(-\frac{t + 120}{95} \right) \right]. \quad (2.3)$$

2.3 Conclusioni

I dati presi possono essere migliorati facendo maggiore attenzione a non toccare gli strumenti o uscendo dalla stanza e assicurandosi che la porta non venga mai aperta.

Per avere informazioni più dettagliate sulla bontà del fit si potrebbe invece procedere applicando il metodo dei minimi quadrati per determinare i tre parametri T_f , t_0 e τ . Per eseguire un fit con due parametri invece che tre, si può migliorare l'accuratezza su T_f continuando a prendere dati il più a lungo possibile. Questo permetterebbe di fissare la temperatura limite con più sicurezza e di determinare t_0 e τ attraverso la linearizzazione della (2.2) in

$$\log \left[1 - \frac{T(t)}{T_f} \right] = \frac{t_0}{\tau} - \frac{1}{\tau} t \iff y(t) = c_1(t_0, \tau) + c_2(t_0, \tau)t.$$

Si potrebbe inoltre provare a utilizzare funzioni diverse dalla (2.2) nel fit per provare a modellare quantitativamente le oscillazioni che si verificano in prossimità della temperatura limite, possibilmente dovute a moti convettivi non del tutto smorzati dai ventilatori indicati al punto §2.1.1.

²I dati in Fig. 2.2 oscillando superano i 25°C ma le ultime misure sono tutte comprese tra 24°C e 24.5°C , motivo per cui assumo quest'ultimo valore come temperatura limite.

³Dal momento che “a occhio” la temperatura di 24.5°C viene già raggiunta dopo 150 min, scegliamo il tempo caratteristico come il tempo necessario a raggiungere il 63 %, ovvero $(1/e)$ -esimo della temperatura finale.

3 Misura di resistenze con un multmetro digitale

TRA LE ESPERIENZE svolte con il multmetro digitale riporto la misura delle resistenze di vari oggetti, tra cui un anello e alcuni resistori, misurati sia individualmente che in parallelo.

Ai resistori dedico una sezione più approfondita in quanto ho preso 50 misure su resistori distinti—ma teoricamente con resistenza uguale—per verificare la distribuzione delle misure.

3.1 Il multmetro

Lo strumento utilizzato per l'interezza dell'esperienza è un multmetro digitale della serie DVM841 della Velleman® [6], Fig. 3.1. Il multmetro è in grado di misurare tensione e corrente continua e alternata, resistenza, frequenza e temperatura. Avendo una risoluzione di 2000 punti, il display del multmetro può visualizzare un massimo di 1999 unità.

L'apparecchio è alimentato a batteria e presenta tre prese a cui si possono collegare due puntali con gli appositi spinotti. Volendo misurare solo resistenze, ho usato solo la presa $V\Omega$ mA e la messa a terra.



Figura 3.1: Il multmetro DVM841 della Velleman® usato per l'esperienza.

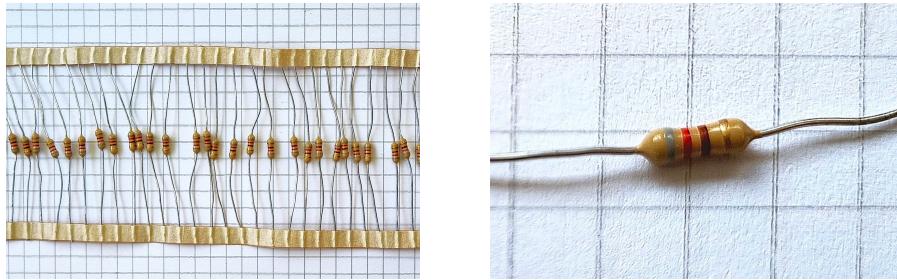


Figura 3.2: A sinistra alcuni dei 50 resistori da 820Ω . A destra un dettaglio dove è visibile il codice colore.

Resistenze (Ω)									
797	806	806	807	807	807	807	807	807	807
807	808	808	808	808	808	808	808	808	808
808	808	808	809	809	809	809	809	809	809
809	809	809	809	809	809	810	810	810	810
810	810	810	811	811	812	812	812	812	813

Tabella 3.1: Misure di resistenza effettuate su 50 resistori distinti.

3.2 Resistori

Il kit presenta $N = 50$ resistori distinti—come quelli in Fig. 3.2—il cui codice colore restituisce un valore¹ teorico di $820\Omega \pm 5\%$, ovvero $820(40)\Omega$.

Ho effettuato le misure impostando il multmetro in modalità *ohm*, alla portata di $2\text{k}\Omega$, poggiando i puntali sui terminali di ciascun resistore e aspettando di volta in volta che la lettura si stabilizzasse. I dati raccolti sono riportati in ordine crescente in Tab. 3.1.

3.2.1 Considerazioni preliminari

Notiamo subito che la resistenza media è $R_m = 808.6\Omega$ con una deviazione standard di $\sigma = 2.3\Omega$; l'errore sul valor medio è quindi $\sigma_R = \sigma/\sqrt{N-1} = 0.33\Omega$, che è confrontabile con la sensibilità dello strumento $\delta R_s = 1\Omega$.

Questi valori rientrano completamente nell'intervallo fornito dal costruttore; tuttavia, il fatto che tutte le misure siano inferiori a 820Ω suggerisce la presenza di un errore sistematico.²

¹Lo si può dedurre da qualunque legenda fedele allo standard IEC 60062.

²Se si trattasse di errori casuali dovuti a imprecisioni di fabbricazione, mi aspetterei letture sia al di sopra che al di sotto del valore di riferimento; è poco probabile che tutte le resistenze devino dal valore teorico allo stesso modo a meno che non si sia verificato un evento che ha alterato tutte le resistenze—un lotto prodotto con lo stesso materiale meno resistente, seppur entro il margine del 5 %, o deterioramento nel tempo.

Intervalli			O_k	E_k
	$R <$	806.5	2	4.133
806.5	$< R <$	807.5	8	5.735
807.5	$< R <$	808.5	12	6.925
808.5	$< R <$	809.5	13	7.278
809.5	$< R <$	810.5	6	6.655
810.5	$< R <$	811.5	2	5.297
811.5	$< R <$	812.5	4	3.668
812.5	$< R$		1	2.211

Tabella 3.2: Suddivisione dei dati per il test del χ^2 . Ometto le unità di misura per chiarezza espositiva e semplicità dei calcoli.

Il dato di resistenza minima di 797Ω può essere scartato secondo il criterio di Chauvenet. Esso dista più di 4σ dal valor medio (ca. 4.17σ) e il numero di dati atteso³ su un campione di $N = 50$ elementi a una distanza maggiore o uguale a 4σ è pari a $0.003 \ll 1/2$. Scartando questo dato la nuova media e la nuova deviazione standard sono:

$$R_m = 808.8\Omega \quad \sigma = 1.633\Omega.$$

La nuova incertezza sul valor medio è $\sigma_R = 0.24\Omega \lesssim \delta R_s = 1\Omega$, che è ancora confrontabile con la sensibilità dello strumento. Se tuttavia consideriamo la somma in quadratura delle due incertezze troviamo

$$\bar{\sigma} = \sqrt{\sigma_R^2 + \delta R_s^2} = 1.02\Omega \simeq 1\Omega,$$

per cui assumo $\bar{\sigma} = 1\Omega$ come incertezza sul valor medio.

3.2.2 Test del χ^2

Supponiamo che le misure seguano, con una significatività $\alpha = 0.05$, la distribuzione normale centrata in R_m e di ampiezza σ :

$$N(x; R_m, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{x - R_m}{\sigma}\right)^2\right].$$

Costruiamo quindi un istogramma dei dati. Visto l'intervallo contenuto in cui le misure variano, ho scelto di raccogliere i dati in bin di ampiezza 1Ω , uno per ciascun valore misurato; ciascun bin si estende da mezza unità *prima* del valore di interesse a mezza unità *dopo*. In Tab. 3.2 sono riportati i bin e le frequenze osservate O_k .

³Per il calcolo di questa probabilità ho fatto riferimento alla tabella fornita in Taylor, J. R. *Introduzione all'analisi degli errori* [5].

n	R_o	δR_s	R_e	δR_e
1	815	1	808.8	1
2	410	1	404.4	0.5
3	272	1	269.6	0.3

Tabella 3.3: Resistenze equivalenti misurate su resistori in parallelo e relative incertezze. Tutti i valori sono espressi in Ω .

Nella stessa tabella sono riportati i valori attesi E_k per ciascun bin, calcolati moltiplicando la dimensione del campione $N = 49$ per l'integrale di ciascun intervallo della gaussiana. Ho ottenuto gli intervalli convertendo gli estremi in variabili normali standardizzate e ho ricavato l'integrale attraverso un foglio di calcolo.

È adesso possibile calcolare il χ^2 per definizione:

$$\chi^2 = \sum_{k=1}^8 \frac{(O_k - E_k)^2}{E_k} = 12.98 .$$

essendo $d = n - c = 6$ il numero di gradi di libertà, $n = 8$ il numero di bin e $c = 2$ il numero di parametri stimati—media e deviazione standard. Il valore critico per il test del χ^2 è $\chi^2_{\text{crit}} = 12.59$ [3], posso quindi rigettare l'ipotesi nulla che le misure seguano la distribuzione gaussiana.

3.3 Altre misure di resistenze

Riporto altre misure eseguite con il multmetro su vari materiali.

3.3.1 Resistenze in parallelo

Usando $n = 1, 2, 3$ resistori scelti casualmente tra quelli studiati al punto §3.2, ho misurato la resistenza equivalente R_o dei resistori montati in parallelo su una breadboard come in Fig. 3.3. I risultati sono riportati in Tab. 3.3, insieme ai valori teorici, calcolati a partire dalla R_m del punto precedente come

$$\frac{1}{R_e} = \frac{n}{R_m} \iff R_e = \frac{R_m}{n} ,$$

e alle incertezze δR_s e δR_e . La prima è ancora la sensibilità dello strumento, mentre la seconda è calcolata propagando l'incertezza $\bar{\sigma}$ sul valor medio R_m trovata al punto §3.2:

$$\delta R_e = \left| \frac{\partial R_e}{\partial R} \right| \bar{\sigma} = \frac{\bar{\sigma}}{n} ,$$

Osservando i dati si nota che la resistenza equivalente osservata R_o è sempre maggiore della resistenza teorica R_e . Questo può essere dovuto alla presenza di una resistenza di contatto con la breadboard, che non ho considerato nel calcolo di R_e .

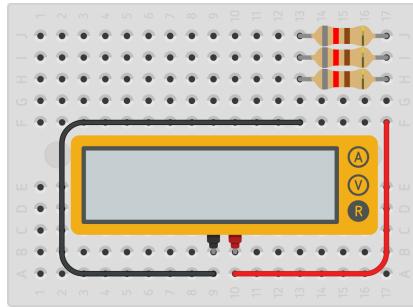


Figura 3.3: Illustrazione rappresentativa dei resistori montati in parallelo su una breadboard.



Figura 3.4: Fotografia dell'anello. I tratti in rosso indicano il punti in cui sono stati poggiate i puntali del multmetro.

3.3.2 Resistività di un anello

Ho misurato la resistenza di un anello metallico che suppongo essere in argento. L'anello è di forma circolare ma presenta un'apertura in basso, per cui ho effettuato la misura della resistenza poggiando i puntali del multmetro in corrispondenza dei punti estremi dell'apertura, come in Fig. 3.4. Ho posto il multmetro alla portata di 200Ω disponendo di una sensibilità $\delta R=0.1\Omega$; la resistenza misurata è quindi $R = 0.5\Omega$.

Dalla seconda legge di Ohm possiamo ricavare la resistività del materiale che compone l'anello, supponendo che esso sia omogeneo e isotropo. La resistività è definita come

$$\rho = R \frac{S}{L}, \quad (3.1)$$

dove S è la sezione dell'anello e L la lunghezza equivalente ottenuta rettificando la circonferenza.

L'anello ha una raggio medio di $r = 2.1(1)\text{ cm}$, uno spessore di $d = 0.3(1)\text{ cm}$ e un'altezza pari a $h = 0.8(1)\text{ cm}$. La sezione dell'anello è quindi $S = hd =$

$0.24(11) \text{ cm}^2$, mentre, denotando con $\ell = 1.0(1) \text{ cm}$ la distanza che separa i due puntali e che quindi non partecipa alla resistenza, la lunghezza equivalente è $L = 2\pi r - \ell = 12.3(7) \text{ cm}$.

Dal momento che l'anello è formato da fili cilindrici intrecciati, la sezione S rappresenta in realtà una sovrastima della sezione effettiva. Il rapporto tra la superficie di un cerchio di raggio $d/2$ e quella di un quadrato di lato d è pari a $\pi/4$, per cui la sezione effettiva è $S_{\text{eff}} = S\pi/4 = 0.19(8) \text{ cm}^2$.

La resistività dell'anello calcolata dalla (3.1) è quindi

$$\rho = R \frac{S_{\text{eff}}}{L} = 0.0077(55) \Omega \text{ cm} = 7.7(55) \times 10^{-5} \Omega \text{ m},$$

valore che si discosta di diversi ordini di grandezza da quello noto di $1.59 \times 10^{-8} \Omega \text{ m}$ [2]. Un risultato così elevato può essere attribuito sia all'incertezza sulle misure geometriche, sia al fatto che l'anello non sia composto da argento puro, sia alla presenza di ossidazione superficiale o di cattivi contatti tra i fili intrecciati che ne aumentano la resistenza complessiva.

4 Accettanza Geometrica di un Rivelatore

L'ultima esperienza consiste nel calcolare l'accettanza geometrica di un rivelatore, definita come il rapporto tra il numero di particelle incidenti e il numero di particelle emesse, sfruttando la generazione di numeri casuali e il metodo montecarlo.

4.1 Il sistema fisico proposto

L'obiettivo che mi sono posto è stato quello di calcolare l'accettanza geometrica di un rivelatore a forma di dischetto con raggio variabile, posto in una posizione arbitraria nel semispazio positivo delle z . Allo stesso modo, la sorgente di particelle è costituita da un secondo dischetto, anch'esso con raggio variabile, centrato nell'origine.

Le ipotesi che ho fatto sono le seguenti:

1. Ciascun punto della sorgente emette in maniera isotropa, ovvero in tutte le direzioni con la stessa probabilità;
2. La probabilità di emissione da parte di un punto della sorgente è uniforme, ovvero tutti i punti hanno la stessa probabilità di emettere una particella;
3. Il rivelatore e la sorgente sono sempre paralleli tra loro e al piano Oxy , e il rivelatore è sempre posto sopra la sorgente.

Per ragioni di simmetria, la posizione del rivelatore sopra la sorgente non causa alcuna perdita di generalità. Se il rivelatore avesse altezza $z = 0$, visto lo spessore nullo sia della sorgente che del rivelatore, nessuna particella verrebbe rivelata. La sorgente inoltre emette in modo simmetrico rispetto al piano Oxy , quindi il caso $z < 0$ è equivalente al caso $z > 0$.

4.1.1 Sorgente puntiforme

Dal momento che la sorgente può emettere da un solo punto alla volta, cominciamo col risolvere il problema nel caso di una sorgente puntiforme nell'origine e un rivelatore che può essere spostato rispettando le ipotesi di cui sopra.

Affinché la distribuzione delle particelle sia uniforme, generare delle coordinate cartesiane casuali non è sufficiente. La scelta che ho fatto è quella di generare delle *direzioni* casuali in coordinate sferiche $(\varrho, \vartheta, \varphi)$, risparmiando la spesa computazionale di generare la distanza ϱ dall'origine. Per le proprietà delle coordinate sferiche possiamo generare uniformemente $\cos\vartheta \in [-1; 1]$, per poi ricavare ϑ prendendone l'arcocoseno, mentre possiamo generare uniformemente $\varphi \in [0; 2\pi]$.

Fissata una direzione (ϑ, φ) , la particella emessa dalla sorgente si muoverà lungo la retta che passa per l'origine avente tale direzione. Ci chiediamo quindi se questa retta intersecherà il rivelatore, e in caso affermativo, in quale punto.¹

Da semplici considerazioni geometriche, se indichiamo con $P \equiv (x, y, z)$ il generico punto dello spazio, con $C \equiv x_c, y_c, z_c$ il centro del rivelatore e con R il suo raggio, il punto appartiene al dischetto se si verificano contemporaneamente le seguenti condizioni:

$$(P - C)^2 \leq R^2, \quad (4.1)$$

$$z = z_c. \quad (4.2)$$

La prima condizione corrisponde a $(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 \leq R^2$, che messa a sistema con la seconda dà $(x - x_c)^2 + (y - y_c)^2 \leq R^2$. Passiamo quindi a coordinate polari per trovare le condizioni che devono rispettare i valori di ϑ e φ :

$$\begin{cases} x = \varrho \sin\vartheta \cos\varphi \\ y = \varrho \sin\vartheta \sin\varphi \\ z = \varrho \cos\vartheta \end{cases}.$$

Sostituendo queste relazioni nella condizione di appartenenza al cerchio troviamo:

$$(\varrho \sin\vartheta \cos\varphi - x_c)^2 + (\varrho \sin\vartheta \sin\varphi - y_c)^2 \leq R^2,$$

che con qualche passaggio si riduce a

$$\varrho^2 \sin^2\vartheta + x_c^2 + y_c^2 - 2\varrho \sin\vartheta (x_c \cos\varphi + y_c \sin\varphi) \leq R^2,$$

che a sua volta, a patto di porre $d^2 = x_c^2 + y_c^2$, $x_c = d \cos\varphi_c$ e $y_c = d \sin\varphi_c$, non è altro che il teorema del coseno applicato al triangolo formato dalle proiezioni sul piano Oxy dei punti O , P e C , che ha un angolo in O proprio pari a $\varphi - \varphi_c$:

$$\varrho^2 \sin^2\vartheta + d^2 - 2\varrho \sin\vartheta d \cos(\varphi - \varphi_c) \leq R^2. \quad (4.3)$$

Se assumiamo $\cos\vartheta \neq 0$ —che è vero se $P \notin Oxy$, e ciò non è restrittivo in quanto il piano è un sottoinsieme a misura nulla dello spazio \mathbb{R}^3 —possiamo invertire la (4.2) scritta in coordinate polari per esplcitare ϱ :

$$\varrho = \frac{z_c}{\cos\vartheta}. \quad (4.4)$$

¹Sapere il punto non è necessario per il calcolo dell'accettanza geometrica, ma è utile allo scopo di generare un'immagine del sistema.

Sostituendo quest'ultima relazione (4.4) nella (4.3) si ottiene

$$z_c \tan\vartheta + d^2 - 2z_c \tan\vartheta d \cos(\varphi - \varphi_c) \leq R^2. \quad (4.5)$$

I valori z_c , d e φ_c sono costanti, per cui una volta generati ϑ e φ è sufficiente verificare la disequazione per sapere se il raggio interseca o no il rivelatore. Resta quindi da trovare esplicitamente la posizione dei punti $P^* \equiv (x^*, y^*, z^*)$ che soddisfano la relazione; il problema può essere ridotto a quello dell'intersezione tra la retta individuata dalla direzione (ϑ, φ) e il piano $z = z_c$. La quantità $z_c \tan\vartheta$ è la proiezione del segmento OP^* sul piano Oxy : il suo prodotto per $\cos\varphi$ e $\sin\varphi$ ci dà rispettivamente le coordinate x^* e y^* . L'altezza z^* è banalmente quella del rivelatore z_c .

È importante notare che il problema non distingue tra “sopra” e “sotto”, per cui un raggio rivolto verso il simmetrico di C rispetto all'origine sarà ugualmente soluzione della disequazione. Per tener conto di ciò basterà dividere per 2 il conteggio totale dei punti che intersecano il rivelatore.

4.1.2 Sorgente estesa

Per estendere quanto sviluppato al punto precedente alla sorgente estesa, è sufficiente fare una semplice traslazione: generato un punto $S \equiv (x_s, y_s, 0)$ arbitrario nella sorgente circolare, è possibile calcolare la posizione relativa del centro C' che avrà quindi coordinate $C' \equiv (x_c - x_s, y_c - y_s, z_c)$. Fatto ciò possiamo generare una direzione casuale e , con gli stessi passaggi del punto precedente, trovare gli eventuali punti di intersezione nel sistema di riferimento del punto sorgente. Trovati tali punti, sarà immediato riportarli nel sistema di riferimento del centro della sorgente con la traslazione inversa. Si trova in particolare

$$\begin{cases} x^* = x_s + z_c \tan\vartheta \cos\varphi \\ y^* = x_s + z_c \tan\vartheta \sin\varphi \\ z^* = z_c \end{cases},$$

dove ϑ e φ sono ancora le direzioni generate casualmente nel sistema del punto sorgente.

4.2 Codice e generazione dei punti

Come per la simulazione dell'assorbimento di particelle cariche esposto al capitolo §1, anche per questo progetto l'interezza del codice è reperibile sul mio profilo GitHub.²

Il programma può essere lanciato scegliendo fin da subito il numero di punti da generare, le dimensioni della sorgente e del rivelatore e la posizione di quest'ultimo. Dopo aver verificato la validità dell'input e caricati i dati in memoria, il programma

²Repository: <https://github.com/ImAure/geometric-acceptance>

genera un punto casuale come sorgente in coordinate polari piane e una direzione casuale per la radiazione in coordinale polari sferiche.

```

1 int rand_polar2D(polar2D_t *ptr, double radius) {
2     if (ptr == NULL) return -1;
3     ptr->rho = (radius < 0) ? 1 : (sqrt((double)rand() / RAND_MAX) *
4         radius);
5     ptr->phi = (radius == 0) ? 0 : ((double)rand() / RAND_MAX) * (2 *
6         M_PI);
7     return 0;
8 }
9
10 int rand_polar3D(polar3D_t *ptr, double radius) {
11     if (ptr == NULL) return -1;
12     ptr->rho = (radius < 0) ? 1 : (cbrt((double)rand() / RAND_MAX) *
13         radius);
14     if (radius == 0) {
15         ptr->theta = 0;
16         ptr->phi = 0;
17     } else {
18         ptr->theta = acos(1 - 2 * ((double)rand() / RAND_MAX));
19         ptr->phi = ((double)rand() / RAND_MAX) * (2 * M_PI);
20     }
21     return 0;
22 }
```

Affinché la distribuzione sia uniforme, nel caso bidimensionale viene generato uniformemente $\rho^2 \in [0; 1]$, ne viene presa la radice quadrata e il risultato è moltiplicato per il raggio massimo. Nel caso tridimensionale la funzione fa la stessa cosa con ρ^3 e la radice terza, mentre come detto prima viene generato³ $\cos\vartheta \in [-1; 1]$ e ne viene preso l'arcocoseno per avere ϑ .

Entrambe le funzioni ammettono come input anche un raggio massimo negativo: in quel caso il raggio verrà posto uguale a 1 e verrà generata solo la direzione. Questo comportamento è sfruttato nella generazione casuale della radiazione, in cui la chiamata a funzione avviene come `rand_polar3D(&ray, -1)`.

La funzione `intercept()` verifica l'intersezione del raggio nel caso di una sorgente puntiforme posta nell'origine. Per utilizzarla correttamente viene calcolata la posizione relativa del rivelatore (x'_c, y'_c, z_c) e questa viene data in input alla funzione insieme alla direzione da verificare. La funzione valuta quindi la disequazione (4.5) e in caso di successo aggiorna il conteggio e stampa su file le coordinate del punto di partenza e del punto di arrivo—riportate di nuovo nel riferimento del centro della sorgente.

```

1 int intercept(file_t *pf, point2D_t src, polar3D_t ray, disc3D_t
2     detector3D, int *hits) {
3     point3D_t hit_point3D;
4     point2D_t dtc_point2D;
5     point3D_t dtc_point3D_hist;
```

³Per essere precisi, il coseno è generato tra 1 e -1 piuttosto che tra -1 e 1. Questo perché per $\vartheta = 0$ si ha $\cos\vartheta = 1$ e per $\vartheta = \pi$ si ha $\cos\vartheta = -1$

```

5     point2D_t dtc_point2D_hist;
6     double u;
7
8     u = detector3D.center.cart.z * tan(ray.theta);
9     if (cart_to_polar2D(&detector3D.center.cart, &dtc_point2D.polar))
10    return -1;
11
12    if ((u * u + dtc_point2D.polar.rho * dtc_point2D.polar.rho - 2 * u
13    * dtc_point2D.polar.rho * cos(ray.phi - dtc_point2D.polar.phi)) <=
14    detector3D.radius * detector3D.radius) {
15        hit_point3D.cart.x = src.cart.x + u * cos(ray.phi);
16        hit_point3D.cart.y = src.cart.y + u * sin(ray.phi);
17        hit_point3D.cart.z = detector3D.center.cart.z;
18
19        dtc_point3D_hist.cart.x = hit_point3D.cart.x - (detector3D.
20        center.cart.x + src.cart.x);
21        dtc_point3D_hist.cart.y = hit_point3D.cart.y - (detector3D.
22        center.cart.y + src.cart.y);
23
24        cart_to_polar2D(&dtc_point3D_hist.cart, &dtc_point2D_hist.polar
25    );
26
27        fprintf(pf, (*hits) ? ("\n%f %f %f %f %f %f %f %f") : ("%f %f %
28        f %f %f %f %f"), src.cart.x, src.cart.y, 0.0, hit_point3D.cart.x,
29        hit_point3D.cart.y, hit_point3D.cart.z, dtc_point2D_hist.polar.rho,
30        dtc_point2D_hist.polar.phi);
31        ++(*hits);
32    }
33    return 0;
34}

```

Sullo stesso file, la funzione `intercept()` si occupa anche stampare la posizione dei punti di intersezione del rivelatore convertendoli in coordinate polari nel sistema di riferimento centrato nel rivelatore stesso. Questi dati possono essere usati per costruire un istogramma e apprezzare la non uniformità della distribuzione della radiazione sul sensore.

L'accettanza geometrica del rivelatore è infine calcolata dividendo il numero di raggi che intersecano il rivelatore per il numero totale di raggi generati. Come giustificato alla fine del punto §4.1.1, questo numero è a sua volta diviso per 2 per tenere conto solo dei raggi che viaggiano nel semispazio $z > 0$. Il programma quindi stampa a schermo l'accettanza geometrica del rivelatore e l'equivalente angolo solido che sottende.

4.3 Esempi e immagini

Riporto qui degli esempi di esecuzione della simulazione. Il nome dell'eseguibile è `sim`, mentre gli argomenti che compaiono sono in ordine il numero di punti da generare, il raggio della sorgente, le coordinate (x, y, z) del centro del rivelato-

re e la stringa da usare per i nomi dei file in cui salvare tutti i punti generati, `<string>.1.txt`, e i punti da graficare, `<string>.2.txt`.

Le immagini sono generate tramite uno script python che estrae i numeri dal secondo file. La prima parte del programma rappresenta i punti di partenza dei raggi in rosso e i punti di arrivo in verde, collegandoli con una riga grigia; la seconda parte del programma crea un istogramma in tre dimensioni i cui bin sono settori di corone circolari. I bin sono generati in modo da avere area costante, per questo quelli più esterni sono più sottili. L'altezza delle barre invece è la densità di frequenza ottenuta dividendo la frequenza relativa per l'area del bin.

Sorgente piccola e rivelatore allineato

```
$ ./sim 100000 0.1 0 0 1 0.5 dati
Hits: 10445/100000
Ratio: 0.0522
Solid angle: 0.6563 (sr) or 0.0522*4pi (sr)
```

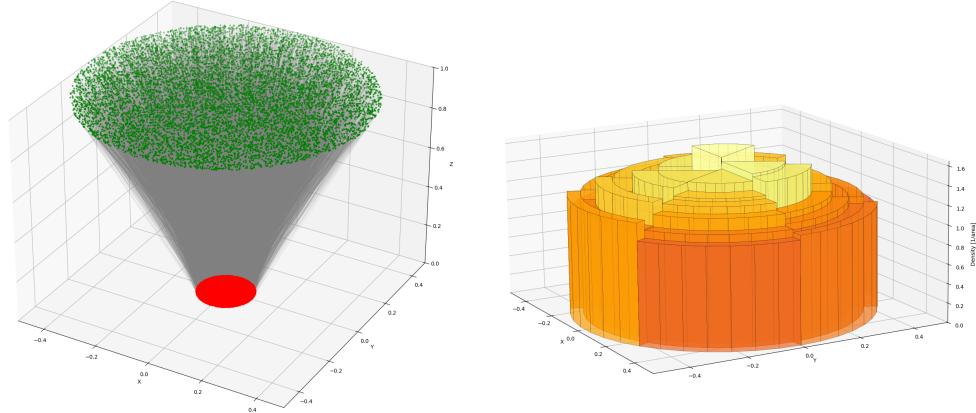


Figura 4.1: Notiamo dall'istogramma che la distribuzione è maggiormente concentrata al centro.

Sorgente puntiforme e rivelatore traslato

```
$ ./sim 100000 0 1 0 0.7 0.4 dati
Hits: 3168/100000
Ratio: 0.0158
Solid angle: 0.1991 (sr) or 0.0158*4pi (sr)
```

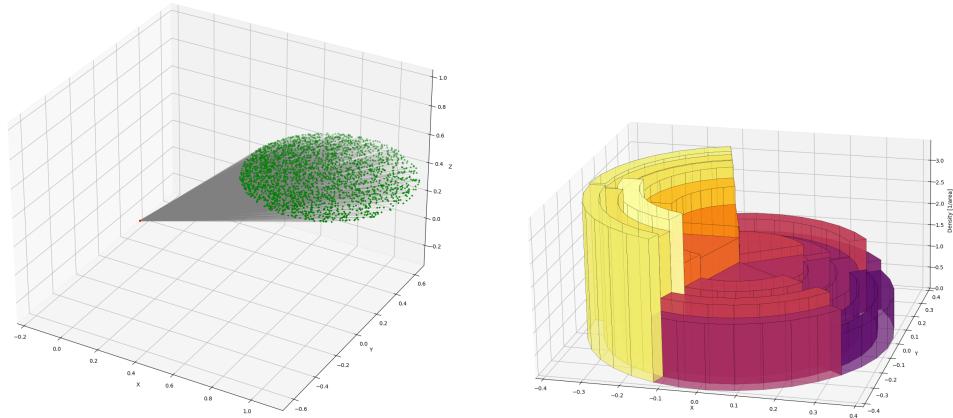


Figura 4.2: In questo caso osserviamo come la distribuzione sia particolarmente concentrata sul lato del rivelatore più vicino alla sorgente

Sorgente e rivelatore confrontabili

```
$ ./sim 10000 0.5 0 0 0.5 0.5 dati
Hits: 2383/10000
Ratio: 0.1192
Solid angle: 1.4973 (sr) or 0.1192*4pi (sr)
```

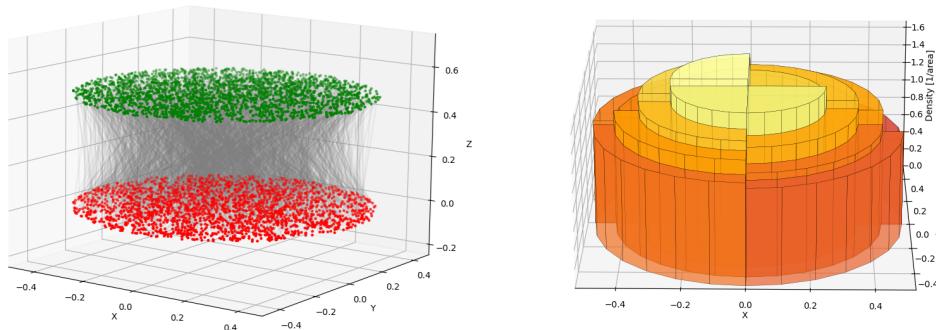


Figura 4.3: Anche qui come nel primo caso la distribuzione dei raggi è più densa al centro e qualitativamente a simmetria cilindrica.

Sorgente grande e rivlatore vicino

```
$ ./sim 50000 1 0 0 0.2 0.4 dati
Hits: 6319/50000
Ratio: 0.0632
Solid angle: 0.7941 (sr) or 0.0632*4pi (sr)
```

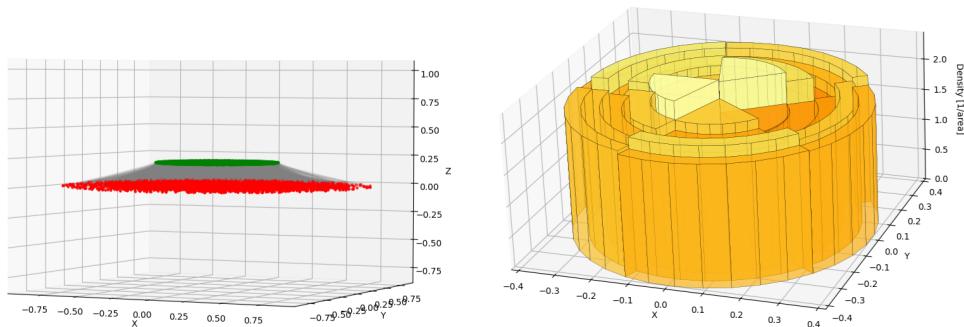


Figura 4.4: Ponendo un rivelatore piccolo molto vicino alla sorgente la distribuzione è in prima approssimazione uniforme.

A Codice per Arduino

Riporto in questa appendice il codice utilizzato per l'esperienza descritta al Capitolo §2.

```
1 #define SENSOR_PIN A0
2 #define MAX_READ    1023.0
3 #define MAX_V       5.0
4 #define N          20
5 #define A          0.5
6 #define B          100
7
8 void setup() {
9     pinMode(SENSOR_PIN, INPUT);
10    Serial.begin(9600);
11 }
12
13 float      tmp;
14 int        i = 0;
15 unsigned long dt = 0;
16
17 void loop() {
18     Serial.print(dt = millis());
19
20     for (i = 0, tmp = 0; i < N; i++) {
21         tmp += (((float)analogRead(SENSOR_PIN) / MAX_READ) * MAX_V -
22 A) * B;
23     }
24     tmp = tmp / N;
25
26     Serial.print(",");
27     Serial.println(tmp);
28
29     dt = millis() - dt;
30     delay(30000 - dt);
31 }
```

Bibliografia

- [1] Analog Devices. *TMP35/TMP36/TMP37 Data Sheet*. Analog Devices. 2015. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/tmp35_36_37.pdf.
- [2] David J Griffiths. *Introduction to Electrodynamics*. 4^a ed. Upper Saddle River, NJ: Pearson, 2012.
- [3] National Institute of Standards and Technology (NIST). *Critical Values of the Chi-Square Distribution*. NIST. URL: <https://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm>.
- [4] S. Navas et al. «Review of Particle Physics». In: *Phys. Rev. D* 110.3 (2024), p. 030001. DOI: [10.1103/PhysRevD.110.030001](https://doi.org/10.1103/PhysRevD.110.030001). URL: <https://link.aps.org/doi/10.1103/PhysRevD.110.030001>.
- [5] John Robert Taylor. *Introduzione all'analisi degli errori. Lo studio delle incertezze nelle misure fisiche*. it. 1999.
- [6] Velleman. *DVM841 Digital Multimeter*. Velleman. 2021. URL: <https://cdn.velleman.eu/downloads/1/dvm841a6v04.pdf>.