

# Effetti sulla resa energetica di un collettore lineare di Fresnel applicato a un collettore solare sottovuoto

Apolloni Davide e Leonardo Tognola

20 Febbraio 2015

# Indice

|          |                                                                 |           |
|----------|-----------------------------------------------------------------|-----------|
| <b>1</b> | <b>Premessa</b>                                                 | <b>3</b>  |
| <b>2</b> | <b>Estratto</b>                                                 | <b>4</b>  |
| <b>3</b> | <b>Introduzione</b>                                             | <b>5</b>  |
| <b>4</b> | <b>Tecnologia</b>                                               | <b>8</b>  |
| 4.1      | Il collettore solare . . . . .                                  | 8         |
| 4.2      | Il metodo di concentrazione della radiazione solare Fresnel . . | 9         |
| <b>5</b> | <b>Il modello matematico</b>                                    | <b>10</b> |
| 5.1      | Il sole nel cielo . . . . .                                     | 10        |
| 5.2      | I raggi e la griglia solare . . . . .                           | 14        |
| 5.3      | La struttura degli specchi . . . . .                            | 15        |
| 5.4      | La riflessione . . . . .                                        | 16        |
| <b>6</b> | <b>Le simulazioni informatiche</b>                              | <b>17</b> |
| 6.1      | Gli angoli solari e il modello matematico . . . . .             | 17        |
| 6.2      | Le ombre . . . . .                                              | 18        |
| <b>7</b> | <b>Costruzione</b>                                              | <b>19</b> |
| 7.1      | Gli specchi . . . . .                                           | 19        |
| 7.2      | L'impianto . . . . .                                            | 20        |
| 7.3      | Materiale . . . . .                                             | 21        |
| <b>8</b> | <b>Risultati</b>                                                | <b>23</b> |
| 8.1      | Risultati ottenuti . . . . .                                    | 23        |
| 8.2      | Discussione dei risultati . . . . .                             | 23        |
| <b>9</b> | <b>Conclusioni</b>                                              | <b>25</b> |
| <b>A</b> | <b>Codice</b>                                                   | <b>28</b> |

# 1 Premessa

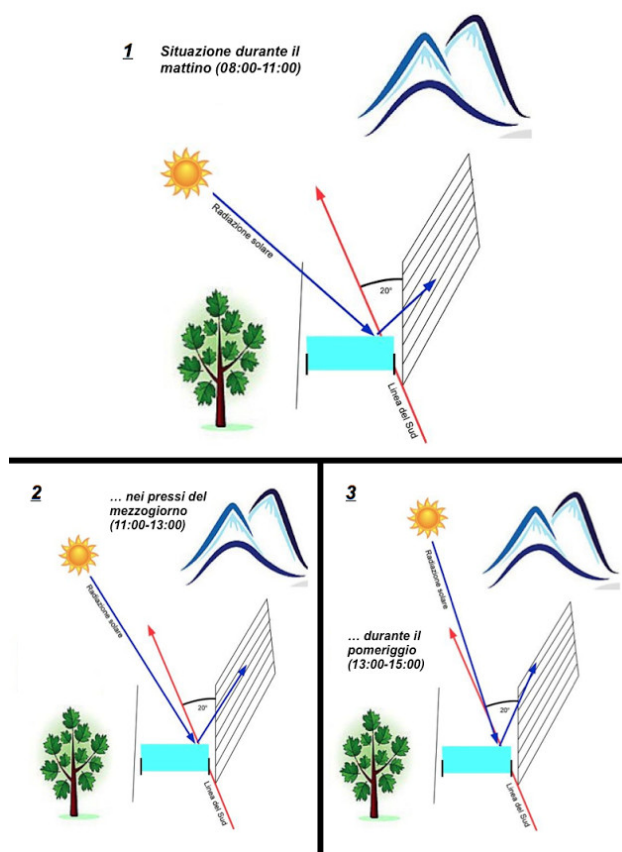
Questo progetto nasce nell'ambito dei lavori di maturità che i liceali sono tenuti a svolgere nel corso dell'ultimo biennio di formazione al Liceo. Durante la scelta della strada da seguire abbiamo cercato perciò di far combaciare le nostre 3 grandi passioni, ossia: matematica, informatica e fisica. Ed è così che siamo giunti alla decisione di seguire il percorso che ci proponeva il professor Cretton nel campo delle energie rinnovabili, percorso che seppur non trattando specificamente le 3 discipline sopracitate, ci ha permesso di unirle e di considerarle assieme. Così è nato il proposito di inserirci in questo settore, settore in continua evoluzione che dona sempre nuovi stimoli e spunti.

## 2 Estratto

Il nostro progetto ha come scopo la costruzione di un impianto semi-mobile di specchi in grado di riflettere i raggi del Sole sui tubi sottovuoto di un collettore solare preesistente con l'obiettivo di aumentarne la resa energetica. Al fine di ottimizzare il complesso, cioè di permettere alla maggior quantità possibile di radiazione solare di giungere, una volta riflessa, sul collettore, la serie di specchi è stata progettata in modo da consentirne la modifica dell'inclinazione zenitale e così il valore del relativo angolo. Il valore dell'angolo zenitale è stato poi ricavato tramite simulazioni estese di molteplici angoli per diversi periodi dell'anno, realizzate tramite un programma informatico elaborato sulla base di un modello matematico in grado di simulare i raggi del Sole, la riflessione sugli specchi e le ombre generate.

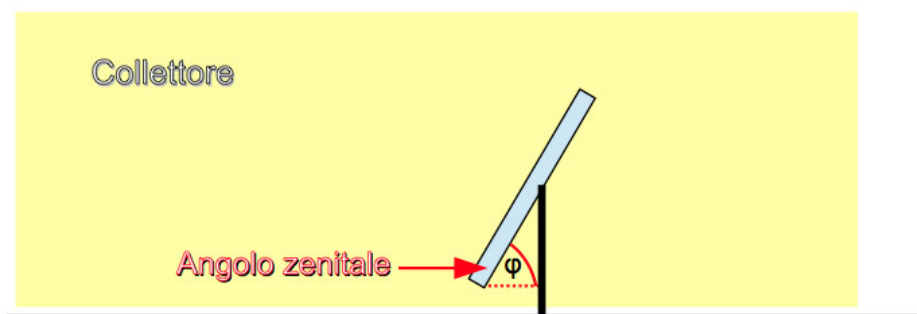
### 3 Introduzione

Il ramo delle energie rinnovabili è un campo in continua espansione ed è in questa direzione che il vasto settore della produzione d'energia sta via via orientandosi. Al momento però, sebbene porti con sé numerosi vantaggi per l'ambiente, esso stenta ad imporsi in maniera decisiva. Le ragioni di questo fenomeno sono da ricercare principalmente nei problemi relativi allo stoccaggio dell'energia ricavata e soprattutto negli elevati costi delle tecnologie utilizzate. Il primo passo verso la definitiva affermazione consiste dunque nel trovare un modo per aggirare quest'ultimo importante ostacolo che frena in particolare il settore del solare. È proprio in questo contesto che si inserisce il nostro progetto.



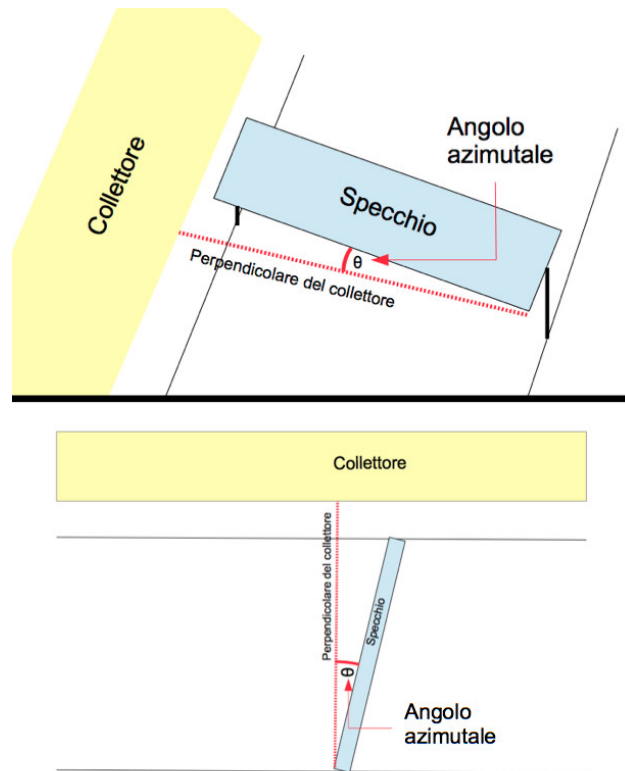
*Figura 1: L'immagine mostra la riflessione della radiazione solare sugli specchi e il successivo arrivo di quest'ultima sul collettore in 3 diversi momenti della giornata: mattino, mezzogiorno e pomeriggio. Si noti in particolare come anche nel pomeriggio (sezione 3) i raggi solari riescono, seppur con un angolo di manovra più ristretto, a giungere sul collettore.*

Avendo a disposizione un collettore solare già installato e collaudato ci siamo chiesti che genere di soluzione fosse possibile proporre al quesito: come è possibile aumentare in maniera tangibile la resa energetica di un impianto del genere in modo da sfruttarne al meglio le potenzialità ed ottimizzando così il rapporto costi totali-resa energetica. La strada che abbiamo intrapreso per rispondere a questa domanda ci ha condotti infine ad una soluzione che prende spunto dal sistema di concentrazione della luce solare chiamato tecnologia solare di Fresnel[1]. Questa interessante proposta consiste nell'integrare ad un lungo tubo collettore trasportante un fluido da surriscaldare, degli specchi esterni posizionati e angolati in modo da approssimare una parabola, specchi così in grado di convogliare la radiazione solare sul tubo stesso in modo tale da aumentare la superficie di intercettamento e così anche la resa energetica dell'impianto[2] (si veda la sezione 4.2 a pagina 9). Partendo da questo presupposto abbiamo perciò ideato un impianto ibrido, che si utilizza specchi per concentrare la luce solare in determinati punti, in particolare sul collettore preesistente, ma che altresì non è strutturato in maniera da formare una parabola, bensì il complesso di specchi è semplicemente posto davanti ai collettori orientati verso sud. Al fine di sfruttare al meglio il complesso di specchi è stato poi deciso, accanto all'aspetto costruttivo concernente quest'ultimi e la struttura atta a sostenerli, di realizzare un programma informatico in grado di identificare con precisione l'angolo zenitale (l'angolo che intercorre tra il lato corto degli specchi e il suolo) migliore in funzione di data e ora, o anche di un periodo determinato in modo che la riflessione sugli specchi induca la maggior quantità di luce possibile a dirigersi verso il collettore e permettendo così di ottenere un'ottimizzazione della resa energetica.



*Figura 2: Rappresentazione grafica dell'angolo zenitale con lo specchio visto di profilo e il collettore osservato frontalmente.*

L'angolo azimutale (vedi figura 3) (cioè l'angolo che intercorre tra il lato lungo degli specchi e la perpendicolare del collettore) non viene considerato per diversi motivi. Anzitutto, a livello di costruzione è molto più semplice realizzare un impianto mobile che rimanga fisso dal punto di vista azimutale. Inoltre, variare quest'angolo non dovrebbe avere un'importante influenza sul risultato finale in quanto, data la conformazione del territorio sul quale sarà installato l'impianto, l'intervallo di tempo per il quale gli specchi sono effettivamente esposti al sole è ristretto a 5-6 ore, e centrato sul mezzogiorno. Essendo il collettore orientato di circa  $20^\circ$  verso est rispetto al nord, e gli specchi perpendicolari al collettore stesso, anche quando il sole procede verso occidente dopo mezzogiorno abbiamo ancora a disposizione un'apertura sufficiente a impedire che i raggi vengano riflessi invece che verso il collettore, dalla parte opposta.



*Figura 3: Le 2 diverse figure mostrano una rappresentazione grafica dell'angolo azimutale degli specchi con prospettive differenti: nell'immagine in alto l'osservatore è posizionato rialzato sul suolo e leggermente distante dall'impianto, in quella in basso invece la prospettiva è a volo d'uccello sul complesso.*

## 4 Tecnologia

### 4.1 Il collettore solare

Nel vasto mondo delle energie rinnovabili i collettori solari sono un valido esempio di come il Sole possa essere sfruttato per produrre energia pulita e ad un costo sostenibile. Essi permettono di trasformare l'energia associata alla radiazione solare in energia termica tramite il riscaldamento di un fluido passante attraverso un tubo esposto alla luce[3]. Nel nostro impianto siamo di fronte ad un particolare tipo di collettore: il collettore solare sottovuoto[4]. Esso è composto da più parti, la principale delle quali è il tubo attraverso il quale viene portato ad evaporazione un liquido, che salendo verso l'alto nel tubo va a riscaldare la parte superiore in contatto con un altro liquido, solitamente un antigelo, che a sua volta va scaldare l'acqua di casa utilizzata per il riscaldamento e l'A.C.S., ovvero l'acqua calda sanitaria[3] (vedi figura 4). La particolarità di questo genere di collettore è il fatto che il tubo destinato a ricevere la luce solare è isolato da un doppio vetro all'interno del quale vi è il vuoto (da qui il termine sottovuoto). In questo modo le perdite di energia vengono ridotte drasticamente permettendo al collettore di avere un'ottima efficienza (si assesta attorno al 70%) anche durante i mesi invernali[5].

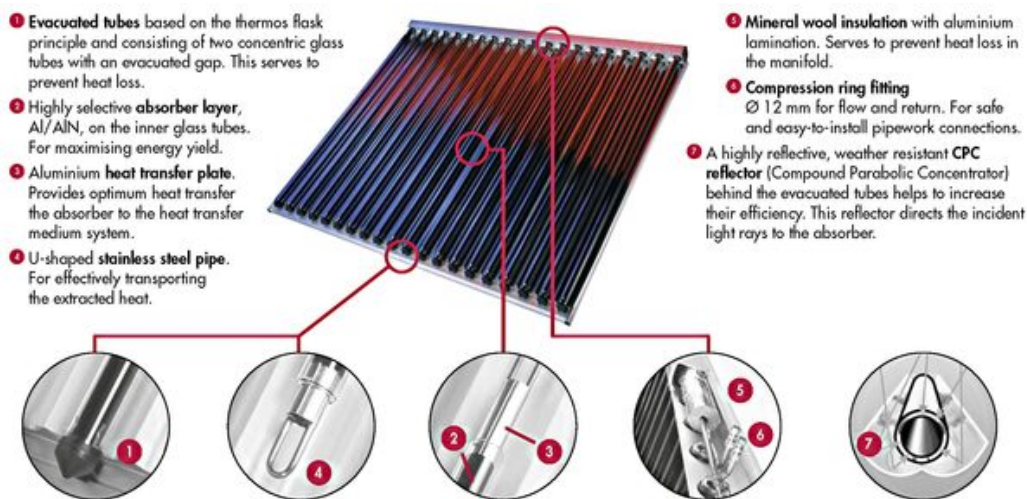


Figura 4: Schema di funzionamento di un collettore solare sottovuoto



## 4.2 Il metodo di concentrazione della radiazione solare Fresnel

La tecnologia solare di Fresnel, sviluppata e brevettata a metà degli anni novanta all'università di Sydney[6], è un'innovativo sistema per accumulare una notevole quantità di energia in determinati punti sensibili come per esempio un lungo tubo nel quale scorre un fluido da riscaldare (vedi figura 4), senza l'utilizzo di costose e complesse strutture paraboliche a gronde. Essa infatti parte dall'idea che, sebbene l'unico modo di far riflettere la luce del Sole in un preciso e determinato punto sia solo attraverso l'uso di una parabola, tramite l'utilizzo di lunghi specchi piani inclinati in un determinato modo è possibile giungere ad un risultato apprezzabile[7]. Il vantaggio di questo sistema è che è economicamente vantaggioso[6] e soprattutto può essere costruito su grandi estensioni senza correre il rischio che l'impianto venga danneggiato da forti venti[8]. Questa tecnologia, ancora in via di sviluppo, sta venendo sempre più utilizzata, specie nei paesi in cui è possibile sfruttare l'energia del Sole in modo efficiente e ve ne sono diversi esempi di impiego nel mondo come l'impianto sviluppato da Areva solar a Bakersfield in California[9] (figura 9).



*Figura 5: AREVA Solar's Compact Linear Fresnel Reflector: esempio di concentratore solare di Fresnel sviluppato dalla multinazionale francese AREVA*

## 5 Il modello matematico

Una delle prime parti del nostro lavoro (fondamentale per arrivare ad uno strumento di simulazione informatica) è stato lo sviluppo di un modello matematico in grado di rappresentare i nostri specchi, il sole e la riflessione dei raggi sugli specchi verso il collettore.

Prima di descrivere a livello matematico il modello occorre parlare delle due possibilità alle quali abbiamo pensato, ovvero un modello continuo o discreto. Nel primo caso avremmo dovuto trovare il modo di rappresentare un flusso uniforme che si muove nello spazio, nel secondo caso avremmo potuto immaginare tanti raggi che si comportano in maniera analoga. Questa seconda soluzione è quella sulla quale abbiamo deciso di lavorare visto che è molto più semplice a livello matematico. È anche vero che il fatto di avere tante strutture sulle quali ripetere in maniera brutale dei calcoli vettoriali (i nostri raggi) richiede tempo al computer per elaborare, tuttavia abbiamo ipotizzato che, data la grande potenza delle macchine moderne, questo non sarebbe stato un grande problema, e infatti, una volta pronte le simulazioni, abbiamo scoperto che il tempo per effettuarle è per nulla eccessivo, nell'ordine (per le simulazioni più ampie) di un paio d'ore.

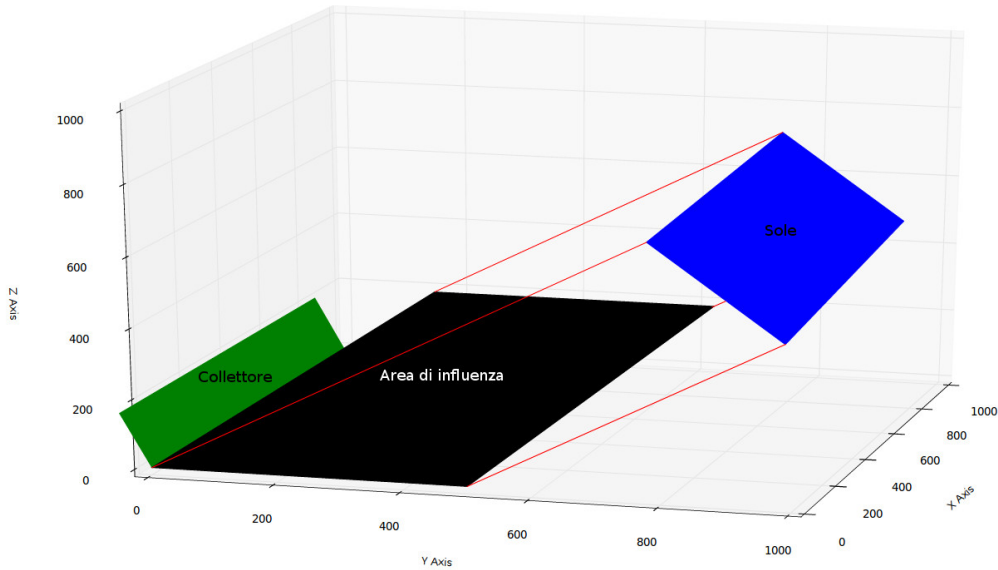
### 5.1 Il sole nel cielo

Per determinare la direzione dei raggi solari è fondamentale conoscere la posizione del sole nel cielo in funzione di data, ora e latitudine. Seppur non sia un compito banale, è fattibile e descrivibile tramite formule note[10]. Una volta ottenuti gli angoli zenitali  $\phi$  e azimutali  $\theta$  (questi ultimi, calcolati in senso orario da nord) possiamo trovare un vettore con la direzione del sole dalla formula

$$\vec{d} = \begin{pmatrix} \sin(\phi) * \cos(\theta) \\ \sin(\phi) * \sin(\theta) \\ \cos(\phi) \end{pmatrix}$$

Notare che, nonostante l'angolo azimutale venga preso in maniera opposta a quella canonica (ovvero, in senso orario invece che in senso antiorario), il segno del valore in  $y$  del nostro vettore non varia rispetto al caso detto. Questo dato che, per comodità, abbiamo invertito la direzione dell'asse  $y$ . Inoltre, per facilitare i calcoli, il sistema di riferimento è orientato nella stessa direzione del collettore solare, il quale è sfasato di circa  $20^\circ$  verso est. Di conseguenza, per i calcoli, dovremo prima sottrarre questo valore dall'angolo azimutale trovato. Una nota: il vettore  $\vec{d}$  "punta" verso il sole; ovvero il suo verso non è a "scendere" verso la terra, bensì a "salire" nella direzione del sole.

Il passo successivo è stato trovare uno strumento matematico in grado di rappresentare e simulare il comportamento del sole. Prima di cominciare precisiamo che il nostro è un modello di tipo discreto, ovvero una simulazione informatica del comportamento di tanti raggi solari in posizioni diverse. Detto questo, la nostra idea è di creare un piano con la caratteristica di avere come vettore normale il vettore  $\vec{d}$  calcolato in precedenza. Dentro questo piano possiamo definire un rettangolo (tramite un punto e due vettori, dove il punto sarà il vertice dei due lati definiti dai vettori), e questo sarà il nostro sole. In più, se definiamo un'area di influenza (ovvero, un'area dentro alla quale troviamo i nostri specchi e dentro alla quale quindi è necessario che cadano i raggi) sul terreno, il nostro rettangolo del sole deve "puntare" precisamente verso quest'area. In altre parole, i vertici del sole devono essere collegati ai vertici dell'area di influenza dalla direzione dei raggi (figura 6). Questo diventa utile quando pensiamo a come definire un raggio solare. A livello fisico ha senso dire che i raggi solari si muovono in linea retta verso la terra, e dunque lo strumento migliore per rappresentarli è tramite una retta, definibile da un punto e un vettore (la sua direzione). Di conseguenza, se noi scegliamo un qualsiasi punto all'interno del nostro rettangolo rappresentante il sole, e ci associamo come direzione il vettore  $\vec{d}$ , otteniamo un raggio che cade nella nostra area di influenza e segue la direzione del sole.



*Figura 6: Rappresentazione grafica del modello. Le 4 linee rosse hanno la direzione dei raggi solari (sono perpendicolari ad esso) e collegano i vertici del sole ai vertici dell'area di influenza.*

Occorre dunque trovare un modo per calcolare questo rettangolo solare. Per semplificare i calcoli partiremo definendo questo rettangolo passante per l'origine (per quanto detto nel paragrafo precedente, vuol dire che uno dei suoi vertici è nell'origine). Abbiamo che il piano  $\pi$  contenente questo rettangolo è definito da tutti i punti  $P = (x, y, z)$  rispettanti l'uguaglianza

$$\pi : \vec{OP} \cdot \vec{n}_\pi = 0$$

dove  $\vec{n}_\pi = \vec{d}$  è la normale del piano. Come detto in precedenza, per definire un rettangolo sono tuttavia necessari non solo un piano, ma anche un punto e due vettori. Il punto fisso lo abbiamo già fissato nell'origine, bisogna adesso trovare i due vettori in modo che il nostro rettangolo rispetti le caratteristiche fissate in precedenza.

La prima fase di questo processo è definire l'area di influenza. Essendo un rettangolo, possiamo usare la stessa idea del rettangolo del sole, ovvero un punto e due vettori. Per comodità il punto (il vertice dei due lati) è ancora una volta l'origine e i due vettori sono paralleli agli assi  $x$  e  $y$ , ovvero

$$\vec{l}_1 = \begin{pmatrix} l_1 \\ 0 \\ 0 \end{pmatrix}$$

$$\vec{l}_2 = \begin{pmatrix} 0 \\ l_2 \\ 0 \end{pmatrix}$$

Dove  $l_1$  e  $l_2$  sono la lunghezza e la larghezza dell'area di influenza. Per semplicità continueremo i calcoli solo per il vettore  $\vec{l}_1$ , gli stessi procedimenti si possono tuttavia applicare anche per  $\vec{l}_2$ . Tornando al problema, per le caratteristiche cercate è evidente che il vettore  $\vec{l}_1$  (ovvero il vettore del nostro rettangolo solare che "punta" verso  $\vec{l}_1$ ) deve essere la proiezione ortogonale di  $\vec{l}_1$  su  $\pi$ . Questo si può ottenere tramite il seguente ragionamento. Calcoliamo ora il vettore

$$\vec{n}_\alpha = \vec{l}_1 \times \vec{n}_\pi$$

e consideriamo il piano  $\alpha$  passante per l'origine e avente come normale il vettore  $\vec{n}_\alpha$ . Per definizione è perpendicolare a  $\pi$  e parallelo a  $\vec{l}_1$ . La retta  $\alpha \cup \pi$  avrà come direzione il vettore  $\vec{r}$  che è una proiezione ortogonale allungata o accorciata di  $\vec{l}_1$  su  $\pi$ . Essendo la retta contenuta sia in  $\alpha$  che in  $\pi$ , la sua direzione deve essere perpendicolare alle normali di entrambi i piani, ovvero

$$\vec{r} = \vec{n}_\pi \times \vec{n}_\alpha$$

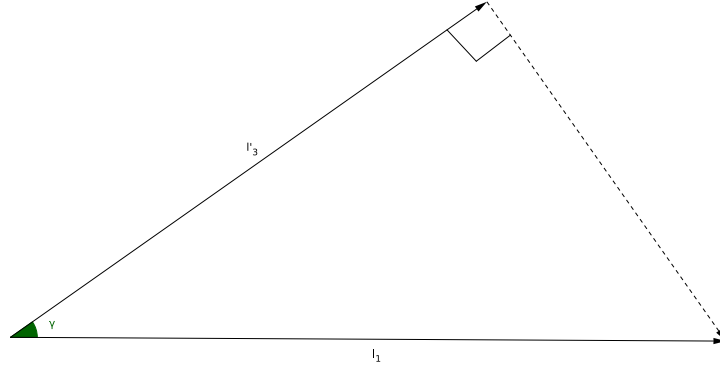


Figura 7: Visione dell'angolo tra  $l'_1$  e  $l_1$ , dove si nota la relazione trigonometrica tra le lunghezze dei due vettori.

Analizziamo il modulo di questo vettore. Data la relazione

$$|\vec{v}_1 \times \vec{v}_2| = |\vec{v}_1| \cdot |\vec{v}_2| \cdot \sin \angle(\vec{v}_1, \vec{v}_2)$$

abbiamo che

$$\begin{aligned} |\vec{r}| &= |\vec{n}_\pi \times \vec{n}_\alpha| \\ &= |\vec{n}_\pi| \cdot |\vec{n}_\alpha| \cdot \sin \angle(\vec{n}_\pi, \vec{n}_\alpha) \\ &= |\vec{l}_1 \times \vec{n}_\pi| \cdot |\vec{n}_\pi| \cdot \sin \angle(\vec{n}_\pi, \vec{n}_\alpha) \\ &= |\vec{l}_1| \cdot |\vec{n}_\pi| \cdot |\vec{n}_\pi| \cdot \sin \angle(\vec{n}_\pi, \vec{n}_\alpha) \cdot \sin \angle(\vec{l}_1, \vec{n}_\pi) \\ &= |\vec{l}_1| \cdot |\vec{n}_\pi|^2 \cdot \sin \angle(\vec{n}_\pi, \vec{n}_\alpha) \cdot \sin \angle(\vec{l}_1, \vec{n}_\pi) \end{aligned}$$

Da questo possiamo semplificare un paio di fattori. Per definizione abbiamo che  $\vec{n}_\pi \perp \vec{n}_\alpha$ , e se poniamo  $\gamma = \angle(\vec{l}_1, \vec{n}_\pi)$ , allora vale che  $\angle(\vec{l}_1, \vec{n}_\pi) = \frac{\pi}{2} \pm \gamma$ . Inserendo questo nell'espressione otteniamo

$$\begin{aligned} |\vec{r}| &= |\vec{l}_1| \cdot |\vec{n}_\pi|^2 \cdot \sin \frac{\pi}{2} \cdot \sin \left( \frac{\pi}{2} \pm \gamma \right) \\ &= |\vec{l}_1| \cdot |\vec{n}_\pi|^2 \cdot \cos \gamma \end{aligned}$$

Come visibile in figura 7 abbiamo tuttavia

$$|\vec{l}'_1| = |\vec{l}_1| \cdot \cos \gamma$$

e di conseguenza

$$\vec{l}'_1 = \frac{1}{|\vec{n}_\pi|^2} \vec{r}$$

Abbiamo dunque trovato un modo per definire il rettangolo solare con tutte le caratteristiche che avevamo fissato.

## 5.2 I raggi e la griglia solare

Il passo successivo è stato sviluppare un modello che permettesse di generare i raggi sul sole, tenendo in conto il fatto che, quando il sole è basso nel cielo, la densità di energia sul terreno è minore rispetto a quando si trova in alto. Per simulare questa caratteristica dobbiamo fare in modo che il numero di raggi che cade nell'area di influenza sia minore quando il sole è basso nel cielo. Per costruzione, sappiamo però che il sole quando ha minore elevazione è anche più piccolo (per il fatto che i vertici del sole puntano a quelli dell'area di influenza). Di conseguenza, se manteniamo costante la distanza tra i raggi, quando il rettangolo solare è più piccolo ci saranno meno raggi non solo su di esso, ma anche dentro la nostra area di influenza. Fare questo è molto semplice. Come detto, per definire i raggi abbiamo bisogno della loro direzione (il vettore  $\vec{d}$  calcolato in 5.1) e un punto per il quale passano. Di conseguenza, la distanza tra due raggi sarà equivalente alla distanza tra i due rispettivi punti. Adesso, mettiamo di volere tutti i nostri raggi ad una distanza  $r$  tra loro. Dobbiamo quindi riempire il nostro rettangolo solare di punti a distanza  $r$  l'uno dall'altro, e ad ognuno di essi associare un raggio (vedi figura 8).

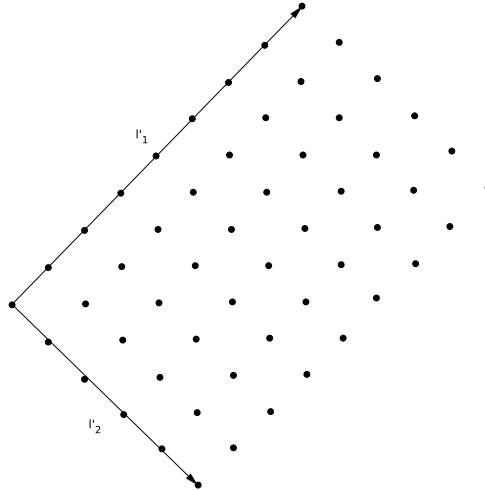


Figura 8: La griglia dei punti di partenza per i raggi. Notare come tutti i punti siano combinazione lineare di  $\vec{l}_1$  e  $\vec{l}_2$  per coefficienti minori di 1.

### 5.3 La struttura degli specchi

Rappresentare gli specchi non è un lavoro complesso. Possiamo usare lo stesso sistema dell'area di influenza, ovvero definire un rettangolo con un punto (un vertice dello specchio) e due vettori, rappresentanti i lati. Per costruzione, gli specchi ruoteranno intorno alla metà del lato corto, quindi è utile considerare il movimento per l'appunto di metà del lato. Ammesso quindi di conoscere la lunghezza di metà del lato corto  $l_1 = |\vec{l}_1|$  possiamo rappresentare questo lato in posizione di "riposo" (quindi verticale) con il vettore

$$\vec{l}_1 = \begin{pmatrix} 0 \\ 0 \\ l_1 \end{pmatrix}$$

Possiamo anche rappresentare il lato lungo di lunghezza  $l_2 = |\vec{l}_2|$  con

$$\vec{l}_2 = \begin{pmatrix} 0 \\ l_2 \\ 0 \end{pmatrix}$$

Questo perché i nostri specchi (per definizione del sistema di coordinate) sono perpendicolari al collettore e dunque si estendono verso l'asse y. Dato dunque l'angolo zenitale  $\alpha$  di inclinazione degli specchi, semplici relazioni trigonometriche mostrano che

$$\vec{l}'_1 = \begin{pmatrix} l_1 \sin \alpha \\ 0 \\ l_1 \cos \alpha \end{pmatrix}$$

mentre  $\vec{l}'_2 = \vec{l}_2$ , con  $\vec{l}'_n$  il vettore lato dopo la rotazione. Possiamo trovare la normale di questo rettangolo semplicemente calcolando il prodotto vettoriale tra  $\vec{l}'_1$  e  $\vec{l}'_2$ . Per svolgere con più semplicità i calcoli in 5.4, è utile considerare questa normale unitaria, dividendola quindi per il suo modulo. Il vertice del rettangolo che possiamo considerare sarà dunque  $\vec{OP} = \vec{OM} - \vec{l}'_1$ , dove  $M$  è il punto mediano fisso del lato corto. Essendo  $\vec{l}'_1$  solo la metà di questo lato, dobbiamo considerare dunque il vettore  $\vec{l}''_1 = 2\vec{l}'_1$  come il primo lato del rettangolo, e  $\vec{l}'_2$  il secondo.

Calcolare l'intersezione tra un rettangolo e un raggio non è complesso. Anzitutto, si può considerare il piano  $\beta$  sul quale giace il rettangolo stesso, e calcolare l'intersezione tra la retta e il piano. Considerando la forma

$$\beta : \vec{OP} + \lambda \vec{l}''_1 + \mu \vec{l}'_2$$

Possiamo trovare i coefficienti  $\lambda$  e  $\mu$  per i quali si verifica l'uguaglianza. È dunque evidente che, per  $0 \leq \lambda, \mu \leq 1$  il raggio cade dentro il rettangolo e deve venire riflesso dallo specchio, rispettivamente intercettato dal collettore.

## 5.4 La riflessione

La riflessione di un raggio è molto semplice da simulare: le leggi dell'ottica[11] dicono che, per una superficie molto riflettente (come uno specchio), dati  $\theta_i$  e  $\theta_f$  gli angoli rispettivamente di incidenza e di riflessione sulla superficie, allora  $\theta_i = \theta_f$ . Questo vuol dire che, preso un piano contenente il vettore del raggio e la normale dello specchio, se facciamo ruotare il raggio di  $\pi$  radianti intorno appunto alla normale otterremo un vettore che rispecchi questa caratteristica (figura MADÒ), ovvero che, se  $\vec{r}$  è il raggio incidente,  $\vec{n}_\alpha$  la normale del piano e  $\vec{r}'$  il raggio riflesso, allora  $\angle(\vec{r}, \vec{n}_\alpha) = \angle(\vec{r}', \vec{n}_\alpha)$ . Questa rotazione non è che un caso particolare della formula di rotazione di Rodrigues[12]

$$v_{rot}(\vec{v}, \vec{k}, \theta) = \vec{v} \cos \theta + \vec{k} \times \vec{v} \sin \theta + \vec{k}(\vec{k} \cdot \vec{v})(1 - \cos \theta)$$

dove  $\vec{v}$  è il vettore da ruotare,  $v_{rot}$  il vettore dopo la rotazione,  $\vec{k}$  l'asse attorno al quale ruotare il vettore  $\vec{v}$  e  $\theta$  l'angolo di rotazione. Otteniamo la riflessione del raggio per  $\theta = \pi$ , ovvero

$$\vec{r}' = -\vec{r} + 2\vec{n}_\alpha(\vec{n}_\alpha \cdot \vec{r})$$



## 6 Le simulazioni informatiche

Una volta definito un modello matematico il passo successivo è stato quello di scrivere un programma in grado di simulare la riflessione del nostro impianto verso il collettore solare. Questo ci permette prima di tutto di trovare quale sia l'angolazione migliore per i nostri specchi, e in secondo luogo (specie riguardo a progetti futuri di ampliamento) di testare nuove idee senza dover per forza costruire fisicamente nuovi impianti per poi scoprire che hanno una scarsa resa. Per maggiori e più puntuali informazioni sull'implementazione riferirsi al codice allegato.

Il nostro programma è scritto interamente in Python 2.7. Questo per il fatto che è molto semplice programmare in Python, è senza dubbio il linguaggio di programmazione con il quale ho più esperienza e dà accesso a varie librerie che permettono di semplificare determinate operazioni (prima tra tutte, la rappresentazione grafica del modello, che ci ha permesso di verificare in maniera visiva che il codice scritto avesse senso). Chiaramente, questa scelta comporta un notevole svantaggio visto che il nostro metodo si basa su un controllo brutale di intersezioni piano-retta rispetto alla scelta di un linguaggio compilato. Tuttavia, quando è stato possibile effettuare le prime simulazioni ci siamo resi conto che in ogni caso il tempo richiesto non è eccessivo, e che in una serata è possibile ricavare l'angolo zenitale ideale del nostro complesso.

### 6.1 Gli angoli solari e il modello matematico

Come già accennato, le equazioni per calcolare la posizione del sole nel cielo sono ben note[10], e possiamo sfruttare questi semplici algoritmi per ricavare angoli zenitali e azimutali per qualsiasi latitudine, longitudine, giorno dell'anno e ora, e anche per prolungati periodi di tempo. Per ognuna di queste coppie di angoli diventa dunque necessario creare un relativo rettangolo solare.

Il modo migliore per rappresentare tutte le nostre strutture matematiche (il sole, gli specchi, ...) è sicuramente tramite la programmazione orientata agli oggetti. In questo modo, ogni classe contiene tutte le caratteristiche che la contraddistinguono (ad esempio, il sole ha come attributi il vertice e i suoi due lati) e possiede tutti i vari metodi che ne definiscono il comportamento. Inoltre, in questo modo possiamo svolgere tutti i calcoli descritti nella sezione 5 direttamente nella costruzione degli oggetti, rendendo il programma più modulare (permettendo il passaggio di parametri come gli angoli solari o l'angolo zenitale degli specchi) e meno ripetitivo.

## 6.2 Le ombre

Una grande comodità della simulazione informatica è il fatto che possiamo simulare le ombre prodotte dai nostri specchi (o volendo, di qualsiasi altro oggetto opaco) senza il minimo sforzo. Infatti, basta semplicemente fare il controllo di intersezione tra i raggi e l'oggetto più a sud (ovvero il primo che incontrerebbe i raggi solari, dato che alle nostre latitudini il sole si trova sempre a sud), fermando il controllo in caso di intersezione o procedendo con gli altri oggetti in caso contrario. Così facendo evitiamo di contare due volte lo stesso raggio, con il rischio di falsare i risultati.

## 7 Costruzione

Come evidenziato già nell'introduzione, nel nostro lavoro ha, accanto all'aspetto matematico-informatico, un importante ruolo anche la parte costruttiva. L'idea di base per la realizzazione dell'impianto è quella di costruire un sistema semi-mobile, ovvero un sistema in grado di poter essere modificato periodicamente (ogni 2 settimane), ma nello stesso tempo solido, in modo da poter essere lasciato in un ambiente esterno durante l'inverno senza incorrere in alcun tipo di problema. Per adeguare poi il tutto alle esigenze che si son venute a creare, si è scelto di realizzare l'impianto interamente "in casa", partendo dagli specchi fino ai supporti per quest'ultimi.

### 7.1 Gli specchi



*Figura 9: Uno dei vari specchi realizzati "in casa", si noti il film riflettente (Mylar) e il profilo in alluminio dove lo specchio verrà fissato alle barre verticali. Il tutto pesa meno di 2.5 kg*

L'aspetto più rilevante per quanto concerne gli specchi da noi realizzati è sicuramente il rivestimento riflettente adottato: il mylar[13]. Questo materiale, utilizzato su larga scala anche dalle aziende nel settore delle energie rinnovabili[14], ha il vantaggio notevole di essere adattabile a qualsiasi superficie e soprattutto di essere molto più leggero rispetto ad uno specchio tradizionale, anche considerando il supporto su cui si stende. Ed è proprio

per quest'ultima caratteristica che ben si adatta al progetto in questione. Oltre al mylar, gli specchi realizzati sono composti da una lastra di alluminio di 160 cm di lunghezza e 50 di larghezza e 8 mm di spessore su cui il mylar poggia, cosa che consente agli specchi di essere leggeri, solidi e soprattutto di non deformarsi sotto il proprio peso, cosa che potrebbe causare una riflessione non perfetta. Oltre a questi due elementi tutti gli specchi sono poi dotati di una cornice che percorre tutti i 4 lati del rettangolo, in alluminio di 2 cm x 2 cm con 2 mm di spessore e a L in modo da essere il più resistente possibile in funzione del peso.

## 7.2 L'impianto

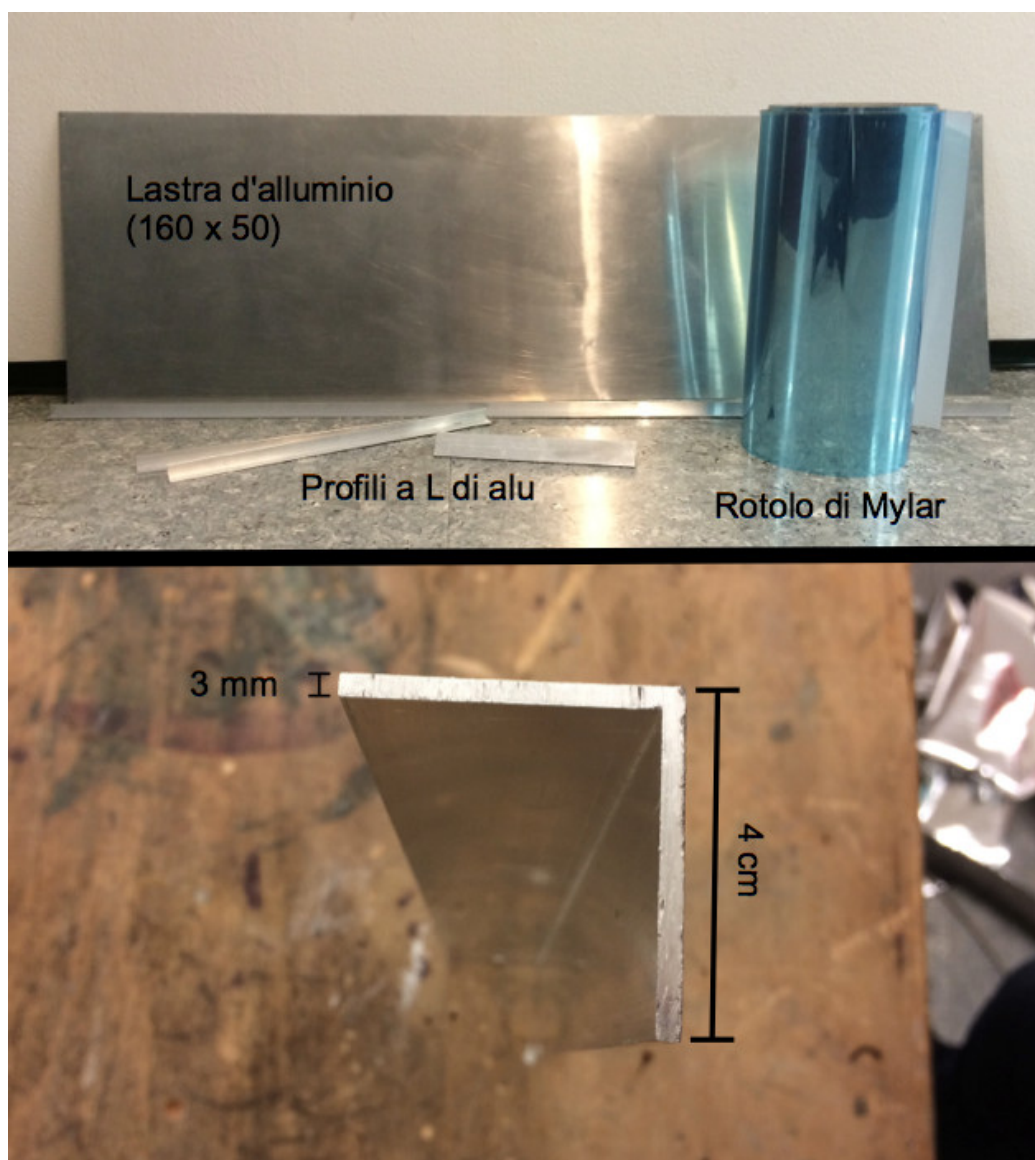
Il complesso comprende più specchi: 6 specchi disposti perpendicolarmente davanti al collettore solare e posizionati uno accanto all'altro ad una distanza reciproca tale da evitare che l'ombra del primo copra il secondo e così via, e un'ulteriore specchio posto piano in orizzontale sul supporto di barre che ha bloccato perciò sia l'angolo zenitale che quello azimutale, caratteristica che lo rende un unicum nell'intero impianto. Per quanto riguarda invece gli altri 6 specchi, per permettere la rotazione periodica e così anche il cambiamento dell'angolo zenitale essi sono sostenuti da alcune barre di alluminio poste verticalmente (4cm x 4cm con 3mm di spessore) che li tengono sollevati dal terreno ad una distanza minima di 5 cm. Quest'ultimo accorgimento è stato preso perché, seppur consapevoli che gli specchi, affinché potessero riflettere la maggior quantità possibile di radiazione solare sul collettore, dovessero avere un angolo zenitale non nullo, l'angolo in questione non era ancora stato definito e perciò al fine di evitare spiacevoli sorprese una volta montato l'impianto, si è optato per realizzare un sostegno verticale agli specchi che permettesse a quest'ultimi di ruotare di 360 gradi mantenendo con ogni angolazione una distanza minima di 5 cm dal suolo. Le barre verticali sono poi affrancate agli specchi con delle viti in prossimità del punto mediano del lato corto del supporto, punto mediano che è perciò anche il punto fisso attorno al quale ruota lo specchio (per approfondimenti consultare la sezione 5.3 a pagina 15). A loro volta poi le barre verticali, siedono su 2 lunghe barre d'alluminio che percorrono tutti gli specchi per garantire una maggiore stabilità e che sono fissate al terreno tramite l'aggancio ad un muretto in cemento. Infine, oltre a tutto ciò, vi è poi un'ulteriore barra di alluminio, lunga anch'essa in modo tale da coprire tutta l'estensione degli specchi, collegata lateralmente ad ognuno di quest'ultimi che ha lo scopo di far ruotare, quando necessario, contemporaneamente tutti gli specchi, così da poter modificare l'angolo zenitale in maniera omogenea per tutte le componenti del complesso ed evitare così erronee differenze di angolazione.



*Figura 10: L'immagine mostra l'impianto ai primi di febbraio e ancora in via di completamento da due differenti angolazioni e in due differenti momenti della giornata.*

### 7.3 Materiale

Il materiale utilizzato per la costruzione di specchi e sostegno, come si può evincere nei paragrafi immediatamente precedenti, è molto semplice, relativamente facile da reperire e a basso costo. L'impianto infatti è costituito essenzialmente dall'assemblaggio di profili a L d'alluminio che costituiscono la cornice degli specchi e il complesso di sostegni verticali e orizzontali, da lastre d'alluminio corrispondenti alla struttura degli specchi e dalla pellicola riflettente mylar, oltre alle numerose viti che tengono assieme il tutto. Si delinea così uno scenario in cui chiunque, con un budget massimo di 250-300 franchi, può procedere alla realizzazione di un progetto costruttivo completo (quindi sia gli specchi che tutta la struttura atta a sostenerli) di questo tipo.



*Figura 11: Nella figura in alto sono mostrati i principali materiali utilizzati per costruire gli specchi, mentre sotto viene mostrato un dettaglio delle barre verticali che sorreggono gli specchi e ne permettono la rotazione.*

## 8 Risultati

I dati sono stati raccontati durante l'arco di 5 giorni ai primi di febbraio, anche se, per questioni meteorologiche e per via di altri test in corso durante quei giorni, sono state solo due le giornate ben soleggiate e prive di vento in cui si è misurato l'aumento della resa energetica dovuto solamente alla presenza del nostro impianto di specchi. Quest'ultima è stata determinata tramite un contatore che misura la resa energetica termica prodotta in  $kWh$ , che si basa sui dati di temperatura e flusso (in  $\frac{m^3}{s}$ ) del fluido riscaldato dall'acetone dei tubi sottovuoto.

### 8.1 Risultati ottenuti

I dati ottenuti nelle 2 giornate per il collettore e l'annesso impianto di specchi (durante i test l'impianto era operativo solo al 55%, quest'ultimo aspetto è spiegato nel paragrafo successivo, vedi paragrafo indicato) indicano con il valore di 31 kWh la resa energetica effettiva misurata. Considerando che la resa energetica con il solo collettore è di 25 kWh, l'aumento di resa dovuto al nostro impianto è stato perciò di 6 kWh, dato corrispondente in percentuale al 19,4% di aumento di resa. Questo dato se relazionato alla superficie di intercettamento mostra come per il solo collettore solare i kWh prodotti per metro quadrato ( $\frac{kWh}{m^2}$ ) sono stati 1.62 mentre per il complesso di specchi ci troviamo di fronte ad un valore più elevato, 1.88.

### 8.2 Discussione dei risultati

Prima di procedere all'analisi dei dati, è necessario fare 3 importanti premesse. La prima riguarda la superficie riflettente del nostro complesso effettivamente presa in considerazione durante i test. Occorre precisare infatti che durante la raccolta dei dati l'impianto è stato operativo solo al 55%, in quanto dei 6 specchi principali che lo costituiscono più lo specchio aggiuntivo posizionato piano sul supporto, solo 4 erano montati andando così a delineare una superficie riflettente di 3.2  $m^2$  invece dei normali 5.85  $m^2$ . La seconda premessa invece mette l'accento sul fatto che di tutti e 6 gli specchi ordinari (quindi escludendo lo specchio aggiuntivo) dell'impianto, il primo, per via della geometria del complesso, cioè a causa sua vicinanza al bordo laterale meridionale del collettore, riflette mediamente meno della metà della radiazione solare rispetto a tutti gli altri 5 specchi. Questa caratteristica del primo specchio dell'impianto è stata determinata durante le simulazioni al computer, ed è un aspetto da mettere in conto nel trarre le giuste conclusioni. La terza ed ultima invece va a considerare il valore della resa energetica del

collettore quando questo è preso singolarmente (cioè senza che nessun impianto di specchi gli venga applicato). Non avendo avuto modo di valutare, per le scadenze richieste, questo valore per il mese di febbraio (il mese in cui sono stati presi i dati) a causa del tempo meteorologico e del fatto che era già montato un'altro impianto di specchi all'infuori dal nostro, per determinarlo abbiamo optato per una strada teorica. Confrontando infatti le ore di luce ai primi di gennaio con le ore di luce ai primi di febbraio e svolgendo una semplice proporzione lineare, conoscendo inoltre il valore richiesto per il mese di gennaio, cioè  $22\text{ kWh}$ , abbiamo determinato che il valore della resa energetica del solo collettore in febbraio si assesta a  $25\text{ kWh}$ . Nel precedente paragrafo è stato perciò preso questo valore come dato di partenza per poi andare a verificare l'effettivo aumento della resa energetica dovuto al nostro impianto. Occorrerà comunque valutare in successivi test questo dato empiricamente per evitare che nel calcolo appena effettuato sia sfuggito qualcosa di non considerato.

Alla luce di questi aspetti perciò appare evidente come le seguenti considerazioni sull'aumento di resa dovuto al nostro impianto sono preliminari, in quanto vi sono da fare ancora ulteriori test dilazionati su un lasso di tempo più esteso. Questo anzitutto per valutare l'aumento di resa con l'impianto completo, poi in modo da correggere eventuali errori teorici di valutazione e soprattutto in modo da aumentare il significato statistico dei dati ottenuti.

Nonostante quando detto sopra rimane comunque importante il dato che mostra l'incremento percentuale della resa energetica del 19.4%, anche in relazione al fatto esposto prima riguardo la riflessione sul collettore non ottimale del primo specchio. Con l'impianto completo questo aspetto avrà di sicuro, proporzionalmente, un impatto più limitato andando così a delineare un aumento di resa più ottimistico di quanto si possa prevedere con una proporzione lineare in questo momento. Si tratta per noi così di un buon risultato in quanto era in questo range che ad inizio progetto avevamo intenzione di inserirci. Il dato sulla resa energetica appena citato assume poi un valore maggiore se rapportato ad un altro dato presentato nel paragrafo precedente che riguarda la resa energetica in funzione della superficie di intercettamento. Nel nostro caso specifico infatti il rapporto resa energetica-superficie di intercettamento è chiaramente a favore del nostro complesso di specchi rispetto al collettore preso singolarmente. Questo è dovuto principalmente alla posizione più vantaggiosa del complesso e probabilmente anche alla migliore angolazione che i raggi solari, una volta riflessi dagli specchi, hanno quando intercettano il collettore, cosa che porta l'impianto di specchi, nel nostro caso, ad essere un impianto molto efficiente sebbene non privo di varie ulteriori migliorie.



## 9 Conclusioni

In poco più di un anno di lavoro siamo riusciti a mettere insieme un programma funzionante, funzionale al contesto e soprattutto agile, che dà all'utente uno strumento utile in molteplici situazioni di questo tipo e capace di indicare la strada da percorrere. Soprattutto però è stato realizzato l'impianto in grado effettivamente di raggiungere lo scopo per cui è stato costruito, un impianto che oltre a permettere un aumento di resa energetica del collettore preesistente del 19,4% con solo il 55% del proprio potenziale operativo, ha una rapporto resa energia prodotta/superficie di intercettazione superiore al solo collettore solare. Proprio questo ultimo aspetto mostra chiaramente come un impianto di specchi come il nostro, grazie soprattutto alla sua duttilità, rappresenta senza dubbio un'interessante soluzione alla questione posta inizialmente a cui ci sentiamo di aver dato infine una risposta convincente. Sappiamo comunque bene che vi sono numerosi elementi da migliorare nel complesso che tenteremo di sistemare e che il risultato ottenuto, sebbene incoraggiante, deve essere poi confermato da ulteriori test fatti in condizioni migliori di quelle avute durante la prima serie di raccolta dati, ma nonostante questo ci riteniamo soddisfatti del lavoro svolto.

## Riferimenti bibliografici

- [1] Energogreen Renewables. *Tecnologia Solare di Fresnel*. [Online; in data 26-Ottobre-2014]. URL: [http://www.energogreen.com/index.php?option=com\\_content&view=article&id=41&Itemid=52&lang=it](http://www.energogreen.com/index.php?option=com_content&view=article&id=41&Itemid=52&lang=it).
- [2] Solar Euromed. *Linear Fresnel*. [Online; in data 26-Ottobre-2014]. URL: <http://www.solareuromed.com/en/miroirs-de-fresnel>.
- [3] Wikipedia. *Solar thermal collector* — *Wikipedia, l'enciclopedia libera*. [Online; in data 26-Ottobre-2014]. URL: [http://en.wikipedia.org/wiki/Solar\\_thermal\\_collector](http://en.wikipedia.org/wiki/Solar_thermal_collector).
- [4] Wikipedia. *Pannello solare termico* — *Wikipedia, l'enciclopedia libera*. [Online; in data 26-Ottobre-2014]. URL: [http://it.wikipedia.org/wiki/Pannello\\_solare\\_termico](http://it.wikipedia.org/wiki/Pannello_solare_termico).
- [5] Solar Panels Plus. *Evacuated Tube Solar Water Heating Collectors*. [Online; in data 26-Ottobre-2014]. URL: <http://www.solarpanelsplus.com/evacuated-tube-collectors>.
- [6] Wikipedia. *Compact linear Fresnel reflector* — *Wikipedia, l'enciclopedia libera*. [Online; in data 27-Ottobre-2014]. URL: [http://en.wikipedia.org/wiki/Compact\\_linear\\_Fresnel\\_reflector](http://en.wikipedia.org/wiki/Compact_linear_Fresnel_reflector).
- [7] SEIA. *Concentrating Solar Power*. [Online; in data 27-Ottobre-2014]. URL: <http://www.seia.org/policy/solar-technology/concentrating-solar-power>.
- [8] Industrial Solar. *Fresnel Collector*. [Online; in data 27-Ottobre-2014]. URL: <http://www.industrial-solar.de/CMS/en/angebot/fresnel-kollektor>.
- [9] Brian Frank. *Suncatchers: A Solar Tech Rundown*. [Online; in data 27-Ottobre-2014]. 20 Gennaio 2012. URL: [http://www.kcet.org/shows/socal\\_connected/content/environment/suncatchers-a-solar-tech-rundown.html](http://www.kcet.org/shows/socal_connected/content/environment/suncatchers-a-solar-tech-rundown.html).
- [10] Jürgen Giesen. *Basics of positional astronomy*. [Online; in data 13-Febbraio-2015]. URL: <http://www.jgiesen.de/elevaz/basics/index.htm>.
- [11] Wikipedia. *Specular reflection* — *Wikipedia, l'enciclopedia libera*. [Online; in data 27-Ottobre-2014]. URL: [http://en.wikipedia.org/wiki/Specular\\_reflection](http://en.wikipedia.org/wiki/Specular_reflection).
- [12] Wikipedia. *Rodrigues' rotation formula* — *Wikipedia, l'enciclopedia libera*. [Online; in data 27-Ottobre-2014]. URL: [http://en.wikipedia.org/wiki/Rodrigues'\\_rotation\\_formula](http://en.wikipedia.org/wiki/Rodrigues'_rotation_formula).

- [13] Wikipedia. *BoPET* — *Wikipedia, l'enciclopedia libera*. [Online; in data 25-Ottobre-2014]. URL: <http://en.wikipedia.org/wiki/BoPET>.
- [14] Generoso Chiaradonna. «Catturare il sole con tecnologia ticinese». In: *La Regione Ticino* (16 Ottobre 2009).

## A Codice

*Listing 1: vector.py*

```
from math import sqrt, cos, sin

class Vector:
    def __init__(self, x, y, z):
        self.x = x
        self.y = y
        self.z = z

    def norm(self):
        return sqrt(self.x ** 2 + self.y ** 2 + self.z **
                    2)

    # Boring stuff to have easier life in operations (
    # python awesomeness)
    def __str__(self):
        return "[" + str(self.x) + ", " + str(self.y) + "
            , " + str(self.z) + "]"

    def __bool__(self):
        if self.x == 0 and self.y == 0 and self.z == 0:
            return False
        return True

    def __getitem__(self, key):
        if key == 0:
            return self.x
        elif key == 1:
            return self.y
        elif key == 2:
            return self.z
        raise IndexError

    def __setitem__(self, key, value):
        if key == 0:
            self.x = value
        elif key == 1:
            self.y = value
        elif key == 2:
            self.z = value
        else:
```

```

        raise IndexError

def __add__(self, other):
    return Vector(self.x + other.x, self.y + other.y,
                  self.z + other.z)

def __sub__(self, other):
    return self + -1 * other

def __mul__(self, other):
    return Vector(self.x * other, self.y * other,
                  self.z * other)

def __rmul__(self, other):
    return self.__mul__(other)

# Products of two vectors
def dot(self, other):
    return self.x * other.x + self.y * other.y + self
        .z * other.z

def cross(self, other):
    return Vector(self.y * other.z - self.z * other.y
                  , self.z * other.x - self.x * other.z, self.x
                    * other.y -
                      self.y * other.x)

# Rodrigues' formula to rotate vectors. Should work
def rotate(self, axis, theta):
    axis = axis * (1 / axis.norm())
    return self * cos(theta) + (axis.cross(self)) *
        sin(theta) + axis * (axis.dot(self)) * (1 -
        cos(theta))

def reflect(self, axis):
    axis = axis * (1 / axis.norm())
    return -1 * self + 2 * axis.dot(self) * axis

```

*Listing 2: line.py*

```
class Line:
    def __init__(self, point, direction):
        # Line is defined by a fixed point on it and its
        # direction in space
        self.point = point
        self.direction = direction

    def get_point(self, k):
        # Return a point with a k coefficient from the
        # line's "origin"
        return self.point + k * self.direction

    def rotate(self, axis, angle):
        # Uses rodrigues' formula implemented in Vector
        # class (direction should be a vector)
        self.direction = self.direction.rotate(axis,
        angle)

    def reflect(self, axis):
        # Just as above
        self.direction = self.direction.reflect(axis)
```

*Listing 3: plane.py*

```
class Plane:
    def __init__(self, point, normal):
        # A plane in space is defined by a point in it
        # and a normal vector to it
        self.point = point
        self.normal = normal

        # Standard linear algebra. Lambda notation is
        # used to keep it clearer
        self.equation = lambda pt: self.normal.dot(pt) -
            self.normal.dot(self.point)
        self.distance = lambda pt: self.equation(pt) /
            self.normal.norm()

    def intersection_point(self, line):
        # If the normal is perpendicular to the line,
        # they either don't intercept or completely
        # overlap
        if line.direction.dot(self.normal) == 0:
            return None
        else:
            # Solved with maxima
            k = (self.normal.dot(self.point) - self.
                normal.dot(line.point)) / self.normal.dot(
                line.direction)
            return line.get_point(k)

class Rectangle:
    def __init__(self, point, width, height):
        # A rectangle can be treated as a plane, in which
        # the point is a vertex
        self.point = point
        self.width = width
        self.height = height

        # Basic linear algebra
        self.normal = self.width.cross(height)
        self.plane = Plane(self.point, self.normal)

    def intersection_point(self, line):
```

```

intersection = self.plane.intersection_point(line
)
if not intersection:
    return None
else:
    # Solved with maxima
    a = -(self.height.x * (intersection.y - self.
        point.y) + self.height.y * self.point.x -
        self.height.y *
            intersection.x) / (self.width.x * self.
                height.y - self.height.x * self.
                    width.y)
    b = (self.width.x * (intersection.y - self.
        point.y) + self.width.y * self.point.x -
        self.width.y *
            intersection.x) / (self.width.x * self.
                height.y - self.height.x * self.width
                    .y)

    # If both a (coefficient for width) and b (
        coefficient for height) are less than 1,
        they point lies
    # in the rectangle
    if 0 <= a <= 1 and 0 <= b <= 1:
        return intersection
    else:
        return None

```



*Listing 4: sun.py*

```
from line import Line
from vector import Vector as v
from math import sin, cos
from plane import Rectangle
from math import pi

class Sun:
    def __init__(self, zenith, azimuth, distance):
        self.distance = distance

        self.rays = []
        self.position = v(0, 0, 0)
        self.l1 = v(1500, 0, 0)
        self.l2 = v(0, 500, 0)

        self.norm = v(sin(pi / 2 - zenith) * cos(azimuth)
            , sin(pi / 2 - zenith) * sin(azimuth), cos(pi
            / 2 - zenith))

        norm_alpha_l1 = self.l1.cross(self.norm)
        self.l1 = self.norm.cross(norm_alpha_l1)
        self.l1 = self.l1 * (1 / self.norm.norm())

        norm_alpha_l2 = self.l2.cross(self.norm)
        self.l2 = self.norm.cross(norm_alpha_l2)
        self.l2 = self.l2 * (1 / self.norm.norm())

    def throw_ray(self):
        # First, let's place the sun far enough from the
        # mirrors
        # self.position = 1000 * self.norm
        # Only useful if trying to visualize everything

        # If rays have already been thrown, better clear
        # it up
        self.rays = []

        self.rectangle = Rectangle(self.position, self.l1
            , self.l2)
        for i in range(0, int(self.rectangle.width.norm()
            /self.distance)):
```

```

for j in range(0, int(self.rectangle.height.
norm()/self.distance)):
    self.rays.append(Line(self.position + i /
        (self.rectangle.width.norm() / self.
        distance) * self.rectangle.width + j /
            (self.rectangle.
                height.norm() /
                self.distance) *
                self.rectangle.
                height, self.
                norm))

```

*Listing 5: solarangles.py*

```
from math import pi, sin, cos, acos

def solar_angles(day_of_year=21, hour_begin=4, hour_end
=20, hour_step=.2, save_to_file=False, longitude=8.9,
    latitude=.803):
    # Returns solar angles for these data
    # TODO: multiple days directly in the function

    # It is not necessary to give this options, but it
    # was useful especially in the beginning, where we
    # had to
    # be sure this program yielded the same output as the
    # basic program we already had
    if save_to_file:
        # Let's give clear filenames
        filename = str(day_of_year) + "-from_" + str(
            hour_begin) + "_to_" + str(hour_end) + "_in_"
            + str(hour_step) + \
                "-at_" + str(longitude) + "_" + str(
                    latitude)
        solardata_hours = open(filename + "_hours", "w")
        solardata_phi = open(filename + "_phi", "w")
        solardata_theta = open(filename + "_theta", "w")
    else:
        solardata = {'hours': [], 'phi': [], 'theta': []}

    minutes = 0

    # Refer to http://www.jgiesen.de/elevaz/basics/index.
    # htm for informations on calculations
    while hour_begin < hour_end:

        fractional_year = 2 * pi / 365 * (day_of_year - 1
            +
                (hour_begin - 12) / 24)
        y = fractional_year

        eqtime = 229.18 * (0.000075 + 0.001868 * cos(y) -
            0.032077 * sin(y) -
                0.014615 * cos(2 * y) - 0.040849 * sin(2
                    * y))
```

```

declin = 0.006918 - 0.399912 * cos(y) + 0.070257
        * sin(y) - 0.006758 * \
            cos(2 * y) + 0.000907 * sin(2 * y) -
            0.002697 * cos(3 * y) + \
            0.00148 * sin(3 * y)

offset = 1
time_offset = eqtime - 4 * longitude + 60 *
              offset

tst = hour_begin * 60 + minutes * 60 +
      time_offset

ha = tst / 4 - 180
ha *= pi / 180

cos_phi = sin(latitude) * sin(declin) + cos(
    latitude) * \
        cos(declin) * cos(ha)
phi = acos(cos_phi)

theta = acos(-(sin(latitude) * cos_phi - sin(
    declin)) /
    (cos(latitude) * sin(phi)))

if save_to_file:
    solardata_hours.write(str(hour_begin) + "\n")
    solardata_phi.write(str(pi / 2 - phi) + "\n")
    solardata_theta.write(str(theta) + "\n")
else:
    solardata['hours'].append(hour_begin)
    solardata['phi'].append(pi / 2 - phi)
    solardata['theta'].append(theta)

minutes += hour_step

minutes = round(minutes, 2)

if minutes >= 1:
    hour_begin += 1
    minutes = 0

if save_to_file:

```

```
        solardata_hours.close()
        solardata_phi.close()
        solardata_theta.close()
    else:
        return solardata
```

*Listing 6: main.py*

```
__author__ = 'davide'

from sun import Sun
from plane import Rectangle
from solarangles import solar_angles
from vector import Vector as v
from math import pi, sin, cos
from collections import OrderedDict

import sys

def get_angles(day, hour_begin, hour_end, hour_step,
               angle_from_north):
    # Get angles from implemented function and modulates
    # by the angle of the system from the north (20
    # degree in our case)
    angles = solar_angles(day, hour_begin, hour_end,
                           hour_step)
    for i in range(len(angles['theta'])):
        angles['theta'][i] -= to_radians(angle_from_north)
    )

    return angles

def to_radians(angle):
    return angle * pi / 180

def to_degrees(angle):
    return angle * 180 / pi

def create_mirror(zenith, position, mirror_length,
                  mirror_height):
    # Generates a mirror knowing its zenith angle,
    # position, and dimension
    zenith = to_radians(zenith)
    half_l1 = v(mirror_height/2 * cos(zenith), 0,
                 mirror_height/2 * sin(zenith))
    l2 = v(0, mirror_length, 0)
    return Rectangle(position - half_l1, 2 * half_l1, l2)

def create_sun(zenith, azimuth, distance):
    return Sun(zenith, azimuth, distance)
```

```

def create_collector(position, length):
    # Collector's zenith shouldn't ever change, so it's
    # hardcoded
    angle = to_radians(18)
    height = 158

    return Rectangle(position, v(length, 0, 0), v(0, -sin
        (angle) * height, cos(angle) * height))

def count_intersections(day, hour_begin, hour_end,
    hour_step, angle_from_north, mirror_zenith,
    mirror_position,
        mirror_length, mirror_height,
        distance):
    # Counts the number of rays that hit the collector
    # after having been reflected by a mirror.
    # every parameter is self-explanatory but distance,
    # which is the distance between two rays
    angles = get_angles(day, hour_begin, hour_end,
        hour_step, angle_from_north)
    counter = 0
    mirror = create_mirror(mirror_zenith, mirror_position
        , mirror_length, mirror_height)
    collector = create_collector(v(0, 0, 0), 975)

    for i in range(len(angles['theta'])):
        sun = create_sun(angles['phi'][i], angles['theta'
            ][i], distance)
        sun.throw_ray()
        for ray in sun.rays:
            if mirror.intersection_point(ray):
                # If there is an intersection, change the
                # ray into its reflection
                ray.point = mirror.intersection_point(ray
                    )
                ray.reflect(mirror.normal)
            else:
                # Otherwise, don't check if it hits the
                # collector
                continue
            if collector.intersection_point(ray):
                counter += 1

```

```

        else:
            continue

    return counter

def simulation(angle_begin, angle_end, angle_step,
              day_begin, day_end, hour_begin, hour_end, hour_step,
              distance):
    # Uses count_intersection to calculate the rays that
    # hit the collector through multiple days for
    # different zenithal angles
    days = {}
    angle_0 = angle_begin
    for i in range(day_begin, day_end):
        angle_begin = angle_0
        days[i] = {}
        while angle_begin <= angle_end:
            days[i][angle_begin] = count_intersections(i,
                hour_begin, hour_end, hour_step, 20,
                angle_begin, v(840, 0, 50), 160, 50,
                distance)
            angle_begin += angle_step

    sums = {}
    for angle in days[day_begin]:
        sums[angle] = 0
        for day in days:
            sums[angle] += days[day][angle]

    return sums
    # Doing this would instead return the ordered list,
    # from the most performing angle to the least
    #return OrderedDict(reversed(sorted(sums.items(), key
    #=lambda t: t[1])))

# Same as above, but for a system of 4 mirrors
def count_intersections_4(day, hour_begin, hour_end,
    hour_step, angle_from_north, mirror_zenith,
    first_position,
                                distance_between_mirrors,
                                mirror_length, mirror_height
                                , distance):
    # first_position is the position of the first mirror

```



```

angles = get_angles(day, hour_begin, hour_end,
                    hour_step, angle_from_north)
counter = 0
mirrors = [create_mirror(mirror_zenith,
                        first_position, mirror_length, mirror_height),
            create_mirror(mirror_zenith,
                        first_position + v(
                            distance_between_mirrors, 0, 0),
                        mirror_length, mirror_height),
            create_mirror(mirror_zenith,
                        first_position + 2 * v(
                            distance_between_mirrors, 0, 0),
                        mirror_length, mirror_height),
            create_mirror(mirror_zenith,
                        first_position + 3 * v(
                            distance_between_mirrors, 0, 0),
                        mirror_length, mirror_height)]

collector = create_collector(v(0, 0, 0), 975)

for i in range(len(angles['theta'])):
    sun = create_sun(angles['phi'][i], angles['theta']
                    ][i], distance)
    sun.throw_ray()
    for ray in sun.rays:
        for mirror in mirrors:
            if mirror.intersection_point(ray):
                ray.point = mirror.intersection_point
                    (ray)
                ray.reflect(mirror.normal)
            else:
                continue
        if collector.intersection_point(ray):
            counter += 1
        break

    return counter

def simulation4(angle_begin, angle_end, angle_step,
                day_begin, day_end, hour_begin, hour_end, hour_step,
                distance):
    days = {}
    angle_0 = angle_begin

```

```

for i in range(day_begin, day_end):
    angle_begin = angle_0
    days[i] = {}
    while angle_begin <= angle_end:
        days[i][angle_begin] = count_intersections_4(
            i, hour_begin, hour_end, hour_step, 20,
            angle_begin, v(140, 0, 50), 140, 160, 50,
            distance)
        angle_begin += angle_step

sums = {}
for angle in days[day_begin]:
    sums[angle] = 0
    for day in days:
        sums[angle] += days[day][angle]

#return sums
return OrderedDict(reversed(sorted(sums.items(), key=
    lambda t: t[1])))

if __name__ == "__main__":
    # We can run every possible simulation here
    distance = 20

    dict1 = simulation(70, 70, 1, 21, 22, 8, 15, 1, 10)

    print(OrderedDict(reversed(sorted(dict1.items(), key=
        lambda t: t[1]))))

    ,,,
    This part would be useful, for instance, when
        calculating the best angle for the entire winter,
        when we need to
    simulate up until day 365 and then cicle back to day
        1.
    dict2 = simulation(10, 80, 1, 1, 60, 8, 14, 1,
        distance)

    final_dict = {}
    for angle in dict1:
        final_dict[angle] = dict1[angle] + dict2[angle]

```

```
print(OrderedDict(reversed(sorted(final_dict.items(),
                                key=lambda t: t[1]))))''')
```