

PROGETTAZIONE OTTICA DI UN DOPPIETTO ACROMATICO

Francesco Lorenzi, Aprile 2020

Sommario

In questa relazione mi propongo di progettare e simulare un doppietto di lenti corretto per aberrazione cromatica. Particolare attenzione sarà posta per la scelta dei vetri ottima per l'ottimizzazione dell'aberrazione cromatica secondaria. Il progetto verrà effettuato all'inizio nell'approssimazione di lenti sottili e poi adattato a lenti spesse. Verranno discusse altre correzioni e infine si stimeranno gli effetti delle tolleranze di fabbricazione.

1 Introduzione e teoria

1.0.1 Aberrazione cromatica e la sua correzione

Il fenomeno indesiderato che il doppietto acromatico si propone di compensare, l'aberrazione cromatica, non è trattabile da un sistema a singola lente. L'aberrazione cromatica è causata dalla dipendenza dell'indice di rifrazione dalla lunghezza d'onda, proprietà dovuta alla presenza di assorbimenti del materiale nelle regioni prossime a quella visibile. Infatti secondo la legge di Sellmeier

$$n^2(\lambda) = 1 + \sum_i \chi_{0i} \frac{\lambda^2}{\lambda^2 - \lambda_i^2}$$

dove χ_{0i} è detto coefficiente di Sellmeier, determinato sperimentalmente, alla lunghezza d'onda λ_i .

La modellizzazione pratica di questo fenomeno passa dalla scelta di un insieme discreto di lunghezze d'onda di interesse nel progetto ottico, che

nel nostro caso saranno 3: appartengono tutte alle lunghezze d'onda di Fraunhofer (le quali sono ottenibili agevolmente in laboratorio rispetto ad altre) e sono scelte in modo da rappresentare in modo ragionevolmente equo l'intera dello spettro visibile

| | |
|---|-----------|
| C | 0.6562725 |
| d | 0.587562 |
| F | 0.4861327 |

Tabella 1: Lunghezze d'onda di progetto in μm , i colori sono solo indicativi.

rispetto ad un sistema di lenti operante con queste lunghezze d'onda, per un oggetto in asse si può calcolare la posizione dell'immagine ai diversi colori: in generale non sarà la stessa proprio a causa della legge di Sellmeier.

Un parametro riassuntivo di come l'indice di rifrazione varia ad ogni lunghezza d'onda, che trova ampie applicazioni nel progetto di lenti, è il numero di Abbe V_d , che è in genere fornito dai produttori di vetri ottici. La sua definizione è

$$V_d := \frac{n_d - 1}{n_F - n_C}$$

Questo parametro permette di rappresentare con facilità le conseguenze dell'aberrazione in termini di lunghezze focali, e sarà uno degli strumenti principali per realizzare il doppietto, come sarà osservato nella prossima sezione.

1.0.2 Leggi ottiche coinvolte nel progetto

La formula che lega la lunghezza focale ai raggi di curvatura delle lenti, è detta "formula dei costruttori di lenti" e fa notare la relazione tra f ed n , cruciale per le proprietà cromatiche del dispositivo

$$\frac{1}{f_\lambda} = (n_\lambda - 1) \left(\frac{1}{R_1} - \frac{1}{R_2} \right).$$

Ci proponiamo di annullare la differenza di lunghezza focale alle lunghezze d'onda F e C , questo perché per un qualsiasi oggetto in asse, porre $f_F = f_C$ significa far coincidere la posizione delle due immagini a F e C , come si può dedurre dalla legge di Gauss per le lenti sottili

$$\frac{1}{f_\lambda} = \frac{1}{s_{i\lambda}} - \frac{1}{s_o}$$

Osserviamo che per una singola lente vale, dato che i raggi sono fissati una volta per tutte

$$f_C(n_C - 1) = f_F(n_F - 1) = f_d(n_d - 1)$$

questa relazione aiuta ad esplicitare il rapporto tra indici di rifrazione e lunghezze focali, e può essere collegata al numero di Abbe per ottenere una condizione per avere $f_F = f_C$. Dato che la relazione appena vista vale anche per un arbitrario sistema di lenti, l'idea su cui questo intero progetto di basa è quella di combinare due lenti in serie, costruite in modo da avere una focale complessiva a d pari a quella designata, e che siano in grado di apporre contributi al Δf differenze tra i fuochi di F e C , tali da rendere la deviazione finale nulla. L'imposizione di questo vincolo implica direttamente la *condizione di Abbe*:

$$f_{d1}V_{d1} + f_{d2}V_{d2} = 0. \quad (1)$$

È chiaro che le due lenti dovranno avere vetri diversi, per garantire due V_d adeguati.

L'ultima relazione che permette di strutturare il disegno, nell'approssimazione di lenti sottili,

è quella che indica la focale di una serie di lenti, ovvero

$$\frac{1}{f_{d\text{serie}}} = \frac{1}{f_{d1}} + \frac{1}{f_{d2}}. \quad (2)$$

Combinando la (1) e la (2) si ottiene, per ogni scelta di vetri, la giusta combinazione di lunghezze focali per avere un sistema detto acromatico, in cui C ed F hanno lo stesso fuoco. Questa condizione è anche detta di assenza di aberrazione cromatica *primaria*.

1.1 Approssimazioni e limiti di validità delle equazioni di progetto

Le equazioni finora presentate sono ricavate in condizioni idealizzate in cui è possibile trascurare gli effetti dovuti allo spessore delle lenti ed alla presenza di raggi lontani dall'asse con diottri sferici. Una versione più completa del modello di lente comprende l'introduzione di due piani (detti principali) che prendono il posto del singolo piano associato alla lente sottile, dove si concentrano tutti gli effetti ottici, ovvero le deviazioni dei raggi

$$\begin{aligned} \frac{1}{f_d} &= (n_d - 1) \left(\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n_d - 1)z}{n_d R_1 R_2} \right) \\ \longrightarrow \frac{1}{f_d} &= (n_d - 1) \left(\frac{1}{R_1} - \frac{1}{R_2} \right) \end{aligned}$$

La prima focale è riferita ai piani principali della lente spessa, mentre la seconda, per lenti sottili, è il limite per lo spessore $z \rightarrow 0$. Nel progetto non valuteremo l'effetto della lente spessa, che verrà conteggiato e corretto nella fase di ottimizzazione al simulatore. Inoltre le formule illustrate considerano la focale parassiale, ovvero trascurano la superficie otticamente non ideale (sferica) del diottro che indica posizioni imperfette dell'immagine quando il fascio incidente investe la lente anche lontano dall'asse ottico. Anche l'indagine di questa approssimazione verrà delegata alla simulazione con del ray-tracer.

2 Progettazione ottica

I vincoli del progetto sono il raggio di 25mm e la lunghezza focale di -90mm . Analizziamo innanzitutto i parametri che siamo liberi di variare.

2.1 Gradi di libertà del progetto

Una volta stabilita la focale del complesso di lenti e imposta la condizione 1, le focali delle due lenti risultano determinate per ogni coppia di vetri scelta.

$$1 \wedge 2 \Leftrightarrow \begin{cases} f_{d2} = f_{dtot} \left(1 - \frac{V_{d1}}{V_{d2}}\right) \\ f_{d1} = f_{dtot} \left(1 - \frac{V_{d2}}{V_{d1}}\right) \end{cases} \quad (3)$$

Restano da determinare i raggi, per cui si ha un raggio a scelta per ogni lente. Tuttavia si sceglie di usare una configurazione cementata, per cui i parametri liberi si riducono ad uno. Una scelta del raggio, che può essere di quello verso l'oggetto, l'immagine oppure quelli centrali, fissa completamente il resto dei parametri. Una volta scelti i vetri si ha quindi un progetto ad un grado di libertà.

Alla luce delle considerazioni fatte sopra ho ritenuto utile indagare su come spendere la libertà di scelta di un raggio e dei vetri in modo da guadagnare una performance migliorata.

Parametri ottimizzabili

Per poter valutare in che direzione è interessante raffinare il doppietto è utile avere una lista delle altre metriche di performance del sistema.

2.1.1 Aberrazione cromatica secondaria

Mentre la correzione dell'aberrazione cromatica primaria si pone di annullare la differenza dei fuochi parassiali delle lunghezze d'onda C ed F , la correzione si può estendere anche al fuoco parassiale per d , avvicinando quindi il sistema ad una risposta ancora più uniforme alle varie frequenze. Per ogni vetro si definisce il coefficiente di dispersione

parziale

$$P_{dC} := \frac{n_d - n_C}{n_F - n_C}.$$

In modo analogo al numero di Abbe, il coefficiente di dispersione parziale permette di valutare il cromatismo secondario per il nostro doppietto, tramite questa espressione approssimata

$$\Delta f_{dC} := f_d - f_C \approx \frac{P_{dC1} - P_{dC2}}{V_{d1} - V_{d2}} f_d.$$

La fase di scelta dei vetri si baserà principalmente sulla minimizzazione di questo parametro, per il quale è facile avere una stima date le caratteristiche dei vetri, tramite la formula appena vista.

2.1.2 Aberrazione sferica

In questo progetto è particolarmente interessante minimizzare l'aberrazione sferica, poichè ci si pone nella situazione di oggetti in asse, in cui le altre aberrazioni del terzo ordine sono assenti. I parametri usati per quantificare l'aberrazione sferica sono W_{040} e il corrispettivo coefficiente di Seidel $S_I = 8W_{040}$. Data la difficoltà però di avere una buona stima per un sistema di lenti, delegherò l'ottimizzazione in fase simulativa, notando che una tale calibrazione non inficia la qualità del cromatismo secondario di una data coppia di vetri.

2.2 Scelte progettuali

Per poter confrontare agevolmente le varie configurazioni ed aiutare le scelte di progetto, ho scritto uno strumento di calcolo in Python.

Grazie al modulo di importazione `agf.py` (gentilmente concesso dal compagno G. De Lazzari), possiamo caricare le caratteristiche in termini di n_d , V_d e P_{dC} direttamente dal catalogo Schott.

La classe `doublet.py`, il cuore del programma, fornisce un set di metodi che permettono, una volta costruito un doppietto di focale complessiva data con vetri dati, di esplicitare automaticamente le caratteristiche più interessanti:

- `focals()` calcola le focali delle due lenti.
- `secondary_chromatism()` calcola il Δf_{dC} .
- `compute_radius(R, R_idx)`: data la specifica di uno dei tre raggi, calcola gli altri due.

Infine lo script `optimizer.py`, dopo aver caricato tutti i vetri Flint in una lista ed i vetri Crown in un'altra, grazie ad `agf.load()`, itera la creazione di doppietti cementati su tutte le coppie possibili, data una specifica iniziale di un raggio fissato. Per ogni iterazione calcola raggi e aberrazione cromatica secondaria. Filtrando solo le coppie di vetri che permettono una costruzione con raggi più grandi di un raggio limite, seleziona quella con l'aberrazione cromatica secondaria minima.

Lo strumento, per quanto migliorabile, è flessibile e molto comodo per analizzare sommariamente le prestazioni di numerosi vetri velocemente. È anche semplice integrare nuove features data la struttura modulare e la classe `Doublet`.

2.2.1 Scelta del raggio e dei vetri

Impongo un filtraggio sui vari doppietti che limiti inferiormente il raggio minimo a 40mm .

La scelta del raggio può essere ben spesa per ottenere un doppietto dalla fabbricazione semplice. Sarebbe per esempio preferibile avere piana la faccia interna di entrambe le lenti, così che la cementazione avvenga alla massima precisione. Con questa condizione, la coppia di vetri migliore (N-FK58, SF57), arriva ad avere uno dei raggi troppo piccolo (-30.28mm , -214.46mm), scarto quindi questa pur allettante configurazione.

Mantenendo la linea guida di avere un diottro piano, stabilisco che sarà quello d'*uscita* del fascio, poichè nel caso fosse quello di ingresso, per la scelta migliore dei vetri uno dei raggi sarebbe addirittura inferiore a 25mm .

La scelta dei vetri allora ricade su quei vetri la cui configurazione con il raggio descritto sopra non presenta raggi più stretti in modulo di 40mm , e rende l'aberrazione cromatica minima rispetto alle altre scelte ammissibili. La soluzione ricade sui seguenti vetri

| | n_d | v_d |
|--------|---------|-------|
| N-PK51 | 1.52855 | 76.98 |
| N-BAF4 | 1.60568 | 43.72 |

(a) Caratteristiche dei vetri scelti

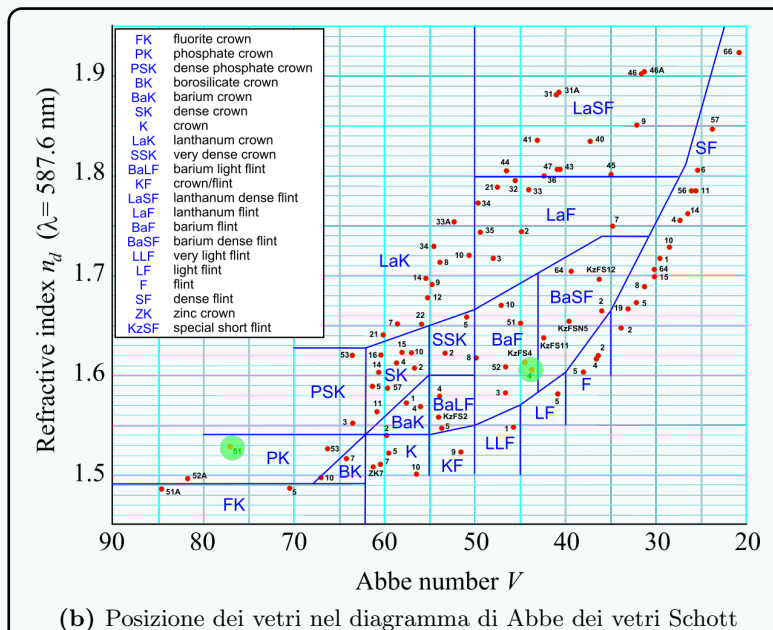


Figura 1: Scelta dei vetri

I raggi di curvatura delle superfici calcolati sono allora

$$\begin{cases} R_1 = -40.748479 \\ R_2 = +41.469408 \\ R_3 = +\infty \end{cases}$$

notiamo come siano molto ravvicinati al limite imposto sui raggi: scegliendo raggi ancora più stretti avremmo potuto migliorare ulteriormente il cromatismo secondario. Con degli spessori arbitrari la lente avrà quest'aspetto. Si nota che non è esattamente biconcava, nonostante la somiglianza fra i raggi

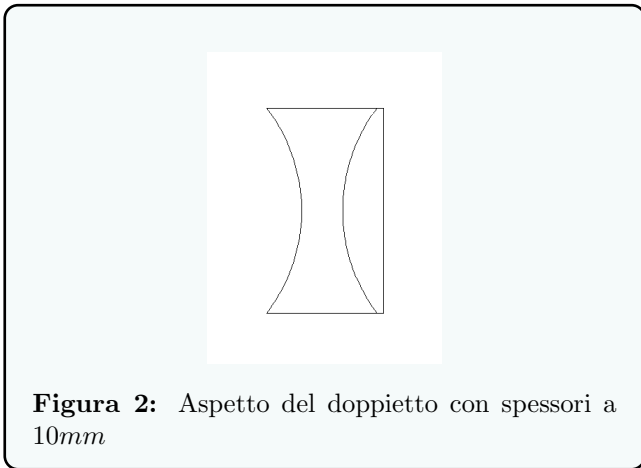


Figura 2: Aspetto del doppietto con spessori a $10mm$

Con questa scelta di raggi il cromatismo secondario calcolato dallo script è:
 $\Delta f_{AC} = 0.02013mm$.

3 Simulazione e ottimizzazione

Inserendo la soluzione trovata nel simulatore OSLO (**O**ptics **S**oftware for **L**ayout and **O**ptimization) si possono analizzare le prestazioni del sistema in modo più dettagliato di quanto possibile analiticamente, grazie al ray tracing.

Nella simulazione si è usata una lente positiva perfetta a valle del doppietto per poterne analizzare lo spot diagram. Grazie a questa lente otteniamo una convergenza del fascio, che presenta uno spot le cui caratteristiche sono esattamente quelle che vogliamo indagare per il doppietto (negativo).

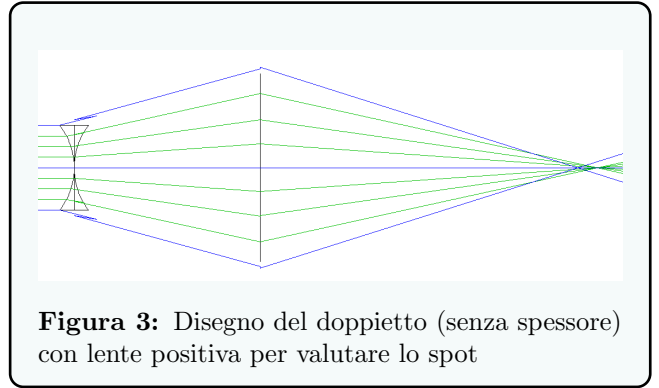


Figura 3: Disegno del doppietto (senza spessore) con lente positiva per valutare lo spot

3.1 Chromatic Focal shift

Il parametro che permette di monitorare l'aberrazione cromatica che viene fornito dal ray-tracer OSLO è la PAC (Primary Axial Color), indicata anche con il simbolo C_L e legata alla Δf secondo questa relazione

$$\Delta f = 4 \left(\frac{f_d}{\#} \right)^2 C_L.$$

È esprimibile, nel caso del doppietto, in termini delle lunghezze focali complessive alle lunghezze d'onda di progetto, congiuntamente al raggio delle due lenti, assunto identico.

Un interessante riassunto delle proprietà cromatiche del sistema è fornito dalla funzione **Chromatic Focal Shift**, che mostra come la lunghezza focale sia dipendente dalla lunghezza d'onda. Il diagramma mostra lo scarto della focale dal valore di EFL (**E**ffective **F**ocal **L**enght), calcolato internamente una volta fornite le superfici. Nel caso delle lenti sottili questo valore coincide con quello ottenuto in fase di progetto.

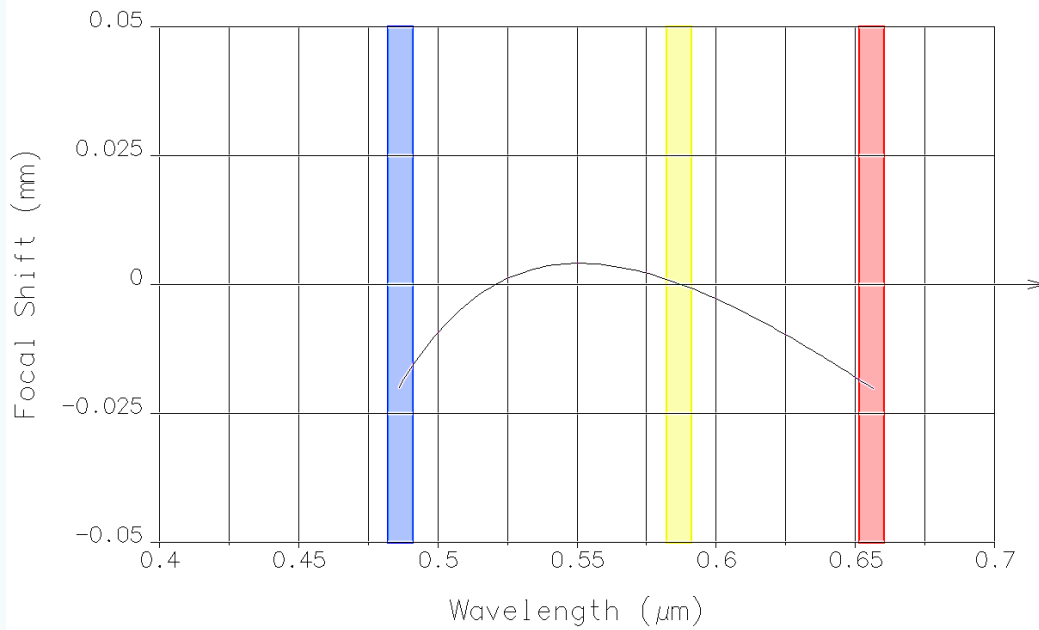


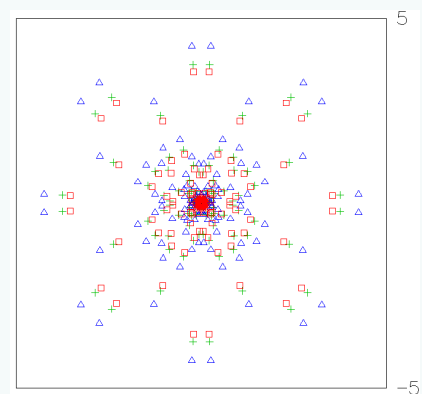
Figura 4: Spostamento della lunghezza focale alle diverse lunghezze d'onda.

Da questo diagramma possiamo verificare come la lunghezza focale sia la stessa per le lunghezze d'onda agli estremi, ovvero C ed F , mentre lo scarto della lunghezza focale sia nullo in d , evidenziata in giallo. L'obiettivo di costruire un sistema acromatico è quindi raggiunto, infatti anche la PAC è nulla.

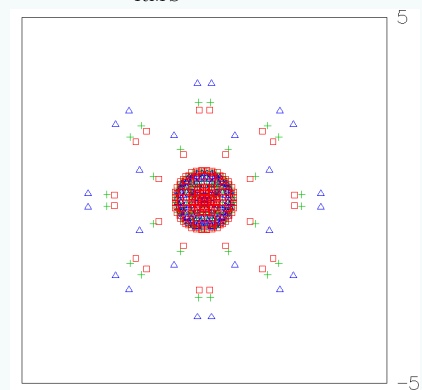
Inoltre si può notare che la differenza di lunghezza focale tra i casi a lunghezza d'onda d e C (quindi anche tra d ed F) è all'incirca di $0.020mm$: in questo modo anche il cromatismo secondario risulta compatibile con quanto previsto in analisi.

3.2 Valutazione preliminare spot diagram

Osserviamo lo spot diagram alla focale nominale della lente: presenta un raggio RMS di $1.262mm$. Infatti anche mantenendo la lente sottile, a causa dell'approssimazione parassiale, per ottenere lo spot di diametro RMS minimo è necessario allontanarsi dal fuoco parassiale. Il minimo è spostato di circa $4mm$ e presenta un raggio RMS di $0.9689mm$.



(a) Spot alla focale nominale
 $R_{RMS} = 1.262mm$



(b) Spot minimizzato
 $R_{RMS} = 0.9689mm$

Figura 5: Valutazione spot con lenti sottili

3.3 Adattamento del progetto a lenti spesse

Ora che i requisiti base del progetto sono soddisfatti, prima di procedere ad ottenere altre ottimizzazioni, rendiamo il progetto realistico, assegnando spessore di 10mm ad entrambe le lenti. Questa variazione porta la focale lontano da quella calcolata per lenti sottili, per cui è necessario un aggiustamento dei raggi, fatto in modo che l'acromatismo guadagnato non vada perso. Per ottenere ciò basta lanciare un'ottimizzazione per la lunghezza focale effettiva e per la PAC. I raggi risultanti sono

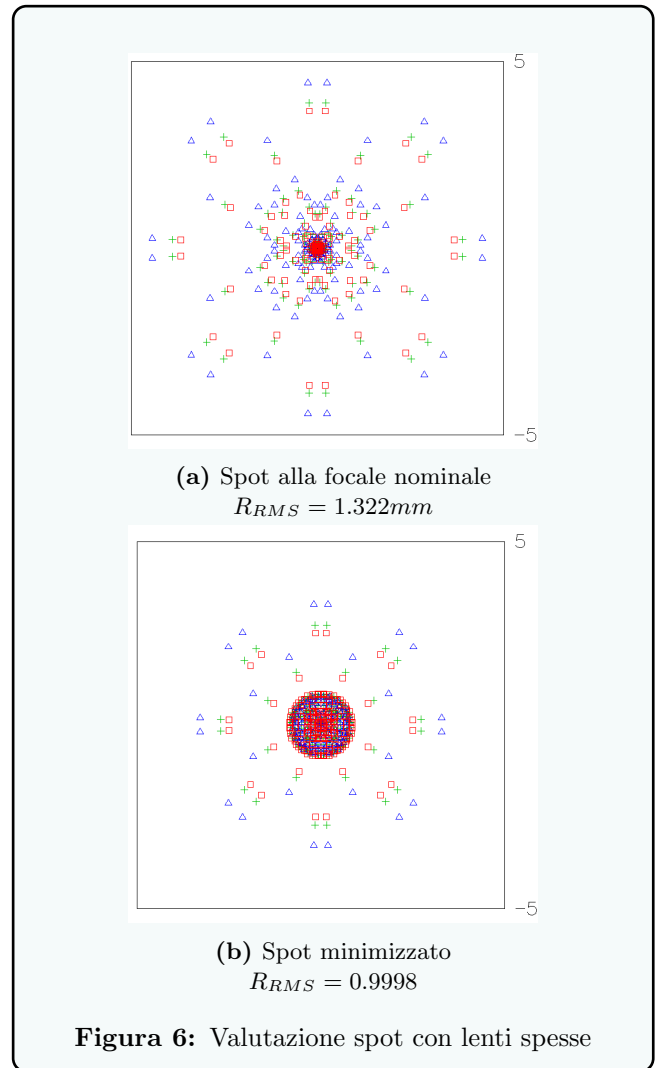
$$\begin{cases} R_1 = -40.848066 \\ R_2 = +45.754367 \\ R_3 = +\infty \end{cases}$$

mentre la performance cromatica è praticamente invariata.

3.4 Spot diagram

A questo punto è interessante osservare le dimensioni dello spot, dato che questo corrisponderebbe alla performance del sistema completo per un primo "prototipo". Dobbiamo ricordare che il nostro sistema è veloce: presenta un $f/\# = 1.8$, perciò soffre particolarmente di aberrazione sferica e aberrazioni di ordine superiore. Come nella visualizzazione preliminare il primo spot è osservato alla

focale nominale, mentre il secondo è l'ottimo per raggio RMS.



Le dimensioni dello spot sono comunque considerevoli, e variate di poco rispetto a prima. Dato che il progetto si avvicina, con l'introduzione degli spessori, ad essere realistico, ci preoccupiamo ora di minimizzare questo spot.

| Lens: No name | | | | | | Efl -90.000000 | | | | | |
|-----------------|------------|------------|-----------------|-------------|---------|----------------|--|---------------|--|----------|--|
| Ent beam radius | | 25.000000 | | Field angle | | 5.7296e-05 | | Primary wavln | | 0.587562 | |
| SRF | RADIUS | THICKNESS | APERTURE RADIUS | GLASS | SPECIAL | | | | | | |
| OBJ | 0.000000 | 1.0000e+20 | 1.0000e+14 | AIR | | | | | | | |
| AST | -40.848067 | 10.000000 | 25.000000 | N-PK51 | | | | | | | |
| 2 | 45.754367 | 10.000000 | 25.000000 | N-BAF4 | | | | | | | |
| 3 | 0.000000 | -90.000000 | 25.000000 | AIR | | | | | | | |
| IMS | 0.000000 | -13.629775 | 0.060308 | S | | | | | | | |

Figura 7: Il doppietto adattato per lenti spesse

```

*CHROMATIC ABERRATIONS
SRF    PAC      SAC      PLC      SLC
SUM  1.1102e-16  0.001625  2.8218e-09  2.2520e-09

*SEIDEL ABERRATIONS
SRF    SA3      CMA3      AST3      PTZ3      DIS3
SUM  -0.416341  2.0185e-06 -7.0221e-12 -4.8615e-12 -1.1287e-18

*fifth-order aberrations
SRF    SA5      CMA5      AST5      PTZ5      DIS5      SA7
SUM  0.482559  1.6647e-06 -2.7837e-24  4.4216e-24  1.1266e-30  0.527418

```

Figura 8: Aberrazioni residue

3.5 Tentativo di correzione dell'aberrazione sferica

Impostando la correzione anche per il parametro di aberrazione sferica (SA3), assieme al vincolo della focale effettiva e della correzione cromatica primaria, non si raggiunge l'obiettivo sperato. Un tale sistema raggiunge uno spot minimo di circa 6mm RMS, decisamente troppo. Possiamo tentare di cedere sull'imposizione del raggio di curvatura d'uscita, per provare a guadagnare l'eliminazione dell'aberrazione sferica del terzo ordine. In questa maniera si riesce a correggere anche l'aberrazione sferica, senza il bisogno di introdurre uno strato

d'aria interno. Si ottengono le misure della figura (9a). Tuttavia pur avendo praticamente eliminato l'aberrazione sferica, il design ha peggiorato il suo spot, che nella posizione del minimo RMS, comunque misura 2.074mm. Indagando ulteriormente ci si accorge di un termine di aberrazione del quinto ordine che è quasi raddoppiato rispetto alla configurazione non ottimizzata! Cercando infine di ottimizzare anche l'aberrazione del quinto ordine, si perde la correzione cromatica, presupposto primario del progetto.

Concludiamo che una miglioria delle aberrazioni sferiche è ottenibile solo introducendo uno strato d'aria.

| Lens: No name | | | | | Efl | -90.000000 |
|-----------------|------------|-------------|-----------------|---------------|----------|------------|
| Ent beam radius | 25.000000 | Field angle | 5.7296e-05 | Primary wavln | 0.587562 | |
| SRF | RADIUS | THICKNESS | APERTURE RADIUS | GLASS | SPECIAL | |
| OBJ | 0.000000 | 1.0000e+20 | 1.0000e+14 | AIR | | |
| AST | -48.156601 | 10.000000 | 25.000000 | N-PK51 | C | |
| 2 | 38.336145 | 10.000000 | 25.000000 | N-BAF4 | C | |
| 3 | 297.783535 | -90.000000 | 25.000000 | AIR | | |
| IMS | 0.000000 | -21.788380 | 2.884161 | S | | |

(a) Configurazione corretta anche per aberrazione sferica

```

*CHROMATIC ABERRATIONS
SRF    PAC      SAC      PLC      SLC
SUM  2.6341e-08  0.001636 -4.4336e-09 -2.8225e-09

*SEIDEL ABERRATIONS
SRF    SA3      CMA3      AST3      PTZ3      DIS3
SUM  -3.4862e-08  8.4423e-07 -7.6068e-12 -4.7672e-12 -3.2756e-18

*fifth-order aberrations
SRF    SA5      CMA5      AST5      PTZ5      DIS5      SA7
SUM  0.847037  1.7447e-06 -3.4976e-24  4.5601e-24  7.0561e-31  0.808909

```

(b) Aumento dell'aberrazione del quinto ordine

Figura 9: Configurazione corretta per SA3 ma difettosa in SA5

4 Analisi delle tolleranze

Consideriamo la possibilità che la lente venga effettivamente realizzata: possiamo fissare una tolleranza compatibile con le tolleranze reali di fabbricazione per verificare che le variazioni dei parametri di interesse non siano eccessive. Ipotizzando un errore di fabbricazione sui raggi di curvatura di $\pm 3mm$, possiamo direttamente simulare le conseguenze a livello di lunghezza focale e PAC, osservando cosa succede nei casi in cui gli errori assumono i valori estremi.

Se considerassimo, come ipotesi, la quantità di materiale asportato in un processo abrasivo come il parametro su cui si ha il controllo durante la fabbricazione, la tolleranza sui raggi di curvatura dovrebbe essere espressa in diverso modo: infatti mano a mano che il raggio di curvatura aumenta, una differenza di $\pm 3mm$ sul suo valore corrisponde a una quantità sempre minore di materiale asportato. Nel nostro caso la superficie 3 del doppietto non viene alterata dalla specifica di tolleranza. Supponiamo inoltre, in uno scenario di produzione in grandi quantità, che sia possibile scegliere le coppie di lenti da cementare in modo che abbiano approssimativamente lo stesso errore sul secondo raggio. Non distingueremo quindi tra il secondo raggio della prima lente e quello della seconda.

Gli intervalli di variazione per focale e PAC risultanti sono

$$f_{MAX} = -98.93mm$$

$$PAC = +0.015 \quad R_1 - 3, R_2 - 3,$$

$$f_{MIN} = -81.70mm$$

$$PAC = -0.015 \quad R_1 + 3, R_2 + 3.$$

Possiamo spingere l'analisi leggermente oltre con osservazioni analitiche: non tutti gli errori di fabbricazione sono identici, dipendono infatti dai parametri nominali scelti. Calcoliamo allora la *sensibilità*, concentrandoci sul caso della lunghezza fo-

cale. La sua sensibilità alla variazione dei raggi di curvatura è definita in questo modo:

$$S_{R_k} = \left. \frac{1}{f} \frac{\partial f}{\partial R_k} \right| \quad k \in \{1, 2\}$$

questo valore si può usare come un indice di variazione relativa in questo modo: possiamo scrivere per $\Delta R_1 \rightarrow 0$ e $\Delta R_2 \rightarrow 0$

$$\frac{\Delta f}{f} = S_{R_1} \Delta R_1 + S_{R_2} \Delta R_2 + o(\Delta R_1) + o(\Delta R_2),$$

il vantaggio di un approccio di questo tipo è quello di farci capire quali sono i parametri che più rendono il progetto "poco robusto" alle variazioni parametriche. Usando l'approssimazione di lenti sottili, per ottenere una stima iniziale, calcoliamo:

$$S_{R_1} = \frac{R_2}{R_1(R_2 - R_1)},$$

$$S_{R_2} = \frac{R_1}{R_2(R_2 - R_1)}.$$

il metodo indica una variazione di $\pm 7\%$, nel nostro caso il simulatore mostra che è appena maggiore. Questa sottostima deriva dal fatto di avere una variazione che non possiamo considerare come piccola, ammontando a quasi il 10% dei raggi iniziali, unitamente al fatto di non aver considerato una lente spessa nell'analisi.

5 Conclusioni

Il progetto di un doppietto acromatico è lungi dall'essere una pratica automatizzabile poichè offre numerose libertà di scelta che un progettista dovrebbe saper ponderare per ottenere un disegno piuttosto che un'altro. L'ottimizzazione di alcune caratteristiche richiede la conoscenza dei parametri su cui si può agire, infatti il processo più efficace di progettazione si aiuta con gli strumenti automatici solamente dopo una comprensione dettagliata del sistema.

Riferimenti bibliografici

- [1] Schott, *Datasheet N-PK51*, https://shop.schott.com/advanced_optics/Products/Optical-Glass/Optical-Glass/Phosphate-Crown/N-PK51/c/optical-glass/glass-N-PK51.
- [2] Schott, *Datasheet N-BAF4*, https://shop.schott.com/advanced_optics/Products/Optical-Glass/Optical-Glass/Barium-Flint/N-BAF4/c/optical-glass/glass-N-BAF4.

Appendice: codice Python usato nel progetto

SCRIPT *optimizer.py*

Script che invoca il metodo `load` di `agf` ed estrae, iterando su tutte le coppie di vetri *crown* e *flint* alcune metriche di performance dei doppietti.

```
import numpy as np
from typing import Dict
import agf
import doublet as dbt
import matplotlib.pyplot as plt
f_tot = -90
aperture = 25
minimum_radius = 40
fix_radius_idx = 1
radius = float('inf')
# projects wavelenghts
d = 0.587562
C = 0.6562725
F = 0.4861327

with open('results.txt', 'a') as out:
    out.write('min radius = ' + str(minimum_radius) + ", fixed surface = " + str(
        fix_radius_idx) + ", fix radius = "+
        str(radius))

    flints = []
    crowns = []
    catalog = agf.load('schott_2017-01-20b.agf')
    for glass in catalog.values():
        if glass.v_d < 50:
            flints.append(glass)
        else:
            crowns.append(glass)

    print("-->", len(flints) * len(crowns), " glasses choice possible")

    couple = []
    for fl in flints:
        for cr in crowns:
            couple.append([cr, fl])

    doublets = []
    second_chrom = []
    k = 0

    d = dbt.Doublet('ciao', f_tot, aperture, 0, 0, couple[0][0], couple[0][1])
    doublets.append(d)
    fs = d.focals()
    radii = d.compute_radius(fix_radius_idx, radius)
    second_chrom.append(d.secondary_chromatism())
    best_sc = second_chrom[0]
    best_doublet = d
    best_radii = radii

    for k in range(1, len(couple)-1):
        d = dbt.Doublet('ciao', f_tot, aperture, 0, 0, couple[k][0], couple[k][1])
        doublets.append(d)
        fs = d.focals()
        # print(cr.n_d, "-K--F-", fl.n_d )
        # print(fs)
        radii = d.compute_radius(fix_radius_idx, radius)
        second_chrom.append(d.secondary_chromatism())
```

```

        if all([abs(r)> minimum_radius for r in radii]) and (abs(second_chrom[-1]) <
                                                           abs(best_sc)):
            print(d, "-----" , radii)
            best_sc = second_chrom[-1]
            best_doublet = d
            best_radii = radii

    print("\n", best_doublet, " best secondary-chromatism glass choice ", best_sc)
    print("radii: ", best_radii)
    print("crown: ", best_doublet.crown.n_d, best_doublet.crown.v_d)
    print("flint: ", best_doublet.flint.n_d, best_doublet.flint.v_d)
    out.write('-->'+ str(best_doublet)+ ' Sec chr = ' + str(best_sc) + '\n -->
                Radii = ' + str(best_radii) + '\n')

    out.close()

```

CLASSE *doublet.py*

Scritta per gestire le caratteristiche dei vetri scelti, ospita alcuni metodi statici per computare le grandezze di interesse.

```

import agf
import math
class Doublet(object):
    def __init__(self, name: str, design_focal: float, aperture: float,
                cr_length: float, fl_length: float,
                crown: agf.Glass, flint: agf.Glass):
        self.name = name
        self.design_f = design_focal
        self.aperture = aperture
        self.cr_length = cr_length
        self.fl_length = fl_length
        self.crown = crown
        self.flint = flint
        self.f1 = 0
        self.f2 = 0
    # all focals referred to d wavelength
    def focals(self) -> [float, float]:
        f_2 = self.design_f * (1 - self.crown.v_d/self.flint.v_d)
        f_1 = -self.flint.v_d/self.crown.v_d * f_2
        self.f1 = f_1
        self.f2 = f_2
        return [f_1, f_2]

    def secondary_chromatism(self):
        return -(self.crown.P_dc() - self.flint.P_dc())/(self.crown.v_d - self.flint.
            v_d) * self.design_f

    @staticmethod
    def __fwd(f, nd, R):
        return 1/ (1/(f*(nd-1)) + 1/R)

    @staticmethod
    def __bkw(f, nd, R):
        return - 1/ (1/(f*(nd-1)) - 1/R)

    def compute_radius(self, R_idx: int, R: float) -> [float, float, float]:
        [f_1, f_2] = self.focals()
        radii = []
        if (R_idx == 1):
            R2 = Doublet.__bkw(f_1, self.crown.n_d, R)
            R3 = Doublet.__bkw(f_2, self.flint.n_d, R2)

```

```

        radii = [R, R2, R3]
    elif (R_idx == 2):
        R1 = Doublet.__fwd(f_1, self.crown.n_d, R)
        R3 = Doublet.__bkw(f_2, self.flint.n_d, R)
        radii = [R1, R, R3]
    elif (R_idx == 3):
        R2 = Doublet.__fwd(f_2, self.flint.n_d, R)
        R1 = Doublet.__fwd(f_1, self.crown.n_d, R2)
        radii = [R1, R2, R]
    else:
        raise("Radius index must be in [1, 2, 3]!")

    return radii

def __str__(self):
    return f"crown = {self.crown.name} flint = {self.flint.name}"

```

CLASSE *agf.py*

Su gentile concessione di Giacomo De Lazzari, autore del modulo di importazione *agf.py*.

```

import math
from typing import Dict

# reference wavelengths
W_C = 0.65627
W_d = 0.58756
W_F = 0.48613

# class representing a glass material
class Glass(object):

    def __init__(self, name: str, n_d: float, v_d: float):
        self.name = name
        self.n_d = n_d
        self.v_d = v_d
        self.n_coeffs = []
        self.comments = ""

    # returns refraction idx at a specific wavelength w (Sellmeier)
    def n(self, w: float) -> float:
        return math.sqrt(1 + sum([m / (1 - d / math.pow(w, 2)) for m, d in self.n_coeffs])
        )

    # returns partial dispersion between d and C reference wavelength
    # computed from glass coefficients
    def P_dc(self) -> float:
        n_C = self.n(W_C)
        n_d = self.n(W_d)
        n_F = self.n(W_F)

        assert(math.isclose(n_d, self.n_d, rel_tol=1e-4))

        return (n_d - n_C) / (n_F - n_C)

    # returns computed Abbe number (v_d) from glass coefficients
    def v_d_computed(self) -> float:
        n_C = self.n(W_C)
        n_d = self.n(W_d)
        n_F = self.n(W_F)

```

```

    assert(math.isclose(n_d, self.n_d, rel_tol=1e-6))

    v_d = (n_d - 1) / (n_F - n_C)

    assert(math.isclose(v_d, self.v_d, rel_tol=1e-6))

    return v_d

def __str__(self) -> str:
    return f"{self.name}\tn_d={self.n_d} v_d={self.v_d} nc={self.n_coefs}"

# loads all glass objects from an AGF file
def load(filename: str) -> Dict[str, Glass]:
    catalog = {}
    current = None

    with open(filename) as f:
        for line in f:
            command, *args = line.replace('\n', '').split(' ')

            if command == 'NM':
                name, _, _, n_d, v_d, *_ = args
                catalog[name] = Glass(name, float(n_d), float(v_d))
                current = name
            elif current is not None:
                if command == 'GC':
                    catalog[current].comments += ' '.join(args) + '\n'
                elif command == 'CD':
                    coeffs = [float(c) for c in args if len(c) > 0]
                    pairs = zip(coeffs[::2], coeffs[1::2])
                    catalog[current].n_coefs = [(m, d) for m, d in pairs if m != 0 and d != 0]

    return catalog

```