Universidade Estadual de Campinas

Projeto

Simulação computacional de reflexão e refração de feixes de luz

Este projeto faz parte do
Desenvolvimento de um dispositivo detector de luz para um LT-STM

Autor: Edivar Felipe de Carvalho

Orientado por
Luiz Fernando Zagonel

Conteúdo

1 Resumo

Este documento sintetiza o projeto que o aluno realizou durante o primeiro semestre de iniciação científica. A IC abrange os temas de ótica geométrica de raios, com reflexão e refração, e também a parte computacional, que abrange a forma que os fenômenos físicos serão convertidos em códigos para o computador. A linguagem Python, de código aberto e com várias bibliotecas com várias funções disponíveis, foi escolhida para a a iniciação científica.

Palavras-chave: Ótica geométrica, programação, Python, instrumentação, catodoluminescência

2 Introdução

A iniciação científica tem como objetivo dimensionar equipamentos a serem usados junto com um LT-STM, ou *Low-Temperature Scanning Tunneling Microscopy*. Dentro do microscópio será colocada uma amostra, devido uma corrente elétrica, ela emitirá luz [1] (mais detalhes serão explicados na seção Experimento). A IC em si consiste em dimensionar os equipamentos de forma a coletar a maior quantidade de luz possível dentro das limitações físicas do microscópio.

3 Metodologia

3.1 Teoria

3.1.1 Refração em lentes

Primeiramente, o aluno focou em revisar a teoria acerca das simulações, ótica geométrica. A ótica de lentes lida com um raio ótico com duas propriedades: A sua distância ao eixo ótico (centro da lente) e o ângulo do feixe com o mesmo [2] [3]. A Figura 1 mostra tais propriedades.

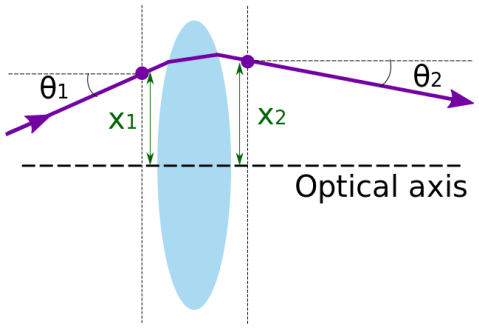

Figura 1: Exemplificação de um feixe com propriedades x e θ antes e depois de passar por uma lente.

Em forma matricial, o feixe pode representado como uma matriz coluna:

$$feixe = \begin{pmatrix} x \\ \theta \end{pmatrix}$$

A propagação do feixe pelo espaço, ou a refração por uma lente, podem ser representados por matrizes atuando no feixe. A matriz que representa a refração por uma lente é, por exemplo:

$$Lente = \begin{pmatrix} 1 & 0 \\ \frac{-1}{f} & 1 \end{pmatrix}$$

Onde f é a distância focal da lente. Para encontrarmos os novos x e θ quando o feixe passa por uma lente, fazemos a multiplicação matricial usual:

$$\begin{pmatrix} x_2 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{-1}{f} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \theta_1 \end{pmatrix} = \begin{pmatrix} x_1 \\ \theta_1 - \frac{x_1}{f} \end{pmatrix}$$

3.1.2 Reflexão no espelho

A Figura 2 ilustra como ocorre a reflexão dos raios no espelho.

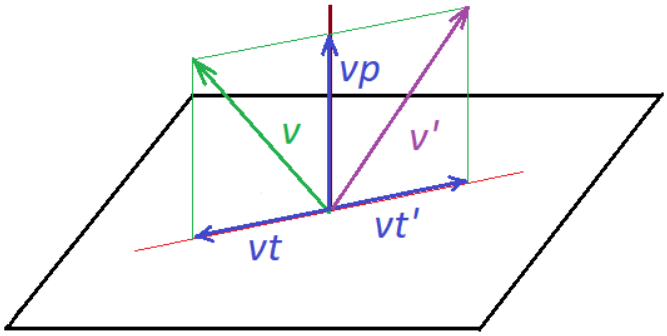

Figura 2: Ilustração da decomposição do vetor que representa um feixe quando ele atinge o espelho.

Podemos pensar no vetor perpendicular a superfície do espelho e no plano que este vetor gera. Para um raio que atinge o espelho, podemos decompô-lo em dois: o na direção do vetor perpendicular (*vp*), e o contido no plano supracitado (*vt*). A reflexão se dá mudando o sentido de *vt*, e sem alterar *vp*.

3.1.3 Programação

Além disto, o aluno revisou fundamentos da linguagem de programação utilizada nas simulações, *Python*. Tais fundamentos são a forma que se usa iterações na linguagem, como o *for* e o *while*, além do uso de outras funções pré-carregadas. Algumas destas são a *range(ini,fim)*, que cria uma lista de números que começa em *ini* e termina em *fim*; e a função *len(vet)*, que retorna o número de elementos (*lenght*) no vetor *vet*.

3.2 Experimento

A experiência a ser realizada consiste em guiar a luz emitida por uma amostra quando esta é submetida a uma corrente; tal amostra está dentro de um microscópio do tipo LT-STM. A corrente que passará na amostra é a corrente de tunelamento utilizada pelo microscópio quando este varrer a amostra. A Figura 3 tem uma esquematização da experiência.

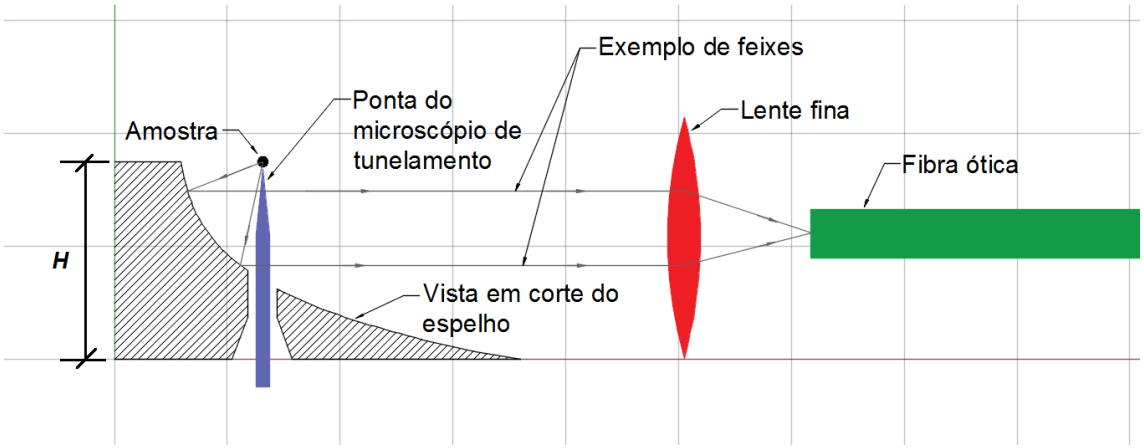

Figura 3: Esquematização da experiência que ocorrerá dentro do microscópio.

O espaço no microscópio para posicionar o espelho é muito limitado; é fundamental realizar simulações para otimizar o uso do pouco espaço disponível. Tais simulações visam guiar a construção de um sistema de coleção de luz de alto desempenho para o microscópio. As simulações procuram variar alguns parâmetros do espelho (como comprimento, largura, etc.) de forma que a quantidade de luz recebida pela fibra ótica seja máxima.

A Figura 4 mostra o espelho em corte na Figura 3.

3.3 Simulações

A iniciação em questão é uma continuação de um trabalho já iniciado. Em Outubro foram realizadas mais simulações com o código que já havia sido desenvolvido. Junto com o código desenvolvido foram criadas mais

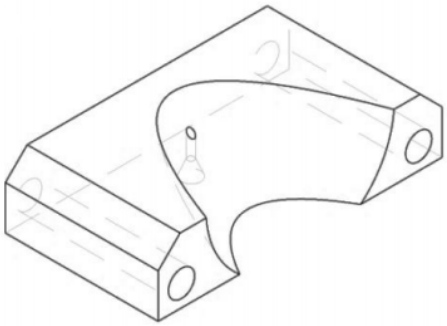

Figura 4: Ilustração do espelho que será utilizado na experiência.

funções para realizar o tratamento dos dados obtidos, gerando gráficos que permitam realizar a escolha que traga mais eficiência do projeto. Tais dados serão discutidos adiante.

3.4 Novo Código

A iniciação em questão pretende dar mais complexidade às simulações do sistema ótico do projeto. Pelo modo como o código foi feito, é difícil realizar mudanças nas posições dos elementos óticos como, por exemplo, introduzir uma falha de alinhamento da lente, e ver como isso afeta a eficiência do circuito ótico.

Em vista destas dificuldades, o estudante reescreveu as funções de propagação dos feixes de luz, tornando-o modular, e utilizando *objetos*, uma entidade que pode adquirir propriedades facilitam tanto a escrita quanto a compreensão do código.

4 Resultados e discussões

4.1 Simulações

Temos a seguir uma função que foi escrita para gerar alguns dos gráficos no experimento. Em tal função mudamos dois parâmetros do espelho, a altura (H na Figura 3) e o parâmetro da parábola, que está relacionado a largura dela.

```
1  import matplotlib.pyplot as plt
2  def graficar(pasta, rf, M):
3
4      tx = len(M[0])
5      ty = len(M)
6      datax = [7.8*x/(tx-1) for x in range(tx)]
7      datay = [5*y/(ty-1) for y in range(ty)]
8
9      imagem = plt.pcolormesh(datax, datay, M)
10
11     plt.xlim(0, datax[tx-1])
12     plt.ylim(0, datay[ty-1])
13     plt.xlabel("Mirror's height (H)")
14     plt.ylabel("Parabola's parameter (p)")
15     plt.text(4, 5.2, "Mirror's efficiency [%], hole with " + str(2*rf),
16     horizontalalignment='center', fontsize=16)
17     plt.colorbar()
18     plt.show()
19     plt.savefig('dados/' +pasta+'/furo '+ str(rf)+ 'mm.png')
20
```

As linhas 4 e 5 tratam de obter quantos valores de altura e do parâmetro da parábola foram utilizados. As linhas 6 e 7 calculam os valores que serão utilizados nos eixos do gráfico. Utilizamos a biblioteca *Matplotlib* para fazer o gráfico, e este é gerado na linha 9. As linhas 11 a 17 são características do gráfico sendo ajustadas. A linha 18 mostra a figura e a 19 o salva em uma imagem.

A Figura 5 mostra um gráfico gerado pela função *graficar*. Considerando a fonte de luz como isotrópica, a eficiência é calculada como o percentual dos feixes de luz saindo da fonte que atingem a fibra ótica.

4.2 Novo Código

Temos a seguir um exemplo de um objeto escrito para linguagem Python:

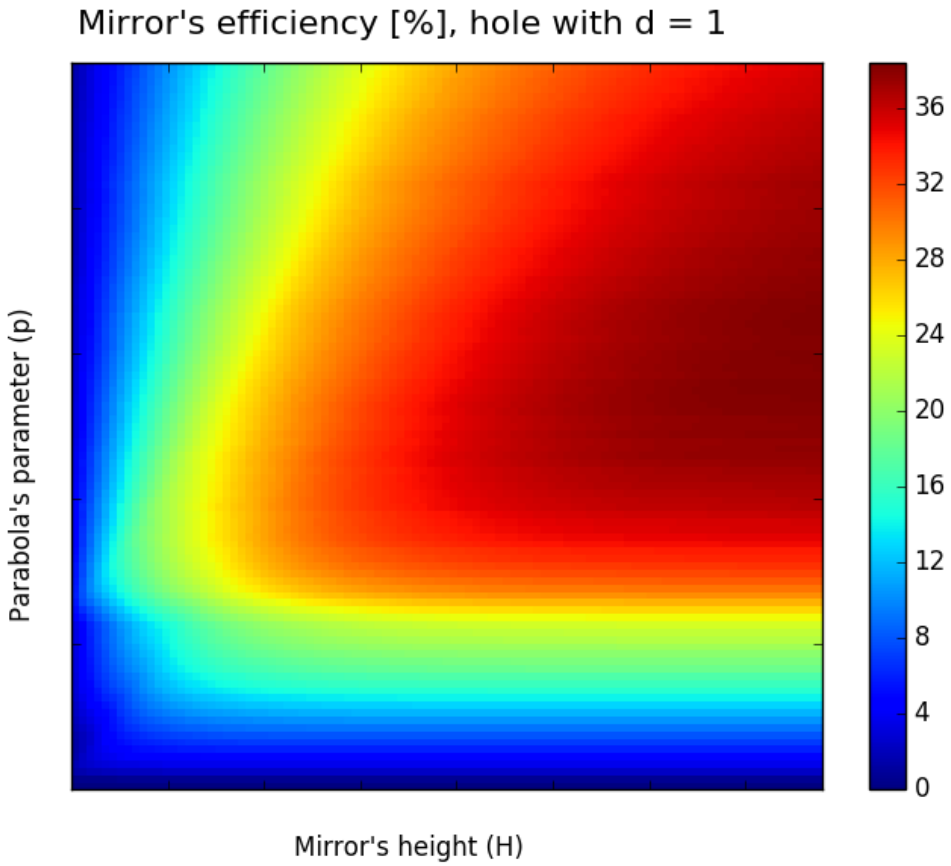

Figura 5: Exemplo de gráfico gerado pelo código feito durante a iniciação científica.

```
class Feixe:

    def __init__(self, theta, phi):
        self.theta = theta
        self.phi = phi
        self.refletido = None
        self.pos_rfl = None
        self.v_rfl = None
        self.refratado = None
        self.pos_rfr = None
        self.v_rfr = None
        self.na_fibra = None

```

Conforme o código vai calculando se o feixe foi refletido pelo espelho, se foi refratado pela lente, etc. ele vai preenchendo as informações dos feixes. Se um feixe é refletido, por exemplo, *self.refletido* (self significa que esta variável pertence ao próprio feixe) recebe o valor *True*, e também uma variável do tipo vetor que salva onde ele foi refletido, *self.pos_rlf*, e a direção que ele irá percorrer, *self.v_rfl*.

Um exemplo é mostrado a seguir:

```
feixe = Feixe(0.1, 0.5)

```

Neste caso, *feixe* tem como ângulo theta (com um sistema de coordenadas) 0.1, e phi, 0.5. As outras variáveis vão sendo calculadas durante a execução do código. A Figura 6 mostra a fonte que foi gerada, com cada ponto representando um feixe.

A Figura 7 mostra a posição em que os feixes foram refletidos no espelho.

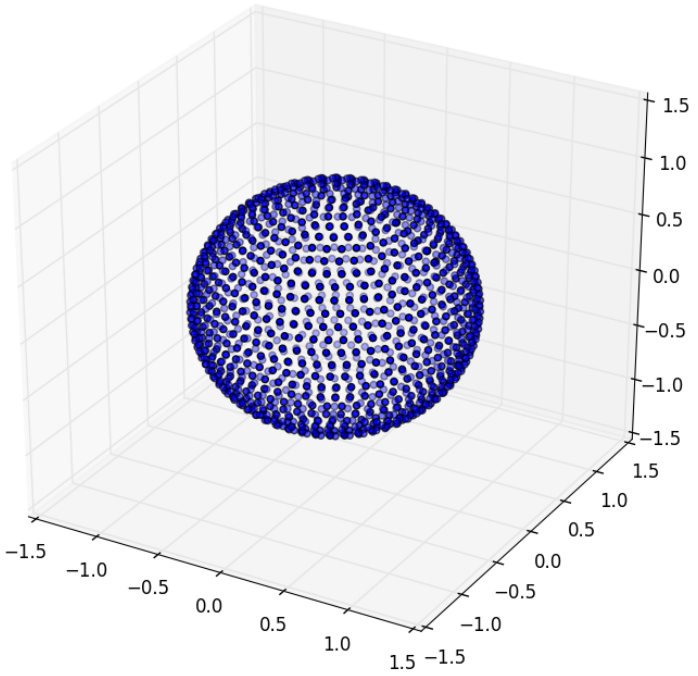

Figura 6: Gráfico com os feixes da fonte pontual.

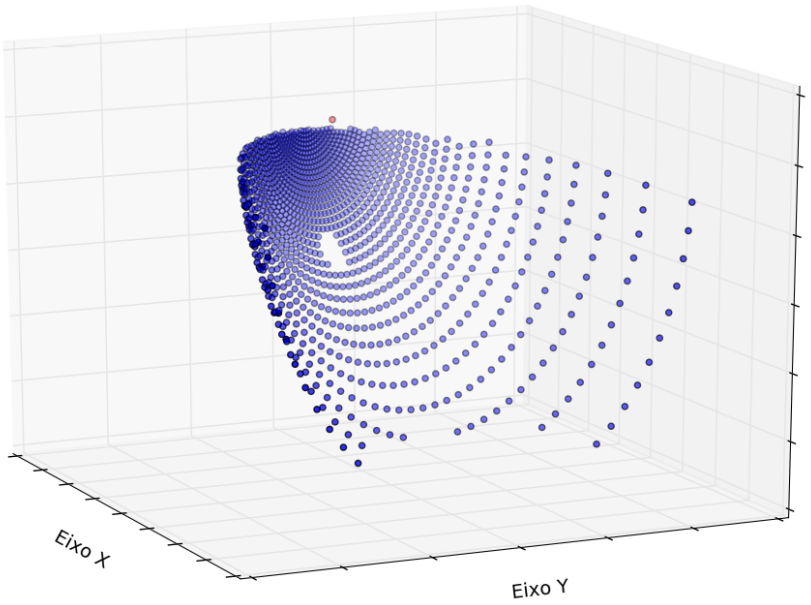

Figura 7: Gráfico a posição em que os feixes foram refletidos no espelho. O ponto vermelho representa a posição da fonte.

Exemplos do que pode ser feito com tal código estão nas Figuras 8 e 9. Na Figura 8 foram plotados os pontos no espaço onde os feixes que sofreram reflexão foram refletidos. Um plot destes é altamente facilitado com o novo código, e ajuda muito na procura por *bugs*. Na Figura 9 foi testada a refração causada pelo código em feixes normais a superfície da lente. Pode-se ver que os feixes não convergem todos exatamente para o mesmo ponto; tal efeito é esperado e é conhecido como *aberração esférica*.

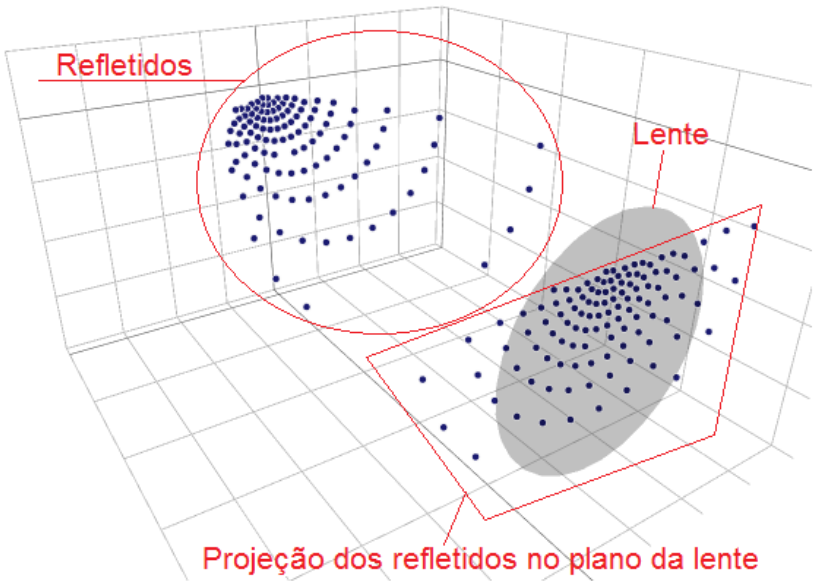

Figura 8: Exemplo de gráfico gerado pelo código feito durante a iniciação científica. Temos os feixes refletidos no espelho e um circulo translúcido representando a lente. Temos também pontos representando a posição dos feixes quando eles estão no mesmo plano que a lente.

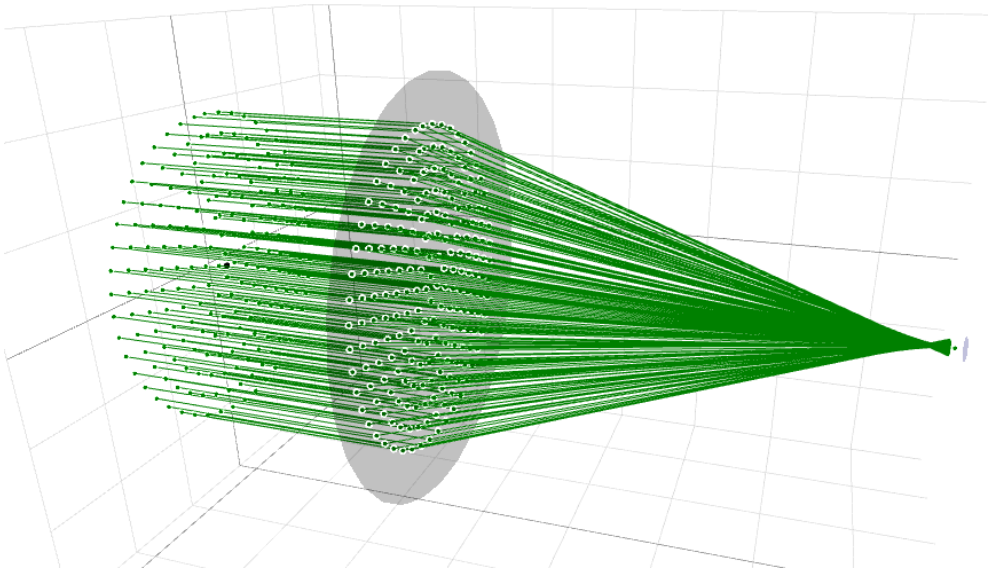

Figura 9: Simulação gerada para verificar se o código estava refratando os feixes corretamente.

5 Conclusões e Perspectivas

Com o novo código várias ferramentas para visualizar as simulações puderam ser implementadas, dando muito mais confiabilidade ao código. Este até agora foi reescrito de forma a alcançar o que já havia sido feito. Junto com as novas ferramentas gráficas, tem-se observado se o novo código condiz com o esperado teoricamente. Os próximos passos envolvem a complexidade adicional que se pretende: Introduzir imprecisões propositais e verificar como tais alterações afetam a eficiência do sistema ótico.

6 Referências

[1] *"Photon emission with the scanning tunnelling microscope"* J. K. Gimzewski, B. Reihl, J. H. Coombs, R. R. Schlittler, Z. Phys. B – Condensed Matter, vol. 72, 497, 1988.

[2] https://en.wikipedia.org/wiki/Ray_transfer_matrix_analysis

[3] http://www.colorado.edu/physics/phys4510/phys4510_fa05/Chapter4.pdf

7 Considerações

Meu orientador concorda com o expressado neste relatório e deu a seguinte opinião:

"O estudante desempenhou as atividades do projeto acima das expectativas. Ele desenvolveu um novo software e implementou ferramentas de visualização inovadoras que ajudam a entender o efeito de desalinhamentos e a resolver e prever problemas. Também executou rotinas que permitiram definir qual é o melhor espelho para ser utilizado no projeto e a compra do mesmo está sendo feita."

E o horário do evento de consulta à comunidade escolhido foi 17-19h.