

# Superpositions of Bessel Beams and Optical Force Modeling in Python

Ivan E. L. Rodrigues, Carlos H. Silva Santos, Leonardo André Ambrosio.

**Resumo**—A complexidade lógica e matemática requerida nas modelagens do eletromagnetismo computacional demandam o desenvolvimento de ferramentas que possibilitem o processamento numérico. Nesta área do conhecimento vem sendo estudada a Teoria Generalizada de Lorenz-Mie (GLMT – Generalized Lorenz-Mie Theory), a qual possibilita descrever o espalhamento de luz criada a partir de partículas ideais atingidas por um laser Gaussiano. Apresentamos então uma nova ferramenta para atender esta complexa tarefa de modelagem numérica em relação a GLMT, especialmente no tema de feixes de Bessel, e voltadas aos cálculos de forças ópticas utilizando a linguagem Python e seus recursos.

**Keywords**— *Eletromagnetismo Computacional, Métodos Numéricos, Teoria Generalizada de Lorenz-Mie, Forças Ópticas, Feixes de Bessel, Linguagem Python.*

## I. INTRODUÇÃO

O Eletromagnetismo Computacional é uma área que requer sofisticados recursos computacionais por haver soluções, tanto analíticas quanto numéricas, que são complexas matematicamente. Também, nesse contexto, destaca-se sua natureza multidisciplinar envolvendo Física, Matemática e Computação que, por consequência, requerem profissionais qualificados para o desenvolvimento. Além disso, sob a perspectiva computacional, passam a ser necessários recursos computacionais que simplifiquem o cotidiano desses profissionais e a capacitação de novos membros das equipes.

Todas essas afirmações se enquadram com a Teoria Generalizada de Lorenz-Mie (GLMT – *Generalized Lorenz-Mie Theory*), a qual possibilita descrever o espalhamento de luz criada a partir de partículas esféricas atingidas por um feixe Gaussiano, de Bessel ou de perfil espacial arbitrário. Para isso, esse trabalho apresenta uma nova ferramenta (*toolbox*) para atender essas modelagens numéricas tidas como complexas e voltadas aos cálculos de forças ópticas utilizando a linguagem Python e seus recursos. Para tanto a seção II traz uma breve descrição numérica da GLMT em *Frozen Waves* e cálculos de força óptica. A seção III a modelagem e os resultados numéricos obtidos com o desenvolvimento aqui proposto. Por fim, a quarta e última seção traz as considerações finais e os trabalhos em desenvolvimento por essa equipe.

## II. FROZEN WAVES E FORÇAS ÓPTICAS COM FEIXES DE BESSEL

A Teoria Generalizada de Lorenz-Mie (GLMT – *Generalized Lorenz-Mie Theory*) possibilita descrever o espalhamento de luz criado a partir de partículas esféricas homogêneas ou heterogêneas (multi-camadas) quando atingidas por um campo óptico de perfil espacial arbitrário (Gaussiano, de Bessel, etc.).

De forma sucinta, ao se considerar partículas de tamanho arbitrário e relacionando-a com equações de Maxwell, surge a

possibilidade de descrever o espalhamento de um feixe incidente arbitrário, requerendo o cálculo de coeficientes conhecidos como coeficientes de fatores de forma (*Beam-shape Coefficients* ou BSC). Isso, por consequência e na ausência de soluções analíticas para os BSCs, requer um elevado poder computacional para resolver esses cálculos que passam a contar com integrais duplas e triplas em coordenadas esféricas. Um feixe de luz alternativo e recentemente introduzido no quadro de trabalho da GLMT ficou conhecido como *Frozen Wave* (FW) [1], construído a partir da superposição discreta de feixes de Bessel (FBs) de mesma frequência e com diferentes números de onda longitudinais.

FWs vêm sendo aplicadas na construção de novos lasers para sistemas de telecomunicações fotônicos, para a detecção, aprisionamento e manipulação de moléculas biológicas ou outras aplicações que estejam diretamente relacionadas às emissões direcionais de átomos e moléculas fotônicas [2].

Nesse trabalho, construímos a teoria a partir da definição de uma função computacional (*def*) para se estabelecer o comportamento dos coeficientes constantes  $A_q$  (1).

$$A_q = \frac{1}{L} \int_{-L}^L F(z) e^{k_{z0}z} e^{i\frac{2\pi q}{L}z} dz \quad (1)$$

sendo que  $L$  é a distância máxima para o padrão de intensidade desejado da FW no eixo de propagação  $z$ ,  $k_{z0}$  é o número de onda longitudinal para  $q = 0$  dado por  $k_{zq} = \frac{n_i}{n_r} (Q + \frac{2\pi q}{L})$ , onde  $n_i$  e  $n_r$  são, respectivamente, a parte imaginária e real do índice de refração do meio externo e  $Q$  é um parâmetro constante da FW (relacionado ao grau de paraxialidade e de concentração radial de campo). A função  $F(z)$  varia dependendo do tipo de FW que está sendo tratada, podendo ter comportamento constante, exponencial, dentre outras descritas em [1]. Pelo comportamento constante dos  $A_q$ 's e seu uso frequente em outras equações na análise de FWs e BSCs, são utilizados os recursos de “listas” do Python para criar uma sequência dos coeficientes de  $q$  entre  $-N$  e  $N$  ( $N = 15$ ). Para tanto, a equação em coordenadas cilíndricas ( $\rho, \phi, z$ ) da componente transversal do campo elétrico de uma FW deslocada ao longo do eixo  $z$  por  $z_0$ , incidente sobre uma partícula centrada na origem, é dada por

$$\Psi(\rho, z; t) = e^{i\omega_0 t} e^{-iQ(z-z_0)} \sum_{q=-N}^N A_q J_0(k_{\rho q} \rho) \times e^{-i(2\pi q/L)(z-z_0)} e^{-ik_{zq} z} \quad (2)$$

Em (2),  $J_0(\cdot)$  é a função de Bessel  $J_\nu(x)$  com  $\nu = 0$  (que é acessada pela função `mpmath.besselj(v,x)` na biblioteca `Sympy`),  $\omega_0$  é a frequência angular no meio externo e  $k_{\rho q}$  é o número de onda radial. Nesse contexto, esse trabalho apresenta o uso do

<sup>1</sup> Ivan Eduardo Lage Rodrigues, Licenciando em Física, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, campus Itapetininga, E-mails: [ivan.elr93@gmail.com](mailto:ivan.elr93@gmail.com).

<sup>2</sup> Carlos Henrique da Silva Santos, Professor, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, campus Itapetininga, E-mails: [carlos.santos@ifsp.edu.br](mailto:carlos.santos@ifsp.edu.br).

<sup>3</sup> Leonardo André Ambrosio, Professor, Departamento de Engenharia Elétrica e Computação (SEL), Escola de Engenharia de São Carlos (EESC), Universidade de São Paulo (USP), E-mail: [leo@sc.usp.br](mailto:leo@sc.usp.br)

Linguagem de Programação *Python* para a modelagem de FBs e FWs, com o necessário cálculo dos BSC. Estes coeficientes que são divididos entre transverso elétricos (TE) e transverso magnéticos (TM) e têm suas formas exatas e aproximadas a partir de vários métodos. Como exemplo para este artigo, apresentamos apenas as expressões para modos TM, a partir da forma exata (*exa*) obtida em [3] e da aproximação localizada integral (*ila*) em [4]:

$$g_{n,TM}^{m,exa} = -(-1)^{\frac{(m-|m|)}{2}} \frac{(n-m)!}{(n+|m|)!} \sum_{q=-N}^N \frac{A_q}{\cos \alpha_q} \times \{i^{\nu-m+1} e^{i(\nu-m+1)\phi_0} J_{\nu-m+1}(\rho_0 k \sin \alpha_q) \times [\tau_n^m(\alpha_q) + m\pi_n^m(\alpha_q)] + i^{\nu-m-1} e^{i(\nu-m-1)\phi_0} J_{\nu-m-1}(\rho_0 k \sin \alpha_q) \times [\tau_n^m(\alpha_q) - m\pi_n^m(\alpha_q)]\} e^{ikz_q z_0} \quad (3)$$

$$g_{n,TM}^{m,ila} = \frac{Z_n^m}{2} \sum_{q=-N}^N A_q e^{ikz_q z_0} \times [J_{1+m}(\omega_q) J_{1+m-\nu}(\rho_0 k \sin \alpha_q) e^{-i(1+m)\phi_0} + J_{-1+m}(\omega_q) J_{-1+m-\nu}(\rho_0 k \sin \alpha_q) e^{-i(-1+m)\phi_0}] \quad (4)$$

Ambas as versões de BSCs em (3) e (4) dependem dos números inteiros  $m$  e  $n$  ( $-n \leq m \leq n$ ), além da posição relativa do feixe,  $(\rho_0, \phi_0, z_0)$ . Além disso,  $\alpha_q$  é o ângulo de axicon do  $q$ -ésimo FB que compõe a FW.  $\pi_n^m(\cdot)$  e  $\tau_n^m(\cdot)$  são funções generalizadas de Legendre,  $Z_n^m$  um fator dependente apenas de  $n$  e  $m$ , e  $\omega_q = (n + 0,5) \sin \alpha_q$  [5]. Na próxima seção são apresentados com mais detalhes a metodologia de desenvolvimento e alguns resultados.

### III. METODOLOGIA DE DESENVOLVIMENTO E RESULTADOS

A inclusão das bibliotecas *Scipy* oferece comandos com foco para a área científica como valor de constantes, equações recorrentes dentre outros, além de *Sympy*, biblioteca avançada de matemática e de boa precisão, *Matplotlib* para a modelagem de gráficos e tabela e *Numpy* com recursos para os métodos numéricos e a organização de dados em forma de *arrays*. Todas elas estão no pacote Anaconda, disponível gratuitamente online (Fig. 1).

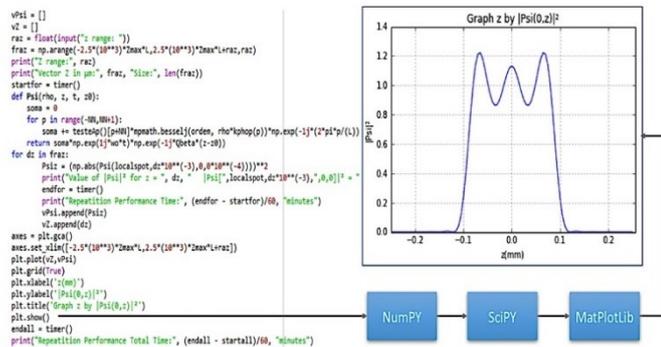
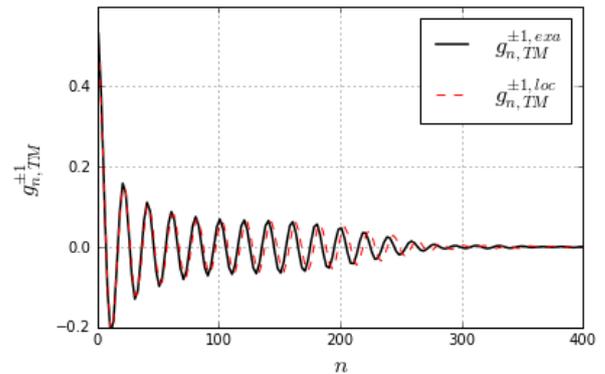


Fig. 1. Esquemático código Python com suas bibliotecas para uma FW.

Para lidar com os somatórios extensos das equações (4) e (5), foi usado um *loop* em *for* que consiste do uso de uma estrutura de repetição para iterar uma sequência de comandos um número arbitrário de vezes, neste caso igual ao número de itens em  $q \in [-N, N]$ , e o resultado de cada iteração é somado a uma variável que é ao fim de todo o *loop* apresentado como resultado do somatório.

Para validar a transcrição das equações (3) e (4) efetuamos diversos testes com diferentes configurações de FWs e valores de  $n, m, \rho_0, \phi_0, z_0$ . Por exemplo, no gráfico a seguir, reproduzimos com sucesso o gráfico da figura 1 de [5] nas mesmas condições, além de compará-lo com  $g_{n,TM}^{m,exa}$ .

Fig. 2.  $g_{n,TM}^{\pm 1}$  exatos e aproximados para uma FW constante e com  $\nu = 0$ .



### IV. CONCLUSÕES

Apesar do rigoroso formalismo matemático e multidisciplinaridade envolvida nesta tarefa, a linguagem *Python* demonstrou, durante o curso deste trabalho, uma alta precisão e flexibilidade para os casos estudados, o que permitiu uma satisfatória análise e modelagem em eletromagnetismo computacional como esperado. Além disso também provou parte do potencial da linguagem para questões relacionadas tal área. Entretanto é preciso considerar o poder de processamento necessário para tais simulações, o que implica na possível necessidade de implementação de recursos de multiprocessamento.

Seria importante futuramente analisar o tempo de compilação e execução do código em comparação com o mesmo código transcrito em outras linguagens computacionais como *Mathematica* ou *Fortran* para uma melhor análise de otimização, assim como testes das forças óticas do BB e torque óticos. Também deve-se analisar os possíveis conflitos entre as várias bibliotecas usadas, um problema que foi recorrente durante o trabalho, visto que os *arrays* do *Numpy* não são aceitos em certas funções do *Sympy*, assim como os mpfs, uma classe de objetos numéricos reais gerados por funções do *Sympy*, não são aceitos em certas funções do *Numpy*. Entretanto, pelo fato de que ambas as bibliotecas têm funções importantes e distintas no código, percebe-se uma grande dificuldade em restringir a escrever com apenas uma delas.

### AGRADECIMENTOS

L. A. Ambrosio agradece a FAPESP. I. E. L. Rodrigues e C. H. S. Santos agradecem o CNPq pela bolsa PIBIC e o IFSP, sob edital 226/2016 da PRP, pelo apoio financeiro ao projeto.

### REFERENCES

- [1] M. ZAMBONI-RACHED. *Stationary optical wave fields with arbitrary longitudinal shape by superposing equal frequency Bessel beams: Frozen waves*, Opt. Express, vol. 12, pp. 4001–4006, 2004.
- [2] D. GAGNON. *Generalized Lorenz-Mie theory: Application to scattering and resonances of photonic complexes*. Diss. Université Laval, 2014.
- [3] J. J. WANG, T. WRIEDT, I. MÄDI, R. AND Y. P. HAN. *Multinole expansion of circularly symmetric Bessel beams of arbitrary order for scattering calculations*. Opt. Communications, vol 387, pp. 102–109, 2017.
- [4] L. A. AMBROSIO, M. ZAMBONI-RACHED. *Optical forces experienced by arbitrary-sized spherical scatterers from superpositions of equal-frequency Bessel beams*. J. Opt. Soc. Am. B, vol. 32, pp. 37–46, 2015
- [5] L. A. AMBROSIO, M. ZAMBONI-RACHED. *Analytical approach of ordinary frozen waves for optical trapping and micromanipulation*. Appl. Opt., vol. 54, pp. 2584–2593, 2015