# Computational and Theoretical Analysis of Structured Light: Quadrupole Interactions in Chiral Molecules

Physics Project `PHY-6004Y` — Final Report

100280312

*27th May 2022*

**Abstract**

This report details the development of computational analysis, including visualization and graphs, of the analytical work in [redacted] PhD thesis, which concerns the inclusion of quadrupole molecular moments in the interaction between structured light (a beam possessing orbital angular momentum) and chiral molecules in a paraxial field. The final part of the report generalizes this model for non-paraxial fields.

---

***Contents***

---

***Acknowledgements***

This report is produced as part of the 3rd year "Physics Project" module, organized by [redacted].

Many thanks to [redacted] and [redacted] for supervising me on this report, especially with their expertise on the background knowledge, and thanks to [redacted] as well for additional assistance.

School of Physics, [redacted].

## 1.1 Fundamentals of QED

Quantum electrodynamics (QED) is a widely applicable theory in which the interaction between electromagnetic radiation (photons) and matter are quantized. The model of the interaction relies on perturbation theory, applicable in this context as energies from radiation fields produce a relatively small perturbation to the system compared to the intramolecular Coulomb energies[1, pp. v, 1], around 8 levels of magnitude lower for a continuous laser with an output of $10^4\,\mathrm{W\,m^{-2}}$[1, p. 74]. In this section, the necessary background knowledge of number states, the Fermi golden rule, tensors, QED methods, Laguerre-Gaussian beams and optical activity as the basis of the quadrupole model are described.

### 1.1.1 The Hamiltonian

The time dynamics in quantum mechanics are governed by the time-dependent Schrödinger equation,

$$i\hbar\frac{\partial}{\partial t}\left|\Psi(t)\right\rangle = H\left|\Psi(t)\right\rangle \tag{1}$$

where $|\Psi(t)\rangle$ is the state vector, the representation of the total system.

Equation (1) is very rarely exactly soluble, and perturbation methods are used to obtain approximate results. In QED, the Hamiltonian is generally defined as follows,

$$H = \underbrace{H_{\mathrm{mol}} + H_{\mathrm{rad}}}_{H_0} + H_{\mathrm{int}} \tag{2}$$

where $H_0$ is the unperturbed Hamiltonian, consisting of the molecular and radiation Hamiltonians and $H_{\mathrm{int}}$, the interaction Hamiltonian, is treated as the perturbation.

### 1.1.2 Number States

The state of the system $|\Psi\rangle$ is typically represented by a wave function. An alternate way, however, is to use number states, also known as "Fock states" in an approach known as *second quantization*. They are denoted as

$$\left|\Psi\right\rangle = \left|n(\boldsymbol{k},\sigma)\right\rangle \tag{3}$$

where $n$ is the occupation number[2, p. 454], the number of particles (assumed to be non-interacting) with wavevector $\boldsymbol{k}$ and polarization $\sigma$.

At the core of the second quantization formalism are the creation and annihilation operators, which add and remove a particle in a system. They are given respectively by[1, p. 36]

$$a^{\dagger(\sigma)}\left|n(\boldsymbol{k},\sigma)\right\rangle = (n+1)^{1/2}\left|(n+1)(\boldsymbol{k},\sigma)\right\rangle \qquad n\in\mathbb{N} \tag{4}$$

$$a^{(\sigma)}\left|n(\boldsymbol{k},\sigma)\right\rangle = \begin{cases} n^{1/2}\left|(n-1)(\boldsymbol{k},\sigma)\right\rangle & n\in\mathbb{Z}_+ \\ 0 & n=0. \end{cases} \tag{5}$$

These operators are identical to the "ladder" operators (raising and lowering operators) in Dirac's method of finding energies in the quantum harmonic oscillator, which add and remove a quanta of energy[3, p. 136].

This is due to the fact that the Hamiltonian for a free electromagnetic field can be written as the sum of the Hamiltonians for a one-dimensional harmonic oscillator[1, p. 35] (see future equations (29) and (59)), and quantized accordingly depending on which mode is occupied.

## 1.2 Fermi Golden Rule

The Fermi Golden rule is a measure of the rate of the probability $P$ of transition between one state and another per unit time, and is defined as[4]

$$\Gamma = \frac{\mathrm{d}P}{\mathrm{d}t} = \frac{2\pi}{\hbar} \sum_{\xi} \left| M_{fi}(\xi) \right|^2 \rho_{\mathrm{s}} \tag{6}$$

where $\hbar$ is the reduced Planck constant equal to $\approx 1.05 \times 10^{-34}$ J s.

Two factors are of importance in the Fermi Golden rule: first, the matrix element, or the quantum amplitude, which is given by

$$M_{fi}(\xi) = \langle \text{final} | \, H_{\text{int}}(\xi) \, | \text{initial} \rangle \tag{7}$$

for the first order (one photon). Secondly, the density of states, which is defined as the number of levels per unit energy[1, p. 84],

$$\rho_{\mathrm{s}} = \frac{\mathrm{d}n}{\mathrm{d}E} = \frac{1}{\hbar} \frac{\mathrm{d}n}{\mathrm{d}\omega}. \tag{8}$$

where $\omega$ is the angular frequency.

## 1.3 The Interaction Hamiltonian and Tensors

The interaction Hamiltonian used in this report is a multipole series in the form known as the Power-Zineau-Woolley Hamiltonian[5],

$$H_{\text{int}} = -\varepsilon_0^{-1} \iiint \boldsymbol{p}^{\perp}(\boldsymbol{r}) \cdot \boldsymbol{\Pi}(\boldsymbol{r}) \, \mathrm{d}^3\boldsymbol{r} - \iiint \boldsymbol{m}(\boldsymbol{r}) \cdot \boldsymbol{b}(\boldsymbol{r}) \, \mathrm{d}^3\boldsymbol{r} + \frac{1}{2} \iiiint\!\!\iiint \mathbf{O}(\boldsymbol{r}, \boldsymbol{r}') \boldsymbol{b}(\boldsymbol{r}) \boldsymbol{b}(\boldsymbol{r}') \, \mathrm{d}^3\boldsymbol{r} \, \mathrm{d}^3\boldsymbol{r}' \tag{9}$$

which can be split into a sum of specific multipole interactions,

$$\begin{aligned} H_{\text{int}} = \sum_{\xi} \big[ & -\varepsilon_0^{-1} \mu_i(\xi) \, d_i^{\perp}(\boldsymbol{R}_\xi) - \varepsilon_0^{-1} Q_{ij}(\xi) \, \nabla_j d_i^{\perp}(\boldsymbol{R}_\xi) - m_i(\xi) \, b_i(\boldsymbol{R}_\xi) \big] \\ & + \frac{e^2}{8m} \sum_{\xi,\alpha} \big[ (\boldsymbol{q}_\alpha(\xi) - \boldsymbol{R}_\xi)_i b_j(\boldsymbol{R}_\xi) \varepsilon_{ijk} \big]^2 + \cdots . \end{aligned} \tag{10}$$

From the left, the terms consist of the electric dipole interaction, the electric quadrupole interaction, the magnetic dipole interaction, and the lowest-order diamagnetic coupling interaction.

For background sections up to §1.8, we will use the electric dipole approximation, which takes only the first term and ignores higher multipole interactions, leaving the Hamiltonian for the interaction coupling as[1, p. 68]

$$H_{\text{int}}(\xi) = -\varepsilon_0^{-1} \mu_i(\xi) d_i^{\perp}(\boldsymbol{R}_\xi) \tag{11}$$

This equation is usually a good approximation as the radiation wavelength is usually larger than the dimensions of the molecules, and so transition probabilities are in large affected by the electric dipole transition moment only. However, this approximation does have caveats, for example, when dealing with shorter wavelengths (like in X-ray spectroscopy) or when dealing with magnetic interactions[6].

In equation (10) we introduce *tensor index notation* (also known as Einstein notation) to denote tensors. Tensors are mathematical objects and can be considered a generalization of scalars (tensor of rank 0) and vectors (tensor of rank 1). A tensor of rank 2 is known as a "dyad", contains a magnitude and two directions with each direction spanning over one dimension of space, and can be represented by matrix. Further ranks can be represented by multi-dimensional arrays. In general, a rank $n$ tensor in a $m$-dimensional space will have $m^n$ components.[7]

A tensor will have indices up to their rank. For example, a vector in 3D space can be written as

$$\boldsymbol{u} = u_1\hat{\boldsymbol{x}} + u_2\hat{\boldsymbol{y}} + u_3\hat{\boldsymbol{z}} = \sum_{n=1}^{3} u_n\hat{\boldsymbol{e}}_n = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{12}$$

where $\{\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}\} \equiv \{\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3\}$ are the basis vectors. A rank 2 tensor in 3D space can similarly be constructed as

$$\mathbf{T} = T_{11}\hat{\boldsymbol{x}}\hat{\boldsymbol{x}} + T_{12}\hat{\boldsymbol{x}}\hat{\boldsymbol{y}} + T_{13}\hat{\boldsymbol{x}}\hat{\boldsymbol{z}} + T_{21}\hat{\boldsymbol{y}}\hat{\boldsymbol{x}} + \cdots = \sum_{n=1}^{3}\sum_{m=1}^{3} T_{mn}\hat{\boldsymbol{e}}_m\hat{\boldsymbol{e}}_n = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \tag{13}$$

where $\{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}\hat{\boldsymbol{y}}, \cdots\}$ are the unit dyads[8].

In tensor index notation, summations are implied, and in isolation, a tensor can simply be referred to as $u_n$ or $T_{mn}$. By supressing the summation sign, certain calculations are made more convenient. For example, certain vector operations can now be expressed using tensor index notation[9]. The dot product of two unit vectors can be expressed using the Kronecker delta,

$$\hat{\boldsymbol{e}}_i \cdot \hat{\boldsymbol{e}}_j = \delta_{ij} \tag{14}$$

where

$$\delta_{ij} = \begin{cases} 1 & i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{15}$$

and so for any two vectors, we have

$$\boldsymbol{u} \cdot \boldsymbol{v} = (a_i\hat{\boldsymbol{e}}_i) \cdot (b_j\hat{\boldsymbol{e}}_j) = a_ib_j\hat{\boldsymbol{e}}_i \cdot \hat{\boldsymbol{e}}_j = a_ib_j\delta_{ij} = a_ib_i \tag{16}$$

where in the final step, we performed *Kronecker delta reduction*, which is done by removing $\delta_{ij}$ and setting $i = j$ in the rest of the equation. This can also be used to reduce expressions such as $\delta_{ij}\delta_{jk}\delta_{ki} = \delta_{ii} = 3$.

The cross product can be expressed using the Levi-Civita epsilon,

$$\hat{\boldsymbol{e}}_i \times \hat{\boldsymbol{e}}_j = \varepsilon_{ijk}\hat{\boldsymbol{e}}_k \tag{17}$$

where

$$\varepsilon_{ijk} = \begin{cases} 1 & \{i,j,k\} = \{1,2,3\} \text{ or } \{2,3,1\} \text{ or } \{3,1,2\}, \\ -1 & \{i,j,k\} = \{3,2,1\} \text{ or } \{2,1,3\} \text{ or } \{1,3,2\}, \\ 0 & \text{otherwise}. \end{cases} \tag{18}$$

and for any two vectors,

$$\boldsymbol{u} \times \boldsymbol{v} = (a_i \hat{\boldsymbol{e}}_i) \times (b_j \hat{\boldsymbol{e}}_j) = a_i b_j \hat{\boldsymbol{e}}_i \times \hat{\boldsymbol{e}}_j = a_i b_j \varepsilon_{ijk} \hat{\boldsymbol{e}}_k. \tag{19}$$

## 1.4 Rotational Averaging of Tensors

Certain tensor quantities have an angular dependence, like the electric dipole moment operator $\boldsymbol{\mu}$ used in the electric dipole approximation (11) for the interaction coupling. This first-rank tensor depends on the orientation of the molecule, which are in general randomly orientated in gases and liquids. By performing a *rotational average* across each molecule $\xi$, this dependence can be removed to obtain an end result which is an expression independent on the angle of incident radiation or of the molecule.

To do this, the components of the tensor are transformed from the laboratory frame, usually denoted with Latin indices $i, j, k, ...$ to the molecular frame, usually denoted with Greek indices $\lambda, \mu, \nu, ...$ using an angle matrix denoted $l_{i_n \lambda_n}$, of which the rotational average can be performed to create a directional cosine $\langle l_{i_n \lambda_n} \rangle$. For a tensor of rank $n$, the directional cosine has $n$ terms $\langle l_{i_1 \lambda_i} \cdots l_{i_n \lambda_n} \rangle$. A non-trigonometric method for calculating the directional cosine, particularly suitable for high $n$ (and low $n$ also), described by Andrews and Thirunamachandran[10], shall be detailed.

Let $\langle l_{i_1 \lambda_i} \cdots l_{i_n \lambda_n} \rangle$ be denoted as $I^{(n)}$. $I^{(n)}$ is rotationally invariant (isotropic), and according to a theorem by Hermann Weyl[11], can be expressed as a linear combination of isotropic tensors. For three-dimensions, the fundamental isotropic unit tensors are the Kronecker delta $\delta_{ij}$, defined in (15), and the Levi-Civita epsilon $\varepsilon_{ijk}$, defined in (18).

Isotropic tensors of even rank are linear combinations of $n/2$ Kronecker deltas, and ones of odd ranks are linear combinations of one Levi-Civita epsilon and $(n-3)/2$ deltas. An "isomer" of a isotropic tensor can be formed by permuting the indices $i_1, i_2, ..., i_n$ in a chosen product.

Using Weyl's theorem from above, it is implied that

$$I^{(n)} = \sum_{r,s} m_{rs}^{(n)} f_r^{(n)} g_s^{(n)} \tag{20}$$

where the products $f^{(n)}$, the tensor in the laboratory frame, and $g^{(n)}$, the tensor in the molecular frame, are summed over the set of their possible isomers, of which their indices are $r$ and $s$. $m_{rs}^{(n)}$ is therefore a numerical coefficient to be determined. In addition, $f^{(n)}$ and $g^{(n)}$ are related for a given isomer $q$ by

$$f_q^{(n)} I^{(n)} = g_q^{(n)}, \tag{21}$$

such that combining equations (20) and (21) and multiplying through by $g_t^{(n)}$ gives

$$f_t^{(n)} f_q^{(n)} = \sum_{r,s} f_t^{(n)} f_r^{(n)} m_{rs}^{(n)} g_s^{(n)} g_q^{(n)} \tag{22}$$

$$s_{tq}^{(n)} = \sum_{r,s} s_{tr}^{(n)} m_{rs}^{(n)} s_{sq}^{(n)} \tag{23}$$

as by index-contraction, $f_u^{(n)} f_v^{(n)} = g_u^{(n)} g_v^{(n)} = s_{uv}^{(n)}$. Equation (23) can be expressed in matrix form

$$S^{(n)} = S^{(n)} M^{(n)} S^{(n)} \tag{24}$$

$$M^{(n)} = (S^{(n)})^{-1}, \tag{25}$$

and it is seen using (20) that $I^{(n)}$ can be found[1, p. 312].

Values of $I^{(n)}$ for $n = 2, \ldots, 6$ can be found in §5.1.

## 1.5 Single Photon Absorption

Using the information described in §1.1.2, we can write the initial and final states of a molecule absorbing a single photon as

$$|\text{initial}\rangle = |n(\boldsymbol{k},\sigma)\rangle \prod_{\xi}^{N} |E_o(\xi)\rangle \tag{26}$$

$$|\text{final}\rangle = |(n-1)(\boldsymbol{k},\sigma)\rangle \, |E_m(\xi)\rangle \prod_{\xi\neq\xi'}^{N} |E_o(\xi')\rangle \tag{27}$$

where the product term represents the molecules (assumed to be non-interacting) in energy level $E_o$ (before absorption) and $E_m$ (after absorption).

However, as any one molecule has an equal probability of absorbing the photon, it is assumed that only molecule $\xi$ absorbs the photon[1, p. 87], and thus the matrix element is

$$M_{fi}(\xi) = \langle E_m(\xi)| \langle (n-1)(\boldsymbol{k},\sigma)| - \varepsilon_0^{-1} \boldsymbol{\mu}(\xi) \cdot \boldsymbol{d}^{\perp}(\boldsymbol{R}_\xi) |n(\boldsymbol{k},\sigma)\rangle |E_o(\xi)\rangle \,, \tag{28}$$

where the interaction term of the Hamiltonian is given by the electric dipole approximation previously seen as equation (11). In addition, we introduce the mode expansion for the field operator $\boldsymbol{d}^{\perp}$, the electric displacement field[1, pp. 36, 68], as

$$\boldsymbol{d}^{\perp}(\boldsymbol{r}) = \sum_{\boldsymbol{k},\sigma} \Omega \{ \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) a^{(\sigma)}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{r}} - \bar{\boldsymbol{e}}^{(\sigma)}(\boldsymbol{k}) a^{\dagger(\sigma)}(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} \}, \tag{29}$$

where $\boldsymbol{e}$ is the polarization vector, $a^{(\sigma)}(\boldsymbol{k})$, $a^{\dagger(\sigma)}(\boldsymbol{k})$ are the annihilation and creation operators for the mode $(\boldsymbol{k},\sigma)$, respectively, and the normalization constant $\Omega$ is

$$\Omega = i \left( \frac{\hbar c k \varepsilon_0}{2V} \right)^{1/2} \tag{30}$$

in which $V$ is the quantization volume.

We can simplify equation (28) as

$$\begin{aligned}
M_{fi}(\xi) &= \langle E_m(\xi)| \langle (n-1)(\boldsymbol{k},\sigma)| - \varepsilon_0^{-1} \boldsymbol{\mu}(\xi) \cdot \boldsymbol{d}^{\perp}(\boldsymbol{R}_\xi) |n(\boldsymbol{k},\sigma)\rangle |E_o(\xi)\rangle \\
&= -\varepsilon_0^{-1} \langle E_m(\xi)| \boldsymbol{\mu}(\xi) |E_o(\xi)\rangle \cdot \langle (n-1)(\boldsymbol{k},\sigma)| \boldsymbol{d}^{\perp}(\boldsymbol{R}_\xi) |n(\boldsymbol{k},\sigma)\rangle \\
&= -\varepsilon_0^{-1} \boldsymbol{\mu}^{mo}(\xi) \cdot \langle (n-1)(\boldsymbol{k},\sigma)| \textstyle\sum_{\boldsymbol{k},\sigma} \Omega \left\{ \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) a^{(\sigma)}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} - \bar{\boldsymbol{e}}^{(\sigma)}(\boldsymbol{k}) a^{\dagger(\sigma)}(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} \right\} |n(\boldsymbol{k},\sigma)\rangle \quad (31) \\
&= -\Omega\varepsilon_0^{-1} \boldsymbol{\mu}^{mo}(\xi) \cdot \langle (n-1)(\boldsymbol{k},\sigma)| \, \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) a^{(\sigma)}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} - \bar{\boldsymbol{e}}^{(\sigma)}(\boldsymbol{k}) a^{\dagger(\sigma)}(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} |n(\boldsymbol{k},\sigma)\rangle \quad (32) \\
&= -\Omega\varepsilon_0^{-1} \boldsymbol{\mu}^{mo}(\xi) \cdot [ \langle (n-1)(\boldsymbol{k},\sigma)| \, \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) a^{(\sigma)}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} |n(\boldsymbol{k},\sigma)\rangle \\
&\qquad\qquad - \langle (n-1)(\boldsymbol{k},\sigma)| \, \bar{\boldsymbol{e}}^{(\sigma)}(\boldsymbol{k}) a^{\dagger(\sigma)}(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} |n(\boldsymbol{k},\sigma)\rangle ] \\
&= -\Omega\varepsilon_0^{-1} \boldsymbol{\mu}^{mo}(\xi) \cdot [ \langle (n-1)(\boldsymbol{k},\sigma)| \, \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} n^{1/2} |(n-1)(\boldsymbol{k},\sigma)\rangle \\
&\qquad\qquad - \langle (n-1)(\boldsymbol{k},\sigma)| \, \bar{\boldsymbol{e}}^{(\sigma)}(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} (n+1)^{1/2} |(n+1)(\boldsymbol{k},\sigma)\rangle ] \\
&= -\Omega\varepsilon_0^{-1} \boldsymbol{\mu}^{mo}(\xi) \cdot [ \langle (n-1)(\boldsymbol{k},\sigma)| \, \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} n^{1/2} |(n-1)(\boldsymbol{k},\sigma)\rangle - 0 ] \quad (33)
\end{aligned}$$

$$= -\Omega\varepsilon_0^{-1} n^{1/2} \boldsymbol{\mu}^{mo}(\xi) \cdot \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} \tag{34}$$

where in (31) we write $\boldsymbol{\mu}^{mo}(\xi)$ as the electric dipole moment for the transition $m \leftarrow o$, in (32) the sum is taken over the singular mode $(\boldsymbol{k}, \sigma)$, and in (33) the second term reduces to zero due to the orthonormality of the number state kets $\langle n' \mid n \rangle = \delta_{n'n}$.

After (34) the transition rate given by the Fermi golden rule (6) is

$$\begin{aligned} \Gamma &= \frac{2\pi}{\hbar}\rho_{\mathrm{s}} \left(\frac{\Omega^2 n}{\varepsilon_0^2}\right) \sum_\xi |\boldsymbol{\mu}^{mo}(\xi) \cdot \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi}|^2 = \frac{2\pi}{\hbar}\rho_{\mathrm{s}} \left(\frac{\Omega^2 n}{\varepsilon_0^2}\right) \sum_\xi |\boldsymbol{\mu}^{mo}(\xi) \cdot \boldsymbol{e}^{(\sigma)}(\boldsymbol{k})|^2 \\ &= \frac{2\pi}{\hbar}\rho_{\mathrm{s}} \left(\frac{\Omega^2 n}{\varepsilon_0^2}\right) e_i^{(\sigma)}(\boldsymbol{k}) \bar{e}_j^{(\sigma)}(\boldsymbol{k}) \sum_\xi \mu_i^{mo}(\xi) \bar{\mu}_j^{mo}(\xi). \end{aligned} \tag{35}$$

The term $\mu_i^{mo}(\xi)\bar{\mu}_j^{mo}(\xi)$, a tensor of rank 2, in equation (35) can be rotationally averaged using the formula in §5.1 to obtain[1, p. 88]

$$\begin{aligned} \langle \Gamma \rangle &= \frac{2\pi}{\hbar}\rho_{\mathrm{s}} \left(\frac{\Omega^2 n}{\varepsilon_0^2}\right) e_i^{(\sigma)}(\boldsymbol{k}) \bar{e}_j^{(\sigma)}(\boldsymbol{k}) \sum_\xi \frac{1}{3} \delta_{ij}\delta_{\lambda\mu} \mu_\lambda^{mo}(\xi)\, \bar{\mu}_\mu^{mo}(\xi) \\ &= \frac{2\pi}{\hbar}\frac{N}{3}\rho_{\mathrm{s}} \left(\frac{\Omega^2 n}{\varepsilon_0^2}\right) e_i^{(\sigma)}(\boldsymbol{k}) \bar{e}_i^{(\sigma)}(\boldsymbol{k}) \mu_\lambda^{mo} \bar{\mu}_\lambda^{mo} \\ &= \frac{2\pi}{\hbar}\frac{N}{3}\rho_{\mathrm{s}} \left(\frac{\Omega^2 n}{\varepsilon_0^2}\right) |\boldsymbol{\mu}^{mo}|^2 \end{aligned} \tag{36}$$

where the summation over $\xi$ is replaced with $N$, the number of absorbers.

## 1.6 Two Photon Absorption

In order to continue with two photon absorption we must make an adjustment to the matrix element formula (7) to two photons. When more than one photon is absorbed, there are a number of intermediate states $|I\rangle$ between the initial state $|i\rangle$ and final state $|f\rangle$. By performing a second-order pertubative analysis[1, p. 109][12, p. 241], the matrix element is found as

$$M_{fi} = \sum_I \frac{\langle f| H_{\mathrm{int}} |I\rangle \langle I| H_{\mathrm{int}} |i\rangle}{E_I - E_i}. \tag{37}$$

In general, the matrix element can be written as

$$M_{fi} = \langle f| H_{\mathrm{int}} |i\rangle + \sum_I \frac{\langle f| H_{\mathrm{int}} |I\rangle \langle I| H_{\mathrm{int}} |i\rangle}{E_I - E_i} + \sum_{I,II} \frac{\langle f| H_{\mathrm{int}} |II\rangle \langle II| H_{\mathrm{int}} |I\rangle \langle I| H_{\mathrm{int}} |i\rangle}{(E_I - E_i)(E_{II} - E_i)} + \cdots \tag{38}$$

but in two-photon absorption, there is only one intermediate state, and higher intermediate states are ignored; the first term cancels to 0 as the occupation numbers differ by 2, and so a transition cannot be made according to the selection rules. The energy from the initial to the intermediate state $E_{iI}$ is calculated as $E_r + (n-1)\hbar\omega - E_o - n\hbar\omega = E_{ro} - \hbar\omega$. After subsituting the initial and final states, and assuming that both photons come from the same beam, we obtain

$$M_{fi} = \sum_r \frac{1}{E_{or} - \hbar\omega} \Big[ \langle E_m| \langle (n-2)(\boldsymbol{k},\sigma)| - \varepsilon_0^{-1} \boldsymbol{\mu}(\xi) \cdot \boldsymbol{d}^\perp(\boldsymbol{R}_\xi) |(n-1)(\boldsymbol{k},\sigma)\rangle |E_r\rangle$$

$$\times \langle E_r| \langle (n-1)(\boldsymbol{k},\sigma)| - \varepsilon_0^{-1} \boldsymbol{\mu}(\xi)\cdot \boldsymbol{d}^{\perp}(\boldsymbol{R}_\xi) |n(\boldsymbol{k},\sigma)\rangle |E_o\rangle \Big] \tag{39}$$

and can be simplified in a similar process as in §1.5 as

$$\begin{aligned} M_{fi} &= \frac{1}{\varepsilon_0^2}\sum_r \frac{\langle E_m| \boldsymbol{\mu}(\xi) |E_r\rangle \langle E_r| \boldsymbol{\mu}(\xi) |E_o\rangle}{E_{or}-\hbar\omega} \cdot \Big[ \langle (n-2)(\boldsymbol{k},\sigma)| \boldsymbol{d}^{\perp}(\boldsymbol{R}_\xi) |(n-1)(\boldsymbol{k},\sigma)\rangle \\ &\qquad \times \langle (n-1)(\boldsymbol{k},\sigma)| \boldsymbol{d}^{\perp}(\boldsymbol{R}_\xi) |n(\boldsymbol{k},\sigma)\rangle \Big] \\ &= \frac{\Omega^2}{\varepsilon_0^2}\sum_r \left( \frac{\mu_i^{mr}(\xi)\mu_j^{ro}(\xi)}{E_{or}-\hbar\omega} \right) \cdot \left[ \left( (n-1)^{1/2} \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} \right) \left( n^{1/2} \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} \right) \right] \\ &= \frac{\Omega^2}{\varepsilon_0^2}\sum_r \left( \frac{\mu_i^{mr}(\xi)\mu_j^{ro}(\xi)}{E_{or}-\hbar\omega} \right) \cdot \{n(n-1)\}^{1/2} e_i^{(\sigma)}(\boldsymbol{k}) e_j^{(\sigma)}(\boldsymbol{k}) e^{2i\boldsymbol{k}\cdot\boldsymbol{R}_\xi}. \end{aligned} \tag{40}$$

Since the term $e_i^{(\sigma)}(\boldsymbol{k})e_j^{(\sigma)}(\boldsymbol{k})$ are symmetric in $i,j$, contraction with the $r$-summed term will leave only the $i,j$-symmetric part, and we can write

$$\begin{aligned} M_{fi} &= \frac{\Omega^2}{2\varepsilon_0^2}\sum_r \left( \frac{\mu_i^{mr}(\xi)\mu_j^{ro}(\xi)}{E_{or}-\hbar\omega} + \frac{\mu_j^{mr}(\xi)\mu_i^{ro}(\xi)}{E_{or}-\hbar\omega} \right) \cdot \{n(n-1)\}^{1/2} e_i^{(\sigma)}(\boldsymbol{k}) e_j^{(\sigma)}(\boldsymbol{k}) e^{2i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} \\ &= \frac{\Omega^2}{2\varepsilon_0^2} \boldsymbol{\alpha}^{mo} \{n(n-1)\}^{1/2} e_i^{(\sigma)}(\boldsymbol{k}) e_j^{(\sigma)}(\boldsymbol{k}) e^{2i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} \end{aligned} \tag{41}$$

where in (41) we introduced a rank 2 tensor $\boldsymbol{\alpha}^{mo}$ which will need to be rotationally averaged after we find the transition rate as

$$\langle \Gamma \rangle = \frac{2\pi}{\hbar} N\rho_{\mathrm{s}} \frac{\Omega^4}{4\varepsilon_0^4} n(n-1) e_i^{(\sigma)}(\boldsymbol{k}) e_j^{(\sigma)}(\boldsymbol{k}) \bar{e}_k^{(\sigma)}(\boldsymbol{k}) \bar{e}_l^{(\sigma)}(\boldsymbol{k}) \left\langle \alpha_{ij}^{mo} \bar{\alpha}_{kl}^{mo} \right\rangle . \tag{42}$$

Using the formula in §5.1 again,

$$\begin{aligned} \left\langle \alpha_{ij}^{mo} \bar{\alpha}_{kl}^{mo} \right\rangle = \frac{1}{30} [ & \delta_{ij}\delta_{kl}(4\delta_{\lambda\mu}\delta_{\nu\pi} - \delta_{\lambda\nu}\delta_{\mu\pi} - \delta_{\lambda\pi}\delta_{\mu\nu}) \\ & + \delta_{ik}\delta_{jl}(-\delta_{\lambda\mu}\delta_{\nu\pi} + 4\delta_{\lambda\nu}\delta_{\mu\pi} - \delta_{\lambda\pi}\delta_{\mu\nu}) \\ & + \delta_{il}\delta_{jk}(-\delta_{\lambda\mu}\delta_{\nu\pi} - \delta_{\lambda\nu}\delta_{\mu\pi} + 4\delta_{\lambda\pi}\delta_{\mu\nu}) ] \alpha_{\lambda\mu}^{mo} \bar{\alpha}_{\nu\pi}^{mo} \end{aligned} \tag{43}$$

in which the Greek indices can be contracted and the $\lambda,\mu$-symmetric property of $\alpha_{\lambda\mu}^{mo}$ be exploited to obtain

$$\left\langle \alpha_{ij}^{mo} \bar{\alpha}_{kl}^{mo} \right\rangle = \frac{1}{30} \left[ \delta_{ij}\delta_{kl}\{4\alpha_{\lambda\lambda}^{mo}\bar{\alpha}_{\mu\mu}^{mo} - 2\alpha_{\lambda\mu}^{mo}\bar{\alpha}_{\lambda\mu}^{mo}\} - (\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})\{\alpha_{\lambda\lambda}^{mo}\bar{\alpha}_{\mu\mu}^{mo} - 3\alpha_{\lambda\mu}^{mo}\bar{\alpha}_{\lambda\mu}^{mo}\} \right] . \tag{44}$$

Combining (44) with the polarization tensors (whose $\sigma$ and $\boldsymbol{k}$ labels are suppressed for brevity),

$$e_i e_j \bar{e}_k \bar{e}_l \left\langle \alpha_{ij}^{mo} \bar{\alpha}_{kl}^{mo} \right\rangle = \frac{1}{15} \left[ (2e_i e_j \bar{e}_k \bar{e}_l - 1)\alpha_{\lambda\lambda}^{mo}\bar{\alpha}_{\mu\mu}^{mo} - (e_i e_j \bar{e}_k \bar{e}_l - 3)\alpha_{\lambda\mu}^{mo}\bar{\alpha}_{\lambda\mu}^{mo} \right] \tag{45}$$

such that the absorption rate can be found to be[1, p. 114]

$$\langle \Gamma \rangle = \frac{\pi}{30\hbar} \frac{\Omega^4}{\varepsilon_0^4} N\rho_{\mathrm{s}} n(n-1) \left[ (2e_i e_j \bar{e}_k \bar{e}_l - 1)\alpha_{\lambda\lambda}^{mo}\bar{\alpha}_{\mu\mu}^{mo} - (e_i e_j \bar{e}_k \bar{e}_l - 3)\alpha_{\lambda\mu}^{mo}\bar{\alpha}_{\lambda\mu}^{mo} \right] . \tag{46}$$

## 1.7 Laguerre-Gaussian Beams

When light is emitted from a laser, it will have a beam profile, determined by the design and fabrication of the laser. Most ideal laser beams will have a Gaussian intensity profile, where the irradiance of the beam away from its centre point is proportional to $e^{-r^2}$, as much of the energy is focused onto the intended, most concentrated spot.[13]

Laguerre-Gaussian beams are a mathematic representation of a Gaussian beam which occur as a modal decomposition of the paraxial scalar Helmholtz equation, a specific representation of the wave equation. In addition, L-G beams have cylindrical symmetry and the electric displacement and magnetic field mode expansions for the L-G beam express a dependence on $\ell$, which determines the twist structure of the orbital angular momentum of the beam, which are used as the basis of structured light[14].

In cylindrical coordinates, the paraxial scalar Helmholtz equation is

$$\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}+\frac{\partial^2}{\partial^2\rho}+\frac{1}{\rho^2}\frac{\partial^2}{\partial\varphi^2}+2ik\frac{\partial}{\partial z}\right)u\left(\rho,\varphi,z\right)=0. \tag{47}$$

The normalized amplitude of the Laguerre-Gaussian beam, a solution to $u\left(\rho,\varphi,z\right)$ above, is:

$$u_{\ell,p}^{\mathrm{LG}}\left(\rho,\varphi,z\right)=\frac{C_p^{|\ell|}}{w(z)}\left(\frac{\sqrt{2}\rho}{w(z)}\right)^{|\ell|}L_p^{|\ell|}\left(\frac{2\rho^2}{w^2(z)}\right)\exp\left(-\frac{\rho^2}{w^2(z)}-ik\frac{k\rho^2 z}{2(z^2+z_{\mathrm{R}}^2)}-i\ell\varphi+i(2p+\ell+1)\arctan\left(\frac{z}{z_{\mathrm{R}}}\right)\right) \tag{48}$$

where $p$ is the radial index, $\ell$ is the topological charge, $w(z)$ is the beam waist, the radius of the beam at $z$, given by

$$w(z)=w(0)\left(1+\frac{z^2}{z_{\mathrm{R}}^2}\right)^{1/2}, \tag{49}$$

$z_{\mathrm{R}}$ is the Rayleigh range (figure 1), a measure of the beam's focal length, $L_p^{|\ell|}$ is the generalized Laguerre polynomial of order $p$, which is given by

$$L_p^{\ell}\left(x\right)=\sum_{n=0}^{p}(-1)^n\binom{p+\ell}{p-n}\frac{x^n}{n!}, \tag{50}$$

and the normalization constant $C_p^{|\ell|}$ is given by

$$C_p^{|\ell|}=\left(2^{|\ell|+1}\frac{p!}{\pi\left(p+|\ell|\right)!}\right)^{1/2}. \tag{51}$$

In most circumstances the Rayleigh range for L-G beams is around several metres, and in considering beam interactions with particles we can approximate equation (48) by considering it in the long Rayleigh range $z_{\mathrm{R}}\gg z$ (i.e. the beam width is constant). By rewriting

$$w\left(z\right)\approx w_0:=w(0) \tag{52}$$

$$\frac{1}{z^2+z_{\mathrm{R}}^2}\approx 0 \tag{53}$$

$$\arctan\left(\frac{z}{z_{\mathrm{R}}}\right)\approx 0 \tag{54}$$

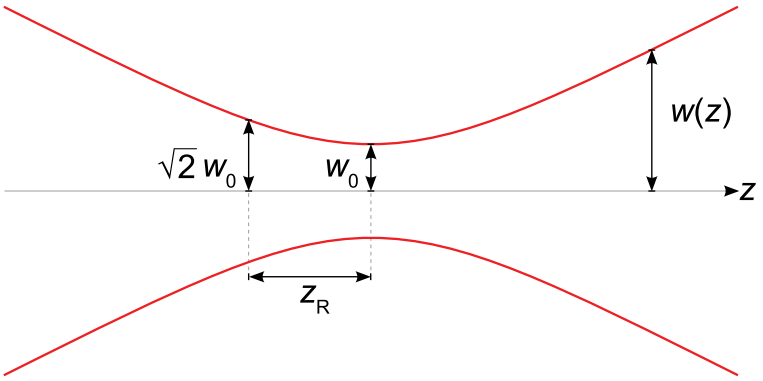


Figure 1: The Rayleigh range $z_{\mathrm{R}}$, which can be defined as the distance from $\sqrt{2}$ times the central beam width. Adapted from Rodolfo Hermans, under GFDL (GNU Free Distribution License).

and defining the *radial distribution function* as

$$f_{\ell,p}(\rho) = \frac{C_p^{|\ell|}}{w_0}\left(\frac{\sqrt{2}\rho}{w_0}\right)^{|\ell|} e^{-\rho^2/w_0^2} L_p^{|\ell|}\left(\frac{2\rho^2}{w_0^2}\right), \tag{55}$$

the amplitude of the Laguerre-Gaussian beam can then be expressed, without $z$ dependence, as

$$u_{\ell,p}^{\mathrm{LG}}(\rho,\varphi) \approx f_{\ell,p}(\rho)\, e^{-i\ell\varphi}. \tag{56}$$

## 1.8 Optical Activity

Optical activity is the difference in the responses of chiral, or optically active molecules, when exposed to light radiation with angular momentum. Chiral molecules are molecules that cannot be superposed onto its mirror image by means of rotation or translation[16], and belong to point groups with no improper axis of rotation $C_n, D_n, T, O, I$[1, p. 183]. The angular momentum of light can manifest itself in two different ways: the first, *spin angular momentum*, quantized by $\sigma\hbar$ per photon and appears in circularly polarized beams; and the second, *orbital angular momentum*, quantized by $\ell\hbar$ per photon in beams with a twisting wavefront[15], and are such called "structured light" or "optical vortices"[17].

The chiroptical properties are examined in this subsection are circular dichroism, which is the differential absorption of circularly polarized light, and then by extension, circular vortex dichroism, the differential absorption of circularly polarized light with optical vortices. Vortex dichroism can also occur in plane polarized beams[18], as is discussed later in §3.

### 1.8.1 Circular Dichroism

Circular dichroism is the phenomenon under which left-circularly polarized and right-circularly polarized light are absorbed at a different rate by chiral molecules. A similar method to what is shown in §1.5 can be used, but the electric dipole approximation for the interaction term is insufficient in this context as circular dichroism arises from an interference of electric dipole and magnetic dipole moments[1, pp. 99, 185], and therefore it is necessary for the first magentic dipole term in the multipolar Hamiltonian to be included:

$$H_{\mathrm{int}} = -\varepsilon_0^{-1}\mu_i(\xi)\, d_i^{\perp}\left(\boldsymbol{R}_\xi\right) - m_i(\xi)\, b_i\left(\boldsymbol{R}_\xi\right). \tag{57}$$

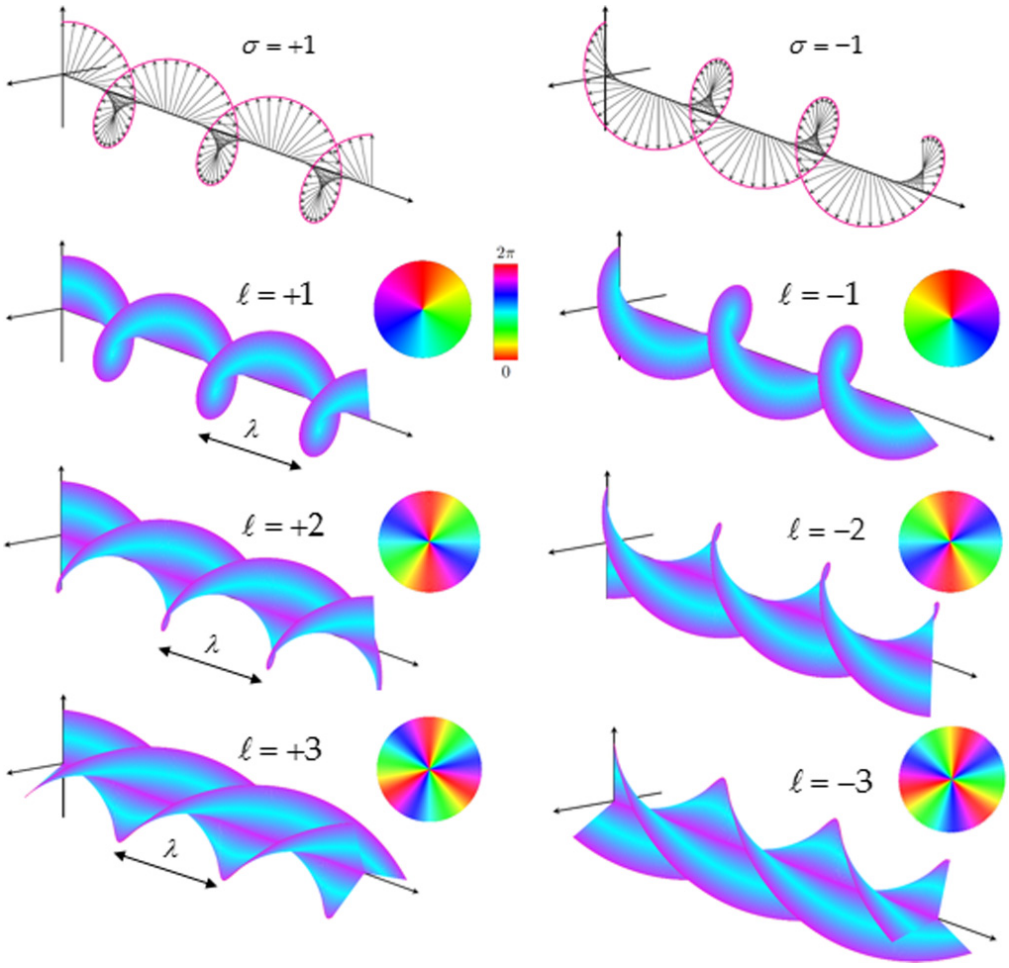


Figure 2: The visualization of the chirality of light. **Top:** Electric field vectors for beams with left- and right-circular polarizations possessing spin angular momentum. **Bottom:** Wavefronts of beams with various topological charges $\ell$, which are generally considered the source of the beam's orbital angular momentum[15].

We can now proceed with the substitution $\sigma = \mathrm{L}(+1)/\mathrm{R}(-1)$, where L and R stand for left- and right-circularly polarized, respectively. Our matrix element is

$$M_{fi}^{\mathrm{L/R}} = -\varepsilon_0^{-1}\boldsymbol{\mu}^{mo}(\xi)\cdot\langle(n-1)(\boldsymbol{k},\mathrm{L/R})|\,\boldsymbol{d}^{\perp}(\boldsymbol{R}_\xi)\,|n(\boldsymbol{k},\mathrm{L/R})\rangle - \boldsymbol{m}^{mo}(\xi)\cdot\langle(n-1)(\boldsymbol{k},\mathrm{L/R})|\,\boldsymbol{b}(\boldsymbol{R}_\xi)\,|n(\boldsymbol{k},\mathrm{L/R})\rangle\,. \tag{58}$$

The mode expansion for the magnetic field $\boldsymbol{b}$ is[1, p. 36]

$$\boldsymbol{b}(\boldsymbol{r}) = \sum_{\boldsymbol{k},\sigma}\frac{\Omega}{c\varepsilon_0}\{\boldsymbol{b}^{(\sigma)}(\boldsymbol{k})a^{(\sigma)}(\boldsymbol{k})\mathrm{e}^{i\boldsymbol{k}\cdot\boldsymbol{r}} - \bar{\boldsymbol{b}}^{(\sigma)}(\boldsymbol{k})a^{\dagger(\sigma)}(\boldsymbol{k})\mathrm{e}^{-i\boldsymbol{k}\cdot\boldsymbol{r}}\}. \tag{59}$$

where $\boldsymbol{b}^{(\sigma)}(\boldsymbol{k})$ is the polarization tensor in the direction of the magnetic field, perpendicular to the electric field per $\boldsymbol{b} = \hat{\boldsymbol{z}}\times\boldsymbol{e}$ such that

$$\boldsymbol{b}^{(\mathrm{L/R})}(\boldsymbol{k}) = \mp i\boldsymbol{e}^{(\mathrm{L/R})}(\boldsymbol{k}). \tag{60}$$

Continuing from (58), and for convenience, dropping photon mode labels for occupation numbers (which

will be done to the end)

$$\begin{aligned}
M_{fi}^{\mathrm{L/R}} &= -\frac{\Omega}{\varepsilon_0} n^{1/2} \boldsymbol{\mu}^{mo}(\xi) \cdot \boldsymbol{e}^{(\mathrm{L/R})}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} - \frac{\Omega}{c\varepsilon_0} \boldsymbol{m}^{mo}(\xi) \cdot [\, \langle n-1|\, \boldsymbol{b}^{(\mathrm{L/R})}(\boldsymbol{k}) a^{(\mathrm{L/R})}(\boldsymbol{k}) \mathrm{e}^{i\boldsymbol{k}\cdot\boldsymbol{R}_{\boldsymbol{\xi}}} |n\rangle \\
&\qquad\qquad - \langle n-1|\, \bar{\boldsymbol{b}}^{(\mathrm{L/R})}(\boldsymbol{k}) a^{\dagger(\mathrm{L/R})}(\boldsymbol{k}) \mathrm{e}^{-i\boldsymbol{k}\cdot\boldsymbol{R}_{\boldsymbol{\xi}}} |n\rangle \,] \\
&= -\frac{\Omega}{\varepsilon_0} n^{1/2} \boldsymbol{\mu}^{mo}(\xi) \cdot \boldsymbol{e}^{(\mathrm{L/R})}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} - \frac{\Omega}{c\varepsilon_0} n^{1/2} \boldsymbol{m}^{mo}(\xi) \cdot \boldsymbol{b}^{(\mathrm{L/R})}(\boldsymbol{k}) \mathrm{e}^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} \\
&= -\frac{\Omega}{\varepsilon_0} n^{1/2} \left( \boldsymbol{\mu}^{mo}(\xi) \cdot \boldsymbol{e}^{(\mathrm{L/R})}(\boldsymbol{k}) + \frac{1}{c} \boldsymbol{m}^{mo}(\xi) \cdot \boldsymbol{b}^{(\mathrm{L/R})}(\boldsymbol{k}) \right) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi} \\
&= -\frac{\Omega}{\varepsilon_0} n^{1/2} \boldsymbol{e}^{(\mathrm{L/R})}(\boldsymbol{k}) \left( \boldsymbol{\mu}^{mo}(\xi) \mp \frac{i}{c} \boldsymbol{m}^{mo}(\xi) \right) e^{i\boldsymbol{k}\cdot\boldsymbol{R}_\xi}.
\end{aligned} \tag{61}$$

The transition rate given by the Fermi golden rule (6) is

$$\Gamma^{\mathrm{L/R}} = \frac{2\pi}{\hbar} N \rho_{\mathrm{s}} \left( \frac{\Omega^2 n}{\varepsilon_0^2} \right) e_i^{(\mathrm{L/R})}(\boldsymbol{k}) \bar{e}_j^{(\mathrm{L/R})}(\boldsymbol{k}) \left( \mu_i^{mo}(\xi) \mp \frac{i}{c} m_i^{mo}(\xi) \right) \left( \bar{\mu}_j^{mo}(\xi) \pm \frac{i}{c} \bar{m}_j^{mo}(\xi) \right) \tag{62}$$

and after rotational averaging, in a similar process to in §1.5, we have

$$\left\langle \Gamma^{\mathrm{L/R}} \right\rangle = \frac{2\pi}{\hbar} \frac{N}{3} \rho_{\mathrm{s}} \left( \frac{\Omega^2 n}{\varepsilon_0^2} \right) \left| \boldsymbol{\mu}^{mo} \mp \frac{i}{c} \boldsymbol{m}^{mo} \right|^2. \tag{63}$$

When the final term is expanded, we will have three terms: the E1E1 ($\boldsymbol{\mu} \cdot \boldsymbol{\mu}$) term, the standard result for one-photon absorption; the M1M1 ($\boldsymbol{m} \cdot \boldsymbol{m}$), which is smaller than the E1E1 term by about $10^{-6}$ and is non-discriminatory (see (75)); and the E1M1 ($\boldsymbol{\mu} \cdot \boldsymbol{m}$) term, which is the leading contributor to circular dichroism—the "interference" mentioned at the beginning of this section.

Using this result, the rotationally averaged differential transition rates can be expressed as[1, p. 186], is

$$\left\langle \Gamma^{(\mathrm{L})} \right\rangle - \left\langle \Gamma^{(\mathrm{R})} \right\rangle = \frac{4i\pi}{\hbar c} \frac{N}{3} \rho_{\mathrm{s}} \left( \frac{\Omega^2 n}{\varepsilon_0^2} \right) [\, \bar{\boldsymbol{\mu}}^{mo} \cdot \boldsymbol{m}^{mo} - \boldsymbol{\mu}^{mo} \cdot \bar{\boldsymbol{m}}^{mo} \,]. \tag{64}$$

Note for real wavefunctions, $\boldsymbol{\mu}^{mo}$ is a real term and $\boldsymbol{m}^{mo}$ is an imaginary term such that the differential transition rate is also real. This differential transition rate can also be expressed in terms of the optical rotary strength $R^{mo} = \operatorname{Im} \mu_i^{om} m_j^{mo}$, an experimental measure of circular dichroism[19]:

$$\left\langle \Gamma^{(\mathrm{L})} \right\rangle - \left\langle \Gamma^{(\mathrm{R})} \right\rangle = -\frac{8i\pi}{\hbar c} \frac{N}{3} \rho_{\mathrm{s}} \left( \frac{\Omega^2 n}{\varepsilon_0^2} \right) R^{mo}. \tag{65}$$

### 1.8.2 Circular Vortex Dichroism

Vortex dichroism results from the differential absorption of light with optical angular momentum which is not factored into the plane wave solutions of Maxwell's equations which gave the mode expansions (29) and (59). Therefore we must construct our field equations from the Laguerre-Gaussian modes mentioned in §1.7 instead, and are, in cylindrical coordinates[20],

$$\boldsymbol{d}_{x,y}^{\perp}(\boldsymbol{\rho}) = \sum_{\boldsymbol{k},\sigma,\ell,p} \frac{\tilde{\Omega}}{\sqrt{2}} \left[ \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) f_{|\ell|,p}(\boldsymbol{\rho}) a_{|\ell|,p}^{(\sigma)}(\boldsymbol{k}) e^{i(kz+\ell\varphi)} - \bar{\boldsymbol{e}}^{(\sigma)}(\boldsymbol{k}) \bar{f}_{|\ell|,p}(\boldsymbol{\rho}) a_{|\ell|,p}^{\dagger(\sigma)}(\boldsymbol{k}) e^{-i(kz+\ell\varphi)} \right] \tag{66}$$

$$\boldsymbol{b}_{x,y}(\boldsymbol{\rho}) = \sum_{\boldsymbol{k},\sigma,\ell,p} \frac{\tilde{\Omega}}{\sqrt{2}\varepsilon_0 c} \left[ \boldsymbol{b}^{(\sigma)}(\boldsymbol{k}) f_{|\ell|,p}(\boldsymbol{\rho}) a^{(\sigma)}_{|\ell|,p}(\boldsymbol{k}) e^{i(kz+\ell\varphi)} - \bar{\boldsymbol{b}}^{(\sigma)}(\boldsymbol{k}) \bar{f}_{|\ell|,p}(\boldsymbol{\rho}) a^{\dagger(\sigma)}_{|\ell|,p}(\boldsymbol{k}) e^{-i(kz+\ell\varphi)} \right] \tag{67}$$

where $\tilde{\Omega}$ includes an additional term $A_{\ell,p}^{-3/2}$ for L-G modes in the normalization factor (30). We now see an azimuthal angle dependence in the exponent: this structure is what gives the orbital angular momentum in the L-G modes. The $x, y$ subscripts indicate that this field is completely transverse, and is thus *paraxial*. In §2.5, we will include a non-paraxial correction which will involve $\boldsymbol{d}_z^{\perp}$ and $\boldsymbol{b}_z$ components.

In this section, we will also include the quadrupole moment interaction in the interaction coupling,

$$H_{\text{int}} = -\varepsilon_0^{-1} \mu_i(\xi)\, d_i^{\perp}\left(\boldsymbol{R}_\xi\right) - m_i(\xi)\, b_i\left(\boldsymbol{R}_\xi\right) - \varepsilon_0^{-1} Q_{ij}(\xi)\, \nabla_j d_i^{\perp}\left(\boldsymbol{R}_\xi\right) \tag{68}$$

which gives the matrix element as

$$M_{fi} = -\varepsilon_0^{-1} \mu_i^{mo}(\xi) \langle n-1|\, d_i^{\perp}(\boldsymbol{R}_\xi)\, |n\rangle - m_i^{mo}(\xi) \langle n-1|\, b_i(\boldsymbol{R}_\xi)\, |n\rangle - \varepsilon_0^{-1} Q_{ij}^{mo}(\xi) \langle n-1|\, \nabla_j d_i^{\perp}(\boldsymbol{R}_\xi)\, |n\rangle\,. \tag{69}$$

Initial simplification is similar to in §1.5 but with an additional $f_{|\ell|,p}$ and $e^{i\ell\varphi}$ factor, and gives

$$\begin{aligned} M_{fi} = &- \frac{\tilde{\Omega} n}{\sqrt{2}\varepsilon_0} e_i^{(\sigma)}(\boldsymbol{k}) \mu_i^{mo}(\xi) f_{|\ell|,p}(\boldsymbol{\rho}) e^{i(kz+\ell\varphi)} - \frac{\tilde{\Omega} n}{\sqrt{2}\varepsilon_0 c} b_i^{(\sigma)}(\boldsymbol{k}) m_i^{mo}(\xi) f_{|\ell|,p}(\boldsymbol{\rho}) e^{i(kz+\ell\varphi)} \\ &- \frac{\tilde{\Omega} n}{\sqrt{2}\varepsilon_0} e_i^{(\sigma)}(\boldsymbol{k}) Q_{ij}^{mo}(\xi) \nabla_j f_{|\ell|,p}(\boldsymbol{\rho}) e^{i(kz+\ell\varphi)}. \end{aligned} \tag{70}$$

In order to calculate $\nabla_j f_{|\ell|,p}(\boldsymbol{\rho}) e^{i(kz+\ell\varphi)}$, the gradient operator in cylindrical coordinates is needed,

$$\nabla_j = \hat{\rho}_j \frac{\partial}{\partial \rho} + \frac{1}{\rho} \hat{\varphi}_j \frac{\partial}{\partial \varphi} + \hat{z}_j \frac{\partial}{\partial z} \tag{71}$$

where

$$\hat{\boldsymbol{\rho}} = \hat{\boldsymbol{x}} \cos\varphi + \hat{\boldsymbol{y}} \sin\varphi \qquad \hat{\boldsymbol{\varphi}} = -\hat{\boldsymbol{x}} \sin\boldsymbol{\varphi} + \hat{\boldsymbol{y}} \cos\varphi \tag{72}$$

and thus

$$\nabla_j f_{|\ell|,p}(\boldsymbol{\rho}) e^{i(kz+\ell\varphi)} = \hat{\rho}_j \frac{\partial f_{|\ell|,p}(\boldsymbol{\rho})}{\partial \rho} e^{i(kz+\ell\varphi)} + i\ell \hat{\varphi}_j \frac{f_{|\ell|,p}(\boldsymbol{\rho})}{\rho} e^{i(kz+\ell\varphi)} + ik\hat{z}_j f_{|\ell|,p}(\boldsymbol{\rho}) e^{i(kz+\ell\varphi)} \tag{73}$$

($e^{i(kz+\ell\varphi)}$ depends on $z$ and $\varphi$, so they are included, but are not affected by the gradient operator).

Therefore, the matrix element can be expressed as

$$\begin{aligned} M_{fi} = &\left[ e_i^{(\sigma)}(\boldsymbol{k}) \mu_i^{mo}(\xi) + \frac{b_i^{(\sigma)}(\boldsymbol{k}) m_i^{mo}(\xi)}{c} + e_i^{(\sigma)}(\boldsymbol{k}) Q_{ij}^{mo}(\xi) \left( \hat{\rho}_j \frac{\partial f_{|\ell|,p}(\boldsymbol{\rho})}{\partial \rho} + i\ell \hat{\varphi}_j \frac{f_{|\ell|,p}(\boldsymbol{\rho})}{\rho} + ik\hat{z}_j f_{|\ell|,p}(\boldsymbol{\rho}) \right) \right] \\ &\times -\frac{\tilde{\Omega} n}{\sqrt{2}\varepsilon_0} f_{|\ell|,p}(\boldsymbol{\rho}) e^{i(kz+\ell\varphi)} \end{aligned} \tag{74}$$

When this expression is squared for the Fermi golden rule, we obtain multiple coupling terms with the electric dipole moment E1, magnetic dipole moment M1, and the electric quadrupole moment E2, leading to a total of nine terms.

We saw in section §1.8.1 that the E1M1 term is the standard result for circular dichroism. As vortex

dichroism will depend of the optical angular momentum given by $\ell$, which is only present in the E2 term (the other term depending on $\ell$ is $e^{i(kz+\ell\varphi)}$ which disappears upon squaring and taking the absolute value), vortex dichroism depends on the quadrupole interaction with itself, the electric dipole, and the magnetic dipole. However, only the E1E2 term is sensitive to the chirality of the molecule and of the light.

This is due to the space parity $\mathcal{P}$ of the multipole moments, in particular,

$$\mathcal{P}\left\{\mathrm{E}n\right\} = (-1)^n, \qquad \mathcal{P}\left\{\mathrm{M}n\right\} = (-1)^{n-1} \tag{75}$$

where the eigenvalue of $+1$ signifies an even parity (do not change sign under a parity transformation), and $-1$ signifies an odd parity (do change sign under a parity transformation)[21]. Therefore the M1E2 and E2E2 have even space parity, and effectively replace a chiral molecule with its mirror image, leaving the multipole coupling invariant, akin to circular dichroism, where the E1E1 and M1M1 also have even space parity and are non-discriminatory.

The transition rate for the E1E2 term only is

$$\mathit{\Gamma} = \frac{2\pi}{\hbar} N \rho_{\mathrm{s}} \left(\frac{\Omega^2 n}{\varepsilon_0^2}\right) \frac{f^2_{|\ell|,p}(\boldsymbol{\rho})}{\rho} e_i^{(\sigma)}(\boldsymbol{k}) \bar{e}_k^{(\sigma)}(\boldsymbol{k}) i\ell\hat{\varphi}_j (\bar{\mu}_k^{mo} Q_{ij}^{mo} - \mu_i^{mo} \bar{Q}_{kj}^{mo}). \tag{76}$$

Notice that the polarization vectors must be complex to produce a non-zero real transition rate, implying that only in conjunction with circular polarization is vortex dichroism able to occur. However, this is only in the E1E2 term—in §3 we will see it is possible for only vortex dichroism to occur using longitudinal fields.

Continuing, we use the relation

$$e_i^{(\mathrm{L})}(\boldsymbol{k}) \bar{e}_k^{(\mathrm{L})}(\boldsymbol{k}) = \frac{(\delta_{ik} - \hat{k}_i \hat{k}_k) - i\varepsilon_{ikm}\hat{k}_m}{2}, \qquad e_i^{(\mathrm{R})}(\boldsymbol{k}) \bar{e}_k^{(\mathrm{R})}(\boldsymbol{k}) = \frac{(\delta_{ik} - \hat{k}_i \hat{k}_k) + i\varepsilon_{ikm}\hat{k}_m}{2} \tag{77}$$

and therefore, retaining the real terms,

$$\mathit{\Gamma} = \frac{\pi}{\hbar} N \rho_{\mathrm{s}} \left(\frac{\Omega^2 n}{\varepsilon_0^2}\right) \frac{f^2_{|\ell|,p}(\boldsymbol{\rho})}{\rho} \varepsilon_{ikm} \hat{k}_m \ell\hat{\varphi}_j (\bar{\mu}_k^{mo} Q_{ij}^{mo} - \mu_i^{mo} \bar{Q}_{kj}^{mo}). \tag{78}$$

To conclude this chapter, we consider the effect of rotational averaging on (78). From §5.1, we take the third rank tensor, such that

$$\left\langle \bar{\mu}_k^{mo} Q_{ij}^{mo} - \mu_i^{mo} \bar{Q}_{kj}^{mo} \right\rangle = \frac{1}{6} \varepsilon_{ijk} \varepsilon_{\lambda\mu\nu} \left( \bar{\mu}_\nu^{mo} Q_{\lambda\mu}^{mo} - \mu_\lambda^{mo} \bar{Q}_{\nu\mu}^{mo} \right) \tag{79}$$

Since the electric quadrupole moment is index-symmetric, and the Levi-Civita epsilon antisymmetric, tensor contraction with the above leads to a null result. As such, the effect of the E1E2 coupling on circular dichroism, or circular vortex dichroism, is zero for fully randomly oriented molecules[1, p. 188].

***Research***

## 2.1 Motive

In this section up to §2.5, we expand upon the information given in §1 by computationally calculating the equations and producing graphs; first, the radial distribution function is plotted as it is a preliminary result required in the multipole couplings, and second, the contributions of the E1E2 and E2E2 couplings to the transition rate, which was discussed in §1.8.2.

While the E2E2 coupling is not supported by the chirality of the material, it can still manifest in conjunction with the polarization of the beam. Together with M1E2, these couplings are still of interest, but unlike M1E2, requires no additional parameters—these plots can be thought of as an additional exercise.

In the final part of this section, §2.5, we introduce a relatively new theory of introducing a longitudinal field to the mode expansions, which describes certain phenomena and can significantly alter the contribution of the E1E2 and E2E2 couplings.

## 2.2 Methodology

The graphing in this section is performed by the `matplotlib` package, with aid from the `numpy` and `scipy` packages. The full code can be seen in the appendix, §5.2. In general, the method for plotting is as follows:

1. Firstly, a function is defined for an equation with a set of values, eg. $\rho, \theta, z, p, \ell, \cdots$

   - The `numpy.einsum` function is used to evaluate expressions in tensor index notation[22]. For example, matrix multiplication can be expressed as $z = A_{ij}B_{jk}$ which using `numpy.einsum` is

     ```
     z = numpy.einsum("ij, jk", A, B)
     ```

   - The `scipy.misc.derivative` function is used to evaluate derivatives, which uses a simple method known as the central difference formula[23],

     $$f'(x) \approx \frac{f(x+\delta x) - f(x-\delta x)}{2\delta x}. \tag{80}$$

2. An array is generated with the values of $x[, y]$ to plot. For contour plots, a two-dimensional array is generated using `numpy.meshgrid`. Each value in the array is fed into the function to create another array with the result.

   - For cylindrical (or polar) coordinates, transformation from Cartesian coordinates are done via

     ```
     rho = numpy.sqrt(x**2 + y**2); phi = numpy.arctan2(y, x)
     ```

     where `numpy.arctan2` is the two-argument arctangent, which gives the angle between the ray to $(x, y)$ and the positive $x$-axis, and corrects the value for the quadrant of $(x, y)$ by compared to the single-argument arctangent. The specific formulae used are[24]

     $$\operatorname{arctan2}(y, x) = \begin{cases} \arctan(y/x) & x > 0, \\ \arctan(y/x) + \pi & x < 0, y > 0, \\ \arctan(y/x) - \pi & x < 0, y < 0, \end{cases} \quad \begin{cases} +\pi/2 & x = 0, y > 0, \\ -\pi/2 & x = 0, y < 0, \\ 0 \text{ or } \pi & x = 0, y = 0. \end{cases} \tag{81}$$

The last case being special values depending on the sign of 0, which are encoded as different values in the IEEE floating point standard[25].

3. The result is plotted with `matplotlib.axes.Axes.plot` for line graphs and `matplotlib.axes.Axes.contourf` for filled contour plots.

## 2.3 Radial Distribution Function

In this section, line graphs and radial contours are plotted for the radial distribution function (55) with variable values of $p$, $\ell$, and a set value of $w_0 = 729\,\mathrm{nm}$. They can be seen in figures 3–8.

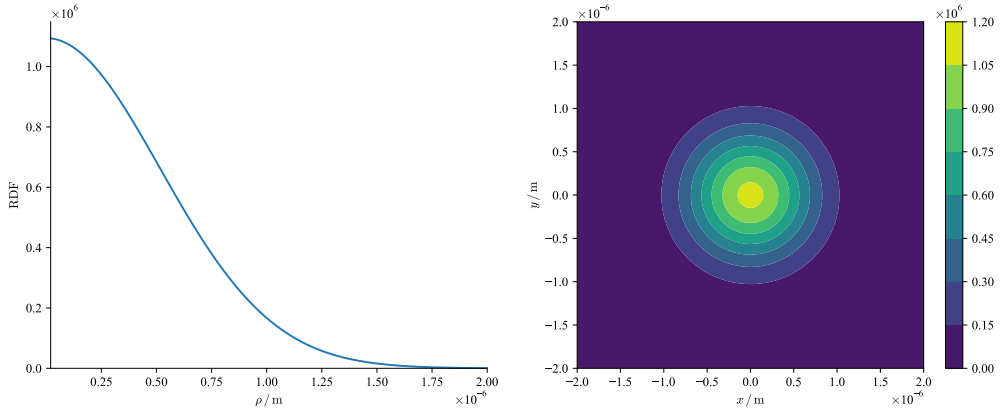


Figure 3: Line graph of the radial distribution function against $\rho$, and contour graph against $x, y$, for the values $p = 0, \ell = 0, w_0 = 729\,\mathrm{nm}$.

In the case of $p = 0, \ell = 0$, the beam amplitude reduces to a standard Gaussian beam.

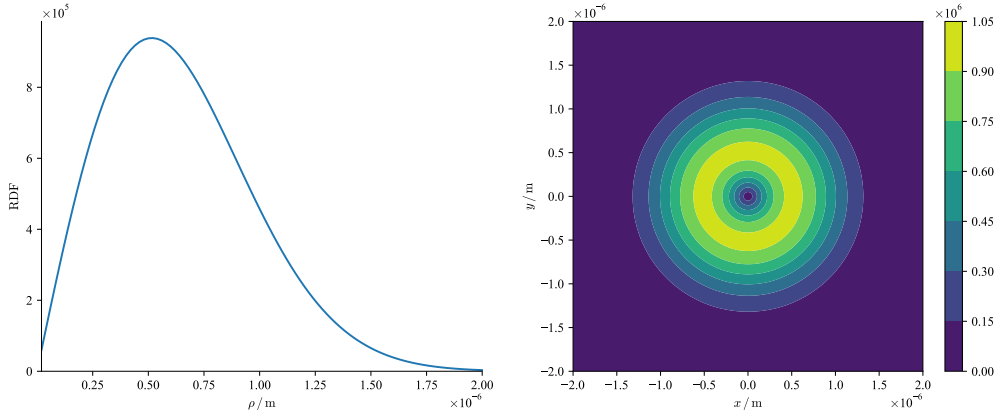


Figure 4: Line graph of the radial distribution function (55) against $\rho$, and contour graph against $x, y$, for the values $p = 0, \ell = \pm 1, w_0 = 729\,\mathrm{nm}$.

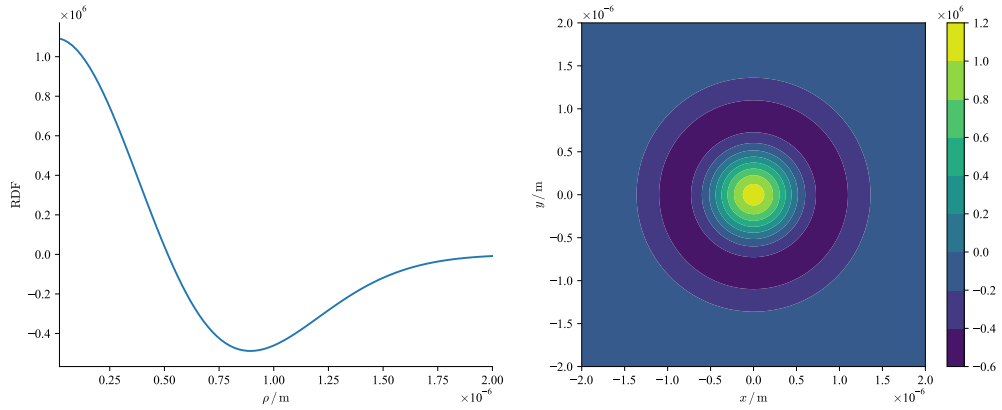


Figure 5: Line graph of the radial distribution function (55) against $\rho$, and contour graph against $x, y$, for the values $p = 1, \ell = 0, w_0 = 729\,\mathrm{nm}$.

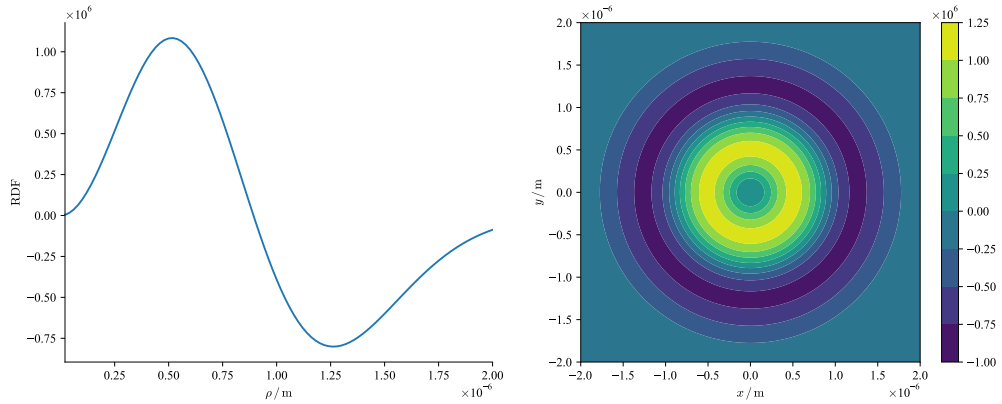


Figure 6: Line graph of the radial distribution function (55) against $\rho$, and contour graph against $x, y$, for the values $p = 1, \ell = \pm 2, w_0 = 729\,\mathrm{nm}$.

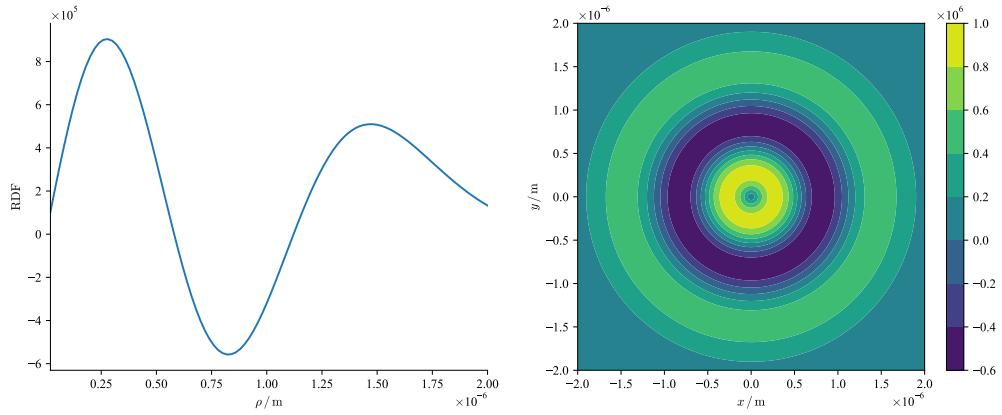


Figure 7: Line graph of the radial distribution function (55) against $\rho$, and contour graph against $x, y$, for the values $p = 2, \ell = \pm 1, w_0 = 729\,\mathrm{m}$.

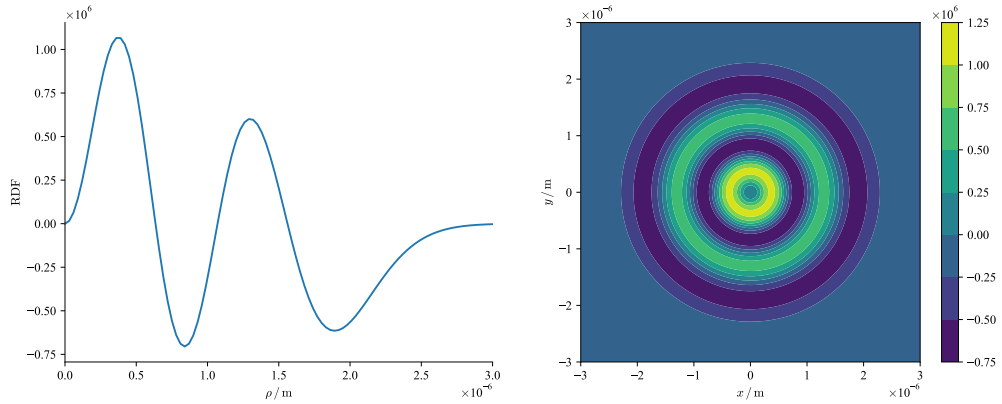


Figure 8: Line graph of the radial distribution function (55) against $\rho$, and contour graph against $x, y$, for the values $p = 3, \ell = \pm 2, w_0 = 729\,\mathrm{m}$.

It is well known that the intensity distribution of the Laguerre-Gaussian beam has $p+1$ "rings"[26]. The intensity is proportional to the amplitude squared, and it is seen in the above figures that there are $p+1$ crests and troughs, with the troughs representing negative amplitude which when squared creates a positive crest.

## 2.4 Interaction with a Paraxial Field

This section is concerned with the E1E2 and E2E2 couplings mentioned in §1.8.2. From this section onward, we denote the polarization $\boldsymbol{e}^{(\sigma)}$ as $(\hat{x} + i\sigma\hat{y})$, as we will be only dealing with circular polarizations; this constraint will simplify the future equations (84) and (85).

### 2.4.1 Multipole Couplings: E1E2 Term

The equation for the E1E2 term is

$$\begin{aligned}\mathrm{E1E2} = &\frac{\tilde{\Omega}^2 n}{\varepsilon_0^2} f_{\ell,p}^2(\rho)\,(\hat{x}+i\sigma\hat{y})_i\,(\hat{x}-i\sigma\hat{y})_k\,\mu_i^{mo}\bar{Q}_{kj}^{mo}\left(\hat{\rho}_j\frac{1}{f_{\ell,p}(\rho)}\frac{\partial f_{\ell,p}(\rho)}{\partial\rho} - i\frac{1}{\rho}\ell\hat{\varphi}_j - ik\hat{z}_j\right)\\ &+\frac{\Omega^2 n}{\varepsilon_0^2} f_{\ell,p}^2(\rho)\,(\hat{x}+i\sigma\hat{y})_i\,(\hat{x}-i\sigma\hat{y})_k\,\bar{\mu}_k^{mo}Q_{ij}^{mo}\left(\hat{\rho}_j\frac{1}{f_{\ell,p}(\rho)}\frac{\partial f_{\ell,p}(\rho)}{\partial\rho} + i\frac{1}{\rho}\ell\hat{\varphi}_j + ik\hat{z}_j\right).\end{aligned} \tag{82}$$

The computational analysis is as follows: a plot is produced for any set of parameters $\{p, \ell, \sigma, w_0, k, \boldsymbol{\mu}, \mathbf{Q}\}$, with the restrictions $p \in \mathbb{N}, \ell \in \mathbb{Z}, \sigma \in \{-1, 1\}$; the quadrupole transition moment tensor must satisfy $\mathrm{tr}(\mathbf{Q}) = 0, Q_{ij} = Q_{ji}$[27]. Each of these values can be changed in turn to see the effect on the contribution to the E1E2 coupling. Additionally, two other methods can be used to analyse the contribution: the first is splitting (82) into two expression in the form $\mathrm{E1}\overline{\mathrm{E}}2 + \overline{\mathrm{E}}\mathrm{1E2}$. The second is splitting (82) into its $\hat{\boldsymbol{\rho}}$, $\hat{\boldsymbol{\varphi}}$, and $\hat{\boldsymbol{z}}$ terms.

The standard set of values which are to be changed are given in figure 9. It is seen that for the E1E2 coupling, the right hand side positively contributes, and the left hand side negatively contributes to the absorption rate.

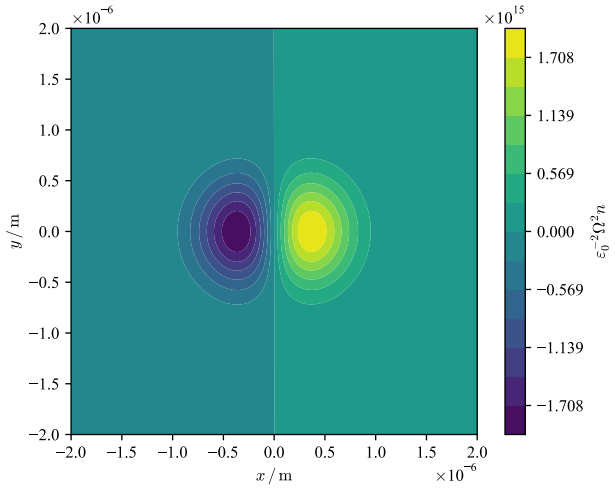


Figure 9: Contour plot of the E1E2 term (82), with the values $p = 0$, $\ell = 1$, $\sigma = 1$, $w_0 = 729\,\mathrm{nm}$, $kw_0 = 2\pi$, $\boldsymbol{\mu} = [\,1, 0, 0\,]$, diagonals of $\mathbf{Q}$ are $10^{-3}\,[\,1, -0.5, -0.5\,]$, non-diagonals $10^{-6}$. $x$ and $y$ ranges shown in figure.

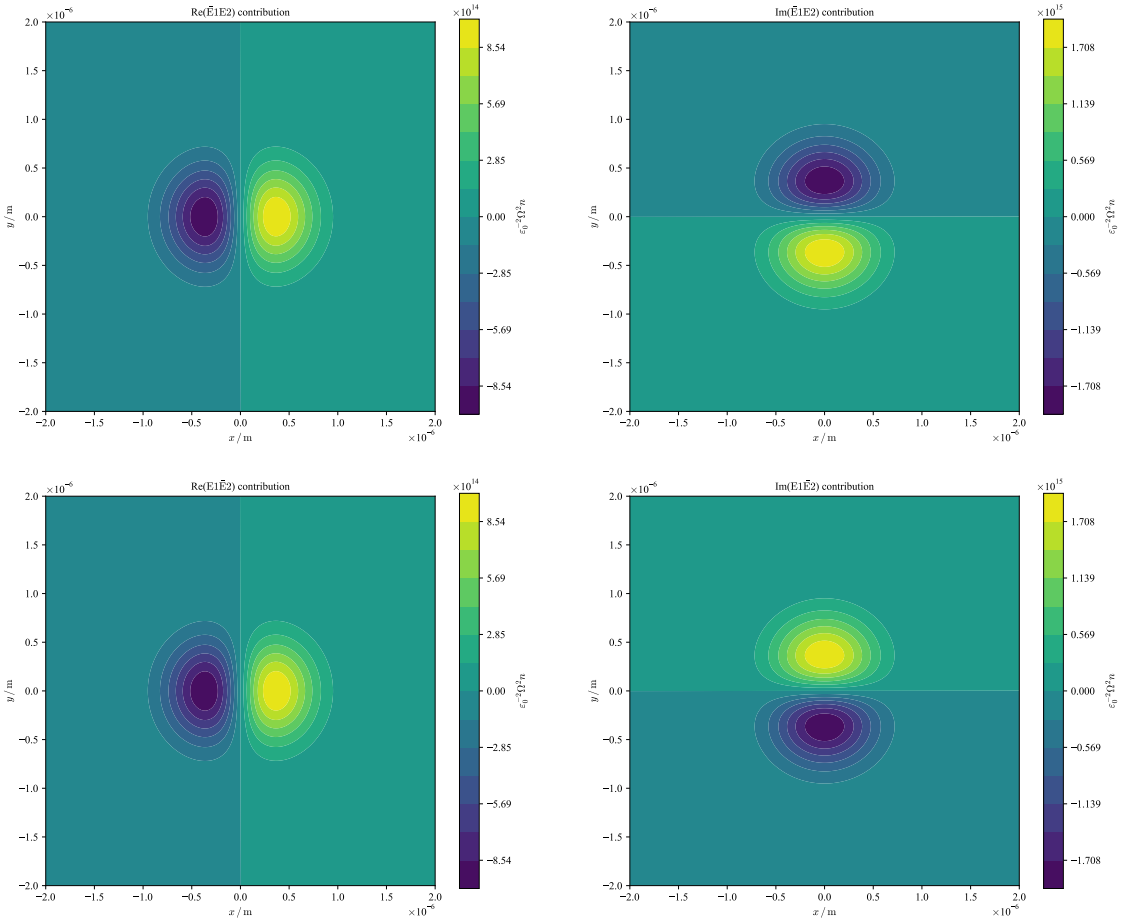


Figure 10: Real (left) and imaginary (right) contour plots of the E1E2 term in figure 9 with the $\overline{\mathrm{E}}1\mathrm{E}2$ expression on the top and the $\mathrm{E}1\overline{\mathrm{E}}2$ on the bottom.

By splitting the E1E2 term in the form $\mathrm{E1}\overline{\mathrm{E}}2 + \overline{\mathrm{E}}\mathrm{E1}2$, it is seen that the imaginary parts of the individual expressions cancel each other out, and the resulting expression is therefore real. The real parts of the expressions are the same and contribute half to the total term.

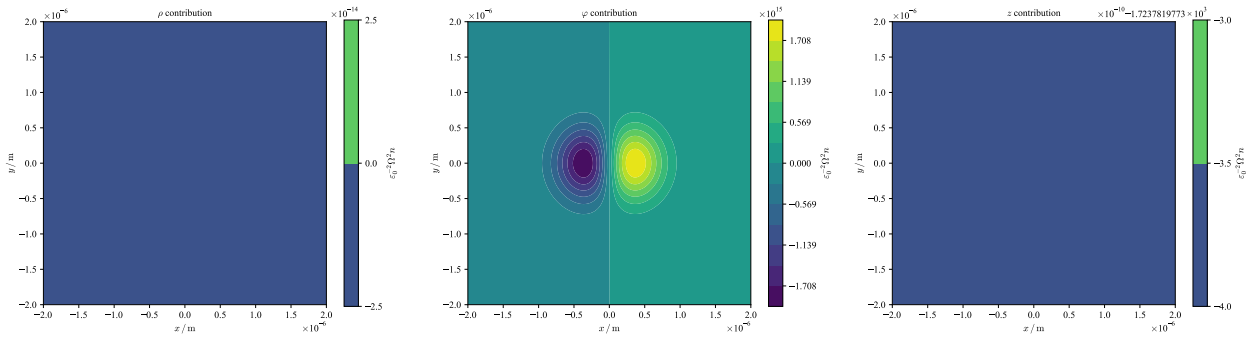


Figure 11: Figure 9 split into $\hat{\boldsymbol{\rho}}$ (left), $\hat{\boldsymbol{\varphi}}$ (middle), and $\hat{\boldsymbol{z}}$ (right) contributions.

By splitting the E1E2 term into $\hat{\boldsymbol{\rho}}$, $\hat{\boldsymbol{\varphi}}$, and $\hat{\boldsymbol{z}}$ terms, it is seen that in the choice of the values in figure 9, that the $\hat{\boldsymbol{\rho}}$ term contributes a null value and that the $\hat{\boldsymbol{z}}$ term contributes a constant of $\approx -1.7 \times 10^3$, which is negligible compared to the order of magnitude of the $\hat{\boldsymbol{\varphi}}$ contribution.

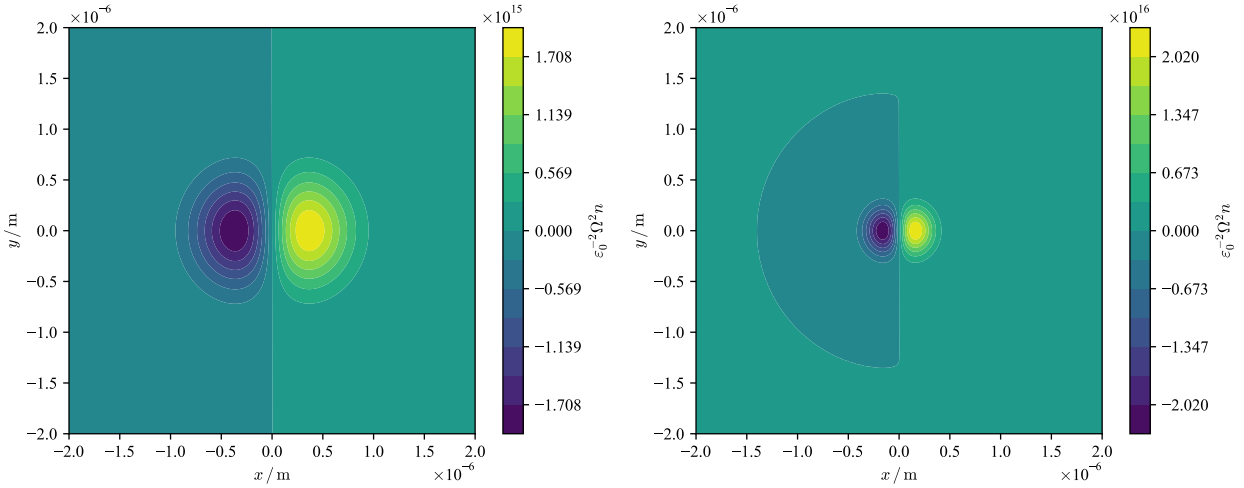


Figure 12: **Left:** E1E2 term with parameters defined in figure 9; **Right:** Changing the beam width $w_0$ to be 320 nm.

The first parameter that can be changed is $w_0$. This has the intuitive effect of narrowing the contribution in terms of its radial size. Similarly, increasing the beam width increases the size. (Note that the large semicircle produced on the right side of figure 12 is assumed to be a floating point error, as the values in this region are close in magnitude to a countour line. This would disappear upon increasing the number of countours, for example.)

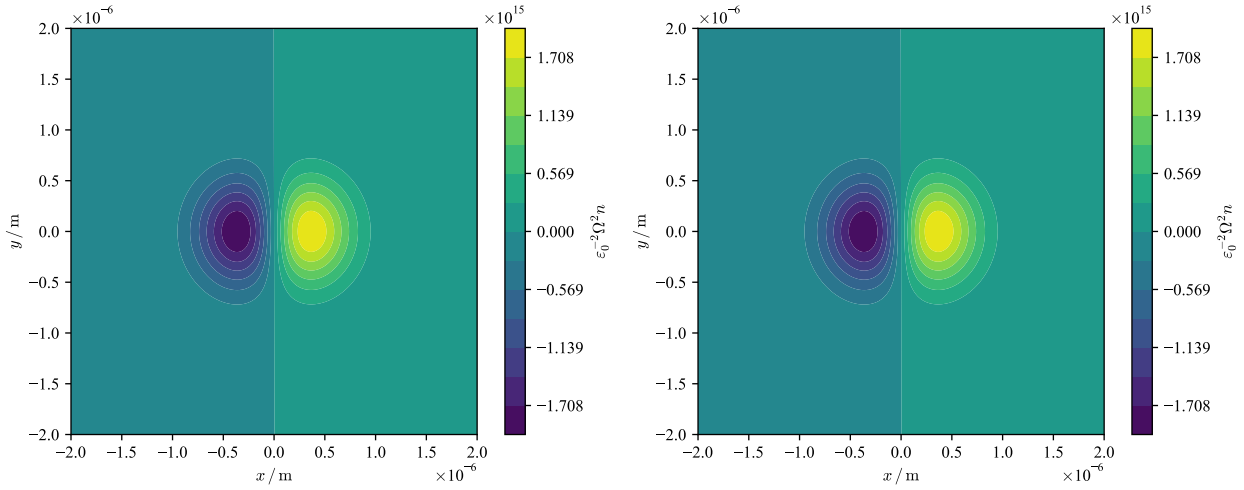


Figure 13: **Left:** E1E2 term with parameters defined in figure 9; **Right:** Changing the wavenumber $k$ to be $20\pi/w_0$.

Changing the parameter $k$ has no effect on the image. As the wavenumber appears only in the $\hat{\boldsymbol{z}}$ contribution of (82), and the effect of the $\hat{\boldsymbol{z}}$ contribution is marginal as seen in figure 11, it may be only until by increasing the wavenumber by tens of orders of magnitude that its effect is seen.

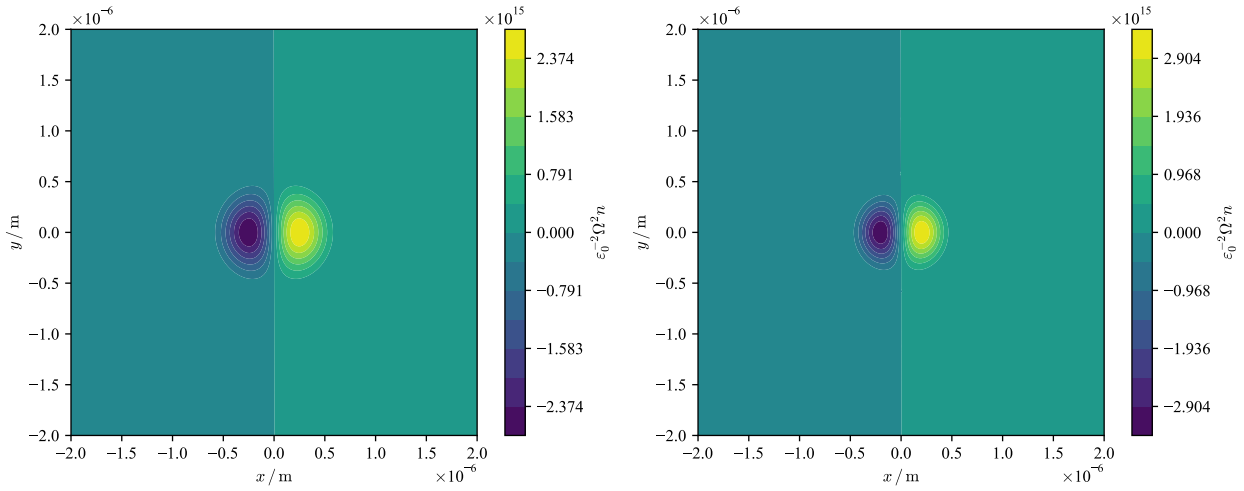


Figure 14: **Left:** E1E2 term with parameters defined in figure 9, with $p = 1$; **Right:** $p = 2$.

Increasing the radial index $p$ has the effect of narrowing the radial size, similarly to decreasing the beam width $w_0$ (figure 12).

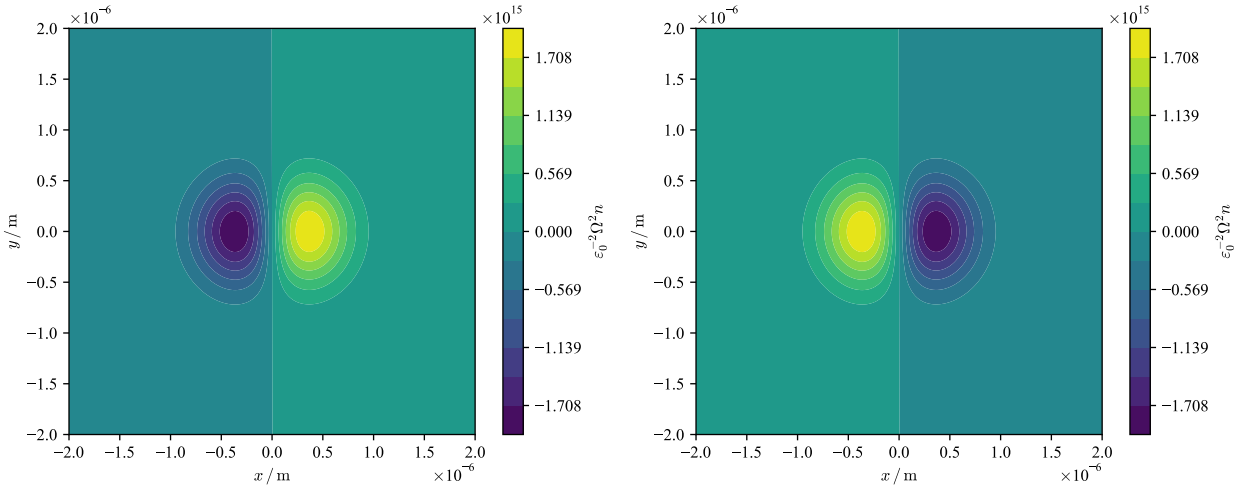


Figure 15: **Left:** E1E2 term with parameters defined in figure 9; **Right:** Changing the polarization $\sigma$ to be $-1$.

Changing the handedness of the circular polarization flips the image in the $x = 0$ axis. The effect can be alternatively described as: "by changing the chirality of the beam, the contributions have switched signs".

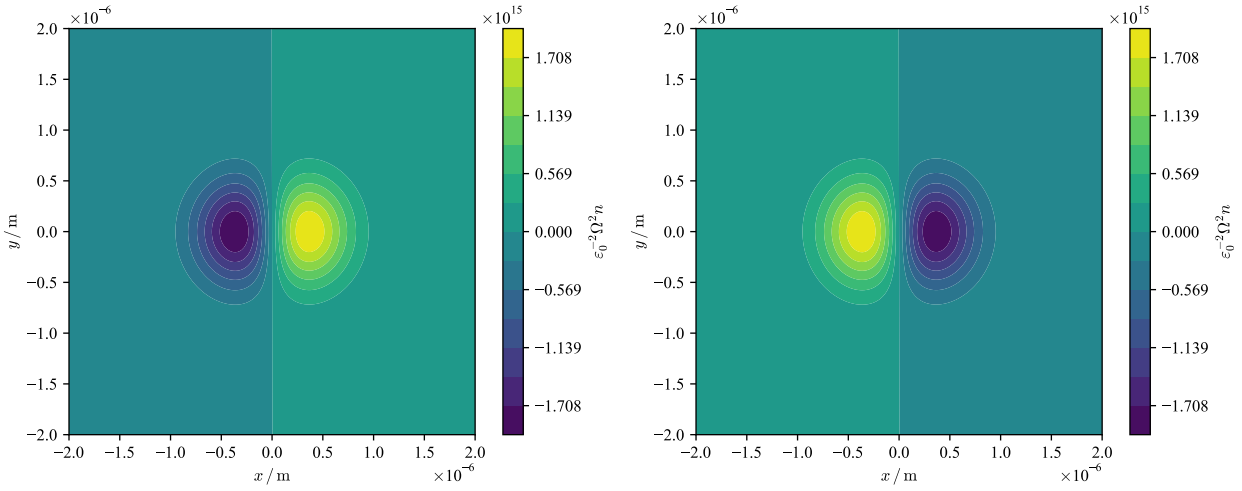


Figure 16: **Left:** E1E2 term with parameters defined in figure 9; **Right:** Changing the topological charge $\ell$ to be $-1$.

An identical effect happens when the handedness of the vortex is flipped, as these parameters both define the chirality of the beam.

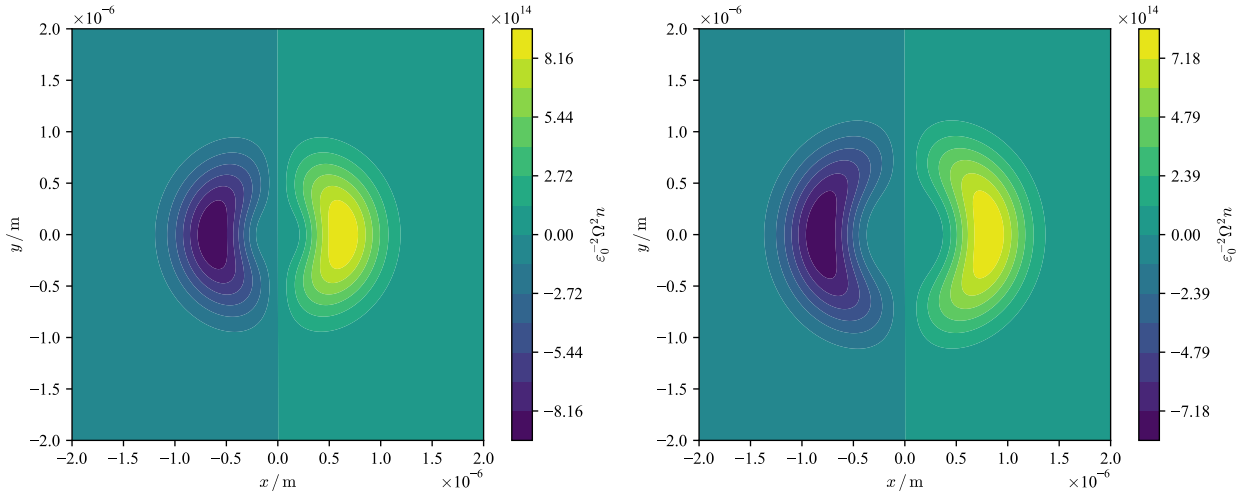


Figure 17: **Left:** E1E2 term with parameters defined in figure 9, with $\ell = 2$; **Right:** $\ell = 3$.

In addition to flipping the vortex handedness, we can also increase the value of the topological charge to any integer. This increases the "gap" in the centre, very similarly to the effect seen in the radial distribution function, figures 3 and 4.

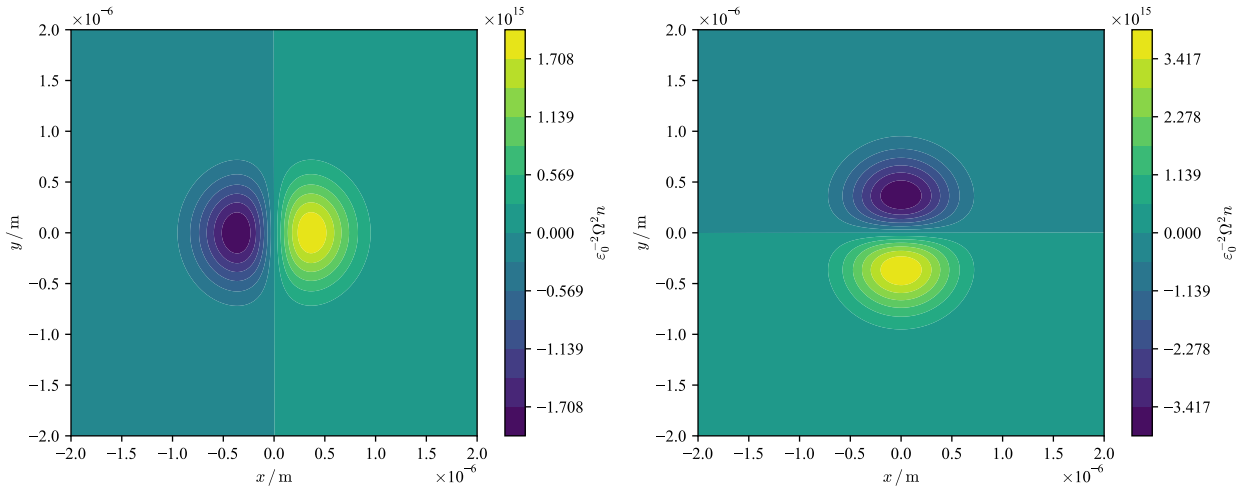


Figure 18: **Left:** E1E2 term with parameters defined in figure 9; **Right:** Changing the topological charge $\boldsymbol{\mu}$ to be $[0, 1, 0]$.

Changing the dipole transition moment to situate on the $y$ axis rotates the contribution by 90°. The effect of doing this to the $z$ axis creates a null result, as the countour graphs only plot across the $x, y$-plane.

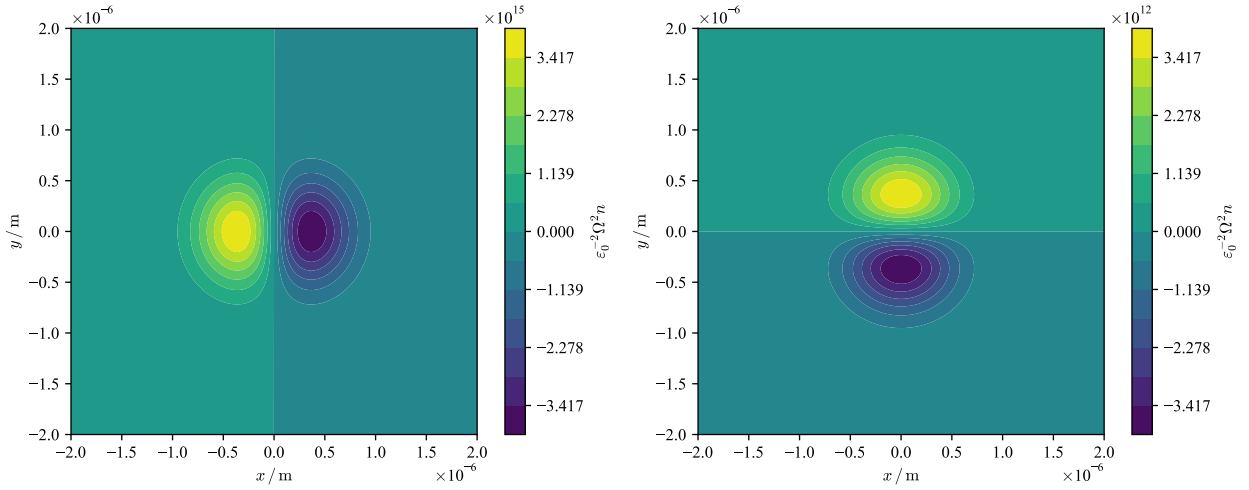


Figure 19: **Left:** E1E2 term with parameters defined in figure 9, with the diagonals of **Q** changed to be $[-0.5, 1, -0.5]$; **Right:** $[1, 0, -1]$.

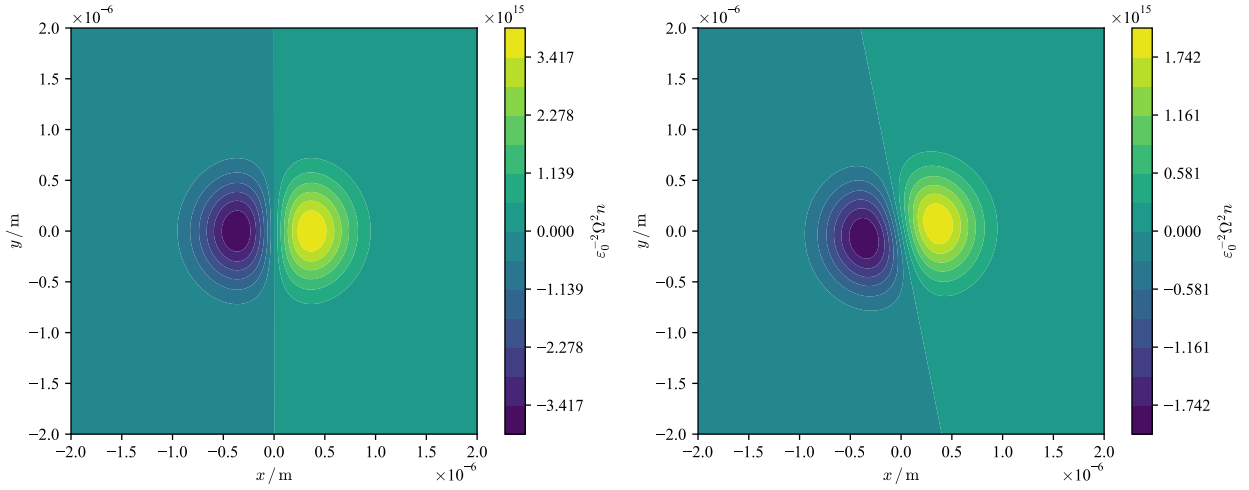


Figure 20: **Left:** E1E2 term with parameters defined in figure 9, with the diagonals of **Q** changed to be $[0.5, -1, 0.5]$; **Right:** The specific values of $Q_{xy}$ and $Q_{yx}$ changed to be $10^{-4}$.

For the values of the quadrupole transition matrix **Q**, the value of $Q_{yy}$ rotates the contribution by 90° is it is 0; else the handedness of the contribution changes depending on whether it is positive or negative. The magnitude of $Q_{yy}$ has no effect, neither do changing the values of $Q_{xx}$ or $Q_{zz}$. As the original set of values in figure 9 has the $Q_{yy}$ value to be negative, the left hand side of figure 20 is identical to figure 9. Additionally, for the non-diagonals, the only values that changes the effect of the contribution are the values $Q_{xy} = Q_{yx}$, which rotates the image by an arbitrary degree depending on its magnitude.

### 2.4.2 Multipole Couplings: E2E2 Term

The equation for the E2E2 term is

$$\mathrm{E2E2} = \frac{\tilde{\Omega}^2 n}{\varepsilon_0^2} f_{\ell,p}^2(\rho)\left(\hat{x}+i\sigma\hat{y}\right)_i\left(\hat{x}-i\sigma\hat{y}\right)_k Q_{ij}^{mo}\bar{Q}_{kl}^{mo}\left(\hat{\rho}_j\frac{1}{f_{\ell,p}(\rho)}\frac{\partial f_{\ell,p}(\rho)}{\partial\rho}-i\frac{1}{\rho}\ell\hat{\varphi}_j+ik\hat{z}_j\right)$$
$$\cdots\times\left(\hat{\rho}_l\frac{1}{f_{\ell,p}(\rho)}\frac{\partial f_{\ell,p}(\rho)}{\partial\rho}-i\frac{1}{\rho}\ell\hat{\varphi}_l-ik\hat{z}_l\right). \tag{83}$$

The countour plot with the values from figure 9 are applied to the E2E2 term:

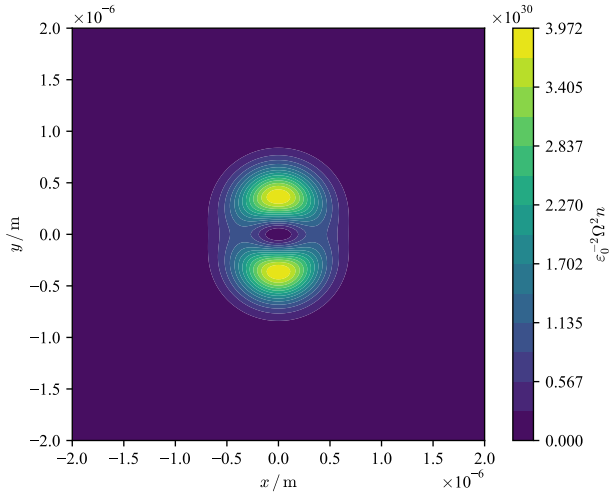


Figure 21: Contour plot of the E2E2 term (83), with the values $p = 0$, $\ell = 1$, $\sigma = 1$, $w_0 = 729\,\mathrm{nm}$, $kw_0 = 2\pi$, $\boldsymbol{\mu} = [1, 0, 0]$, diagonals of $\mathbf{Q}$ are $10^{-3}\,[1, -0.5, -0.5]$, non-diagonals $10^{-6}$. $x$ and $y$ ranges shown in figure.

Many of the transformations seen in figures 9 to 20 also apply to the E2E2 term, although ones of notable interest are changing the value of $\ell$ and the values of the quadrupole transition matrix $\mathbf{Q}$.

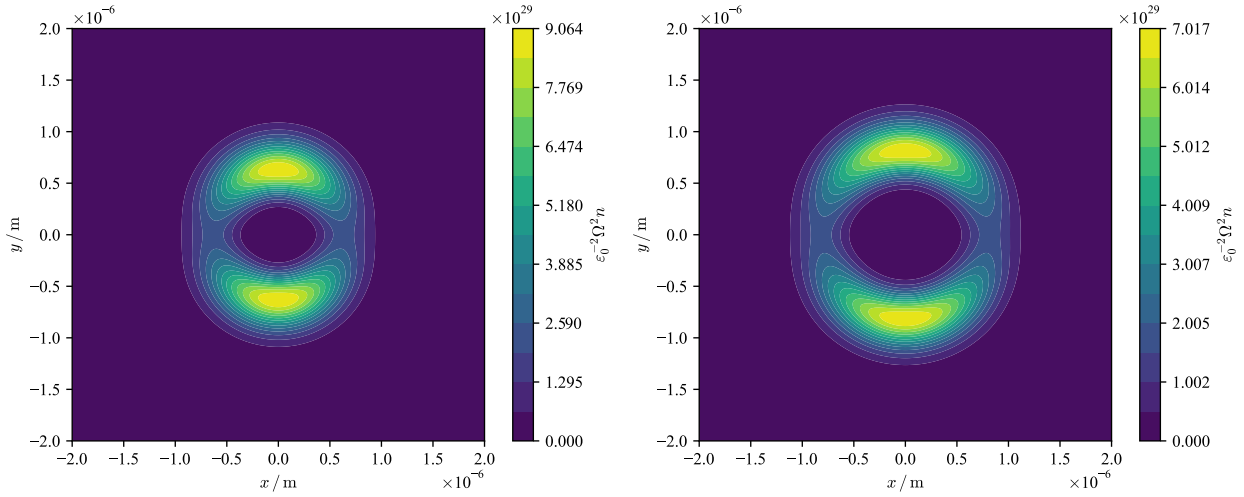


Figure 22: **Left:** E2E2 term with parameters defined in figure 21, with $\ell = 2$; **Right:** $\ell = 3$.

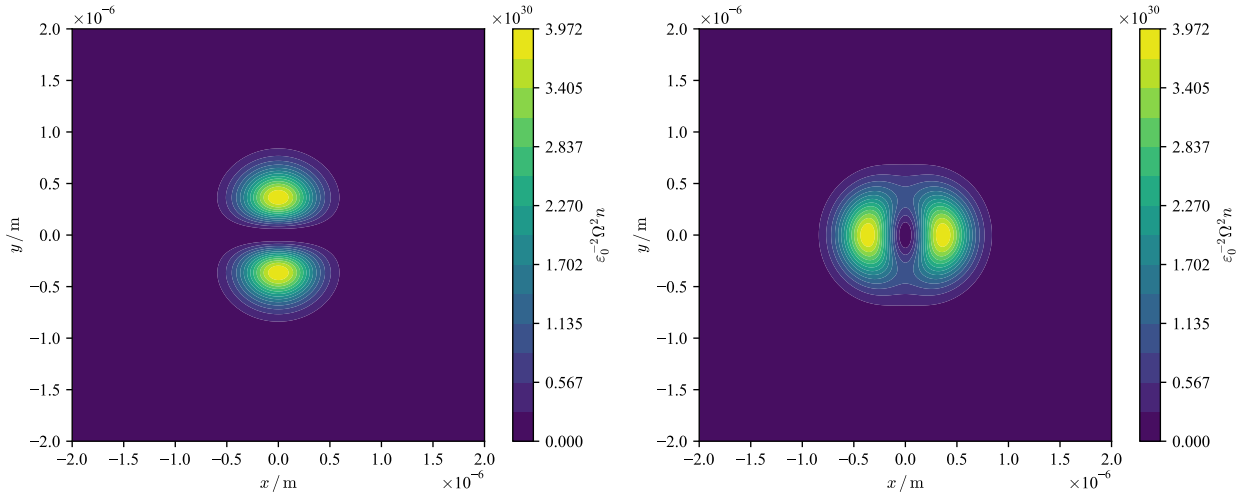


Figure 23: **Left:** E2E2 term with parameters defined in figure 21, with the diagonals of $\mathbf{Q}$ changed to be $[\,1, 0, -1\,]$; **Right:** $[\,-0.5, 1, -0.5\,]$.

## 2.5 Interactions with a Non-Paraxial Field

In §1.8.2, it is mentioned that the electric displacement and magnetic field mode expansions are completely transverse. This is the case when the mode expansions are performed using plane wave solutions to Maxwell's equations, but for any beam-like solution to the Helmholtz equation, (66) and (67) are only the zeroth order approximations; the total field is equal to $\boldsymbol{d}^{\perp} = \boldsymbol{d}^{\perp}_{x,y} + \hat{\boldsymbol{z}} d^{\perp}_{z}$ (and similarly for the magnetic field).

When the first order longitudinal corrections are included in the mode expansions, the result is[20]

$$\boldsymbol{d}^{\perp}(\boldsymbol{\rho}) = \sum_{\boldsymbol{k},\sigma,\ell,p} \frac{\tilde{\Omega}}{\sqrt{2}} \left[ \left\{ \boldsymbol{e}^{(\sigma)}(\boldsymbol{k}) + \frac{i}{k} \left( \frac{\partial}{\partial \rho} - \frac{\ell\sigma}{\rho} \right) e^{i\sigma\varphi} \hat{\boldsymbol{z}} \right\} f_{|\ell|,p}(\boldsymbol{\rho}) a^{(\sigma)}_{|\ell|,p}(\boldsymbol{k}) e^{i(kz+\ell\varphi)} - \texttt{H.C.} \right] \tag{84}$$

$$\boldsymbol{b}(\boldsymbol{\rho}) = \sum_{\boldsymbol{k},\sigma,\ell,p} \frac{\tilde{\Omega}}{\sqrt{2}\varepsilon_0 c} \left[ \left\{ \boldsymbol{b}^{(\sigma)}(\boldsymbol{k}) + \frac{i}{k} \left( \sigma \frac{\partial}{\partial \rho} - \frac{\ell}{\rho} \right) e^{i\sigma\varphi} \hat{\boldsymbol{z}} \right\} f_{|\ell|,p}(\boldsymbol{\rho}) a^{(\sigma)}_{|\ell|,p}(\boldsymbol{k}) e^{i(kz+\ell\varphi)} - \texttt{H.C.} \right] \tag{85}$$

where H.C. is the Hermitian conjugate of the previous expression. These equations are only valid for circular polarizations, but other types of polarizations can also be considered in the method of calculating the longitudinal components[28].

When these mode expansions are substituted into the interaction Hamiltonian (68), matrix element (7), and the Fermi golden rule (6), we arrive at the non-paraxial equations for the E1E2 and E2E2 couplings. Dropping the notational dependencies of the radial distribution function $f$, we have[17]

$$\begin{aligned}
\text{E1E2} = -\frac{\tilde{\Omega}^2 n}{2} \Bigg[ & \left\{ (\hat{x} + i\sigma\hat{y})_i f + \frac{i}{k} \left( \frac{\partial f}{\partial \rho} - \frac{\ell\sigma}{\rho} f \right) e^{i\sigma\varphi} \hat{z}_i \right\} \left\{ \left( \hat{\rho}_l \frac{\partial f}{\partial \rho} - \frac{i\ell}{\rho} f \hat{\varphi}_l - ifk\hat{z}_l \right) (\hat{x} - i\sigma\hat{y})_k \right. \\
& + \left( -\hat{\rho}_l \frac{i}{k} \frac{\partial}{\partial \rho} - \frac{(\ell+\sigma)}{k\rho} \hat{\varphi}_l - \hat{z}_l \right) \left( \frac{\partial f}{\partial \rho} - \frac{\ell\sigma}{\rho} f \right) e^{-i\sigma\varphi} \hat{z}_k \Big\} \mu^{mo}_i \bar{Q}^{mo}_{kl} \\
& + \left\{ (\hat{x} - i\sigma\hat{y})_k f - \frac{i}{k} \left( \frac{\partial f}{\partial \rho} - \frac{\ell\sigma}{\rho} f \right) e^{-i\sigma\varphi} \hat{z}_k \right\} \left\{ \left( \hat{\rho}_j \frac{\partial f}{\partial \rho} + \frac{i\ell}{\rho} f \hat{\varphi}_j + ifk\hat{z}_j \right) (\hat{x} + i\sigma\hat{y})_i \right. \\
& \left. + \left( \hat{\rho}_j \frac{i}{k} \frac{\partial}{\partial \rho} - \frac{(\ell+\sigma)}{k\rho} \hat{\varphi}_j - \hat{z}_j \right) \left( \frac{\partial f}{\partial \rho} - \frac{\ell\sigma}{\rho} f \right) e^{i\sigma\varphi} \hat{z}_i \Big\} \bar{\mu}^{mo}_k Q^{mo}_{ij} \right]
\end{aligned} \tag{86}$$

and

$$\mathrm{E2E2} = -\frac{\tilde{\Omega}^2 n}{2}\left[\left\{\left(\hat{\rho}_j\frac{\partial f}{\partial\rho}+\frac{i\ell}{\rho}f\hat{\varphi}_j+ifk\hat{z}_j\right)(\hat{x}+i\sigma\hat{y})_i+\left(\hat{\rho}_j\frac{i}{k}\frac{\partial}{\partial\rho}-\frac{(\ell+\sigma)}{k\rho}\hat{\varphi}_j-\hat{z}_j\right)\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)e^{i\sigma\varphi}\hat{z}_i\right\}\right.$$
$$\left.\times\left\{\left(\hat{\rho}_l\frac{\partial f}{\partial\rho}-\frac{i\ell}{\rho}f\hat{\varphi}_l-ifk\hat{z}_l\right)(\hat{x}-i\sigma\hat{y})_k+\left(-\hat{\rho}_l\frac{i}{k}\frac{\partial}{\partial\rho}-\frac{(\ell+\sigma)}{k\rho}\hat{\varphi}_l-\hat{z}_l\right)\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)e^{-i\sigma\varphi}\hat{z}_k\right\}\right]Q_{ij}^{mo}\bar{Q}_{kl}^{mo} \tag{87}$$

For the purposes of clarity in the structure of the code, we split this into individual expressions as

$$\mathrm{E1E2} = -\frac{\tilde{\Omega}^2 n}{2}[\{\mathtt{a1}\}_i\{(\mathtt{a2p1})_l(\hat{x}-i\sigma\hat{y})_k+(\mathtt{a2p2})_l(\mathtt{a2p3})_k\}\mu_i\bar{Q}_{kl}$$
$$+\{\mathtt{b1}\}_k\left\{(\mathtt{b2p1})_j(\hat{x}+i\sigma\hat{y})_i+(\mathtt{b2p2})_j(\mathtt{b2p3})_i\right\}]\bar{\mu}_k Q_{ij} \tag{88}$$
$$\mathrm{E2E2} = -\frac{\tilde{\Omega}^2 n}{2}[\left\{(\mathtt{b2p1})_j(\hat{x}+i\sigma\hat{y})_i+(\mathtt{b2p2})_j(\mathtt{b2p3})_i\right\}$$
$$\times\{(\mathtt{a2p1})_l(\hat{x}-i\sigma\hat{y})_k+(\mathtt{a2p2})_l(\mathtt{a2p3})_k\}]Q_{ij}\bar{Q}_{kl} \tag{89}$$

where

$$(\mathtt{a1})_i = (\hat{x}+i\sigma\hat{y})_i f+\frac{i}{k}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)e^{i\sigma\varphi}\hat{z}_i \tag{90}$$
$$(\mathtt{a2p1})_l = \hat{\rho}_l\frac{\partial f}{\partial\rho}-\frac{i\ell}{\rho}f\hat{\varphi}_l-ifk\hat{z}_l \tag{91}$$
$$(\mathtt{a2p2})_l = \left(-\hat{\rho}_l\frac{i}{k}\frac{\partial}{\partial\rho}-\frac{(\ell+\sigma)}{k\rho}\hat{\varphi}_l-\hat{z}_l\right)\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)$$
$$= \left[-\hat{\rho}\frac{i}{k}\frac{\partial}{\partial\rho}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)-\frac{(\ell+\sigma)}{k\rho}\hat{\varphi}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)-\hat{z}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)\right]_l$$
$$= \left[\left(-\hat{\rho}\frac{i}{k}\frac{\partial^2 f}{\partial r^2}\right)-\left(-\hat{\rho}\frac{i}{k}\ell\sigma\frac{\partial}{\partial\rho}\frac{f}{\rho}\right)-\frac{(\ell+\sigma)}{k\rho}\hat{\varphi}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)-\hat{z}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)\right]_l \tag{92}$$
$$(\mathtt{a2p3})_k = e^{-i\sigma\varphi}\hat{z}_k \tag{93}$$

and

$$(\mathtt{b1})_k = (\hat{x}-i\sigma\hat{y})_k f-\frac{i}{k}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)e^{-i\sigma\varphi}\hat{z}_k \tag{94}$$
$$(\mathtt{b2p1})_j = \hat{\rho}_j\frac{\partial f}{\partial\rho}+\frac{i\ell}{\rho}f\hat{\varphi}_j+ifk\hat{z}_j \tag{95}$$
$$(\mathtt{b2p2})_j = \left(\hat{\rho}_j\frac{i}{k}\frac{\partial}{\partial\rho}-\frac{(\ell+\sigma)}{k\rho}\hat{\varphi}_j-\hat{z}_j\right)\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)$$
$$= \left[\hat{\rho}\frac{i}{k}\frac{\partial}{\partial\rho}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)-\frac{(\ell+\sigma)}{k\rho}\hat{\varphi}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)-\hat{z}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)\right]_j$$
$$= \left[\left(\hat{\rho}\frac{i}{k}\frac{\partial^2 f}{\partial r^2}\right)-\left(\hat{\rho}\frac{i}{k}\ell\sigma\frac{\partial}{\partial\rho}\frac{f}{\rho}\right)-\frac{(\ell+\sigma)}{k\rho}\hat{\varphi}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)-\hat{z}\left(\frac{\partial f}{\partial\rho}-\frac{\ell\sigma}{\rho}f\right)\right]_j \tag{96}$$
$$(\mathtt{b2p3})_i = e^{i\sigma\varphi}\hat{z}_i \tag{97}$$

### 2.5.1 Multipole Couplings: E1E2 Term

We now begin the comparison of paraxial and non-paraxial couplings. Perhaps suprisingly, these countour plots are significantly different to their paraxial equivalents.

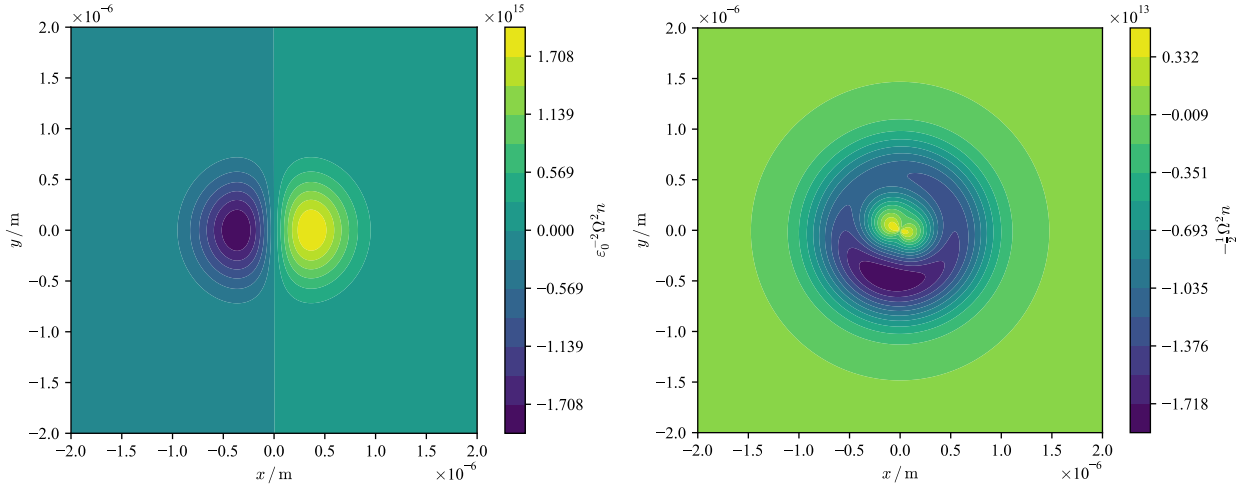


Figure 24: **Left:** Paraxial E1E2 term with parameters defined in figure 9; **Right:** Non-paraxial equivalent.

The effect of changing the beam width is identical and will not be shown in the report.

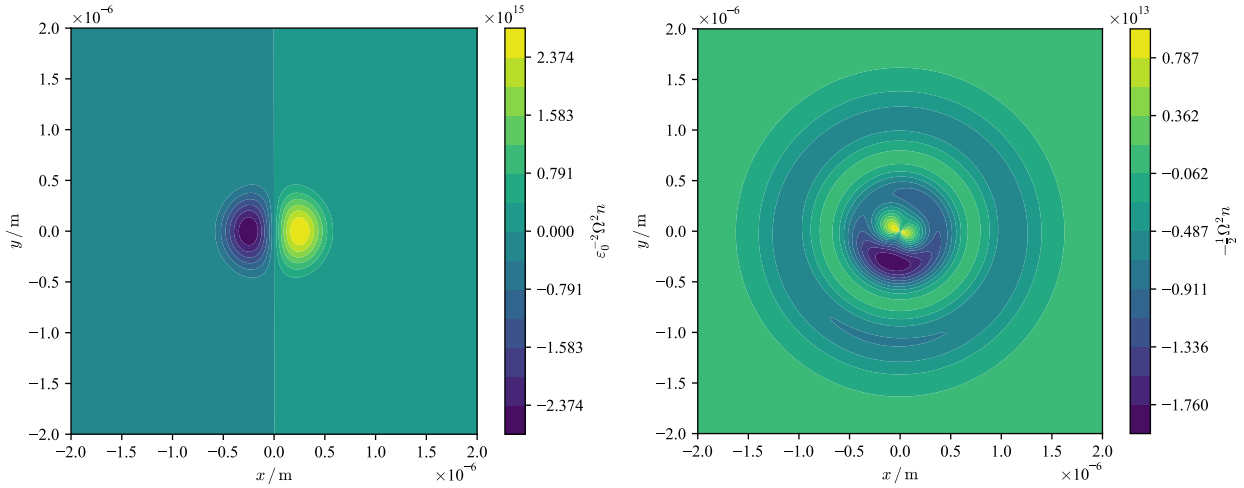


Figure 25: **Left:** Paraxial E1E2 term with parameters defined in figure 9 except $p = 1$; **Right:** Non-paraxial equivalent.

The effect of increasing $p$ is similar in the non-paraxial version, but there are some residual "blemishes" in the outer rings.

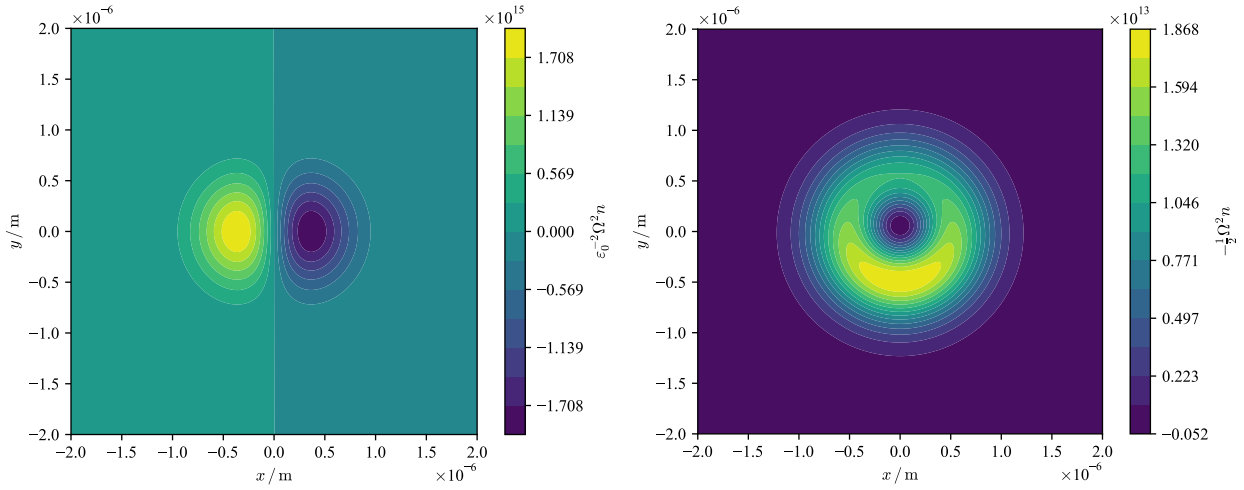


Figure 26: **Left:** Paraxial E1E2 term with parameters defined in figure 9 except $\sigma = -1$; **Right:** Non-paraxial equivalent.

The effect of changing the polarization of the beam has somewhat of a similar effect with the changing of the contribution's sign, except that the structure in the center of the ring appears to be missing compared to in figure 24.

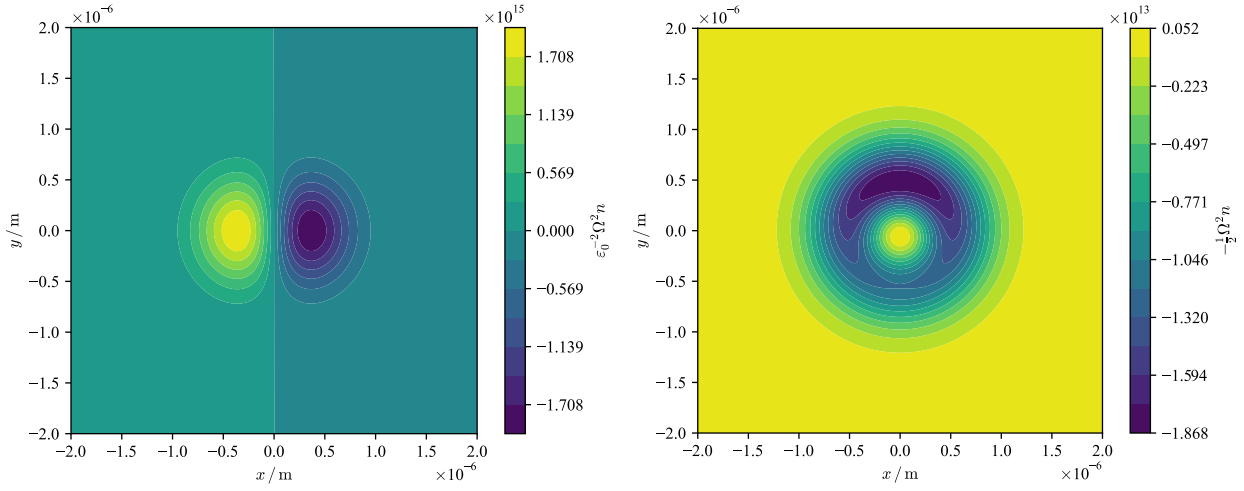


Figure 27: **Left:** Paraxial E1E2 term with parameters defined in figure 9 except $\ell = -1$; **Right:** Non-paraxial equivalent.

The same cannot be said by changing the handedness of the beam vortex. It appears that the image is flipped in the $y = 0$ axis, and the magnitude of the contribution is slightly different.

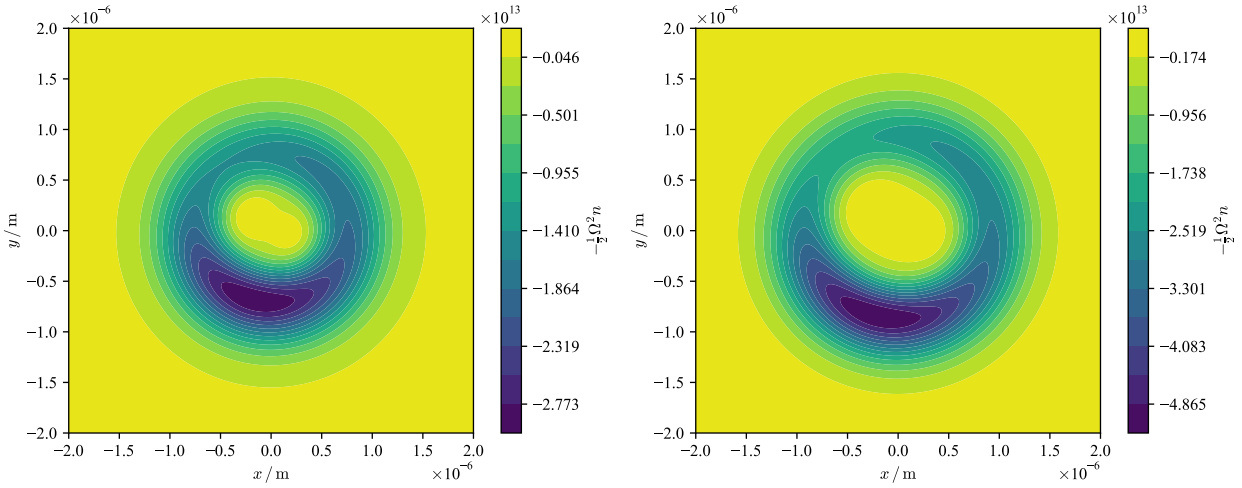


Figure 28: **Left:** Non-paraxial E1E2 term with $\ell = 2$; **Right:** $\ell = 3$.

Increasing $\ell$ to other integer values has a similar effect of increasing the gap in the centre.

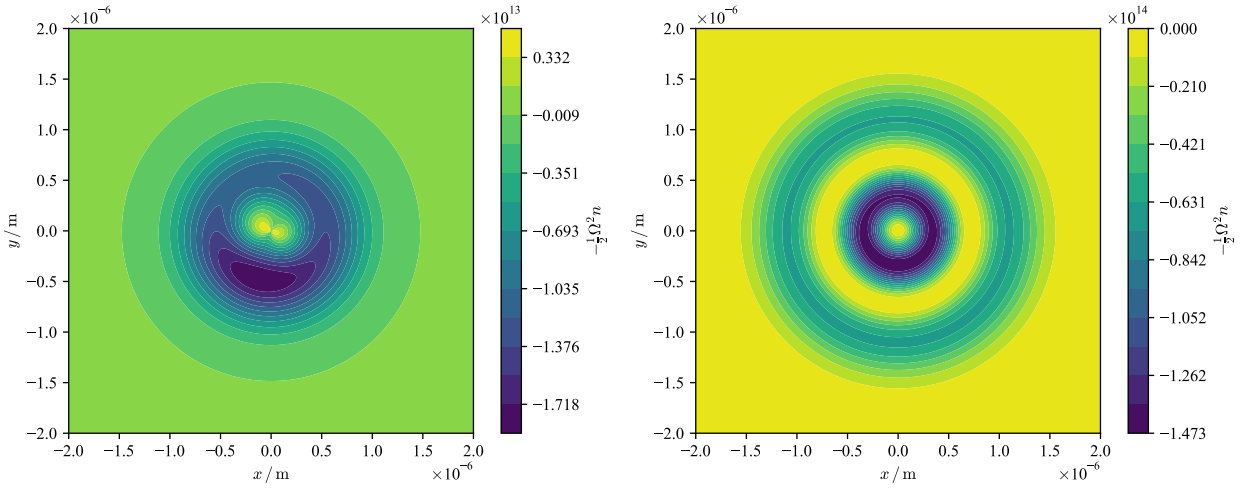


Figure 29: **Left:** Non-paraxial E1E2 term with parameters defined in figure 9; **Right:** The value of $kw_0$ to $kw_0 = 20\pi$.

Unlike the paraxial equivalent, changing the wavenumber has a noticeable effect. However, increasing the wavenumber further has smaller and smaller effect, contrary to expectations that as $kw_0$ increases, the non-paraxial contributions will resemble more like the paraxial equivalent.

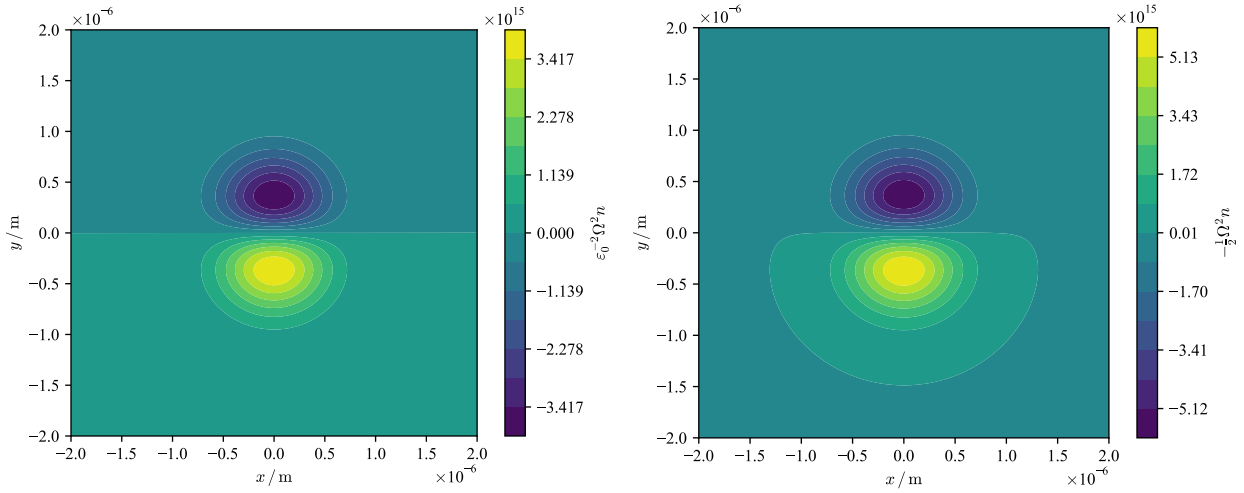


Figure 30: **Left:** Paraxial E1E2 term with parameters defined in figure 9 except $\boldsymbol{\mu} = [0, 1, 0]$; **Right:** Non-paraxial equivalent.

When changing the electric dipole moment to situate on the $y$ axis, the paraxial and non-paraxial plots look similar.

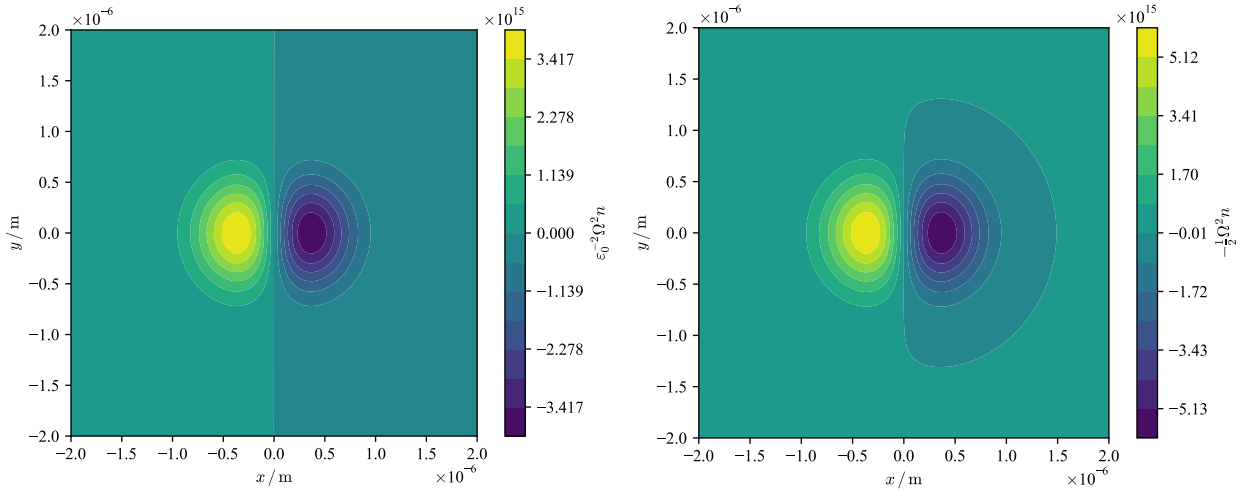


Figure 31: **Left:** Paraxial E1E2 term with parameters defined in figure 9 except diagonals of $\mathbf{Q}$ are $[-0.5, 1, -0.5]$; **Right:** Non-paraxial equivalent.

All the effects of changing the value of $Q_{yy}$ result in a similar image to the right of figure 31, which is very similar to its paraxial equivalent. The only exception is when $Q_{yy}$ is changed to a number close to $-0.5$. It is unknown the importance of the number, but as can be seen in figure 32, when $Q_{yy}$ approaches this value, the non-paraxial plots "transform" into the plot shown in figure 24.

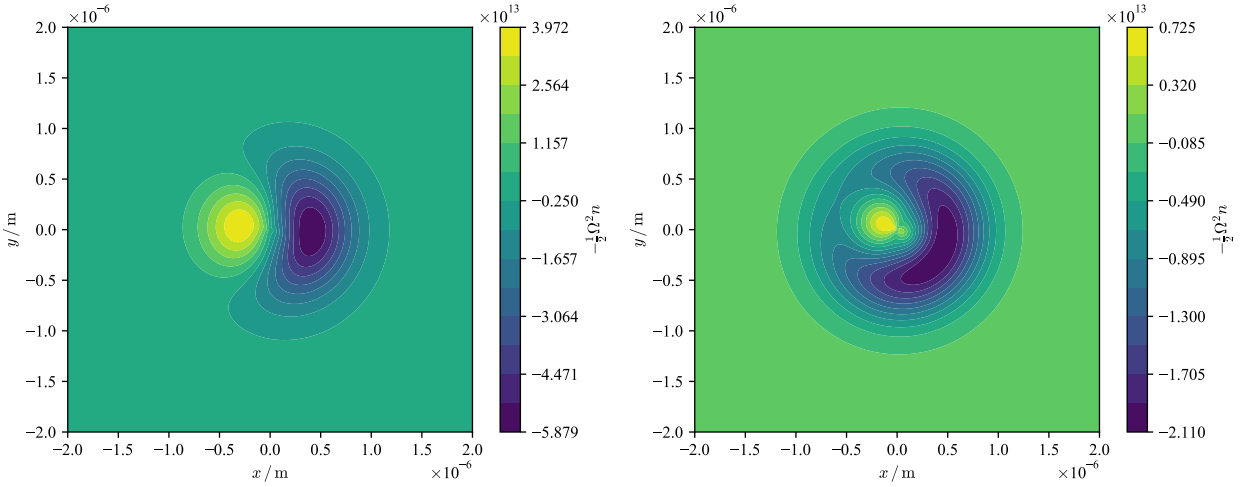


Figure 32: **Left:** Non-Paraxial E1E2 term with parameters defined in figure 9 except diagonals of $\mathbf{Q}$ are $[1, -0.496, -0.504]$; **Right:** $[1, -0.499, -0.501]$

### 2.5.2 Multipole Couplings: E2E2 Term

When plotting the non-paraxial E2E2 terms, all images appear to be similar to the figure shown to the left of figure 33: Zeroes everywhere except a small point in the centre of large magnitude. The only value to which this does not happen is increasing the wavenumber by a large amount.

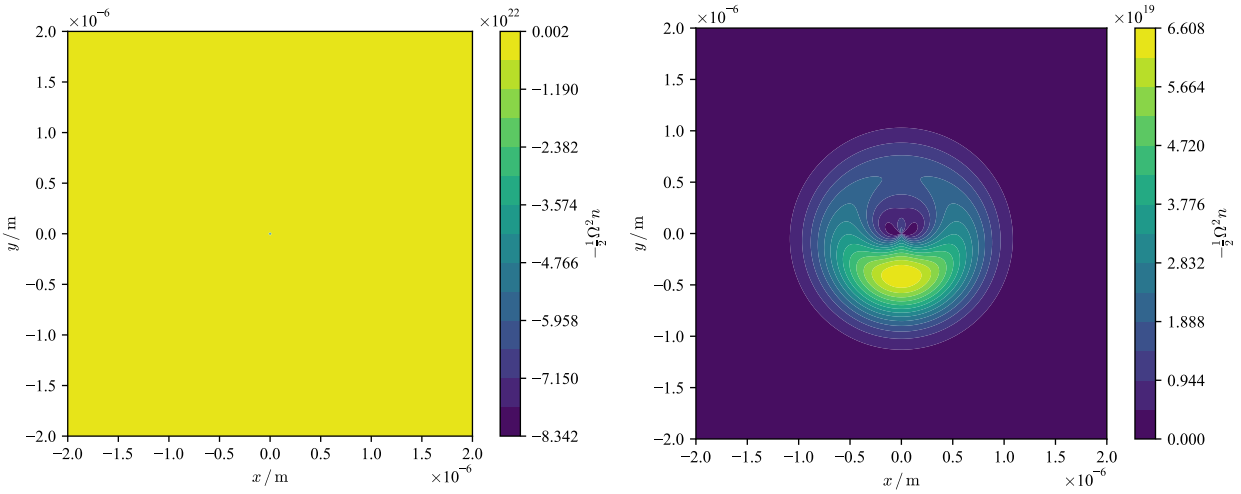


Figure 33: **Left:** Non-Paraxial E2E2 term with parameters defined in figure 9; **Right:** the value of $kw_0$ changed to be $1000\pi$.

*3*

***Discussion***

The effect of the inclusion of longitudinal fields explain certain phenomena such as vortex dichroism without the presence of circularly polarized beams. By continuing with the method described in §2.5, vortex dichroism terms appear in the E1M1 coupling and survive the effect of rotational averaging (in comparison to the end of §1.8.2)[17, §B]:

$$\left\langle \Gamma^{|\sigma|}_{|\ell|} \right\rangle - \left\langle \Gamma^{|\sigma|}_{-|\ell|} \right\rangle \propto \frac{N}{3} \frac{n\hbar k}{\varepsilon_0 A^2 V} \frac{2|\ell|}{k^2 \rho} f \frac{\partial f}{\partial \rho} R^{mo} = \left\langle \Gamma^{\sigma=0}_{|\ell|} \right\rangle - \left\langle \Gamma^{\sigma=0}_{-|\ell|} \right\rangle \tag{98}$$

We saw in §2.5.1 and §2.5.2 that the differences in the paraxial and non-paraxial couplings are significantly different. While some of the transformations applied to the paraxial couplings are intuitive, the non-paraxial equivalents are somewhat unexpected, to which there are still some unexplained properties. As seen in figure 32, the cause of this was the quadrupole tensor $\mathbf{Q}$.

It was seen that by analytical calculation, the E1E2 contributions disappear upon rotational averaging (§1.4), when the molecules in the substance are randomly orientated. This effect however, only occurs in liquids and gases; the exception are in some structured solids. Therefore the contributions seen in the plots of §2.4 and §2.5 do affect the absorption rate of light photons.

As there was only limited space in the report, the analysis of changing parameters is limited and we only changed one parameter at a time. As such, there are also many other combinations of parameters that could have an effect on the contributions of the E1E2 and E2E2 couplings, especially with respect to the non-paraxial fields. Additionally, the contributions of $\hat{\boldsymbol{\rho}}$, $\hat{\boldsymbol{\varphi}}$ and $\hat{\boldsymbol{z}}$ terms could also be applied to the non-paraxial fields; this would require the restructuring of equations (86) and (87).

The work in this report can easily be expanded upon to other couplings, including M1E2, and even higher order terms such as the second magnetic multipole moment M2 and the electric octopule E3, although as is the property of multipole expansions, the effect that higher moments give are expected to be very small.

*4*

***Bibliography***


[1] D. P. Craig and T. Thirunamachandran. *Molecular Quantum Electrodynamics*. Dover Books on Chemistry. Dover Publications, Mineola, New York, 2012. ISBN 9780486135632. URL `https://books.google.co.uk/books?id=S6DDAgAAQBAJ`.

[2] J. J. Sakurai and J. Napolitano. *Modern Quantum Mechanics*. Cambridge University Press, Cambridge, 3rd edition, 2020. ISBN 9781108473224. URL `https://books.google.co.uk/books?id=GdX7DwAAQBAJ`.

[3] Paul Adrien Maurice Dirac et al. *The Principles of Quantum Mechanics*. Number 27 in the International Series of Monographs on Physics. Oxford University Press, Great Clarendon Street, Oxford, 4th edition, 1981. ISBN 9780198520115. URL `https://books.google.co.uk/books?id=XehUpGiM6FIC`.

[4] Paul Adrien Maurice Dirac and Niels Henrik David Bohr. The quantum theory of the emission and absorption of radiation. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 114(767):243–265, 1927. doi:10.1098/rspa.1927.0039.

[5] Edwin Albert Power and T. Thirunamachandran. The multipolar hamiltonian in radiation theory. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 372(1749):265–273, 1980. doi:10.1098/rspa.1980.0112.

[6] Patrick J. Lestrange, Franco Egidi, and Xiaosong Li. The consequences of improperly describing oscillator strengths beyond the electric dipole approximation. *The Journal of Chemical Physics*, 143(23):234103, 2015. doi:10.1063/1.4937410.

[7] Todd Rowland and Eric W. Weisstein. Tensor. `https://mathworld.wolfram.com/Tensor.html`, 2022. [Online].

[8] Joseph C. Kolecki. An introduction to tensors for students of physics and engineering. Technical report, National Aeronautics and Space Administration, John H. Glenn Research Centre, Cleveland, Ohio, July 2002. URL `https://www.grc.nasa.gov/www/k-12/Numbers/Math/documents/Tensors_TM2002211716.pdf`.

[9] Ilan Ben-Yaacov and Francesc Roig. Index notation for vector calculus. `http://www.ees.nmt.edu/outside/courses/GEOP523/Docs/index-notation.pdf`, 2006.

[10] David L. Andrews and Thuraiappah Thirunamachandran. On three-dimensional rotational averages. *The Journal of Chemical Physics*, 67(11):5026–5033, 1977. doi:10.1063/1.434725.

[11] Hermann Weyl. *The Classical Groups: Their Invariants and Representations*, volume 45. Princeton University Press, 1946. ISBN 9781400883905. doi:10.1515/9781400883905.

[12] Kevin F. Brennan. *The Physics of Semiconductors: With Applications to Optoelectronic Devices*. Cambridge University Press, 1999. doi:10.1017/CBO9781139164214.

[13] J. Magnes, D. Odera, J. Hartke, M. Fountain, L. Florence, and V. Davis. Quantitative and qualitative study of gaussian beam visualization techniques, 2006.

[14] Les Allen, Marco W. Beijersbergen, R. J. C. Spreeuw, and J. P. Woerdman. Orbital angular momentum of light and the transformation of Laguerre-Gaussian laser modes. *Physical review A*, 45(11):8185, 1992. doi:10.1103/PhysRevA.45.8185.

[15] Kayn A. Forbes and David L. Andrews. Orbital angular momentum of twisted light: chirality and optical activity. *Journal of Physics: Photonics*, 3(2):022007, March 2021. doi:10.1088/2515-7647/abdb06.

[16] David S. Bradshaw, Jamie M. Leeder, Matt M. Coles, and David L. Andrews. Signatures of material and optical chirality: Origins and measures. *Chemical Physics Letters*, 626:106–110, 2015. ISSN 0009-2614. doi:10.1016/j.cplett.2015.02.051.

[17] Kayn A. Forbes and Garth A. Jones. Optical vortex dichroism in chiral particles. *Phys. Rev. A*, 103: 053515, May 2021. doi:10.1103/PhysRevA.103.053515.

[18] Ward Brullot, Maarten K. Vanbel, Tom Swusten, and Thierry Verbiest. Resolving enantiomers using the optical angular momentum of twisted light. *Science Advances*, 2(3):e1501349, 2016. doi:10.1126/sciadv.1501349.

[19] John A. Schellman. Circular dichroism and optical rotation. *Chemical Reviews*, 75(3):323–331, 1975. doi:10.1021/cr60295a004.

[20] Kayn A. Forbes, Dale Green, and Garth A. Jones. Relevance of longitudinal fields of paraxial optical vortices. *Journal of Optics*, 23(7):075401, May 2021. doi:10.1088/2040-8986/abff96.

[21] David L. Andrews. Quantum formulation for nanoscale optical and material chirality: symmetry issues, space and time parity, and observables. *Journal of Optics*, 20(3):033003, 2018. doi:10.1088/2040-8986/aaaa56.

[22] NumPy Developers. numpy.einsum. `https://numpy.org/doc/stable/reference/generated/numpy.einsum.html`, 2022. [Online].

[23] Patrick Walls. Mathematical python. `https://github.com/patrickwalls/mathematical-python/`, 2022. [Online].

[24] NumPy Developers. numpy.arctan2. `https://numpy.org/doc/stable/reference/generated/numpy.arctan2.html`, 2022. [Online].

[25] IEEE. IEEE standard for floating-point arithmetic. *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pages 1–84, 2019. doi:10.1109/IEEESTD.2019.8766229.

[26] William N. Plick and Mario Krenn. Physical meaning of the radial index of Laguerre-Gauss beams. *Phys. Rev. A*, 92:063841, December 2015. doi:10.1103/PhysRevA.92.063841.

[27] Eric W. Weisstein. Electric quadrupole moment. `https://scienceworld.wolfram.com/physics/ElectricQuadrupoleMoment.html`, 2017. [Online].

[28] Oleg V. Angelsky, Aleksandr Y. Bekshaev, Steen G. Hanson, Claudia Yu Zenkova, Igor I. Mokhun, and Zheng Jun. Structured light: Ideas and concepts. *Frontiers in Physics*, 8, 2020. ISSN 2296-424X. doi:10.3389/fphy.2020.00114.

*5*

***Appendices***

## 5.1 Values of $I^{(n)}$ for various $n$

Rotational averages for tensors of rank 2 to 6, expressed in the form of equation (24), can be found in this subsection[1, p. 312].

$$
\begin{aligned}
I^{(2)} &= \frac{1}{3}\delta_{i_1i_2}\delta_{\lambda_1\lambda_2} \\
I^{(3)} &= \frac{1}{6}\varepsilon_{i_1i_2i_3}\varepsilon_{\lambda_1\lambda_2\lambda_3} \\
I^{(4)} &= \frac{1}{30}\begin{bmatrix}\delta_{i_1i_2}\delta_{i_3i_4}\\ \delta_{i_1i_3}\delta_{i_2i_4}\\ \delta_{i_1i_4}\delta_{i_2i_3}\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}4&-1&-1\\-1&4&-1\\-1&-1&4\end{bmatrix}\begin{bmatrix}\delta_{\lambda_1\lambda_2}\delta_{\lambda_3\lambda_4}\\ \delta_{\lambda_1\lambda_3}\delta_{\lambda_2\lambda_4}\\ \delta_{\lambda_1\lambda_4}\delta_{\lambda_2\lambda_3}\end{bmatrix} \\
I^{(5)} &= \frac{1}{30}\begin{bmatrix}\varepsilon_{i_1i_2i_3}\delta_{i_4i_5}\\ \varepsilon_{i_1i_2i_4}\delta_{i_3i_5}\\ \varepsilon_{i_1i_2i_5}\delta_{i_3i_4}\\ \varepsilon_{i_1i_3i_4}\delta_{i_2i_5}\\ \varepsilon_{i_1i_3i_5}\delta_{i_2i_4}\\ \varepsilon_{i_1i_4i_5}\delta_{i_2i_3}\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}3&-1&-1&1&1&0\\-1&3&-1&-1&0&1\\-1&-1&3&0&-1&-1\\1&-1&0&3&-1&1\\1&0&-1&-1&3&-1\\0&1&-1&1&-1&3\end{bmatrix}\begin{bmatrix}\varepsilon_{\lambda_1\lambda_2\lambda_3}\delta_{\lambda_4\lambda_5}\\ \varepsilon_{\lambda_1\lambda_2\lambda_4}\delta_{\lambda_3\lambda_5}\\ \varepsilon_{\lambda_1\lambda_2\lambda_5}\delta_{\lambda_3\lambda_4}\\ \varepsilon_{\lambda_1\lambda_3\lambda_4}\delta_{\lambda_2\lambda_5}\\ \varepsilon_{\lambda_1\lambda_3\lambda_5}\delta_{\lambda_2\lambda_4}\\ \varepsilon_{\lambda_1\lambda_4\lambda_5}\delta_{\lambda_2\lambda_3}\end{bmatrix}
\end{aligned}
$$

$$
I^{(6)} = \frac{1}{210}\begin{bmatrix}\delta_{i_1i_2}\delta_{i_3i_4}\delta_{i_5i_6}\\ \delta_{i_1i_2}\delta_{i_3i_5}\delta_{i_4i_6}\\ \delta_{i_1i_2}\delta_{i_3i_6}\delta_{i_4i_5}\\ \delta_{i_1i_3}\delta_{i_2i_4}\delta_{i_5i_6}\\ \delta_{i_1i_3}\delta_{i_2i_5}\delta_{i_4i_6}\\ \delta_{i_1i_3}\delta_{i_2i_6}\delta_{i_4i_5}\\ \delta_{i_1i_4}\delta_{i_2i_3}\delta_{i_5i_6}\\ \delta_{i_1i_4}\delta_{i_2i_5}\delta_{i_3i_6}\\ \delta_{i_1i_4}\delta_{i_2i_6}\delta_{i_3i_5}\\ \delta_{i_1i_5}\delta_{i_2i_3}\delta_{i_4i_6}\\ \delta_{i_1i_5}\delta_{i_2i_4}\delta_{i_3i_6}\\ \delta_{i_1i_5}\delta_{i_2i_6}\delta_{i_3i_4}\\ \delta_{i_1i_6}\delta_{i_2i_3}\delta_{i_4i_5}\\ \delta_{i_1i_6}\delta_{i_2i_4}\delta_{i_3i_5}\\ \delta_{i_1i_6}\delta_{i_2i_5}\delta_{i_3i_4}\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}16&-5&-5&-5&2&2&-5&2&2&2&2&-5&2&2&-5\\-5&16&-5&2&-5&2&2&2&-5&-5&2&2&2&-5&2\\-5&-5&16&2&2&-5&2&-5&2&2&-5&2&-5&2&2\\-5&2&2&16&-5&-5&-5&2&2&2&-5&2&2&-5&2\\2&-5&2&-5&16&-5&2&-5&2&-5&2&2&2&2&-5\\2&2&-5&-5&-5&16&2&2&-5&2&2&-5&-5&2&2\\-5&2&2&-5&2&2&16&-5&-5&-5&2&2&-5&2&2\\2&2&-5&2&-5&2&-5&16&-5&2&-5&2&2&2&-5\\2&-5&2&2&2&-5&-5&-5&16&2&2&-5&2&-5&2\\2&-5&2&2&-5&2&-5&2&2&16&-5&-5&-5&2&2\\2&2&-5&-5&2&2&2&-5&2&-5&16&-5&2&-5&2\\-5&2&2&2&2&-5&2&2&-5&-5&-5&16&2&2&-5\\2&2&-5&2&2&-5&-5&2&2&-5&2&2&16&-5&-5\\2&-5&2&-5&2&2&2&2&-5&2&-5&2&-5&16&-5\\-5&2&2&2&-5&2&2&-5&2&2&2&-5&-5&-5&16\end{bmatrix}\begin{bmatrix}\delta_{\lambda_1\lambda_2}\delta_{\lambda_3\lambda_4}\delta_{\lambda_5\lambda_6}\\ \delta_{\lambda_1\lambda_2}\delta_{\lambda_3\lambda_5}\delta_{\lambda_4\lambda_6}\\ \delta_{\lambda_1\lambda_2}\delta_{\lambda_3\lambda_6}\delta_{\lambda_4\lambda_5}\\ \delta_{\lambda_1\lambda_3}\delta_{\lambda_2\lambda_4}\delta_{\lambda_5\lambda_6}\\ \delta_{\lambda_1\lambda_3}\delta_{\lambda_2\lambda_5}\delta_{\lambda_4\lambda_6}\\ \delta_{\lambda_1\lambda_3}\delta_{\lambda_2\lambda_6}\delta_{\lambda_4\lambda_5}\\ \delta_{\lambda_1\lambda_4}\delta_{\lambda_2\lambda_3}\delta_{\lambda_5\lambda_6}\\ \delta_{\lambda_1\lambda_4}\delta_{\lambda_2\lambda_5}\delta_{\lambda_3\lambda_6}\\ \delta_{\lambda_1\lambda_4}\delta_{\lambda_2\lambda_6}\delta_{\lambda_3\lambda_5}\\ \delta_{\lambda_1\lambda_5}\delta_{\lambda_2\lambda_3}\delta_{\lambda_4\lambda_6}\\ \delta_{\lambda_1\lambda_5}\delta_{\lambda_2\lambda_4}\delta_{\lambda_3\lambda_6}\\ \delta_{\lambda_1\lambda_5}\delta_{\lambda_2\lambda_6}\delta_{\lambda_3\lambda_4}\\ \delta_{\lambda_1\lambda_6}\delta_{\lambda_2\lambda_3}\delta_{\lambda_4\lambda_5}\\ \delta_{\lambda_1\lambda_6}\delta_{\lambda_2\lambda_4}\delta_{\lambda_3\lambda_5}\\ \delta_{\lambda_1\lambda_6}\delta_{\lambda_2\lambda_5}\delta_{\lambda_3\lambda_4}\end{bmatrix}
$$

## 5.2 Python Code

Figures set in §2 are produced by the python code below.

../src/rdf.py

```python
# -*- coding: utf-8 -*-
"""
Coder: James Jia
Latest Modification: 2022-05-16
Description: Coding of the radial distribution function given by _________
"""

import numpy as np
import scipy.special as sp
```

```
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'Times New Roman'
plt.rcParams['mathtext.fontset'] = 'cm'
plt.rcParams["axes.formatter.limits"] = [-5,5]

def normalizationFactor(p:int, l:int) -> float:
    '''
    p : Radial Index
    l : Topological charge

    C : Normalization factor
    '''
    Csq = 2**(abs(l) + 1)\
        * np.math.factorial(p)\
        * 1/(np.pi * np.math.factorial(p + abs(l)))
    C = np.sqrt(Csq)
    return C

def radialDistributionFunction(p:int, l:int, w0:float, r:float) -> float:
    '''
    p (int) : Radial Index
    l (int): Topological charge
    w0 : Beam waist at z = 0, in m
    r : Position, in m

    f : Radial Distribution Function
    '''
    C = normalizationFactor(p, l)
    f = (C/w0)\
      * ((np.sqrt(2) * r)/(w0))**np.abs(l)\
      * np.exp(-r**2/w0**2)\
      * sp.genlaguerre(p, np.abs(l))(2*r**2/w0**2)
    return f

def visualizeRDF(p:int, l:int, w0:float, rmin:float, rmax:float,\
                 saveSVG=False) -> None:
    '''
    function (object): Function to visualize
    p (int): Radial Index
    l (int): Topological charge
    w0 : Beam waist at z = 0, in m
    rmin : Minimum x, y position to plot, in m
    rmax : Maximum x, y position to plot, in m
    saveSVG (bool): Save an SVG of the result?
    '''
    interval = (rmax - rmin)/200
    x = np.arange(rmin, rmax+interval, interval)
    y = np.arange(rmin, rmax+interval, interval)
    r = x[x>=0]
    x, y = np.meshgrid(x, y)
    rho = np.sqrt(x**2 + y**2)
    line = radialDistributionFunction(p, l, w0, r)
    countour = radialDistributionFunction(p, l, w0, rho)
```

```
64
65      fig, (ax1, ax2) = plt.subplots(1, 2)
66
67      ax1plot = ax1.plot(r, line)
68      ax1.set_xlim(r[0], r[-1])
69      if line[line<=0].size == 0: #set bottom y-lim to 0 if all values are +ive
70          ax1.set_ylim(0,)
71      ax1.set(xlabel=r'$\rho\,/\,\mathrm{m}$', ylabel=r'$\mathrm{RDF}$')
72      ax1.spines['top'].set_visible(False)
73      ax1.spines['right'].set_visible(False)
74      ax1.ticklabel_format(useMathText=True)
75      ax2plot = ax2.contourf(x, y, countour)
76      ax2.set_aspect('equal', 'box')
77      ax2.set(xlabel=r'$x\,/\,\mathrm{m}$', ylabel=r'$y\,/\,\mathrm{m}$')
78      ax2.ticklabel_format(useMathText=True)
79      cbar = fig.colorbar(ax2plot, ax=ax2)
80      cbar.ax.ticklabel_format(useMathText=True)
81
82      fig.set_size_inches(13.5,5)
83      fig.set_dpi(200)
84      if saveSVG:
85          plt.savefig(f"RDF_{p}_{l}.svg", bbox_inches="tight")
```

../src/paraxial.py

```
1   # -*- coding: utf-8 -*-
2   """
3   Coder: James Jia
4   Description: Coding of paraxial equations given by _________
5   """
6
7   import numpy as np
8   from scipy.misc import derivative
9   from rdf import radialDistributionFunction
10
11  multiplier = r'$\varepsilon_0^{-2}\Omega^2n$'
12
13  def e1e2Term(p:int, l:int, w0:float, k:float, sigma:int,\
14               rho:"np.array", phi:"np.array", **kwargs) -> float:
15      '''
16      p (int): Radial Index
17      l (int): Topological charge
18      w0 : Beam waist at z = 0, in m
19      k : Wavenumber
20      sigma : polarization
21      rho (2D array): rho position, in m
22      phi (2D array): phi position, in rad
23
24      Required keyword arguments:
25       mu (vect.): electric dipole transition moment
26       Q (matr.): quadrupole transition moment
27
28      Optional keyword arguments:
```

```
29          rfzseperate (bool): send e1e2 as an array with [rho,phi,z] values
30          e1e2seperate (bool): send e1e2 as an array with [bar(e)1e2, e1bar(e)2] values
31
32      e1e2 : E1E2 term of the matrix element
33      '''
34      # Get values from kwargs. The kwargs exist for compatibility in the
35      # visualizeFunct function in FinalReport_Visualize file
36      mu = kwargs["mu"]
37      Q = kwargs["Q"]
38      rfzseperate = False if kwargs.get("rfzseperate") is None\
39          else kwargs.get("rfzseperate")
40      e1e2seperate = False if kwargs.get("e1e2seperate") is None\
41          else kwargs.get("e1e2seperate")
42
43      def rdf(rho):
44          return radialDistributionFunction(p, l, w0, rho)
45      # Arrays need to be consistent, so "0" needs to have the same shape
46      # as phi and rho.
47      assert np.shape(phi) == np.shape(rho)
48      arrayShape = np.shape(phi)
49      zeroArray = np.zeros(arrayShape)
50
51      # Calculations of values as follows
52      derf = derivative(rdf, rho, dx=0.01)
53      barQ = np.conjugate(Q)
54      barmu = np.conjugate(mu)
55      xpisy = np.array([1,1j*sigma,0])
56      xmisy = np.array([1,-1j*sigma,0])
57
58      rhohatTerm = rdf(rho)*derf *\
59               np.array([np.cos(phi), np.sin(phi), zeroArray])
60      phihatTerm = rdf(rho)**2/rho * (1j/l) *\
61               np.array([-np.sin(phi), np.cos(phi), zeroArray])
62      zhatTerm = 1j*k *\
63             np.array([zeroArray, zeroArray, np.ones(arrayShape)])
64
65      if rfzseperate == False and e1e2seperate == False:
66          rfzarray1 = rhohatTerm + phihatTerm + zhatTerm
67          rfzarray2 = rhohatTerm - phihatTerm - zhatTerm
68
69          # Compute einsum. "rfzarray" is a rank 1 tensor, but its components (x,y,z)
70          # are 2D arrays, which makes einsum think it is rank 3. So there needs to
71          # be extraction of "rfzarray" into individual "rfz" terms, calculate each
72          # einsum with a "rfz" term seperately, and reassemble them back into an
73          # array.
74          e1e2 = np.empty(arrayShape, dtype = 'complex_')
75          for i in range(arrayShape[0]):
76              for j in range(arrayShape[1]):
77                  rfz1 = [rfzarray1[0][i][j],rfzarray1[1][i][j],rfzarray1[2][i][j]]
78                  rfz2 = [rfzarray2[0][i][j],rfzarray2[1][i][j],rfzarray2[2][i][j]]
79                  e1e2[i][j] = np.einsum("i,k,k,ij,j",xpisy,xmisy,barmu,Q,rfz1)\
80                               + np.einsum("i,k,i,kj,j",xpisy,xmisy,mu,barQ,rfz2)
81      elif rfzseperate == True and e1e2seperate == False:
82          rfzarray1 = [rhohatTerm, phihatTerm, zhatTerm]
```

```
        rfzarray2 = [rhohatTerm, -phihatTerm, -zhatTerm]

        # now has an additional argument for rho, phi, zhat
        e1e2 = np.empty((3,arrayShape[0],arrayShape[1]), dtype = 'complex_')
        for i in range(arrayShape[0]):
            for j in range(arrayShape[1]):
                rho1 = [rfzarray1[0][0][i][j],rfzarray1[0][1][i][j],rfzarray1[0][2][i][j]]
                rho2 = [rfzarray2[0][0][i][j],rfzarray2[0][1][i][j],rfzarray2[0][2][i][j]]
                phi1 = [rfzarray1[1][0][i][j],rfzarray1[1][1][i][j],rfzarray1[1][2][i][j]]
                phi2 = [rfzarray2[1][0][i][j],rfzarray2[1][1][i][j],rfzarray2[1][2][i][j]]
                z1 = [rfzarray1[2][0][i][j],rfzarray1[2][1][i][j],rfzarray1[2][2][i][j]]
                z2 = [rfzarray2[2][0][i][j],rfzarray2[2][1][i][j],rfzarray2[2][2][i][j]]

                e1e2[0][i][j] = np.einsum("i,k,k,ij,j",xpisy,xmisy,barmu,Q,rho1)\
                              + np.einsum("i,k,i,kj,j",xpisy,xmisy,mu,barQ,rho2)
                e1e2[1][i][j] = np.einsum("i,k,k,ij,j",xpisy,xmisy,barmu,Q,phi1)\
                              + np.einsum("i,k,i,kj,j",xpisy,xmisy,mu,barQ,phi2)
                e1e2[2][i][j] = np.einsum("i,k,k,ij,j",xpisy,xmisy,barmu,Q,z1)\
                              + np.einsum("i,k,i,kj,j",xpisy,xmisy,mu,barQ,z2)
    else:
        rfzarray1 = rhohatTerm + phihatTerm + zhatTerm
        rfzarray2 = rhohatTerm - phihatTerm - zhatTerm

        # has an additional argument for bar(e)1e2, e1bar(e)2
        e1e2 = np.empty((2,arrayShape[0],arrayShape[1]), dtype = 'complex_')
        for i in range(arrayShape[0]):
            for j in range(arrayShape[1]):
                rfz1 = [rfzarray1[0][i][j],rfzarray1[1][i][j],rfzarray1[2][i][j]]
                rfz2 = [rfzarray2[0][i][j],rfzarray2[1][i][j],rfzarray2[2][i][j]]
                e1e2[0][i][j] = np.einsum("i,k,k,ij,j",xpisy,xmisy,barmu,Q,rfz1)
                e1e2[1][i][j] = np.einsum("i,k,i,kj,j",xpisy,xmisy,mu,barQ,rfz2)

    return e1e2


def e2e2Term(p:int, l:int, w0:float, k:float, sigma:int,\
             rho:"np.array", phi:"np.array", **kwargs) -> float:
    '''
    p (int): Radial Index
    l (int): Topological charge
    w0 : Beam waist at z = 0, in m
    k : Wavenumber
    rho (2D array): rho position, in m
    phi (2D array): phi position, in rad
    e (opt., vect.): polarization operator

    Required keyword arguments:
     Q (matr.): quadrupole transition moment

    Optional keyword arguments:
     rfzseperate (bool): send e1e2 as an array with [rho,phi,z] values

    e2e2 : E2E2 term of the matrix element
    '''
```

```
    # Get values from kwargs. The kwargs exist for compatibility in the
    # visualizeFunct function in FinalReport_Visualize file
    Q = kwargs["Q"]
    rfzseperate = False if kwargs.get("rfzseperate") is None\
        else kwargs.get("rfzseperate")

    def rdf(rho):
        return radialDistributionFunction(p, l, w0, rho)
    # Arrays need to be consistent, so "0" needs to have the same shape
    # as phi and rho.
    assert np.shape(phi) == np.shape(rho)
    arrayShape = np.shape(phi)
    zeroArray = np.zeros(arrayShape)

    # Calculations of values as follows
    derf = derivative(rdf, rho, dx=0.01)
    barQ = np.conjugate(Q)
    xpisy = np.array([1,1j*sigma,0])
    xmisy = np.array([1,-1j*sigma,0])

    rhohatTerm = rdf(rho)*derf *\
             np.array([np.cos(phi), np.sin(phi), zeroArray])
    phihatTerm = rdf(rho)**2/rho * (1j/l) *\
             np.array([-np.sin(phi), np.cos(phi), zeroArray])
    zhatTerm = 1j*k *\
           np.array([zeroArray, zeroArray, np.ones(arrayShape)])

    if rfzseperate == False:
        rfzarray1 = rhohatTerm + phihatTerm + zhatTerm
        rfzarray2 = rhohatTerm - phihatTerm - zhatTerm

        # Compute einsum. "rfzarray" is a rank 1 tensor, but its components (x,y,z)
        # are 2D arrays, which makes einsum think it is rank 3. So there needs to
        # be extraction of "rfzarray" into individual "rfz" terms, calculate each
        # einsum with a "rfz" term seperately, and reassemble them back into an
        # array.
        e2e2 = np.empty(arrayShape, dtype = 'complex_')
        for i in range(arrayShape[0]):
            for j in range(arrayShape[1]):
                rfz1 = [rfzarray1[0][i][j],rfzarray1[1][i][j],rfzarray1[2][i][j]]
                rfz2 = [rfzarray2[0][i][j],rfzarray2[1][i][j],rfzarray2[2][i][j]]
                e2e2[i][j] = np.einsum("i,k,ij,kl,j,l",xpisy,xmisy,Q,barQ,rfz1,rfz2)
    else:
        rfzarray1 = [rhohatTerm, phihatTerm, zhatTerm]
        rfzarray2 = [rhohatTerm, -phihatTerm, -zhatTerm]

        # now has an additional argument for rho, phi, zhat
        e2e2 = np.empty((3,arrayShape[0],arrayShape[1]), dtype = 'complex_')
        for i in range(arrayShape[0]):
            for j in range(arrayShape[1]):
                rho1 = [rfzarray1[0][0][i][j],rfzarray1[0][1][i][j],rfzarray1[0][2][i][j]]
                rho2 = [rfzarray2[0][0][i][j],rfzarray2[0][1][i][j],rfzarray2[0][2][i][j]]
                phi1 = [rfzarray1[1][0][i][j],rfzarray1[1][1][i][j],rfzarray1[1][2][i][j]]
                phi2 = [rfzarray2[1][0][i][j],rfzarray2[1][1][i][j],rfzarray2[1][2][i][j]]
```

```
                z1 = [rfzarray1[2][0][i][j],rfzarray1[2][1][i][j],rfzarray1[2][2][i][j]]
                z2 = [rfzarray2[2][0][i][j],rfzarray2[2][1][i][j],rfzarray2[2][2][i][j]]

                e2e2[0][i][j] = np.einsum("i,k,ij,kl,j,l",xpisy,xmisy,Q,barQ,rho1,rho2)
                e2e2[1][i][j] = np.einsum("i,k,ij,kl,j,l",xpisy,xmisy,Q,barQ,phi1,phi2)
                e2e2[2][i][j] = np.einsum("i,k,ij,kl,j,l",xpisy,xmisy,Q,barQ,z1,z2)

    return e2e2
```

../src/nonparaxial.py

```
# -*- coding: utf-8 -*-
"""
Coder: James Jia
Description: Coding of Non-paraxial equations given by _________
"""

import numpy as np
from scipy.misc import derivative
from rdf import radialDistributionFunction

multiplier = r'$-\frac{1}{2}\Omega^2n$'

def e1e2Term(p:int, l:int, w0:float, k:float, sigma:float,\
             rho:"np.array", phi:"np.array", **kwargs) -> float:
    '''
    p (int): Radial Index
    l (int): Topological charge
    w0 : Beam waist at z = 0, in m
    k : Wavenumber
    sigma : polarization
    rho (2D array): rho position, in m
    phi (2D array): phi position, in rad

    Required keyword arguments:
     mu (vect.): electric dipole transition moment
     Q (matr.): quadrupole transition moment

    e1e2 : E1E2 term of the matrix element
    '''
    # Get values from kwargs. The kwargs exist for compatibility in the
    # visualizeFunct function in FinalReport_Visualize file
    mu = kwargs["mu"]
    Q = kwargs["Q"]

    def rdf(rho):
        return radialDistributionFunction(p, l, w0, rho)
    def rdfOverr(rho):
        return radialDistributionFunction(p, l, w0, rho)/rho
    # Arrays need to be consistent, so "0" needs to have the same shape
    # as phi and rho.
    assert np.shape(phi) == np.shape(rho)
    arrayShape = np.shape(phi)
```

```
    zeroArray = np.zeros(arrayShape)

    # some terms defined below
    derf = derivative(rdf, rho, dx=0.01)
    dderf = derivative(rdf, rho, dx=0.01, n=2)
    barmu = np.conjugate(mu)
    barQ = np.conjugate(Q)
    xhat = np.array([np.ones(arrayShape), zeroArray, zeroArray])
    yhat = np.array([zeroArray, np.ones(arrayShape), zeroArray])
    zhat = np.array([zeroArray, zeroArray, np.ones(arrayShape)])
    rhohat = np.array([np.cos(phi), np.sin(phi), zeroArray])
    phihat = np.array([-np.sin(phi), np.cos(phi), zeroArray])
    xpisyarray = xhat + 1j*sigma*yhat
    xmisyarray = xhat - 1j*sigma*yhat

    # start calculation of the first half of the equation ({a1}{a2})
    a1array = xpisyarray*rdf(rho) + (1j/k)*(derf - (l*sigma/rho)*rdf(rho))*\
         np.e**(1j*sigma*phi)*zhat
    a2p1array = rhohat*derf - (1j*l/rho)*rdf(rho)*phihat - 1j*rdf(rho)*k*zhat
    a2p2array = (-rhohat*(1j/k)*dderf) -\
           (-rhohat*(1j/k)*l*sigma*derivative(rdfOverr,rho,dx=0.01)) -\
           ((l+sigma)/(k*rho))*phihat*(derf-((l*sigma)/rho)*rdf(rho)) -\
           zhat*(derf-((l*sigma)/rho)*rdf(rho))
    a2p3array = np.e**(-1j*sigma*phi)*zhat

    # a2 is a rank-two tensor with (x, y) values which internally makes it a rank
    # 4 tensor, so the shape of the array will be ([no. of x values],[no. of y
    # values], 3, 3).
    a2array = np.empty((arrayShape[0],arrayShape[1],3,3), dtype = 'complex_')
    for i in range(arrayShape[0]):
        for j in range(arrayShape[1]):
            a2p1 = [a2p1array[0][i][j],a2p1array[1][i][j],a2p1array[2][i][j]]
            a2p2 = [a2p2array[0][i][j],a2p2array[1][i][j],a2p2array[2][i][j]]
            a2p3 = [a2p3array[0][i][j],a2p3array[1][i][j],a2p3array[2][i][j]]
            xmisy = [xmisyarray[0][i][j],xmisyarray[1][i][j],xmisyarray[2][i][j]]
            a2array[i][j] = np.einsum('l,k',a2p1,xmisy) + np.einsum('l,k',a2p2,a2p3)

    a = np.empty(arrayShape, dtype = 'complex_')
    for i in range(arrayShape[0]):
        for j in range(arrayShape[1]):
            a1 = [a1array[0][i][j],a1array[1][i][j],a1array[2][i][j]]
            # the (0,1,2) values are at the end since the shape of a2 is also
            # with the (...3, 3) at the end.
            a2 = [a2array[i][j][0],a2array[i][j][1],a2array[i][j][2]]
            a[i][j] = np.einsum('i,lk,i,kl',a1,a2,mu,barQ)

    # start calculation of the second half of the equation ({b1}{b2})
    b1array = xmisyarray*rdf(rho) - (1j/k)*(derf - (l*sigma/rho)*rdf(rho))*\
         np.e**(-1j*sigma*phi)*zhat
    b2p1array = rhohat*derf + (1j*l/rho)*rdf(rho)*phihat + 1j*rdf(rho)*k*zhat
    b2p2array = (rhohat*(1j/k)*dderf) -\
           (rhohat*(1j/k)*l*sigma*derivative(rdfOverr,rho,dx=0.01)) -\
           ((l+sigma)/(k*rho))*phihat*(derf-((l*sigma)/rho)*rdf(rho)) -\
           zhat*(derf-((l*sigma)/rho)*rdf(rho))
```

```
    b2p3array = np.e**(1j*sigma*phi)*zhat

    # similarly with the above.
    b2array = np.empty((arrayShape[0],arrayShape[1],3,3), dtype = 'complex_')
    for i in range(arrayShape[0]):
        for j in range(arrayShape[1]):
            b2p1 = [b2p1array[0][i][j],b2p1array[1][i][j],b2p1array[2][i][j]]
            b2p2 = [b2p2array[0][i][j],b2p2array[1][i][j],b2p2array[2][i][j]]
            b2p3 = [b2p3array[0][i][j],b2p3array[1][i][j],b2p3array[2][i][j]]
            xpisy = [xpisyarray[0][i][j],xpisyarray[1][i][j],xpisyarray[2][i][j]]
            b2array[i][j] = np.einsum('j,i',b2p1,xpisy) + np.einsum('j,i',b2p2,b2p3)

    b = np.empty(arrayShape, dtype = 'complex_')
    for i in range(arrayShape[0]):
        for j in range(arrayShape[1]):
            b1 = [b1array[0][i][j],b1array[1][i][j],b1array[2][i][j]]
            # the (0,1,2) values are at the end since the shape of a2 is also
            # with the (...3, 3) at the end.
            b2 = [b2array[i][j][0],b2array[i][j][1],b2array[i][j][2]]
            b[i][j] = np.einsum('k,ji,k,ij',b1,b2,barmu,Q)

    return a + b

def e2e2Term(p:int, l:int, w0:float, k:float, sigma:float,\
             rho:"np.array", phi:"np.array", **kwargs) -> float:
    '''
    p (int): Radial Index
    l (int): Topological charge
    w0 : Beam waist at z = 0, in m
    k : Wavenumber
    sigma : polarization
    rho (2D array): rho position, in m
    phi (2D array): phi position, in rad

    Required keyword arguments:
     Q (matr.): quadrupole transition moment

    e1e2 : E1E2 term of the matrix element
    '''

    # Get values from kwargs. The kwargs exist for compatibility in the
    # visualizeFunct function in FinalReport_Visualize file
    Q = kwargs["Q"]

    def rdf(rho):
        return radialDistributionFunction(p, l, w0, rho)
    def rdfOverr(rho):
        return radialDistributionFunction(p, l, w0, rho)/rho
    # Arrays need to be consistent, so "0" needs to have the same shape
    # as phi and rho.
    assert np.shape(phi) == np.shape(rho)
    arrayShape = np.shape(phi)
    zeroArray = np.zeros(arrayShape)

```

```
    # some terms defined below
    derf = derivative(rdf, rho, dx=0.01)
    dderf = derivative(rdf, rho, dx=0.01, n=2)
    barQ = np.conjugate(Q)
    xhat = np.array([np.ones(arrayShape), zeroArray, zeroArray])
    yhat = np.array([zeroArray, np.ones(arrayShape), zeroArray])
    zhat = np.array([zeroArray, zeroArray, np.ones(arrayShape)])
    rhohat = np.array([np.cos(phi), np.sin(phi), zeroArray])
    phihat = np.array([-np.sin(phi), np.cos(phi), zeroArray])
    xpisyarray = xhat + 1j*sigma*yhat
    xmisyarray = xhat - 1j*sigma*yhat

    # start calculation...
    b2p1array = rhohat*derf + (1j*l/rho)*rdf(rho)*phihat + 1j*rdf(rho)*k*zhat
    b2p2array = (rhohat*(1j/k)*dderf) -\
           (rhohat*(1j/k)*l*sigma*derivative(rdfOverr,rho,dx=0.01)) -\
           ((l+sigma)/(k*rho))*phihat*(derf-((l*sigma)/rho)*rdf(rho)) -\
           zhat*(derf-((l*sigma)/rho)*rdf(rho))
    b2p3array = np.e**(1j*sigma*phi)*zhat

    b2array = np.empty((arrayShape[0],arrayShape[1],3,3), dtype = 'complex_')
    for i in range(arrayShape[0]):
        for j in range(arrayShape[1]):
            b2p1 = [b2p1array[0][i][j],b2p1array[1][i][j],b2p1array[2][i][j]]
            b2p2 = [b2p2array[0][i][j],b2p2array[1][i][j],b2p2array[2][i][j]]
            b2p3 = [b2p3array[0][i][j],b2p3array[1][i][j],b2p3array[2][i][j]]
            xpisy = [xpisyarray[0][i][j],xpisyarray[1][i][j],xpisyarray[2][i][j]]
            b2array[i][j] = np.einsum('j,i',b2p1,xpisy) + np.einsum('j,i',b2p2,b2p3)

    a2p1array = rhohat*derf - (1j*l/rho)*rdf(rho)*phihat - 1j*rdf(rho)*k*zhat
    a2p2array = (-rhohat*(1j/k)*dderf) -\
           (-rhohat*(1j/k)*l*sigma*derivative(rdfOverr,rho,dx=0.01)) -\
           ((l+sigma)/(k*rho))*phihat*(derf-((l*sigma)/rho)*rdf(rho)) -\
           zhat*(derf-((l*sigma)/rho)*rdf(rho))
    a2p3array = np.e**(-1j*sigma*phi)*zhat

    a2array = np.empty((arrayShape[0],arrayShape[1],3,3), dtype = 'complex_')
    for i in range(arrayShape[0]):
        for j in range(arrayShape[1]):
            a2p1 = [a2p1array[0][i][j],a2p1array[1][i][j],a2p1array[2][i][j]]
            a2p2 = [a2p2array[0][i][j],a2p2array[1][i][j],a2p2array[2][i][j]]
            a2p3 = [a2p3array[0][i][j],a2p3array[1][i][j],a2p3array[2][i][j]]
            xmisy = [xmisyarray[0][i][j],xmisyarray[1][i][j],xmisyarray[2][i][j]]
            a2array[i][j] = np.einsum('l,k',a2p1,xmisy) + np.einsum('l,k',a2p2,a2p3)

    ba = np.empty(arrayShape, dtype = 'complex_')
    for i in range(arrayShape[0]):
        for j in range(arrayShape[1]):
            b2 = [b2array[i][j][0],b2array[i][j][1],b2array[i][j][2]]
            a2 = [a2array[i][j][0],a2array[i][j][1],a2array[i][j][2]]
            ba[i][j] = np.einsum('ji,lk,ij,kl',b2,a2,Q,barQ)
    return ba
```

../src/visualize.py

```python
# -*- coding: utf-8 -*-
"""
Coder: James Jia
Description: Visualization of functions in paraxial.py and nonparaxial.py using
             matplotlib
"""

import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'Times New Roman'
plt.rcParams['mathtext.fontset'] = 'cm'
plt.rcParams['axes.labelsize'] = 10
plt.rcParams['axes.titlesize'] = 10

def visualizeFunct(funct:object, p:int, l:int, w0:float, k:float, sigma:int,
    rmin:float, rmax:float, mu:"np.array"=np.array([1, 0, 0]),
    Q:"np.array"=np.array([[1,0.001,0.001], [0.001,1,0.001], [0.001,0.001,1]]),
    zticklabel:str="", rfzseperate:bool=False, e1e2seperate:bool=False,
    saveSVG:bool=False) -> None:
    '''
    p (int): Radial Index
    l (int): Topological charge
    w0 : Beam waist at z = 0, in m
    k : Wavenumber
    sigma : polarization
    rmin : Minimum x, y position to plot, in m
    rmax : Maximum x, y position to plot, in m
    mu (opt., vect.): electric dipole transition moment
    Q (opt., matr.): quadrupole transition moment
    zticklabel (str.): the label for the colourbar
    rfzseperate (bool): display [rho,phi,z] contributions as seperate graphs
        (paraxial only)
    e1e2seperate (bool): display bar(e)1e2, e1bar(e)2 as seperate graphs
        (paraxial only)
    saveSVG (bool): Save an SVG of the result?
    '''
    interval = (rmax - rmin)/200
    x = np.arange(rmin, rmax+interval, interval)
    y = np.arange(rmin, rmax+interval, interval)
    x, y = np.meshgrid(x, y)
    rho = np.sqrt(x**2 + y**2)
    phi = np.arctan2(y, x)
    res = funct(p, l, w0, k, sigma, rho, phi, mu=mu, Q=Q,\
                rfzseperate=rfzseperate, e1e2seperate=e1e2seperate)

    if res.ndim == 2:
        axplot = plt.contourf(x, y, res,
                              levels=np.linspace(np.min(res),np.max(res),15))
        ax = plt.gca()
        fig = plt.gcf()
        ax.set_aspect('equal', 'box')
        ax.set(xlabel=r'$x\,/\,\mathrm{m}$',ylabel=r'$y\,/\,\mathrm{m}$')
```

```
        ax.ticklabel_format(useMathText=True)
        cbar = plt.colorbar(axplot)
        cbar.set_label(zticklabel)
        cbar.ax.ticklabel_format(useMathText=True)
        fig.set_size_inches(6,4.5)
    elif rfzseperate:
        fig, (ax1, ax2, ax3) = plt.subplots(1, 3)
        try:
            ax1plot = ax1.contourf(x, y, res[0],
                           levels=np.linspace(np.min(res[0]),np.max(res[0]),15))
        except:
            ax1plot = ax1.contourf(x, y, res[0])
        ax1.set_aspect('equal', 'box')
        ax1.set(xlabel=r'$x\,/\,\mathrm{m}$',ylabel=r'$y\,/\,\mathrm{m}$',
                title=r'$\rho$ contribution')
        ax1.ticklabel_format(useMathText=True)
        cbar1 = fig.colorbar(ax1plot, ax=ax1)
        cbar1.set_label(zticklabel)
        cbar1.ax.ticklabel_format(useMathText=True)
        try:
            ax2plot = ax2.contourf(x, y, res[1],
                           levels=np.linspace(np.min(res[1]),np.max(res[1]),15))
        except:
            ax2plot = ax2.contourf(x, y, res[1])
        ax2.set_aspect('equal', 'box')
        ax2.set(xlabel=r'$x\,/\,\mathrm{m}$',ylabel=r'$y\,/\,\mathrm{m}$',
                title=r'$\varphi$ contribution')
        ax2.ticklabel_format(useMathText=True)
        cbar2 = fig.colorbar(ax2plot, ax=ax2)
        cbar2.set_label(zticklabel)
        cbar2.ax.ticklabel_format(useMathText=True)
        try:
            ax3plot = ax3.contourf(x, y, res[2],
                           levels=np.linspace(np.min(res[2]),np.max(res[2]),15))
        except:
            ax3plot = ax3.contourf(x, y, res[2])
        ax3.set_aspect('equal', 'box')
        ax3.set(xlabel=r'$x\,/\,\mathrm{m}$',ylabel=r'$y\,/\,\mathrm{m}$',
                title=r'$z$ contribution')
        ax3.ticklabel_format(useMathText=True)
        cbar3 = fig.colorbar(ax3plot, ax=ax3)
        cbar3.set_label(zticklabel)
        cbar3.ax.ticklabel_format(useMathText=True)
        fig.set_size_inches(22.5,5.5)
    elif e1e2seperate:
        fig, [[ax1, ax2], [ax3, ax4]] = plt.subplots(2, 2)
        ax1plot = ax1.contourf(x, y, np.real(res[0]),
                               levels=np.linspace(np.min(np.real(res[0])),
                                                  np.max(np.real(res[0])),
                                                  15))
        ax1.set_aspect('equal', 'box')
        ax1.set(xlabel=r'$x\,/\,\mathrm{m}$',ylabel=r'$y\,/\,\mathrm{m}$',
                title=r'Re($\bar{\mathrm{E}}1\mathrm{E}2$) contribution')
        ax1.ticklabel_format(useMathText=True)
```

```
        cbar1 = fig.colorbar(ax1plot, ax=ax1)
        cbar1.set_label(zticklabel)
        cbar1.ax.ticklabel_format(useMathText=True)
        ax2plot = ax2.contourf(x, y, np.imag(res[0]),
                               levels=np.linspace(np.min(np.imag(res[0])),
                                                  np.max(np.imag(res[0])),
                                                  15))
        ax2.set_aspect('equal', 'box')
        ax2.set(xlabel=r'$x\,/\,\mathrm{m}$',ylabel=r'$y\,/\,\mathrm{m}$',
                title=r'Im($\bar{\mathrm{E}}1\mathrm{E}2$) contribution')
        ax2.ticklabel_format(useMathText=True)
        cbar2 = fig.colorbar(ax2plot, ax=ax2)
        cbar2.set_label(zticklabel)
        cbar2.ax.ticklabel_format(useMathText=True)
        ax3plot = ax3.contourf(x, y, np.real(res[1]),
                               levels=np.linspace(np.min(np.real(res[1])),
                                                  np.max(np.real(res[1])),
                                                  15))
        ax3.set_aspect('equal', 'box')
        ax3.set(xlabel=r'$x\,/\,\mathrm{m}$',ylabel=r'$y\,/\,\mathrm{m}$',
                title=r'Re($\mathrm{E}1\bar{\mathrm{E}}2$) contribution')
        ax3.ticklabel_format(useMathText=True)
        cbar3 = fig.colorbar(ax3plot, ax=ax3)
        cbar3.set_label(zticklabel)
        cbar3.ax.ticklabel_format(useMathText=True)
        ax4plot = ax4.contourf(x, y, np.imag(res[1]),
                               levels=np.linspace(np.min(np.imag(res[1])),
                                                  np.max(np.imag(res[1])),
                                                  15))
        ax4.set_aspect('equal', 'box')
        ax4.set(xlabel=r'$x\,/\,\mathrm{m}$',ylabel=r'$y\,/\,\mathrm{m}$',
                title=r'Im($\mathrm{E}1\bar{\mathrm{E}}2$) contribution')
        ax4.ticklabel_format(useMathText=True)
        cbar4 = fig.colorbar(ax4plot, ax=ax4)
        cbar4.set_label(zticklabel)
        cbar4.ax.ticklabel_format(useMathText=True)
        fig.set_size_inches(18,15)
    # Image parameters
    fig.set_dpi(200)
    if saveSVG:
        plt.savefig(f"{funct.__module__}.{funct.__name__}_{p}_{l}_{sigma}.svg",
                    bbox_inches="tight")

if __name__ == "__main__":
    import rdf
    import paraxial as parax
    import nonparaxial as nonparax

    p = 0
    l = 1
    sigma = 1
    k = 2*np.pi/729e-9
    w0 = 729e-9
    lim = 2e-6
```

```
162     mu = np.array([1, 0, 0])
163     Q = 1e-3*np.array([[1,1e-3,1e-3], [1e-3,-0.5,1e-3], [1e-3,1e-3,-0.5]])
164 
165     # rdf.visualizeRDF(p, l, w0, -lim, lim, False)
166     visualizeFunct(nonparax.e1e2Term, p, l, w0, k, sigma, -lim, lim, mu, Q,
167                    "", False, False, False)
```