

Report on Numerical Simulation of Thomson Scattering of an Electron from Multiple Beams

Arthur Campello

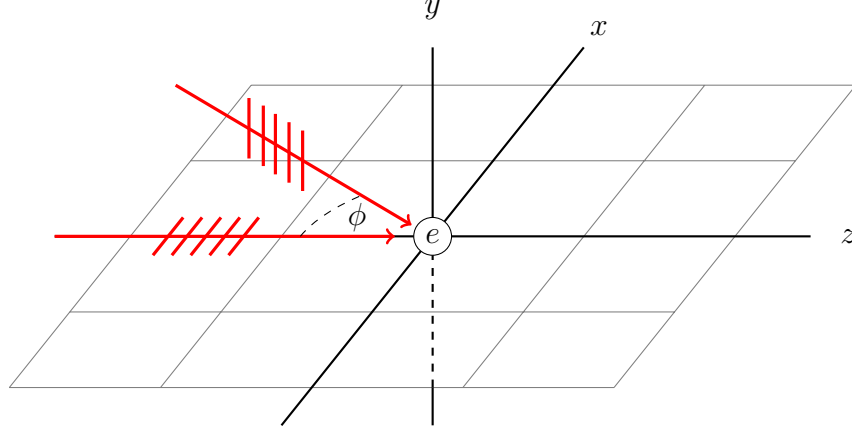
June 14, 2019

1 Introduction

The Tender X-ray Instrument (TXI) is a new LCLS-II instrument with unique capabilities. It will overlap two independent x-ray free electron light (XFEL) sources of perpendicular polarizations in a non-co-linear geometry. The unprecedented nature of this new setup will improve current fundamental models of X-ray matter interaction. Specifically, the coupling of the crossed fields can alter the polarization and intensity of the radiated field in ways not currently taken into consideration. Although one can analytically determine the time-dependent radiated electric field arising from such light scattering with ease, finding this for the case of multiple intersecting beams poses a harder problem. The complexity arises because the term $\mathbf{k} \cdot \mathbf{x}$ no longer remains zero. The non-parallel polarized beam condition, thus entails solving differential equations of the form $\ddot{\mathbf{x}}(t) \propto \cos[at + b\mathbf{x}(t)]$ instead of the far simpler $\ddot{\mathbf{x}}(t) \propto \cos(at)$. For this reason, practically finding the radiated electric field requires a numerical simulation. In addition to the radiated intensity, the polarization and time-like behavior of the field matter greatly to potential experiments at the future beamline. Currently, we project this beamline to cross two beams at an angle of 0.1 radians where one has an energy of 1keV and a polarization perpendicular to both the gravity acceleration and the direction of the beam propagation and the other has an energy of 2.1 and a polarization perpendicular to gravity. This report details a MATLAB-based program aimed at numerically simulating the electric fields produced by this setup and those of similar natures. It includes a description of the assumptions made and the parameters involved, a physics-based justification for the mathematical problem it solves, numerical considerations made to ensure the accuracy of the mathematical solutions, and a detailed structure map of the functions and files involved in producing certain outputs.

2 Assumptions and Parameters

The program relies on an assumption that the electron is at rest at the retarded time, when the beams both reach the particle simultaneously. It additionally assumes that the "first" beam is polarized along the x -axis and travels along the z -axis while the "second" beam is polarized along the y -axis and travels in the xz -plane and an angle ϕ (towards the positive x -direction) from the z -axis. The diagram below illustrates this setup with the incoming beams as red arrows and the polarizations as lines perpendicular to the beams. The parameters E_1 and E_2 the relative electric fields of the first and second beams, respectively. Additionally, ω_1 and ω_2 represent their relative frequencies (proportional to their energies). Finally, η represents the phase shift between the oscillations of the electric fields of the beams. The parameters ϕ , E_1 , E_2 , ω_1 , ω_2 , and η are all provided by the user as inputs. These parameters can vary from experiment to experiment. For the version of this program detailed in this report, all physical constants are set to one, meaning the results appear in relative quantities.



3 Physical Calculations

All outputs from the program stem from E_{rad} , the electric field radiated from the acceleration of the electron. The physical equations referenced in this section to calculate E_{rad} all come from Section 11.2 of Chapter 11 in the textbook Introduction to Electrodynamics, Third Edition, by David J. Griffiths. Given the overlapping and crossing input electric fields described in the previous section and letting the vector \mathbf{x} represent the position of the electron relative to its starting position, one can write

$$m\ddot{\mathbf{x}} = eE_1 \cos(\omega_1 t + \hat{z} \cdot \mathbf{x}) \hat{x} + eE_2 \cos(\omega_2 t + [\sin(\phi), 0, \cos(\phi)] \cdot \mathbf{x} + \eta) \hat{y} \quad (1)$$

with assumptions $\mathbf{x}(t=0) = [0, 0, 0]$ and $\dot{\mathbf{x}}(t=0) = [0, 0, 0]$ and use a finite difference approximation to find $\mathbf{x}(t)$. At this stage, the numerical correction detailed in Section 4 is applied. From the corrected $\mathbf{x}(t)$, one can find $\ddot{\mathbf{x}}(t)$. From here, one can make use of radiation from an accelerating particle initially at rest to find \mathbf{E}_{rad} at some position in space \mathbf{r} and time t as

$$\mathbf{E}_{rad}(\mathbf{r}, t) = \frac{\mu_0 e}{4\pi |\mathbf{r}|} [(\mathbf{r} \cdot \ddot{\mathbf{x}}(t))\mathbf{r} - \ddot{\mathbf{x}}(t)]. \quad (2)$$

While this function encodes all necessary information for the electric field, it becomes more practical if written as $\mathbf{E}_{rad}(\theta, \alpha, t)$, where θ and α represent angles of rotation about the x -axis and y -axis, respectively that transform \hat{z} into $\frac{\mathbf{r}}{|\mathbf{r}|}$. This coordinate transformation is needed since the radiated field is observed in the far field at a fixed distance from the sample and as a function of the scattering angle. Since rotation in three-dimensions is non-commutative, I will write this as a rotation of α followed by one of θ . Given some $\mathbf{r} \in \mathbb{R}^{3 \times 1}$, one can find these angles as

$$R_x(\theta)R_y(\alpha)\frac{\mathbf{r}}{|\mathbf{r}|} = \hat{z} \implies \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \frac{1}{|\mathbf{r}|} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (3)$$

From this, one can easily find the radiated intensity $I_{rad}(\theta, \alpha)$ by finding $|\mathbf{E}_{rad}(\theta, \alpha, t)|$ averaged over one period in time and squaring it; i.e.

$$I_{rad}(\theta, \alpha) = \frac{1}{\mu_0 c} \langle |\mathbf{E}_{rad}(\theta, \alpha, t)|^2 \rangle_t \quad (4)$$

The scalar nature of the radiated intensity along some direction means the coordinate system does not alter it. This is not the case, however, for $\mathbf{E}_{rad}(\theta, \alpha, t)$ itself. Since the light radiated from the electron must travel in the radial direction, \mathbf{E}_{rad}' in coordinates with respect to the radial vector serve to represent the field as "observed" from some angle. More specifically, for some \mathbf{r} , this coordinate

transformation would arise from transforming \mathbf{E}_{rad}' according to the angles θ and α found earlier. This transformation means $\frac{\mathbf{r}}{|\mathbf{r}|} = \hat{z}'$ and, thus, $\mathbf{E}_{\text{rad}}'_z = 0$. One can use θ and α along with \mathbf{E}_{rad} (in the original coordinate system) to find \mathbf{E}_{rad}' as

$$\mathbf{E}_{\text{rad}}' = R_y(-\alpha)R_x(-\theta)\mathbf{E}_{\text{rad}} \implies \begin{bmatrix} E'_{\text{rad}_x} \\ E'_{\text{rad}_y} \\ E'_{\text{rad}_z} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} E_{\text{rad}_x} \\ E_{\text{rad}_y} \\ E_{\text{rad}_z} \end{bmatrix}. \quad (5)$$

the transformation constitutes the exact "reverse" transformation from that in equation X.X as the angles are negative and the order of the rotations is swapped. This concludes the description of how the output quantities are determined. The following section describes the aforementioned numerical correction and the last section outlines how the equations appear in the program and how the program presents the calculated outputs.

4 Numerical Considerations

The only numerically relevant consideration made by the user involves the choice of the finite difference resolution (often denoted as a variable "sampps" in the program, which specifies sampling per second). a low resolution will likely amplify numerical errors and produce an incorrect result, while an extremely high resolution will greatly limit the speed of the program. A recommended value for this falls in the range between 100 and 10,000.

For a successful finite difference approximation, the file "TwoBeamsField.m" begins by estimating the first two time series values and, from this, it estimates the third as a difference of difference between points (mimicking the second derivative for acceleration). Because of this, only the third value of the time series (indexed at 3) and those beyond are modified. For this reason, and since the acceleration is based on a function of the iterate, terms of $(k - 2.5)$, where k is the iterate, are included, for an optimally accurate estimate of the acceleration rate for that iteration. This has been tested to return correct approximations for functions of the form $y''(t) = \cos(at + b)$, for which an analytic solution exists to compare.

5 Program Structure

This section lists the files included in the program as well as their general roles. The files themselves contain comments that reference the equations as numbered in this report. Additionally, I provide a "map" of which files reference other files, where an arrow from one file to another means the second utilizes the first.

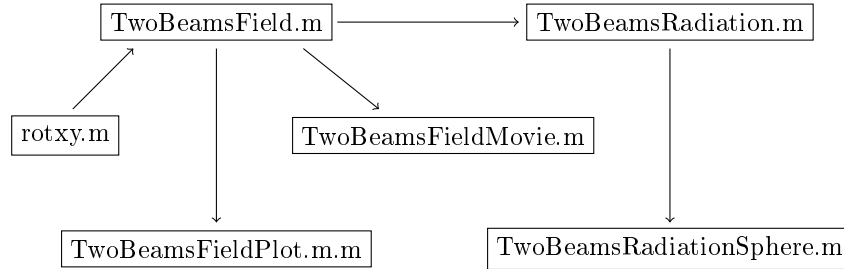
Descriptions of files:

- **rotxy.m** This function-file takes in two angles and a vector representing a spatial Cartesian coordinate in three dimensions, and a parameter for the multiplication order. If the order parameter is 0, It produces the vector arising from first rotating the the vector by the first given angle about the first axis (the x -axis) and then by the second angle about the second axis (the y -axis). Otherwise, the vector is produced by the two operations in the alternative order.
- **TwoBeamsField.m** This function-file takes in parameters ϕ , $\omega = [\omega_1, \omega_2]$, η , $E = [E_1, E_2]$ as defined in section 2, sampps as defined in section 4 and a pair of angles represengint θ and α as defined in section three. From the frequencies, it determines the time for two periods of the $\mathbf{x}(t)$ functions and returns the calculated $\mathbf{E}'_{\text{rad}}(\theta^*, \alpha^*, t)$, where θ^* and α^* are the fixed input angles. It makes use of equations 1, 2, 3, and 5. The function outputs this as a $3 \times n$ matrix

where each column represents the \mathbf{E}'_{rad} elements (in the transformed coordinate system) and the column index is the time.

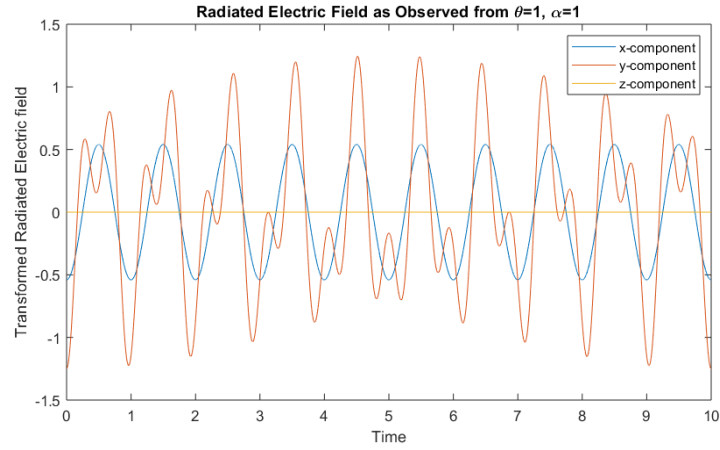
- **TwoBeamsFieldPlot.m** This file is not a function-file, but instead a script that allows a user to hard code the parameters used as inputs for the TwoBeamsField.m plot and plots the three components of the electric field over time for two periods. As explained in section 3, the \hat{z}' component should always be zero.
- **TwoBeamsFieldMovie.m** This script allows a user to hard code all parameters used in TwoBeamsField.m except the angle pair. The user also hard codes two ranges of angles for θ and α and a number of angles to sample (at evenly-spaced angles between the specified intervals). The program creates a meshgrid from this information and produces a movie of the x' and y' components of the outgoing field at each angle pair displayed in a grid (the z' component should, again, be zero). The movie spans two periods in time.
- **TwoBeamsRadiation.m** This function file takes in the same parameters as TwoBeamsField.m does, uses that function to produce an electric field, and makes use of equation 3 to return the the intensity radiated at the input angle pair.
- **TwoBeamsRadiationSphere.m** This script allows a user to hard code all parameters used in TwoBeamsField.m except the angle pair. The user also hard codes two ranges of angles for θ and α and a number of angles to sample (at evenly-spaced angles between the specified intervals). The program creates a meshgrid from this information and produces a heatmap of the relative radiated intensity for the specified angles. It artificially sets a corner value to zero to ensure the inclusion of no radiation in the color bar.

Here I include a map detailing which files listed above are used by which others:

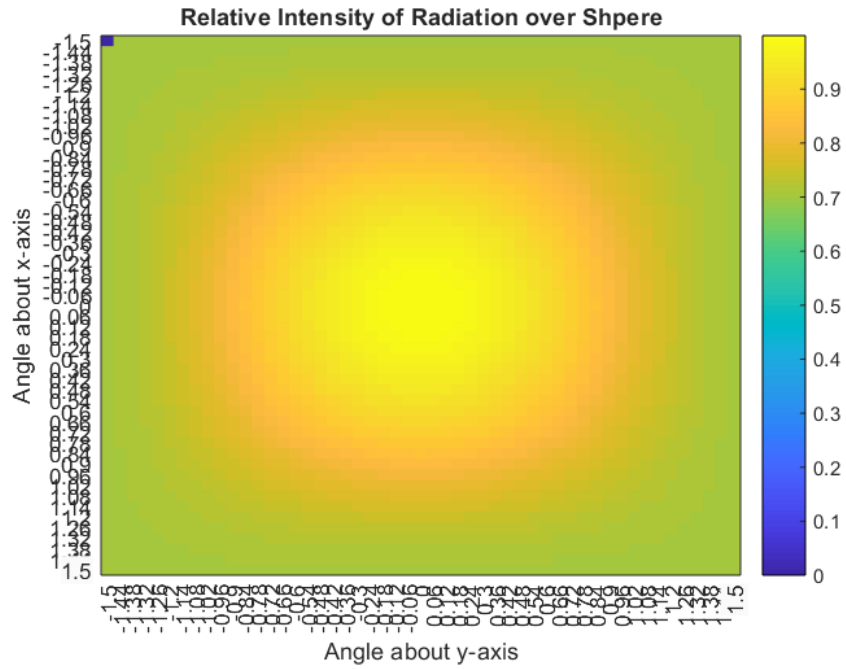


6 Sample Outputs from Estimated Parameters

This section shows the outputs generated from different files for the parameters listed as follows: $\phi = 0.1$, $[\omega_1, \omega_2] = [1, 2.1]$, $\eta = 0$, and $[E_1, E_2] = [1, 1]$. These parameters represent the best estimates of the true parameters at the time of the writing of this report. The first plot shows the output from the file TwoBeamsFieldPlot.m calculated at $[\theta, \alpha] = [1, 1]$. This is shown below



The second plot, included below, shows the output of `TwoBeamsRadiationSphere.m` sampled at θ and α both ranging from -1.5 to 1.5 at 51 intervals.



The final output from the program is that from the file `TwoBeamsFieldMovie.m`. Since this outputs a movie that plays in time, only a snapshot can be included in this report. This is done in the plot below, showing the output at some point in time for the electric field components sampled at θ and α both ranging from -1.5 to 1.5 at 51 intervals.

