Module 5F - Decontamination with Lasers

Omar Betancourt, Payton Goodrich, Emre Mengi

March 2021

Contents

Objectives: Understanding irradiation, simulating the decontamination in industrial workplaces.
Prerequisite Knowledge: N/A
Prerequisite Modules: 1A - Calculus, 1B - Linear Algebra, 2A - Heat Transfer, 2D - Optics, 3A - Geometric Ray Tracing, 3C - Generic Time Stepping
Difficulty: Hard
Summary: This work develops an efficient and rapid computational method to simulate a UV pulse in order to ascertain the decontamination efficacy of UV irradiation for a surface. It is based on decomposition of a pulse into a groups of rays, which are then tracked as they progress towards the target contact surface. The algorithm computes the absorption at the point of contact and color codes it relative to the incoming irradiation. This allows one to quickly quantify the decontamination efficacy across the topology of a structure.

1 Theory

1.1 Introduction

Viral decontamination based on UV technology has become ubiquitous, with many variants now being proposed, in response to the outbreak of COVID-19 in 2020. UV light varies in wavelength from 10 nm to 400 nm, thus making it shorter than visible wavelengths and larger than x-rays. Short wave UV light (UV-c) can damage DNA and sterilize surfaces making it useful in the medical industry. This was first noted in 1878 (Downes and Blunt [6]) when the effect of short-wavelength light killing bacteria was discovered. By 1903 it was known the most effective wavelengths were around 250 nm (UV-c), for which Niels Finsen won a Nobel Prize (for skin-based tuberculosis eradication using UV light). By approximately 1960, the effect that ultraviolet radiation can destroy DNA in living microorganisms was established (see Bolton and Colton [3] for reviews). While many types of decontamination technologies are of interest (see references Anderson et al [2], Battelle [2], Boyce et. al [4], Card et al [5], Heimbuch and Harish [8], Heimbuch et al. [9], Ito and Ito [10], Lin et al. [13], Kanemitsu [12], Lindsley et al [28], Lore et al. [15], Marra et al. [16], Mills et al. [17], Nerandzic et al. [18], Viscusi et al [20] and Tseng and Li [19]), this work will focus on UV-c technologies. The literature asserts that UV-c irradiation dose of above 1 J/cm^2 at 254 nm peak wavelength inactivates SARS-CoV-2 and achieves above a 99% biocidal efficacy on Bacillus subtilis spores.[1] However, the literature also presents evidence that it is difficult to ensure that all surfaces are completely decontaminated, due to shadowing effects. Accordingly, this work focuses on viral decontamination by ultraviolet (UV) irradiation technologies, specifically on developing an efficient and rapid computational method to simulate a UV pulse, in order to ascertain the decontamination efficacy of UV irradiation for a surface.

Remarks: Purely UV-c protocols should be adopted if there is no other choice, but can be a key component of a multistage process involving a combination of decontamination processes such as (1) hydrogen peroxide vapor and gas plasma, (2) heat and humidity and (3) UV-c irradiation.

[1] While 1 J/cm2 is effective for decontamination, it is harmful to humans.

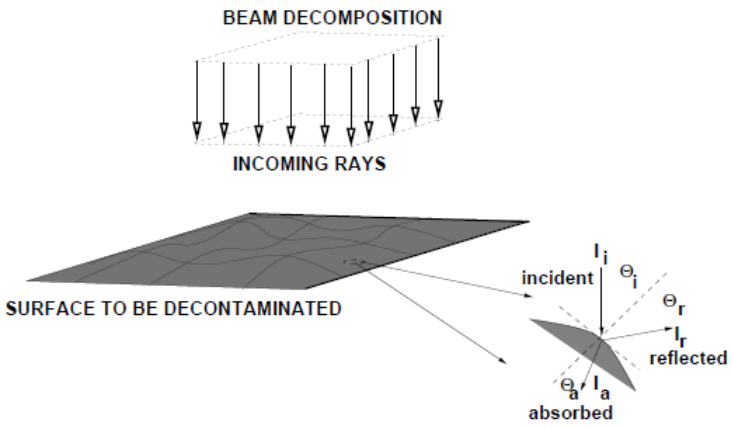

Figure 1.1: An electromagnetic pulse applied to a surface.

1.1.1 Objectives

This work develops an efficient and rapid computational method to simulate a UV pulse in order to ascertain the decontamination efficacy of UV irradiation for a surface. It is based on decomposition of a pulse into a groups of rays, which are then tracked as they progress towards the target contact surface. The algorithm computes the absorption at the point of contact and color codes it relative to the incoming irradiation. This allows one to rapidly quantify the decontamination uncertainty by identifying regions where the absorption is inadequate and serves as a guide for practitioners to ascertain where problems may occur a priori to experiments. Additionally, the reflections are calculated, and can be used in ascertaining safety to a bystander. The interest here is on the absorption of an initially coherent pulse (Figure 1.1), represented by multiple collimated (parallel) rays (initially forming a planar wave front), where each ray is a vector in the direction of the flow of energy (the rays are parallel to the initial wave's propagation vector). We make the following observations:

- It is assumed that the features of the surface to be irradiated are at least an order of magnitude larger than the wavelength of the incident radiation (essentially specular surfaces), therefore "geometrical" ray tracing theory is applicable, and is well-suited for the systems of interest. It is important to emphasize the regimes of validity of such a model are where the surface features are larger than the UV wavelengths. For example, if we were to use UV-rays ($10^{-8}m \leq \lambda \leq 4 \times 10^{7}m$), the features in this analysis would be assumed to possess scales larger than approximately $4 \times 10^{-6}m$. For systems containing features smaller than this, one can simply use the model as a qualitative guide.
- Ray-tracing is a method that is employed to produce rapid approximate solutions to waveequations for high-frequency/small-wavelength applications where the primary interest is in the overall propagation of energy.[2]
- Ray-tracing methods proceed by initially representing wave fronts by an array of discrete rays. *Thereafter, the problem becomes one of a primarily geometric character*, where one tracks the changing trajectories and magnitudes of individual rays which are dictated by the reflectivity and the Fresnel conditions (if a ray encounters a material interface).
- Ray-tracing methods are well-suited for computation of scattering in complex systems that are difficult to mesh/discretize, relative to procedures such as the Finite Difference Time Domain Method or the Finite Element Method.
- Other high frequency irradiation regimes can also be considered in the same manner, such as X-rays and gamma rays, provided that the scattering target has the appropriate (larger) lengthscale. Even in

[2]Resolving diffraction (which ray theory is incapable of describing) is unimportant for the applications of interest.

the case where this clear separation of length scales is not present, this model still provides valuable information on the propagation of the beam and the reflected response of the dispersed system.

1.2 Electromagnetic energy propagation

The 'Optics' module supplies the theory underpinning electromagnetic wave propagation and rays, therefore, refer to that module for details. The summary of the derived relations are given in the following section.

1.2.1 Reflectivity

To observe the dependency of $\mathbb{R}$ on $\hat{n}$ and θ_i we can explicitly write

$$\mathbb{R} = \frac{1}{2}\left(\left(\frac{\frac{\hat{n}^2}{\hat{\mu}}\cos\theta_i - \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}{\frac{\hat{n}^2}{\hat{\mu}}\cos\theta_i + \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}\right)^2 + \left(\frac{\cos\theta_i - \frac{1}{\hat{\mu}}\left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}{\cos\theta_i + \frac{1}{\hat{\mu}}\left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}\right)^2\right). \tag{1.1}$$

We observe:

- As $\hat{n} \to \infty, \mathbb{R} \to 1$, no matter what the angle of incidence's value. We note that as $\hat{n} \to 1$, provided that $\hat{\mu} = 1, \mathbb{R} \to 0$, i.e. all incident energy is absorbed (it is transparent).
- With increasing $\hat{n}$, the angle for minimum reflectance grows larger. As mentioned previously, for the remainder of the work, we shall take $\hat{\mu} = 1\,(\mu_o = \mu_i = \mu_a)$, thus

$$\hat{n} = \frac{n_a}{n_i} = \sqrt{\frac{\epsilon_a\mu_a}{\epsilon_i\mu_i}} \Rightarrow \epsilon_a\mu_a = (\hat{n})^2\epsilon_i\mu_i \Rightarrow \epsilon_a = (\hat{n})^2\epsilon_i. \tag{1.2}$$

- The previous assumption yields

$$\mathbb{R} = \frac{I_r}{I_i} = \frac{1}{2}\left(\left(\frac{\hat{\hat{n}}^2\cos\theta_i - \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}{\hat{n}^2\cos\theta_i + \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}\right)^2 + \left(\frac{\cos\theta_i - \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}{\cos\theta_i + \left(\hat{n}^2 - \sin^2\theta_i\right)^{\frac{1}{2}}}\right)^2\right) \tag{1.3}$$

- Since $\bar{I}$ is the energy per unit area per unit time, we obtain the energy associated with an entire pulse/beam by multiplying the irradiance by the cross-sectional area of an initially coherent beam, $\bar{I}A^b$, where A^b is the cross-sectional area of the beam (comprising all of the rays).
- The energy per unit time (power) for a ray in the pulse/beam is then given by $I = \bar{I}A^r = \bar{I}A^b/N_r$, where N_r is the number of rays in the beam (Figure **??**) and A^r can be considered the area associated with a ray.
- The reflection relation, Equation 1.1 can then be used to compute changes in the magnitude of the reflected rays (and the amount absorbed), with directional changes given by the laws of reflection.

We refer the reader to Gross [1] and Zohdi [3-9] for details.

2 Example

From this point forth, we assume that the ambient medium behaves as a vacuum. Accordingly, there are no energetic losses as the rays move through the surrounding medium.

2.1 Tracking of beam-decomposed rays

Starting at $t = 0$ and ending at $t = T$, the simple overall algorithm to track rays is as follows, at each time increment:

1. Check for intersections of rays with surfaces (hence a reflection), and compute the ray magnitudes and orientation if there are reflections (for all rays that are experiencing a reflection, $I_j^{ref}, j = 1, 2, \ldots Rays$)
2. Increment all ray positions ($\boldsymbol{r}_j(t + \Delta t) = \boldsymbol{r}_j(t) + \Delta t \boldsymbol{v}_j(t), j = 1, 2, \ldots,$ Rays)
3. Increment time forward ($t = t + \Delta t$) and repeat the process for the next time interval.

In order to capture all of the ray reflections that occur:

- The time step size Δt is dictated by the offset height of the source. A somewhat ad-hoc approach is to scale the time step size by the speed of ray propagation according to $\Delta t = \xi \frac{\mathcal{H}}{\|\boldsymbol{v}\|}$, where $\mathcal{H}$ is the height of the source and $0.0001 \leq \xi \leq 0.01$. Typically, the results are insensitive to ξ that are smaller than this range.
- Although outside the scope of this work, one can also use this algorithm to compute the thermal response by combining it with heat transfer equations via staggering schemes (Zohdi [3,6]).

2.2 Test surface

The discrete-ray approach is flexible enough to simulate a wide variety of systems. As a test surface, we consider a topology to be irradiated described by $F(x_1, x_2, x_3) = 0$. The outward surface normals, $\boldsymbol{n}$, needed during the scattering calculations, are easy to characterize by writing

$$\boldsymbol{n} = \frac{\nabla F}{\|\nabla F\|} \tag{2.1}$$

The components of the gradient are

$$\nabla F = \frac{\partial F}{\partial x_1}\boldsymbol{e}_1 + \frac{\partial F}{\partial x_2}\boldsymbol{e}_2 + \frac{\partial F}{\partial x_3}\boldsymbol{e}_3 \tag{2.2}$$

It is advantageous to write the surface in parametric form:

$$F(x_1, x_2, x_3) = G(x_1, x_2) - x_3 = 0 \tag{2.3}$$

which leads to

$$x_3 = G(x_1, x_2) \tag{2.4}$$

The gradient becomes

$$\nabla F = \frac{\partial G}{\partial x_1}\boldsymbol{e}_1 + \frac{\partial G}{\partial x_2}\boldsymbol{e}_2 - \boldsymbol{e}_3 \tag{2.5}$$

In order to determine whether a ray has made contact with a surface domain, one checks if the x_3 component of a ray ($\boldsymbol{r}_j$) is less that x_3 of the surface.

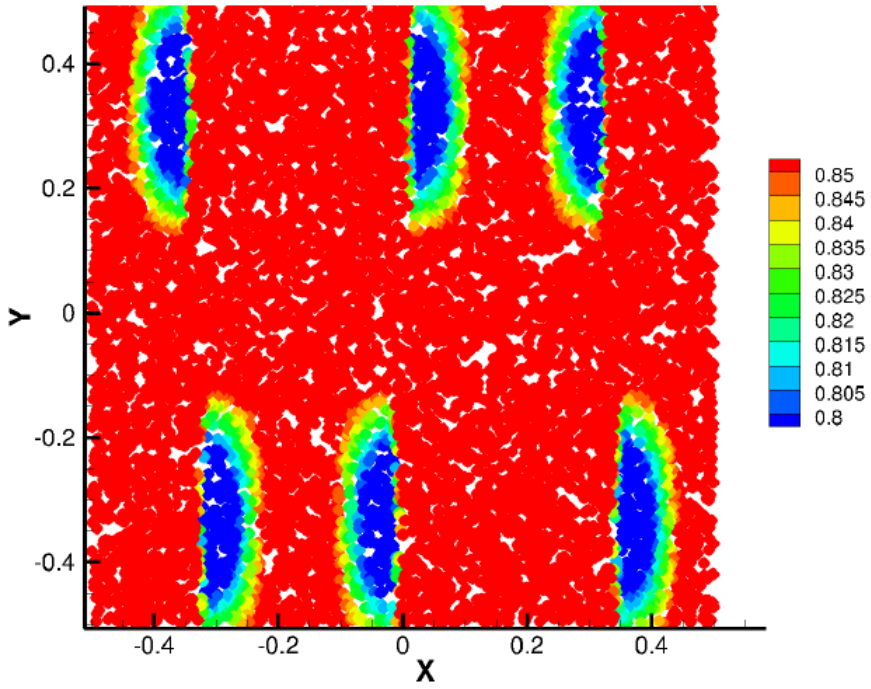

Figure 2.1: Top view for a surface with amplitude A = 0.3. Colors indicate the absorption, normalized by the incoming radiation level.

Table 2.1: The loss of absorption efficacy with contact surface amplitude oscillation (waviness).

Surface Amplitude	Average Surface Absorption
0.0	0.8888
0.1	0.8879
0.2	0.8808
0.3	0.8675
0.4	0.8507
0.5	0.8327
0.6	0.8126
0.7	0.7878
0.8	0.7679
0.9	0.7485
1.0	0.7315

2.3 Numerical/Quantitative examples

We have the following set up for a series of tests:

- The initial velocity vector for all initially collimated (parallel) rays comprising the beam was $v = (c, 0, 0)$, where $c = 3 \times 10^8 m/s$ is the speed of light in a vacuum.
- We used a parametrized test surface given by

$$x_3 = 2 + A\left(\sin\frac{2\omega_1\pi x_1}{L_1}\right)\sin\left(\frac{2\omega_2\pi x_2}{L_2}\right) \tag{2.6}$$

with $L_1 = L_2 = 1, \omega_1 = 1.5$ and $\omega_2 = 0.75$, where we vary A. We also added a flat cut-off so that the surface had a half-sine wave character (Figure 2.2).

- The number of rays in the beam were steadily increased from $N_r = 100, 200$, etc, until the results were insensitive to further refinements. This approach indicated that between approximately $9500 \leq N_r \leq 10000$ parallel rays in rectangular cross-sectional plane of the beam. The rays were randomly placed within the beam (Figures 1.1 and 2.2), to correspond to unpolarized incoming energy, and yielded stable results across the parameter study range.

- Figure 2.2 shows a sequence of frames of the detailed response of a surface to 10000 rays. Figure 2.1 shows a top view. Table 2.1 shows the steady loss of absorption efficacy with contact surface amplitude oscillation (waviness). The algorithm computes the absorption at the point of contact and color codes it relative to the incoming irradiation. This allows one to quickly quantify the decontamination across the topology of the structure.

- This approach also allows an analyst to explore nonuniform beam profiles, for example exponential central irradiance decay: $I(d) = I(d = 0)e^{-ad}$, where d is the distance from the center of the initial beam, where in the case of $a = 0$, one recaptures a flat beam, $I(d) = I(d = 0)$.[1]

2.4 Summary and discussion

In closing, the pandemic of 2020 has led to a gigantic increase research in modeling and simulation of infectious diseases. There are numerous aspects associated with this literally epoch-changing event that is now facing humanity, such as (a) Disease propagation, (b) Immune-responses, (c) Logistical responses, (d) Political responses and (e) Decontamination protocols. Because UV-c based decontamination methods are becoming widely used in industry, with many variants being proposed, fast computational analysis and design tools are needed to ascertain their effectiveness. Accordingly, this work developed a discrete-ray model to allow for propagation of energy encountering a surface, based on the decomposition of irradiation into a groups of rays, which are then tracked as they progress towards the target. This facilitates:

- Quick quantification of the decontamination efficacy across the topology of the structure (color coding the efficacy relative to the incoming irradiation).

- Parametric studies to the changes in absorption as a function of changes in surface geometry.

The simulations take on the order of one minute on a laptop. This type of approach makes it quite suitable for use in conjunction with mobile decontamination systems and allows provides a simpler alternative to a direct, computationally intensive, discretization of a continuum description using Maxwell's equations with a Finite Element or Finite Difference method.

[1] Note that algorithmically, we can the set total initial irradiance via $\sum_{i=1}^{N_r} I_i^{inc}(t = 0)\mathcal{A}_r = P$ Watts. To achieve this distribution, one would first place rays randomly in the plane, and then scale the individual I^{inc} by e^{-ad} and the normalized the average so that the total was P watts.

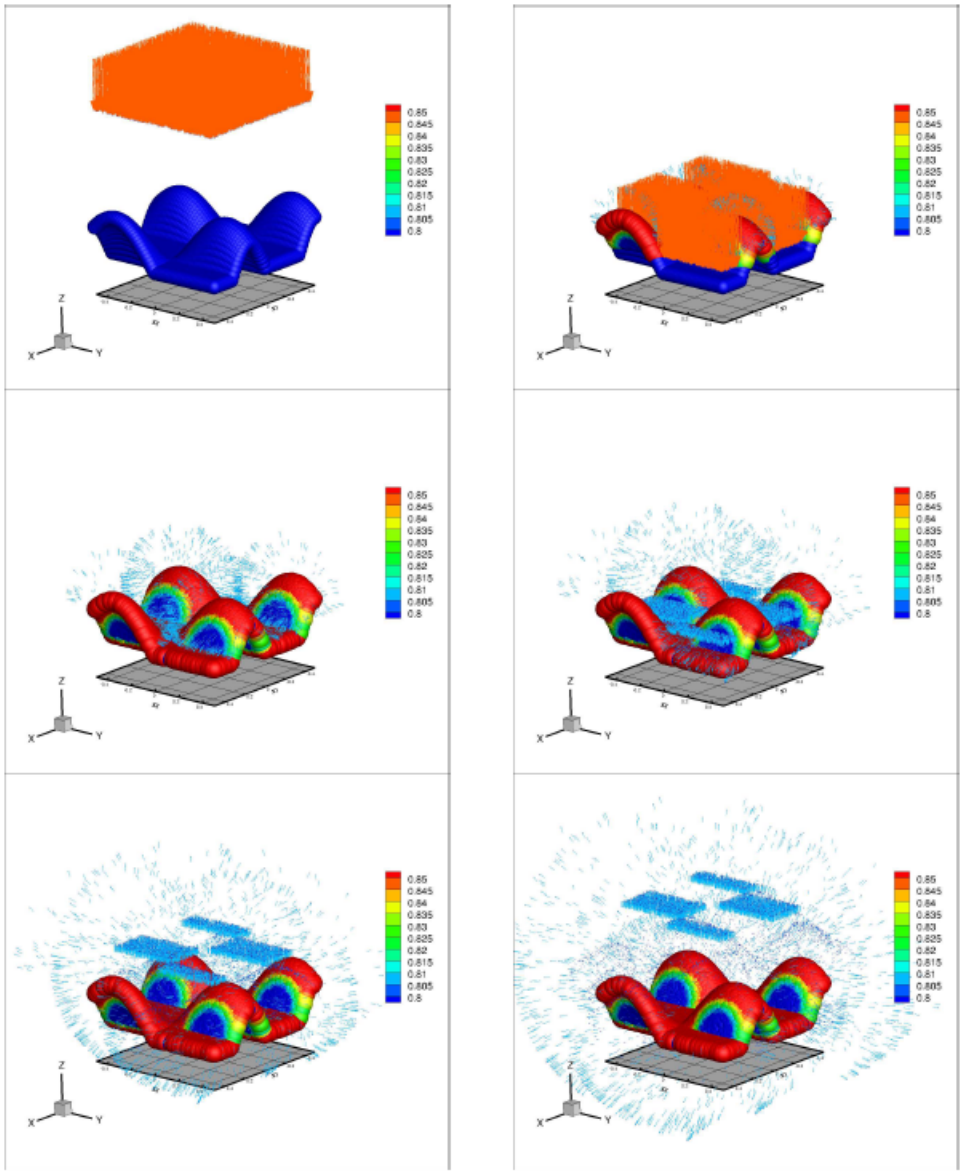

Figure 2.2: Sequence of frames for a surface with amplitude A = 0.3 (the amplitude was enhanced by a factor of 20 in graphics to more easily see the effects of the topology on absorption). Colors indicate the absorption, normalized by the incoming radiation level.

3 Assignment

BASICS OF UV-C DECONTAMINATION OF FOOD SURFACES (100 POINTS)

In this assignment, you will be introduced to the basics of light-based decontamination systems. *The typical report format will not be required for this assignment, and unless otherwise specified, either hand-written or typed information is acceptable.* If you use Matlab or another language for simplifying answers or evaluating specific numbers, clearly set up the equations before showing your numerical results. Correct answers without supporting work will be penalized, and incorrect answers without supporting work will receive zero credit.

For the given test surfaces below, calculate:

- (10 points) Discretize the test surface and represent the surface as a scatter plot. *Hint: Use your engineering judgement to determine the adequate mesh size to accurately capture the physics of the system.*
- (10 points) Calculate surface normals.
- (50 points) Use Forward Euler time-stepping scheme to simulate a beam hitting the given test surfaces, write the plots. created in every 100th time step to a video frame and save the video. *Hint: To speed up the collision detection, you can assign one beam per surface point.*
- (30 points) Calculate the absorption values of each surface point and represent it using a color map.

The test surfaces are described in mathematical equations:

(a)

$$x_3 = 2 + (0.3)\left(\sin\frac{2(1.5)\pi x_1}{1}\right)\sin\left(\frac{2(0.75)\pi x_2}{1}\right) \tag{3.1}$$

(b)

$$x_3 = \sqrt{\left(5 - \sqrt{x_1^2 + x_2^2}\right)^2} - 4 - 1.75 \tag{3.2}$$

A variable glossary is provided at the end of the assignment. *Hint: The equations required to model the system is provided in the theory section.*

Table 3.1: Model Parameters.

Symbol	Type	Units	Value	Description
μ_o	Scalar	Wb/Am	$4\pi \times 10^{-7}$	free-space magnetic permeability
ϵ_o	Scalar	C/Nm	8.8542×10^{-7}	free-space electric permittivity
c	Scalar	m/sr	$\frac{1}{\sqrt{\mu_o \epsilon_o}}$	speed of light
$\hat{\mu}$	Scalar	none	$\frac{\mu_a}{\mu_i} = 1$	magnetic permeability ratio (incident/absorbing)
$\hat{\epsilon}$	Scalar	none	$\frac{\epsilon_a}{\epsilon_i} = 25$	electric permittivity ratio (incident/absorbing)
$\hat{n}$	Scalar	none	$\frac{n_a}{n_i} = 1.5$	refractive index ratio (incident/absorbing)
v_{beam}	Vector	m/s	$[0 \quad 0 \quad -c]$	beam initial velocity
h_{beam}	Scalar	m	3	beam initial height
x_1	Scalar	m	$-0.5 \leq x_1 \leq 0.5$	surface length limits
x_2	Scalar	m	$-0.5 \leq x_1 \leq 0.5$	surface width limits
t_f	Scalar	s	7.5×10^{-9}	final time for simulation
Δt	Scalar	s	8.66×10^{-13}	time step size

4 Solution

The assignment solution is encoded in Matlab below.

```
1
2  clc; clear all; close all; % housekeeping
3
4  % User input for recording
5
6  prompt = 'Do you want to save the video? [Y/N]';
7  str = input(prompt, 's');
8  if strcmp('Y',str)
9      v = VideoWriter('newradicircular.avi');
10     open(v);
11 elseif strcmp('N',str)
12
13 else
14     fprintf('WARNING: Enter Y/N.')
15     return
16 end
17
18
19 % Constants
20 mu_0 = 4*pi*10^-7; %free-space magn. permeability [Wb/(Am)]
21 eps_0 = 8.8542*10^-12; %free-space elec. permittivity [C/(Nm)]
22 c = 1/sqrt(mu_0*eps_0); %speed of light [m/s]
23 %mu_hat: ratio of magn. permeability scattering/absorbing (mu_a)...
24 mu_hat = 1;
25 eps_hat = 25; % electric permitivity (chosen)
26 %n_hat = sqrt(mu_hat * eps_hat);
27 n_hat = 1.5;
28
29 % Set up the surface
30 L1 = 1;
31 L2 = 1;
32 w1 = 1.5;
33 w2 = 0.75;
34 A = 0.3;
35 test_lim = 0.5; %limits of test surface
36 x1 = linspace(-test_lim,test_lim,100);
37 x2 = linspace(-test_lim,test_lim,100);
38 [X1, X2] = meshgrid(x1);
39 %X3 = 2 + A.*sin(2.*w1.*pi.*X1./L1).*sin(2.*w2.*pi.*X2./L2); % surface a
40 X3 = sqrt((5 - sqrt(X1.^2 + X2.^2)).^2 - 4)-1.75; % surface b
41 % for i=1:numel(X3) %uncomment for surface a
42 %     if (X3(i)-2)<0
43 %         X3(i)=2;
44 %     end
45 % end
46 scatter3(X1(:),X2(:),X3(:),[],[0.5 0.5 0.5],'filled')
47
48 % Calculate normals
49 %
50 % %Manual Method
51 % %delF
```

```
% delGx1 = A.*(2.*w1.*pi./L1).*cos(2.*w1.*pi.*X1./L1).*sin(2.*w2.*pi.*X2./L2);
% delGx2 = A.*(2.*w2.*pi./L2).*sin(2.*w1.*pi.*X1./L1).*cos(2.*w2.*pi.*X2./L2);
%
% delFx1 = delGx1;
% delFx2 = delGx2;
% delFx3 = ones(size(delGx1)).*-1;
%
% magdelF = sqrt(delFx1.^2 + delFx2.^2 + delFx3.^2);
%
% norm1 = -delFx1 ./ magdelF;
% norm2 = -delFx2 ./ magdelF;
% norm3 = -delFx3 ./ magdelF;

% Create rays
beam.initvel = [0,0,-c]; %initial velocity of beam
%beam.initvel = [0,c/sqrt(2),-c/sqrt(2)]; %initial velocity of beam
beam.N = 1000; %number of rays
beam.initH = 3; %initial height
beam.N = round(sqrt(beam.N))^2; % updated number of rays to have it square
fprintf('Updated number of rays is: %1.0d\n',beam.N)

%Different beam types
%full beam
[beam.posx1,  beam.posx2] = meshgrid(linspace(-0.5,0.5,round(sqrt(beam.N))));
%center beam (square)
%[beam.posx1,  beam.posx2] = meshgrid(linspace(-0.25,0.25,round(sqrt(beam.N)))
    );
%center beam (circular)
% Theta = linspace(-pi,pi,round(sqrt(beam.N)));
% R = linspace(0,0.5,round(sqrt(beam.N)));
% [ThetaG,RG] = meshgrid(Theta,R);
% beam.posx1 = RG.*cos(ThetaG);
% beam.posx2 = RG.*sin(ThetaG);

%offset beam
%beam.posx2 = beam.posx2 - 1;

%for absorption plotting
X1_final = beam.posx1;
X2_final = beam.posx2;
%X3_final = 2 + A.*sin(2.*w1.*pi.*X1_final./L1).*sin(2.*w2.*pi.*X2_final./L2);
    %surface a
X3_final = sqrt((5 - sqrt(X1_final.^2 + X2_final.^2)).^2 - 4) - 1.75; %
    surface b
% for i=1:numel(X3_final)%uncomment for surface a
%     if (X3_final(i)-2)<0
%         X3_final(i)=2;
%     end
% end
%calculate normals
%Alternative - MATLAB Function
[nx1, nx2, nx3] = surfnorm(X1_final,X2_final,X3_final);
norm1 = nx1;
norm2 = nx2;
```

```
103 norm3 = nx3;
104
105 beam.posx3 = ones(sqrt(beam.N)).*beam.initH;
106 beam.vel1 = ones(sqrt(beam.N)).*beam.initvel(1);
107 beam.vel2 = ones(sqrt(beam.N)).*beam.initvel(2);
108 beam.vel3 = ones(sqrt(beam.N)).*beam.initvel(3);
109
110 % Time-stepping
111 totaltime = 0.0000000075; % simulation time
112 %xi = 0.0001; % time-stepping constant
113 xi = 0.000000005; % time-stepping constant
114 dt = xi * beam.initH / sqrt(c);
115 Ntimestep = round(totaltime / dt);
116
117
118 figure(1)
119
120 h = gca;
121 view(135, 30);
122 axis equal;
123 grid on;
124
125 xlim([-1 1]);
126 ylim([-1 1]);
127 zlim([2 5]);
128 xlabel('X[m]');
129 ylabel('Y[m]');
130 zlabel('Z[m]');
131 %hold(gca, 'on');
132 hold on
133 IR = zeros(1,numel(beam.posx1));
134 theta_i = zeros(1,numel(beam.posx1));
135
136 X1_final = NaN(numel(beam.posx1),1);
137 X2_final = NaN(numel(beam.posx2),1);
138
139 tic;
140 for i = 1:Ntimestep
141     beam.posx1 = beam.posx1 + beam.vel1 * dt;
142     beam.posx2 = beam.posx2 + beam.vel2 * dt;
143     beam.posx3 = beam.posx3 + beam.vel3 * dt;
144
145
146
147 % check reflection
148     %I_surf = NaN(1,numel(beam.posx2));
149     %I_ray = NaN(1,numel(beam.posx2));
150
151     %if min(beam.posx3,[],'all') < min(X3,[],'all')
152         for k=1:numel(beam.posx1)
153             %check if below surface
154             %X3_calc = 2 + A.*sin(2.*w1.*pi.*beam.posx1(k)./L1).*sin(2.*w2.*pi
                .*beam.posx2(k)./L2); %surface a
```

```
155             X3_calc = sqrt((5 - sqrt(beam.posx1(k).^2 + beam.posx2(k).^2)).^2
                    - 4) - 1.75; % surface b
156 %             for z=1:numel(X3_calc) %uncomment for surface a
157 %                 if (X3_calc-2)<0
158 %                     X3_calc = 2;
159 %                 end
160 %             end
161             if beam.posx1(k) < test_lim && beam.posx1(k) > -test_lim &&...
162                     beam.posx2(k) < test_lim && beam.posx2(k) > -test_lim
163                 if X3_calc>beam.posx3(k)
164                     if IR(k) == 0
165                         %find normal and ray velocity vector
166                         norm_k = [norm1(k) norm2(k) norm3(k)];
167                         beam_vel = [beam.vel1(k) beam.vel2(k) beam.vel3(k)];
168                         theta_i = acos(dot(norm_k,beam_vel)/(norm(beam_vel)))-
                                pi;
169 %                       if theta_i > pi/2
170 %                           theta_i = theta_i - pi/2;
171 %                       end
172                         if beam.vel3(k)>0
173 
174                         else
175                             %record point of contact with surface
176                             X1_final(k) = beam.posx1(k);
177                             X2_final(k) = beam.posx2(k);
178 
179                             beam_vel = beam_vel - 2.*dot(norm_k,beam_vel) .*
                                    norm_k;
180                             beam.vel1(k) = beam_vel(1);
181                             beam.vel2(k) = beam_vel(2);
182                             beam.vel3(k) = beam_vel(3);
183                             IR(k) = (1/2)*(((n_hat^2*cos(theta_i)-sqrt(n_hat^2-
                                    sin(theta_i)^2))/...
184                             (n_hat^2*cos(theta_i)+sqrt(n_hat^2-sin(theta_i)^2))
                                    )^2+...
185                             ((cos(theta_i)-sqrt(n_hat^2-sin(theta_i)^2))/...
186                             (cos(theta_i)+sqrt(n_hat^2-sin(theta_i)^2)))^2);
187                         end
188                     end
189                 end
190             end
191         end
192     %end
193 
194     if mod(i,100) == 0 % update movie frame
195         figure(1)
196         %surf(X1,X2,X3)
197         scatter3(X1(:),X2(:),X3(:),100,[0.5 0.5 0.5],'filled')
198 
199         hold on
200         h = gca;
201         view(135, 30);
202         set(gcf,'pos', [0 200 1000 1000]);
203         axis equal;
```

```
204             grid on;
205
206             xlim([-1 1]);
207             ylim([-1 1]);
208             zlim([2 5]);
209             xlabel('X[m]');
210             ylabel('Y[m]');
211             zlabel('Z[m]');
212             %hold(gca, 'on');
213             hold on
214             quiver3(beam.posx1,beam.posx2,beam.posx3,beam.vel1,beam.vel2,beam.vel3
                    );
215
216             if strcmp('Y',str)
217                 frame = getframe(gcf);
218                 writeVideo(v,frame);
219             end
220         end
221         hold off
222     end
223
224     %recalculate surface points for absorbed surfaces
225     %X3_final = 2 + A.*sin(2.*w1.*pi.*X1_final./L1).*sin(2.*w2.*pi.*X2_final./L2);
            %surface a
226     X3_final = sqrt((5 - sqrt(X1_final.^2 + X2_final.^2)).^2 - 4) - 1.75; %
            surface b
227     % for i=1:numel(X3_final) %uncomment for surface a
228     %     if (X3_final(i)-2)<0
229     %         X3_final(i)=2;
230     %     end
231     % end
232
233
234     % Re-plot surface with absorption color-coding
235     %
236
237     absorb = 1 - IR;
238
239
240     rng = max(absorb)-min(absorb);
241     %rng = 1-0;
242     interv = rng/4;
243
244     figure(1)
245
246     h = gca;
247     view(135, 30);
248     axis equal;
249     grid on;
250
251     xlim([-1 1]);
252     ylim([-1 1]);
253     zlim([2 5]);
254     xlabel('X[m]');
```

```
ylabel('Y[m]');
zlabel('Z[m]');
%hold(gca, 'on');
hold on

X1_1 = NaN(1,numel(X1_final));
X2_1 = NaN(1,numel(X1_final));
X3_1 = NaN(1,numel(X1_final));
X1_2 = NaN(1,numel(X1_final));
X2_2 = NaN(1,numel(X1_final));
X3_2 = NaN(1,numel(X1_final));
X1_3 = NaN(1,numel(X1_final));
X2_3 = NaN(1,numel(X1_final));
X3_3 = NaN(1,numel(X1_final));
X1_4 = NaN(1,numel(X1_final));
X2_4 = NaN(1,numel(X1_final));
X3_4 = NaN(1,numel(X1_final));
ind = 0;

for t=1:numel(beam.posx1)
    ind = ind + 1;
    if absorb(t) <= min(absorb)+interv
        X1_1(ind) = X1_final(t);X2_1(ind) = X2_final(t);X3_1(ind) = X3_final(t
            );
    elseif absorb(t) <= min(absorb)+2*interv
        X1_2(ind) = X1_final(t);X2_2(ind) = X2_final(t);X3_2(ind) = X3_final(t
            );
    elseif absorb(t) <= min(absorb)+3*interv
        X1_3(ind) = X1_final(t);X2_3(ind) = X2_final(t);X3_3(ind) = X3_final(t
            );
    elseif absorb(t) <= min(absorb)+4*interv
        X1_4(ind) = X1_final(t);X2_4(ind) = X2_final(t);X3_4(ind) = X3_final(t
            );
    end
end

h = gca;
view(135, 30);
set(gcf,'pos', [0 200 1000 1000]);
axis equal;
grid on;

xlim([-1 1]);
ylim([-1 1]);
zlim([2 5]);
xlabel('X[m]');
ylabel('Y[m]');
zlabel('Z[m]');
%hold(gca, 'on');
addition = min(min(X3))/100; %plot radiated portion above original surface
scatter3(X1_1,X2_1,X3_1+addition,100,'b','filled')
scatter3(X1_2,X2_2,X3_2+addition,100,'g','filled')
scatter3(X1_3,X2_3,X3_3+addition,100,'y','filled')
scatter3(X1_4,X2_4,X3_4+addition,100,'r','filled')
```

```
305
306 % for t=1:numel(beam.posx1)
307 %     if absorb(t) <= min(absorb)+interv
308 %         scatter3(X1(t),X2(t),X3(t),100,'b','filled')
309 %     elseif absorb(t) <= min(absorb)+2*interv
310 %         scatter3(X1(t),X2(t),X3(t),100,'g','filled')
311 %     elseif absorb(t) <= min(absorb)+3*interv
312 %         scatter3(X1(t),X2(t),X3(t),100,'y','filled')
313 %     elseif absorb(t) <= min(absorb)+4*interv
314 %         scatter3(X1(t),X2(t),X3(t),100,'r','filled')
315 %     end
316 % end
317
318 if strcmp('Y',str)
319     frame = getframe(gcf);
320     writeVideo(v,frame);
321 end
322 toc;
323 hold off
324 if strcmp('Y',str)
325     close(v);
326 end
```

5 Ethical Considerations for this Project

A goal of this project is to enable advancements in science and engineering through to address critical national challenges associated with next generation food systems. There are deep ethical considerations associated with any technology, in particular for food systems. While technology has tremendous potential to identify greater efficiencies, when it is created without appropriate consideration for who will have access to and control over new resources, or how the new technologies will impact those who work in the system, the efficiencies identified may come at the cost of greater societal inequity. It is important to pursue harnessing technology to disrupt existing inequities, rather than further entrench existing power structures. The following areas should be considered:

- Labor: 1) occupational health, 2) food manufacturing, and 3) outdoor agriculture labor;
- Producers: 1) Small- to mid-size farms, 2) urban agriculture, and 3) research in farm transitions; Technology: 1) research in technology and democracy;
- Health Human Rights: 1) land rights, 2) social justice, and 3) decolonization in agriculture;

Please consider the following questions:

- What are the societal implications of the technology that you are developing?
- Can this technology be distributed fairly and equitably to a wide variety of entities in agricultural industry?
- Are there any potential unintended consequences of this technology becoming available?
- Are there any harmful "spinoffs" of this technology?
- Are there any useful "spinoffs" of this technology?

6 References

1. Anderson, J. G., Rowan, N. J., MacGregor, S. J., Fouracre, R. A., Farish, O. (2000). Inactivation of food-borne enteropathogenic bacteria and spoilage fungi using pulsed-light. IEEE Transactions on Plasma Science , 28 (1), 83-88.

2. Battelle (2020). Instructions for Healthcare Personnel: Preparation of Compatible N95 Respirators for Decontamination by the Battelle Memorial Institute Using the Battelle Decontamination System. https://www.fda.gov/media/137032/download

3. Bolton, J. and Colton, C. (2008). The Ultraviolet Disinfection Handbook, American Water Works Association. ISBN 978 1 58321 584 5, pp. 3-4.

4. Boyce, JM (2016). Modern technologies for improving cleaning and disinfection of environmental surfaces in hospitals. Antimicrobial Resistance and Infection Control. 5: 10. doi:10.1186/s13756-016-0111-x. PMC 4827199. PMID 27069623.

5. Card, K. J., Crozier, D., Dhawan, A., Dinh, M., Dolson, E., Farrokhian, N., Gopalakrishnan, V., Ho, E., King, E. S., Krishnan, N., Kuzmin, G., Maltas, J., Pelesko, J., Scarborough, J. A., Scott, J. G., Sedor, G., Weaver, D. T. (2020). UV Sterilization of Personal Protective Equipment with Idle Laboratory Biosafety Cabinets During the Covid-19 Pandemic [Preprint]. Occupational and Environmental Health. https://doi.org/10.1101/2020.03.25.20043489

6. Downes, A. and Blunt, T. P. (1878). On the Influence of Light upon Protoplasm. Proceedings of the Royal Society of London. 28 (190-195): 199-212. Bibcode:1878RSPS...28..199D. doi:10.1098/rspl.1878.0109

7. Gross, H. (2005). Handbook of optical systems. Fundamental of technical optics. H. Gross, Editor. Wiley-VCH.

8. Heimbuch, B. and Harnish, D. (2019). Research to Mitigate a Shortage of Respiratory Protection Devices During Public Health Emergencies (Report to the FDA No. HHSF223201400158C). Applied Research Associate, Inc.

9. Heimbuch, B. K., Wallace, W. H., Kinney, K., Lumley, A. E., Wu, C.-Y., Woo, M.-H., Wander, J. D. (2011). A pandemic influenza preparedness study: Use of energetic methods to decontaminate filtering facepiece respirators contaminated with H1N1 aerosols and droplets. American Journal of Infection Control , 39 (1), e1-e9.

10. Ito, A., Ito, T. (1986). Absorption spectra of deoxyribose, ribosephosphate, ATP and DNA by direct transmission measurements in the vacuum-UV (150-190 nm) and far-UV (190-260 nm) regions using synchrotron radiation as a light source. Photochemistry and Photobiology , 44 (3), 355-358.

11. Jackson, J. D. (1998). Classical Electrodynamics.

12. Kanemitsu, K. et al, (2005) Does incineration turn infectious waste aseptic?, Journal of Hospital Infection, 60(4):304-306.

13. Lin, T.-H., Tang, F.-C., Hung, P.-C., Hua, Z.-C., Lai, C.-Y. (2018). Relative survival of Bacillus subtilis spores loaded on filtering facepiece respirators after five decontamination methods. Indoor Air , 28 (5), 754-762.

14. Lindsley,W. G., Martin, S. B., Thewlis, R. E., Sarkisian, K., Nwoko, J. O., Mead, K. R., Noti, J. D. (2015). Effects of Ultraviolet Germicidal Irradiation (UVGI) on N95 Respirator Filtration Performance and Structural Integrity. Journal of Occupational and Environmental Hygiene , 12 (8), 509-517. https://doi.org/10.1080/15459624.2015.1018518

15. Lore, M. B., Heimbuch, B. K., Brown, T. L., Wander, J. D., Hinrichs, S. H. (2011). Effectiveness of Three Decontamination Treatments against Influenza Virus Applied to Filtering Facepiece Respirators. The Annals of Occupational Hygiene , 56 (1), 92-101.

16. Marra, A. R., Schweizer, M. L., Edmond, M. B. (2018). No-Touch Disinfection Methods to Decrease Multidrug-Resistant Organism Infections: A Systematic Review and Meta-analysis. Infection Control Hospital Epidemiology , 39 (1), 20-31.

17. Mills, D., Harnish, D. A., Lawrence, C., Sandoval-Powers, M., Heimbuch, B. K. (2018). Ultraviolet germicidal irradiation of influenza-contaminated N95 filtering facepiece respirators. American Journal of Infection Control , 46 (7), e49-e55.

18. Nerandzic, M. M., Cadnum, J. L., Pultz, M. J., Donskey, C. J. (2010). Evaluation of an automated ultraviolet radiation device for decontamination of Clostridium difficile and other healthcare-associated pathogens in hospital rooms. BMC Infectious Diseases , 10 (1), 197.

19. Tseng, C.-C., Li, C.-S. (2007). Inactivation of Viruses on Surfaces by Ultraviolet Germicidal Irradiation. Journal of Occupational and Environmental Hygiene , 4 (6), 400-405.

20. Viscusi, D. J., Bergman, M. S., Eimer, B. C., Shaffer, R. E. (2009). Evaluation of Five Decontamination Methods for Filtering Facepiece Respirators. The Annals of Occupational Hygiene , 53 (8), 815-827.

21. Zohdi, T. I. (2006a). Computation of the coupled thermo-optical scattering properties of random particulate systems. Computer Methods in Applied Mechanics and Engineering. Volume 195, 5813-5830.

22. Zohdi, T. I. (2006b). On the optical thickness of disordered particulate media. Mechanics of Materials. Volume 38, 969-981.

23. Zohdi, T. I and Kuypers, F. A. (2006c). Modeling and rapid simulation of multiple red blood cell light scattering. Proceedings of the Royal Society Interface. Volume 3, Number 11 Pages 823-831.

24. Zohdi, T. I. (2012). Electromagnetic properties of multiphase dielectrics. A primer on modeling, theory and computation. Springer-Verlag.

25. Zohdi, T. I. (2015). A computational modeling framework for high-frequency particulate obscurant cloud performance. The International Journal of Engineering Science. 89, 75-85.

26. Zohdi, T. I. (2016). On high-frequency radiation scattering sensitivity to surface roughness in particulate media. Computational Particle Mechanics. http://dx.doi.org/10.1007/s40571-016- 0118-3

27. Zohdi, T. I. (2019). Rapid simulation-based uncertainty quantification of flash-type time-offlight and Lidar-based body-scanning processes. Computer Methods in Applied Mechanics and Engineering. https://doi.org/10.1016/j.cma.2019.03.056

28. Zohdi, T.I. (2020) Rapid simulation of viral decontamination efficacy with UV irradiation. Computer Methods Appl. Mech. Eng. https://doi.org/10.1016/j.cma.2020.113216