```
% Raytrace Function
function [radiation,Power_at_receiver,no_elements,avg] =
raytrace_function(files,data)
x = zeros(length(files), 1000000); % preallocate Using 1 million because it
% was defined to have maximum 1 million photons per file exported by Tonatiuh
y = zeros(length(files), 1000000); % preallocate
z = zeros(length(files), 1000000); % preallocate
for i=1:length(files)
ID = fopen(files(i).name); % opening files
file_raw_data = fread(ID,'double', 'b'); % reading files
file_photon_set = reshape(file_raw_data,5,[]); % partition of 5-tuples
% Selecting only rays that reach from the bottom
for j=1:length(file_photon_set)
if file_photon_set(5,j) == 0
x(i,j) = file_photon_set(2,j);% use this
y(i,j) = file_photon_set(3,j);
z(i,j) = file_photon_set(4,j);% use this
end
end
clear ID;
clear file_raw_data;
clear file_photon_set;
end
% Gathering all photons that reach the surface & Changing axis for having
% a clear plot.
if length (files) == 1
y_coord=transpose(x(1,1:j));
x_coord=transpose(z(1,1:j));
else
if length (files) == 2
y_coord=transpose([x(1,1:1000000) x(2,1:j)]);
x_coord=transpose([z(1,1:1000000) z(2,1:j)]);
else
if length (files) == 3
y_coord=transpose([x(1,1:1000000) x(2,1:1000000) x(3,1:j)]);
x_coord=transpose([z(1,1:1000000) z(2,1:1000000) z(3,1:j)]);
else
if length (files) == 4
y_coord=transpose([x(1,1:1000000) x(2,1:1000000) x(3,1:1000000) x(4,1:j)]);
x_coord=transpose([z(1,1:1000000) z(2,1:1000000) z(3,1:1000000) z(4,1:j)]);
else
if length (files) == 5
y_coord=transpose([x(1,1:1000000) x(2,1:1000000) x(3,1:1000000) x(4,1:1000000) x.
(5,1:j)]);
x_coord=transpose([z(1,1:1000000) z(2,1:1000000) z(3,1:1000000) z(4,1:1000000) z.
(5,1:j)]);
else
if length (files) == 6
y_coord=transpose([x(1,1:1000000) x(2,1:1000000) x(3,1:1000000) x(4,1:1000000) x.
(5,1:1000000) x(6,1:j)]);
x_coord=transpose([z(1,1:1000000) z(2,1:1000000) z(3,1:1000000) z(4,1:1000000) z.
(5,1:1000000) z(6,1:j)]);
else
end
end
end
end
end
end
```

```
%importfile('Receiver_parameters.txt');
power_per_photon = data; %W per photon
no_photons_reaching = length(x_coord);
Power_at_receiver = power_per_photon*no_photons_reaching; % W
coord = [x_coord y_coord];
%coord_rotated= [y_coord x_coord];
width_cell = .148;
length_string = .988;
cell_divisions_length = 1520;
length_element = length_string/cell_divisions_length;
%width_element = width_cell/cell_divisions_width;
width_cell_1 = .022;
width_cell_2 = .104;
cell_divisions_width_1 = 22;
cell_divisions_width_2 = 104;
width_element_1 = width_cell_1/cell_divisions_width_1;
width_element_2 = width_cell_2/cell_divisions_width_2;
no_elements =
cell_divisions_length*(cell_divisions_width_1*2+cell_divisions_width_2);
area_element_1 = length_element*width_element_1;
area_element_2 = length_element*width_element_2;
factor_1 = (power_per_photon)/area_element_1; %W/m^2 per photon
factor_2 = (power_per_photon)/area_element_2;
% Establishing the edges of the histogram
edge_x = -length_string/2:length_element:length_string/2;
edge_y = [-width_cell/2:width_element_1:-.052 -.051:width_element_2:.052 .053:.
width_element_1:width_cell/2];
%edge_y = -width_cell/2:.00148:width_cell/2;
edges = (edge_x),(edge_y);
% Establishing ticks and labels for graph
tick_y = [-.494, -.45:.05:.45, .494];
yticklabel = [-494, -450:50:450, 494];
tick_x = -.07:.01:.07;
xticklabel = -70:10:70;
distribution = hist3(coord,'Edges',edges);
%photons_rot = hist3(coord_rotated, bins_rot); %Rotated plot
radiation_graph = zeros(1521,149);
for i=1:23
radiation_graph(:,i) = distribution(:,i)*factor_1;
end
for i=24:127
radiation_graph(:,i) = distribution(:,i)*factor_2;
end
for i=128:149
radiation_graph(:,i) = distribution(:,i)*factor_1;
end
%radiation_graph = [rad_; %kW/m^2
radiation = radiation_graph;
radiation(end,:) = [];
radiation(:,end) = [];
avg = mean(mean(radiation));
% Radiation distribution on the module.
%set(figure(1),'Position', [300 246 800 420]);
h = pcolor(edge_y,edge_x,radiation_graph);
colorbar;
set(h, 'EdgeColor', 'none');
ylabel(colorbar,'Radiation (W m^-2)')
set(gca,'XTick',tick_x);
set(gca,'YTick',tick_y);
```

```
set(gca,'YTickLabel',yticklabel)
set(gca,'XTickLabel',xticklabel)
title 'Radiation Distribution'
xlabel 'Cell Width (mm)'
ylabel 'String length (mm)'
box off
%axis equal tight
end
```