**Simulation code for Fig. 3B**

# small\_box\_plane\_wave\_setup\_03212024.lsf

####

# This script sets up the FDTD environment

# i.e. geometry of scatterers, plane wave illumination, boundary condition, field distribution monitor

# Author: Yi-Shiou Duh (allenduh@stanford.edu)

# 3/21/2024

##############################################

## FDTD Set up

##############################################

### Initialization

switchtolayout;

selectall;

clear; # script workspace

deleteall;

### File export

filename\_base = " C:\Users\allen\Desktop\plane\_wave\_smallbox\_03212024\_index\_";

### Const

nmUnit = 1e-9;

umUnit = 1e-6;

c0 = 3e8;

# Mesh setting

mesh = "conformal variant 0"; # How to choose see https://support.lumerical.com/hc/en-us/articles/360034382614

meshacc = 4; # mesh accuracy = 4

meshOverRidenm = 10;

### FDTD Domain size

Xspan = 10 \* umUnit;

Yspan = 12 \* umUnit;

# FDTD time

simulationTime = 3 \* Yspan / c0;

pmlLayers = 5;

dt = 0.99; # DT STABILITY FACTOR. Larger: faster simulation time. For long propagation set to 0.9

autoShut = 1e-08;

# Monitor setup

downSample = 1; # or 3 (coarse)

# Plane wave source location

sourceY = 0.1 \* umUnit;

# Define wavelength

lambdaStart = 600 \* nmUnit; # Initial wavelength

lambdaEnd = 680 \* nmUnit; # Final wavelengt

lambdaStep = 40 \* nmUnit; # Wavelength step

##############################################

## Scatter

# Radius -- 0.5 micron

# SiO2 (1.43) in water (1.33, 1.37, 1.41, 1.45, 1.49)

# Density 0.3 particles / um^2

# We made these scattering phantoms by uniformly mixing silica nanospheres with an n' of 1.43 (close to that of lipids and collagen)

# in an optically transparent hydrogel with a background n' of 1.33 (the same as water)

##############################################

scatter\_radius = 500 \* nmUnit;

particleDensity = 0.3;

blurry\_top\_Y = 11 \* umUnit;

blurry\_down\_Y = 1 \* umUnit;

numOfParticles = round((blurry\_top\_Y - blurry\_down\_Y) \* Xspan / umUnit ^ 2 \* particleDensity);

# Background index list

background\_index = [1.31, 1.32, 1.33, 1.34, 1.35, 1.36, 1.37, 1.38, 1.39, 1.4, 1.41, 1.42, 1.43, 1.44, 1.45, 1.46, 1.47, 1.48, 1.49, 1.5];

numOfBackgroundIndex = length(background\_index);

# Add material

if (materialexists("Dye\_index\_vary") != 1) {

setmaterial(addmaterial("(n,k) Material"), "name", "Dye\_index\_vary");

}

if (materialexists("Hydrogel\_1.33") != 1) {

setmaterial(addmaterial("(n,k) Material"), "name", "Hydrogel\_1.33");

setmaterial("Hydrogel\_1.33", {"Refractive Index": 1.33, "Imaginary Refractive Index": 0});

}

if (materialexists("Silica\_1.43") != 1) {

setmaterial(addmaterial("(n,k) Material"), "name", "Silica\_1.43");

setmaterial("Silica\_1.43", {"Refractive Index": 1.43, "Imaginary Refractive Index": 0});

}

loaddata("small\_box\_scatters\_location");

##############################################

## If scatter's location (xLocList, yLocList) not determined

##############################################

#xLocList = matrix(numOfParticles, 1);

#yLocList = matrix(numOfParticles, 1);

#particleIndex = 0;

#for (i = 1 : numOfParticles + 100) {

#if (particleIndex < numOfParticles) {

#xLoc = rand(-Xspan/2 + scatter\_radius, Xspan/2 - scatter\_radius);

#yLoc = rand(blurry\_down\_Y + scatter\_radius, blurry\_top\_Y - scatter\_radius);

### Check new added particle not touch previous generate

#touch = false;

#for (j = 1 : particleIndex) {

#xOtherParticle = xLocList(j, 1);

#yOtherParticle = yLocList(j, 1);

#interDist = sqrt((xOtherParticle - xLoc)^2 + (yOtherParticle - yLoc)^2);

#if (interDist < 2 \* scatter\_radius) { ## Touch

#touch = true;

#}

#}

### If new particle not touch previous generated, add into (xLocList, yLocList)

#if (touch == false) {

#xLocList(particleIndex + 1) = xLoc;

#yLocList(particleIndex + 1) = yLoc;

#particleIndex = particleIndex + 1;

#}

#}

#}

#savedata("small\_box\_scatters\_location", xLocList, yLocList);

# In the front

#xLocList = matrix(numOfParticles, 1);

#yLocList = matrix(numOfParticles, 1);

#select("SiO2");

#for (i=1:getnumber) {

#xLoc=get("x",i);

#yLoc=get("y",i);

#xLocList(i) = xLoc ;

#yLocList(i) = yLoc ;

#}

#savedata("small\_box\_scatters\_location", xLocList, yLocList);

# small\_box\_plane\_wave\_run\_03212024.lsf

########

# This script runs FDTD at different background index and export field distribution data

# When plane wave passes through few particle scatterers, wavefront distorts

# As we sweep refractive index of the background, wavefront becomes plane wave when index matches

# Export field distribution at 600, 640, 680nm to .mat

# Author: Yi-Shiou Duh (allenduh@stanford.edu)

# 3/21/2024

########

for (idxDye = 1 : numOfBackgroundIndex) {

switchtolayout;

deleteall;

RI\_dye = background\_index(idxDye);

?RI\_dye;

filename = filename\_base + num2str(idxDye);

############################

###### Add FDTD

############################

addfdtd;

set("dimension", 1); # 1 = 2D, 2 = 3D

set("simulation time", simulationTime);

set("simulation temperature", 300);

set("x", 0);

set("x span", Xspan);

set("y", Yspan / 2);

set("y span", Yspan);

set("mesh accuracy", meshacc);

set("x min bc", "periodic"); # Simple BC for plane wave illumination, X Span 30um >> wavelength

set("x max bc", "periodic");

set("y min bc", "PML");

set("y max bc", "PML");

set("mesh refinement", mesh);

set("index", RI\_dye);

set("pml layers", pmlLayers);

set("dt stability factor", dt);

set("auto shutoff min", autoShut);

############################

###### Add Background dye

############################

addrect;

set("name", "Liquid");

set("x", 0);

set("x span", Xspan + umUnit); # Liquid rectangle surpass boundary to avoid error

set("y", Yspan / 2);

set("y span", Yspan + umUnit);

# Set Dye refractive index

setmaterial("Dye\_index\_vary", {"Refractive Index": RI\_dye, "Imaginary Refractive Index": 0});

set("material", "Dye\_index\_vary");

############################

###### Add scatterers

############################

for(k = 1 : numOfParticles) {

addcircle;

set("name","SiO2");

set("x", xLocList(k));

set("y", yLocList(k));

set("radius", scatter\_radius);

#set("material","SiO2 - Quartz");

#set("material","SiO2 (Glass) - Palik");

set("material","Silica\_1.43");

}

############################

##### Add Plane wave

############################

addplane;

set("name", "Plane wave");

set("injection axis", "y");

set("direction", "forward");

set("x", 0);

set("x span", Xspan);

set("y", sourceY);

set("wavelength start", lambdaStart);

set("wavelength stop", lambdaEnd);

############################

##### Add field distribution monitor

############################

setglobalmonitor("frequency points", (lambdaEnd - lambdaStart) / lambdaStep + 1);

setglobalmonitor("use wavelength spacing", 1);

addpower;

set("name","FieldDistribution");

set("monitor type", 7); # 1 = point, 2 = linear x, 3 = linear y, 4 = linear z, 5 = 2D x-normal, 6 = 2D y-normal, 7 = 2D z-normal, 8 = 3D

set("x", 0);

set("x span", Xspan);

set("y min", sourceY);

set("y max", Yspan);

set("down sample X", downSample);

set("down sample Y", downSample);

############################

## Run simulation and file save

############################

run;

# Extract values to save

fullField = getresult("FieldDistribution","E");

fullField\_Ex = fullField.Ex;

fullField\_Ey = fullField.Ey;

fullField\_lambda = fullField.lambda;

fullField\_xCoordinate = fullField.x;

fullField\_yCoordinate = fullField.y;

matlabsave(filename, fullField\_Ex, fullField\_Ey, fullField\_lambda, xLocList, yLocList, fullField\_xCoordinate, fullField\_yCoordinate);

print(filename);

}

**Simulation code for Fig. 4K-M**

# image\_deep\_scatters\_setup\_03212024.lsf

####

# This script sets up FDTD environment

# i.e. Geometry of scatters, gaussian illumination, boundary condition, field distribution monitor

# Author: Yi-Shiou Duh (allenduh@stanford.edu)

# 3/21/2024

##############################################

## FDTD Set up

##############################################

### Initialization

switchtolayout;

selectall;

clear; # script workspace

deleteall;

### File export

filename\_base = "C:\Users\allen\Desktop\image\_deep\_scatters\_03212024\_";

matlabload("C:/Users/allen/Desktop/1800\_scatters.mat"); # Load scatters x, y locations

### Const

nmUnit = 1e-9;

umUnit = 1e-6;

c0 = 3e8;

# Mesh setting

mesh = "conformal variant 0"; # How to choose see https://support.lumerical.com/hc/en-us/articles/360034382614

meshacc = 2; # mesh accuracy

### FDTD Domain size

Xspan = 600 \* umUnit;

Yspan = 1100 \* umUnit;

# FDTD time

simulationTime = 7500 \* 1e-15; #2.05 \* Yspan / c0;

pmlLayers = 5;

dt = 0.9; # DT STABILITY FACTOR. Larger: faster simulation time. For long propagation set to 0.9

autoShut = 1e-08;

# Monitor setup

downSample = 3; # 1 or 3 (coarse)

# Gaussian Source

sourceY = 5 \* umUnit;

NA\_source = 0.25;

focus\_offset = 0;

# Define wavelength

lambdaStart = 600 \* nmUnit; # Initial wavelength

lambdaEnd = 680 \* nmUnit; # Final wavelengt

lambdaStep = 40 \* nmUnit; # Wavelength step

##############################################

## Scatter

# Radius -- 0.5 micron

# SiO2 (1.43) in water

# Density 0.003 particles / um^2

scatter\_radius = 500 \* nmUnit;

particleDensity = 0.003;

blurry\_top\_Y = 1010 \* umUnit; # 1mm scattering medium thickness

blurry\_down\_Y = 10 \* umUnit;

numOfParticles = round((blurry\_top\_Y - blurry\_down\_Y) \* Xspan / umUnit ^ 2 \* particleDensity);

# Add material

if (materialexists("Dye\_index\_vary") != 1) {

setmaterial(addmaterial("(n,k) Material"), "name", "Dye\_index\_vary");

}

if (materialexists("Hydrogel\_1.33") != 1) {

setmaterial(addmaterial("(n,k) Material"), "name", "Hydrogel\_1.33");

setmaterial("Hydrogel\_1.33", {"Refractive Index": 1.33, "Imaginary Refractive Index": 0});

}

if (materialexists("Dye\_0.62M\_600nm") != 1) {

setmaterial(addmaterial("(n,k) Material"), "name", "Dye\_0.62M\_600nm");

setmaterial("Dye\_0.62M\_600nm", {"Refractive Index": 1.41, "Imaginary Refractive Index": 1.5e-3});

}

if (materialexists("Silica\_1.43") != 1) {

setmaterial(addmaterial("(n,k) Material"), "name", "Silica\_1.43");

setmaterial("Silica\_1.43", {"Refractive Index": 1.43, "Imaginary Refractive Index": 0});

}

# image\_deep\_scatters\_run\_03212024.lsf

########

# This script runs FDTD at different background index and exports field distribution data

# When Gaussian beam passes through scattering medium, wavefront distort

# We simulate the field distribution across 1mm thickness

# Export field distribution and selective cross section at 600, 640, 680nm to .mat

# Author: Yi-Shiou Duh (allenduh@stanford.edu)

# 3/21/2024

########

background\_medium = "dye"; # water or dye

switchtolayout;

deleteall;

?background\_medium;

filename = filename\_base + background\_medium;

############################

###### Add FDTD

############################

addfdtd;

set("dimension", 1); # 1 = 2D, 2 = 3D

set("simulation time", simulationTime);

set("simulation temperature", 300);

set("x", 0);

set("x span", Xspan);

set("y", Yspan / 2);

set("y span", Yspan);

set("mesh accuracy", meshacc);

set("x min bc", "PML");

set("x max bc", "PML");

set("y min bc", "PML");

set("y max bc", "PML");

set("mesh refinement", mesh);

set("pml layers", pmlLayers);

set("dt stability factor", dt);

set("auto shutoff min", autoShut);

############################

###### Add Background dye

############################

addrect;

set("name", "Liquid");

set("x", 0);

set("x span", Xspan + umUnit); # Liquid rectangle surpass boundary to avoid error

set("y", Yspan / 2);

set("y span", Yspan + umUnit);

# Set RI

if (background\_medium == "water"){

set("material","Hydrogel\_1.33");

}

if (background\_medium == "dye"){

set("material","Dye\_0.62M\_600nm");

}

############################

###### Add scatter

############################

#hide; # show will enable GUI

print("adding " + num2str(numOfParticles) + " particles");

for(k = 1 : numOfParticles) {

addcircle;

set("name", "SiO2");

set("x", xLocList(k));

set("y", yLocList(k));

set("radius", scatter\_radius);

set("material","Silica\_1.43");

}

############################

##### Add Plane wave

############################

addgaussian;

set("name", "Gaussian");

set("injection axis", "y");

set("direction", "forward");

set("x", 0);

set("x span", Xspan);

set("y", sourceY);

set("wavelength start", lambdaStart);

set("wavelength stop", lambdaEnd);

set("optimize for short pulse", 0);

# Thin lens for point source

set("use thin lens", 1);

set("NA", NA\_source);

set("distance from focus", focus\_offset);

set("number of plane waves", 400);

############################

##### Add field distribution monitor

############################

setglobalmonitor("frequency points", (lambdaEnd - lambdaStart) / lambdaStep + 1);

setglobalmonitor("use wavelength spacing", 1);

addpower;

set("name","FieldDistribution");

set("monitor type", 7); # 1 = point, 2 = linear x, 3 = linear y, 4 = linear z, 5 = 2D x-normal, 6 = 2D y-normal, 7 = 2D z-normal, 8 = 3D

set("x", 0);

set("x span", Xspan);

set("y min", sourceY);

set("y max", Yspan);

set("down sample X", downSample);

set("down sample Y", downSample);

# forward monitor

addpower;

set("name","forwardScatter");

set("monitor type", 2); # 1 = point, 2 = linear x, 3 = linear y, 4 = linear z, 5 = 2D x-normal, 6 = 2D y-normal, 7 = 2D z-normal, 8 = 3D

set("x", 0);

set("x span", Xspan);

set("y", Yspan);

set("down sample X", 1);

############################

## Run simulation and file save

############################

run;

fullField = getresult("FieldDistribution","E");

forwardScatter = getresult("forwardScatter","E");

fullField\_Ex = fullField.Ex;

fullField\_Ey = fullField.Ey;

fullField\_lambda = fullField.lambda;

fullField\_xCoordinate = fullField.x;

fullField\_yCoordinate = fullField.y;

matlabsave(filename, fullField\_Ex, fullField\_Ey, fullField\_lambda, xLocList, yLocList, fullField\_xCoordinate, fullField\_yCoordinate, forwardScatter);