README:

A clustering diffused particle method

Lang Wang,
Ilia L. Rasskazov,
P. Scott Carney

August 26, 2020

Optics & electromagnetics

Diffused particle method calculates the scattered field from point particles with both electric and magnetic responses. The computational complexity is reduced by hierarchical clustering techniques to enable simulations with on the order of 10^{10} particles. The calculation is based on a generalized Foldy-Lax equation in [2]. With so many particles we are able to see the transition to bulk media behavior of the fields. Thus the method is also used to verify a generalized effective medium theory [1].

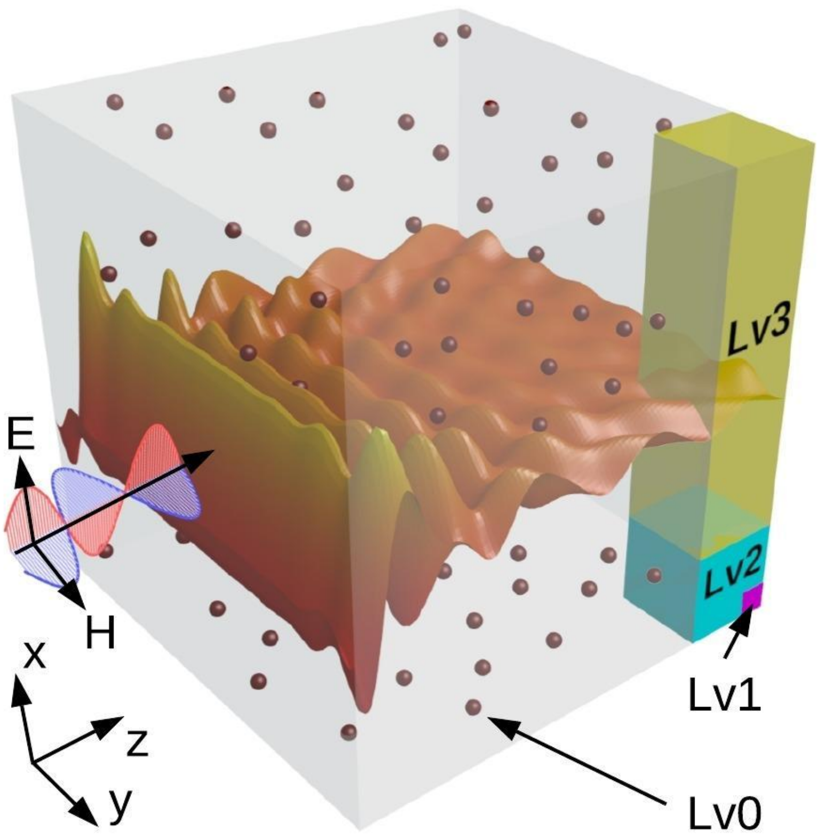

Figure 1: The simulation model of the diffused-particle method.

Contents

1 Examples

1.1 *elec_field.m*

The script *elec_field.m* plots the electric field in the middle plane of the simulation region. It reproduces Fig. 5 in [1] and Fig. 4 in [2] with the input parameters given in the corresponding references. The input parameters are listed in the table below:

parameter	unit	description
k0	λ^{-1}	wave number
rho	λ^3	concentration of particles
alpha_e	λ^3	electric polarizability
alpha_m	λ^3	magnetic polarizability
Ys	λ	Y size of the sampling region
Zs	λ	Z size of the sampling region
n0		initial guess of refractive index n
L	λ	side length of the cubic simulation region
D1	λ	side length of a Lv1 voxel
D2	λ	side length of a Lv2 voxel

1.2 *sweep_beta.m*

The script *sweep_beta.m* plots the values of ε, μ and n with different β_{e} and β_{m} values, based on [2]. It reproduces Fig. 4 in [1] and Fig. 5 in [2](a) with the input parameters given in the corresponding references. The input parameters different from the script *elec_field.m* are listed in the table below:

parameter	unit	description
alpha_e_sweep	λ^3	swept electric polarizability values
alpha_m_sweep	λ^3	swept magnetic polarizability values

1.3 *sweep_n0.m*

The script *sweep_n0.m* plots the number of iterations before the convergence of the algorithm with different n_0 values, based on [2]. It reproduces Fig. 5 in [2](b) with the input parameters given in the corresponding references. The input parameters different from the script *elec_field.m* are listed in the table below:

parameter	unit	description
n0_re		swept real part of n_0
n0_im		swept imaginary part of n_0

2 Functions

2.1 Effective medium theories

These functions under the folder */functions/eff_medium_theory* calculate theoretical macroscopic parameters ε and μ given microscopic parameters β_{e} and β_{m}.

2.1.1 *CM.m*

Computes permittivity and permeabiity by the Clausius-Mossotti relation. The input and output parameters are given below:

parameter	unit	description
Inputs		
be		β_{e}
bm		β_{m}
Outputs		
ep		ε
mu		μ

2.1.2 *GCM_m_branch.m*

Computes theoretical ("-" branch) effective permittivity and permeabiity based on cite1. The input and output parameters are the same as the function *CM.m*.

2.1.3 *GCM_p_branch.m*

Computes theoretical ("+" branch) effective permittivity and permeabiity based on cite1. The input and output parameters are the same as the function *CM.m*.

2.2 Green's functions

These functions under the folder */functions/Greens_func* calculate Green's functions.

2.2.1 *G2d.m*

Calculates the 2D Green's function. The input and output parameters are given below:

parameter	unit	description
Inputs		
k0	λ^{-1}	free-space wave number
rs	λ	3D location of the source
rt	λ	3D location of the observing point
Outputs		
G		the value of the 2d Green's function

2.2.2 *G2d.m*

Calculates the 3D Green's function $\times 4\pi k_0^2$. The input and output parameters are the same as the function *G2d.m*.

2.2.3 *Gxx.m*

Calculates the xx component of 3D Green's function $\times 4\pi k_0^2$. The input and output parameters are the same as the function *G2d.m*.

2.3 Dipole and mesh generators

These functions under the folder */functions/cube_creator* generate 3D locations of the dipoles or the cubic mesh.

2.3.1 *cube_mesh.m*

Generates 3D cubic mesh in a cube. The input and output parameters are given below:

parameter	unit	description
		Inputs
ax	λ	side length of the cube in x
ay	λ	side length of the cube in y
az	λ	side length of the cube in z
dr	λ	distance between neighbor mesh nodes
		Outputs
r	λ	3D locations of mesh nodes

2.3.2 *ran_cube.m*

Generates random 3D locations of dipoles in a cube. The input and output parameters are given below:

parameter	unit	description
		Inputs
X	λ	side length of the cube in x
Y	λ	side length of the cube in y
Z	λ	side length of the cube in z
n		number of dipoles
		Outputs
r	λ	random 3D locations of the dipoles

2.4 Plot tools

These functions under the folder */functions/plot_tool* provide plot tool for different requirements.

2.4.1 *plot_DPM.m*

Plots the simulated values of epsilon and mu by diffused particle method, for example: Fig. 4 in [1]. The input parameters are given below:

parameter	unit	description
beta_e		β_{e}
ep		ε
mu		μ
fig_numb		figure number of the plot

2.4.2 *plot_E_colormap.m*

plots the electric field in the middle plane of the simulation region, for example Fig. 1 and Fig. 4 in [2]. The input parameters are given below:

parameter	unit	description
L	λ	side length of the cubic simulation region
D2	λ	side length of the lv2 voxel
E		electric field distributed in the middle plane

2.4.3 *plot_EMT.m*

Plots the theoretical values of ε and μ vs. $Re(\beta_{\mathrm{e}})$, for example: Fig. 4 in [1]. The input parameters are given below:

parameter	unit	description
beta_e_start		start value of β_{e} array
beta_e_end		end value of β_{e} array
beta_m_start		start value of β_{m} array
beta_m_end		end value of β_{m} array
fig_numb		figure number of the plot

References

[1] L. Wang, I. L. Rasskazov, and P. S. Carney, Clausius-mossotti relation revisited: Media with electric and magnetic response, arXiv preprint: 2008.09178 (2020) .

[2] L. Wang, I. L. Rasskazov, and P. S. Carney, Clustering diffused-particle method for scattering from large ensembles of electromagnetically polarizable particles, arXiv preprint: 2008.09185 (2020) .