



# **KEMP User Guide & Tutorials**

Diffraction of a dielectric single hole

Contributors: Min Gon Lee and SeokJae Yoo

Last Update: 09/25/2015

## Contents

- 1) Basic setting
- 2) Geometry setting
- 3) Others setting
- 4) Execution
- 5) Data collection

여기서는 사용자의 편의성을 위해 이미 만들어져 있는 파일을 이용해서 설명한다. `example_singlerectangularhole.py` 파일을 이용하면 Single hole 구조를 쉽게 만들 수 있다.

# Diffraction of a dielectric single hole

## 1) Basic Setting

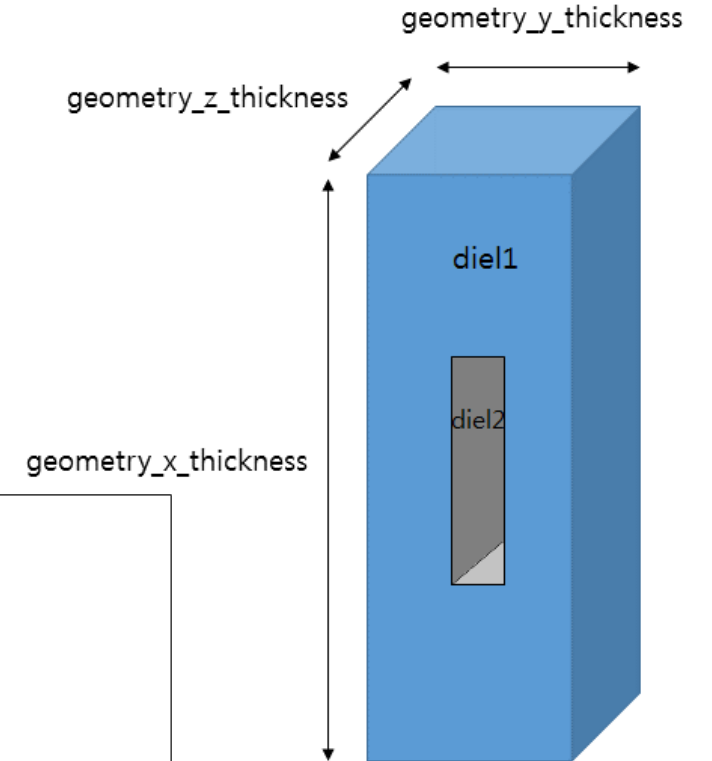
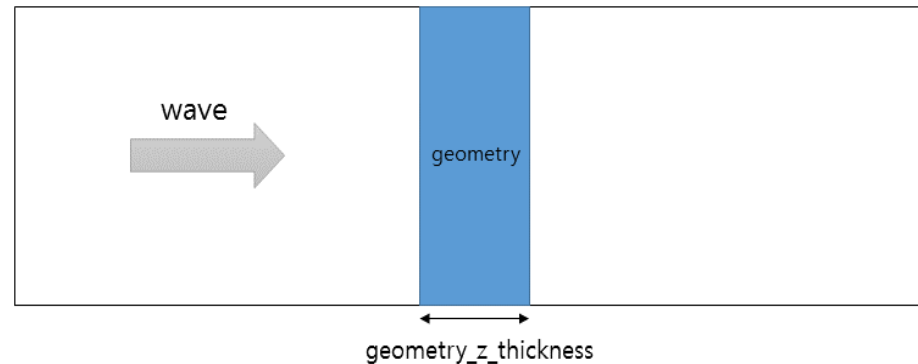
```
#-----general setting-----  
  
nm = 1e-9  
um = 1e-6  
dx, dy, dz = [10.*nm, 5*nm, 5*nm]  
nx, ny, nz = [700, 200, 200]  
x = np.ones(nx, dtype=np.float64)*dx  
y = np.ones(ny, dtype=np.float64)*dy  
z = np.ones(nz, dtype=np.float64)*dz  
lx, ly, lz = dx*nx, dy*ny, dz*nz  
space_grid = (x, y, z)  
fdtd = Basic_FDTD('3D', space_grid, dtype=np.complex64, engine='nvidia_cuda')  
  
pml_apply = {'x':'+-' , 'y':'+-' , 'z':'+-' }  
pbc_apply = {'x':False, 'y':False, 'z':False}  
  
save_name = 'examplehole'
```

- FDTD 공간 설정과 simulation을 돌릴 device 그리고 PML, PBC를 설정한다.

# Diffraction of a dielectric single hole

## 2) Geometry Setting

```
#-----geometry configure-----  
  
geometry_on = 1      #put in geometry if 1  
  
geometry_x_thickness = 5000.*nm  
geometry_y_thickness = 500.*nm  
geometry_z_thickness = 500.*nm  
  
diel1 = Dielectric(to_epr(fDTD, n=1.))  
diel2 = Dielectric(to_epr(fDTD, n=10**20))
```



- 위 그림과 같이 geometry를 설정한다.
- Geometry를 적용하지 않을 경우  $geometry\_on = 0$  으로 설정한다.

# Diffraction of a dielectric single hole

## 3) Others Setting

```
#-----others setting-----  
  
tmax = 30000  
  
source_polarized = 'ey'  
  
configure_wavelength = np.array([5, 15, 0.1])*um
```

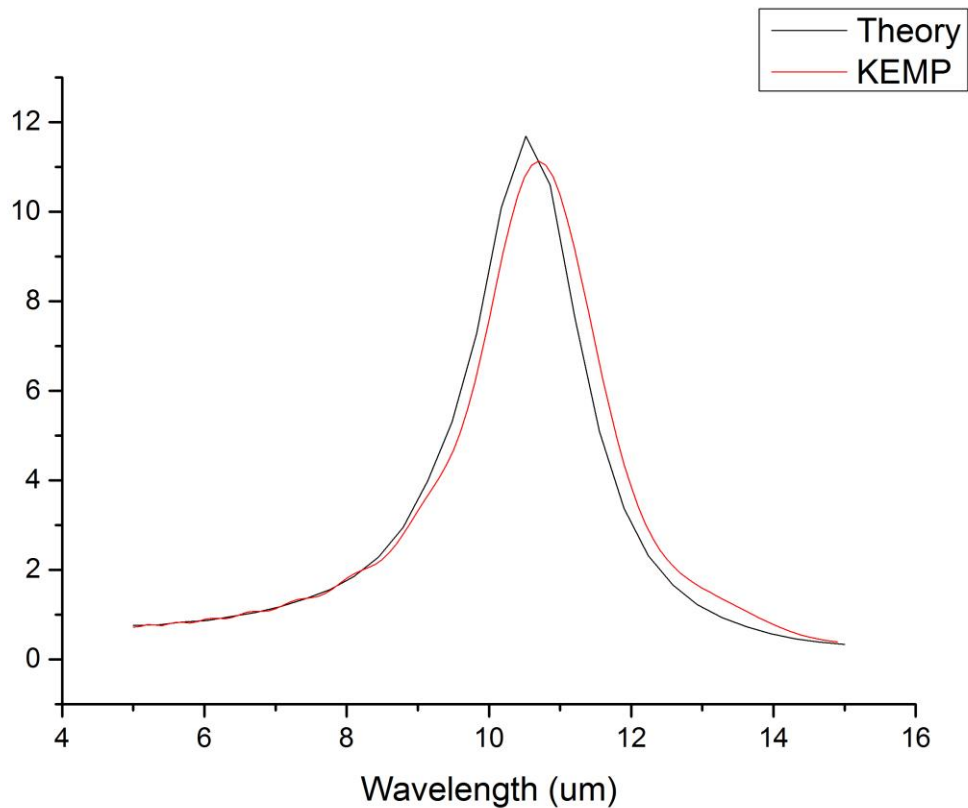
- FDTD step(wave가 진행되는 시간)을 설정한다.
- Source의 polarized direction과 wavelength의 범위를 설정한다.
- Wavelength의 범위는 [처음 wavelength, 마지막 wavelength, wavelength 간격]\*(unit of wavelength)로 설정한다. 결과에서 마지막 wavelength는 포함되지 않는다.

## 4) Execution

- Window
  - 저장된 파일을 실행한다.
- Linux
  - Python 파일이 있는 폴더에서 접속해 명령어 “python 파일명”로 실행한다.

# Diffraction of a dielectric single hole

## 5) Data Collection



- 실행후 HDF5 파일을 통해 poynting vector의 z성분을 얻는다. 여기서 얻은 결과는 z축의 중앙, 즉 hole내 z축 중앙에서 xy평면에서 얻은 값들을 이용해 이들을 모두 더한 값이 최종적으로 HDF5 파일에 저장된다. 이 값을  $Sz\_geometry$ 라고 하고, geometry를 적용하지 않은 free space에서 얻은 결과를  $Sz\_free$ 라고 하자. 왼쪽 그래프는  $Re[Sz\_geometry]/Re[Sz\_free]$ 로 normalization 한 것이다.