The FDTD program is called like this:

fdtd_sw(mdl,exc,out,sw,nmax,n_disp);

'mdl' is geometry model for FDTD, 'exc' is excitation, 'out' is output, 'sw' is Schwarzschild parameters. These four arguments are all structures, and their fields are introduced as below. 'nmax' is the maximum time step, 'n_disp' is the time step interval for display in command window.

The fields of 'mdl', 'exc', 'out' and 'sw':

mdl 1 × 1 struct: Geometry model

- mdl.**Nmedium** 1 × 1: Number of medium types except air and PEC. If Nmedium > 0, there should be two fields 'epsr' and 'mur':
- mdl.**epsr** Nmedium × 1: Relative permittivity
- mdl.**mur** Nmedium × 1: Relative permeability
- mdl.**NXi** 1 × 1: Number of mesh node in x direction
- mdl.**NYi** 1 × 1: Number of mesh node in y direction
- mdl.**NZi** 1 × 1: Number of mesh node in z direction
- mdl.**x** NXi × 1: Local x coordinates of mesh node (m)
- mdl.**y** NYi × 1: Local y coordinates of mesh node
- mdl.**z** NZi × 1: Local z coordinates of mesh node
  The global coordinates are sw.X0+[mdl.x,mdl.y,mdl.z]
- mdl.**tEx** NXi-1 × NYi × NZi: Medium type in Ex location
- mdl.**tEy** NXi × NYi-1 × NZi: Medium type in Ey location
- mdl.**tEz** NXi × NYi × NZi-1: Medium type in Ez location
- mdl.**tHx** NXi × NYi-1 × NZi-1: Medium type in Hx location
- mdl.**tHy** NXi-1 × NYi × NZi-1: Medium type in Hy location
- mdl.**tHz** NXi-1 × NYi-1 × NZi: Medium type in Hz location
  Type number: PEC=-1, air=0, medium=1~Nmedium

exc 1 × 1 struct: Excitation

- exc.**f_dipole** logical: Is there dipole excitation? If f_dipole is true, there should be the field 'dipole':
- exc.**dipole** 1 × 1 struct: Dipole excitation in z direction, and it occupies one grid
  - exc.dipole.**co** 3 × 1: Coordinates
  - exc.dipole.**wave** 1 × 1 struct: Waveform
- exc.**f_v** 1 × 1: Is there voltage excitation? If f_v is true, there should be the field v:
- exc.**v** 1 × 1 struct: Voltage excitation
  - exc.v.**dir** 1 × 1 char: Direction = 'X' or 'Y' or 'Z'
  - exc.v.**co** 4 × 1: Coordinates.
    If dir='X', co(1:2) are x coordinates, co(3) is y coordinate, co(4) is z coordinate
    If dir='Y', co(1) is x coordinate, co(2:3) are y coordinates, co(4) is z

coordinate

If dir='Z', co(1) is x coordinate, co(2) is y coordinate, co(3:4) are z coordinates

exc.v.**type** 1×1 char : 'S' or 'H'

exc.v.**wave** 1×1 struct: Waveform

wave 1×1 struct: Waveform

wave.**type** string: Type of waveform: 'SINE', 'GAUSS', 'DIFFGAUSS' or 'MODGAUSS'

wave.**A** 1×1: Amplitude

if type='SINE', there should be :

wave.**F** 1×1: Frequency (Hz)

if type='GAUSS' or 'DIFFGAUSS' or 'MODGAUSS', there should be:

wave.**BW** 1×1: Bandwidth (Hz)

if type='MODGAUSS', there should be:

wave.**F0** 1×1: Center frequency (Hz)

out 1×1 struct: Output

out.**nv** 1×1: Number of output voltage, if nv > 0, there should be the field v:

out.**v** $1 \times nv$ struct: output voltage

out.v.**dir** 1×1 char: Direction='X','Y','Z'

out.v.**co** 4×1: Coordinates, the same as exc.v.co

out.v.**fname** string: File name of the output file

out.**np** 1×1: Number of the output plane, if np > 0, there should be the field plane:

out.**plane** $1 \times np$ struct: output plane

out.plane.**p** char: 'x','y','z'

out.plane.**co** 1×1: Coordinate

out.plane.**nt** $n \times 1$: Time steps

out.plane.**e** string: 'Ex','Ey','Ez'. Output e in plane p=co at time step nt.

out.plane.**fname** string: File name of the output file

sw 1×1 struct: Schwarzschild parameters

sw.**Rs** 1×1: Schwarzschild Radius

sw.**X0** 3×1: The coordinates of origin of local coordinates system

%%

The Green function program is called like this:

[ct,U,U_u,U_0,V,S_0,S_00,S_u,S_u0]=GF_sw_cart(X0,X1,Rs,cdt);
e=dE(U,U_u,U_0,V,S_0,S_00,S_u,S_u0,JgS,J0gS);

X0 3×1: The coordinates of source positon

X1 3×1: The coordinates of receive positon

Rs 1×1: Schwarzschild Radius

cdt 1×1:Time step

ct 1×1: Time interval between X0 and X1

e 3×1: Differential element of electric field

The following arguments are quantities in Eq. (55):

U 4×4: $U_{\alpha\beta'}$

U_u $4 \times 4 \times 4$: $U_{\alpha\beta';\gamma}$

U_0 4×4: $U_{\alpha\beta',0'}$

V 4×4: $V_{\alpha\beta'}$

S_0 1×1: $\sigma_{0'}$

S_00 1×1: $\sigma_{0',0'}$

S_u 4×1: σ_{α}

S_u0 4×1: $\sigma_{\alpha,0'}$

JgS 4×1: $J^{\alpha'}\sqrt{-g'}d^3x'$

J0gS 4×1: $J^{\alpha'}{}_{,0'}\sqrt{-g'}d^3x'$