This program has been tested in MATLAB R2016b, Windows 7 Ultimate.
Please run 'RunMe.m' directly, and you will see the numerical results of the second example in my paper (Fig. 3).

The program 'RunMe.m' will output five files: 'v1.txt', 'v2.txt', 'v3.txt','v4.txt', 'dipole.mat'. These files have been saved in folder 'outputdata'.
'dipole.mat' is the result of Green function method. It includes 8 variables: 'E1','ct1','E2','ct2','E3','ct3','E4','ct4'. Ei(i=1~4) is the electric field at point Pi(i=1~4), Ei(:,n) is the value at time cti+(n-1)* 0.0029.
'v1.txt'~'v4.txt' are results of FDTD method. There are two columns in each file. Data of the first column is time (not multiplied by light speed), and the second column is voltage. By dividing the voltage by 0.005 we will get the electric field E_z.

%%

All files/folders: 'conf_dipole.mat', 'dipole_sw.m', 'RunMe.m', 'ReadMe.pdf', 'User Manual.pdf', 'fdtd_sw', 'GF_sw', 'outputdata'.

The file 'conf_dipole.mat' includes the input data for 'RunMe.m'. The file 'dipole_sw.m' is the program to calculate the radiation of an electric dipole by Green function method, and this program will be called by 'RunMe.m'.

The program for FDTD method is placed in the folder 'fdtd_sw'.
The program for Green function method is placed in the folder 'GF_sw'.

The files in the folder 'fdtd_sw':

bound_init.m	: Initialize the FDTD boundary
dipole_add.m	: Add dipole source into the FDTD simulation
dipole_init.m	: Initialize the dipole parameters
epsrmur_sw.m	: The relative permittivity and permeability in Schwarzschild space-time
exc_init.m	: Initialize the excitation
fdtd_sw.m	: Main program
grid_init.m	: Initialize the FDTD grid
ind_init.m	: Convert type numbers to array indices
init.m	: Initialize the FDTD parameters
med_bound.m	: Find the medium parameter on the PML interface (see Fig. 1 in my paper)
OB_conf.m	: Output Boundary configuration
out_init.m	: Initialize the output parameters
out_post.m	: Post-processing for output
output.m	: Output
plane_init.m	: Initialize the output plane
plane_out.m	: Output electric field in the specified plane

plane_read.m : Read electric field in the specified plane from the output file
PML_conf.m : PML configuration
pml_init_sw.m : Initialize the PML parameters
PML_param.m : PML parameter
pml_update_e.m : Update electric field in PML domain
pml_update_h.m : Update magnetic field in PML domain
source_add.m : Add sources into the FDTD simulation
update_e_sw.m : Update electric field in non-PML domain
update_h_sw.m : Update magnetic field in non-PML domain
volt_add.m : Add voltage source into the FDTD simulation
volt_co_init.m : Initialize the coordinate of the voltage source
volt_out.m : Output voltage between the two specified points
wave_fun.m : Waveform function
wave_init.m : Initialize waveform parameters

The files in the folder 'GF_sw':

cart_equator.m : Rotate Cartesian coordinate so that the specified two points fall in the equatorial plane
cart2sph_sw.m : Convert Cartesian coordinate to spherical coordinate
Christoffel_sw.m : Christoffel symbols in spherical coordinate
Christoffel_sw_cart.m : Christoffel symbols in Cartesian coordinate
Contraction.m : Tensor contraction
ContracU2.m : Get $U_{\alpha\beta'}$ from $U_{\alpha\beta';\gamma\delta}$
dE.m : Differential element of electric filed
dot_TT.m : Contraction between two tensors
dot_TV.m : Contraction between a tensor and a vector
dxs_dxc.m : Transform matrix - from spherical coordinate to Cartesian coordinate
EulerTrans.m : Euler transform matrix
geodesic_sw_bvp.m : Solve geodesic equation
GF_sw_cart.m : Green function in Cartesian coordinate
GF_sw_sph.m : Green function in spherical coordinate
Int_null.m : Integral alone the null geodesic
Int_S_up.m : Solve $\sigma_{\alpha\beta'}$
Int_Sp_p.m : Solve $\sigma^{\alpha'}{}_{\beta'}$
Int_Sp_pp.m : Solve $\sigma^{\alpha'}{}_{\beta'\gamma'}$
Int_Su_u.m : Solve $\sigma^{\alpha}{}_{\beta}$
Int_Su_uu.m : Solve $\sigma^{\alpha}{}_{\beta\gamma}$

Int_Su_uuu.m : Solve $\sigma^{\alpha}{}_{\beta\gamma\delta}$

Int_U.m : Solve $U_{\alpha\beta'}$

Int_U_.m : Solve $U_{\alpha\beta'}$ alone the inverse direction

Int_U_p.m : Solve $U_{\alpha\beta';\gamma'}$

Int_U_u.m : Solve $U_{\alpha\beta';\gamma}$

Int_U_uu.m : Solve $U_{\alpha\beta';\gamma\delta}$

Int_V0.m : Solve $V^{0}_{\alpha\beta'}$

inv_metric_sw.m : Inverse metric

L0_s3.m : $\left(\frac{\sigma^{\alpha}{}_{\beta\gamma}}{t}\right)_{t\to 0}$

L0_U1.m : $\left(\frac{U_{\alpha\beta';\gamma}}{t}\right)_{t\to 0}$

metric_sw.m : Metric

prod_tensor.m : Tensor product

Riemann_sw.m : Riemann curvature

RM1_sw.m : The first order covariant derivative of Riemann curvature

RM2_sw.m : The second order covariant derivative of Riemann curvature