2018 狗 年 大 吉

微信公众号：视学算法

```
function varargout = PlaneWaveGUI(varargin)
%
% function varargout = PLANEWAVEGUI(varargin)
%
%   The function PLANEWAVEGUI creates the Graphical User Interface for the
%   TE/TM wave propagation through multilayered structures. See the file
%   'Help.html' or click on the button 'Help' of the GUI for its use.
%
% Copyright: Lorenzo Luini, DEI-Department of Electronics and Information,
%                Politecnico di Milano, Milano, Italy;
%                email: luini@elet.polimi.it
%
% Release: version 1.0, last update: 05-Oct-2007 10:00:00
%
% Remarks: the GUI has been created using GUIDE. It requires Matlab release
%              R2007a to run properly.


% Begin initialization code - DO NOT EDIT
```

视学算法

```
gui_Singleton = 1;

gui_State = struct('gui_Name',       mfilename, ...

                   'gui_Singleton',  gui_Singleton, ...

                   'gui_OpeningFcn', @PlaneWaveGUI_OpeningFcn, ...

                   'gui_OutputFcn',  @PlaneWaveGUI_OutputFcn, ...

                   'gui_LayoutFcn',  @PlaneWaveGUI_LayoutFcn, ...

                   'gui_Callback',   []);

if nargin && ischar(varargin{1})

    gui_State.gui_Callback = str2func(varargin{1});

end




if nargout

    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});

else

    gui_mainfcn(gui_State, varargin{:});

end

% End initialization code - DO NOT EDIT
% --- Executes just before PlaneWaveGUI is made visible.

function PlaneWaveGUI_OpeningFcn(hObject, eventdata, handles, varargin)



% Choose default command line output for PlaneWaveGUI

handles.output = hObject;



% Update handles structure

guidata(hObject, handles);



% Run the script with default values after initialization



% Retrieve input data from the GUI

teta=get(handles.edit5);

teta=str2num(teta.String);



freq=get(handles.edit1);

freq=str2num(freq.String);
```

```
er=get(handles.edit2);

er=str2num(er.String);



mr=get(handles.edit3);

mr=str2num(mr.String);



sigma=get(handles.edit4);

sigma=str2num(sigma.String);



Zlay=get(handles.edit6);

Zlay=str2num(Zlay.String);

Zlay=[0 Zlay];



fH(1)=handles.axes1;

fH(2)=handles.axes4;

fH(3)=handles.axes5;
```

念学算法

```
fH(4)=handles.axes6;



A0=get(handles.edit7);

A0=str2num(A0.String);



anim=get(handles.radiobutton1);

anim=anim.Value;



RealorAbs=get(handles.popupmenu1,'Value');

TETM=get(handles.popupmenu2,'Value');



zval=get(handles.edit9);

zval=str2num(zval.String);



nsam=get(handles.edit10);

nsam=str2num(nsam.String);
```

渊学算法

```
% Define an alias of the function
PlaneW=@PlaneWave_MultLayer;


% Check if input data are correct
ResInp=CheckInputs(teta,freq,er,mr,sigma,Zlay,3,0,fH,A0,zval,zval,nsam,anim,RealorAbs,TETM,0);


% If check is passed, run the code
if ResInp==1
    PlaneWave_MultLayer(teta,freq,er,mr,sigma,Zlay,3,0,fH,A0,zval,zval,nsam,anim,RealorAbs,TETM,0)
end


% --- Outputs from this function are returned to the command line.
function varargout = PlaneWaveGUI_OutputFcn(hObject, eventdata, handles)
% varargout   cell array for returning output args (see VARARGOUT);
```

视学算法

```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Get default command line output from handles structure
varargout{1} = handles.output;



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double
% --- Executes during object creation, after setting all properties.

function edit1_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit1 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called




% Hint: edit controls usually have a white background on Windows.

%       See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

    set(hObject,'BackgroundColor','white');

end





function edit2_Callback(hObject, eventdata, handles)

% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double




% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end




function edit3_Callback(hObject, eventdata, handles)

% hObject    handle to edit3 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)



% Hints: get(hObject,'String') returns contents of edit3 as text

%        str2double(get(hObject,'String')) returns contents of edit3 as a double




% --- Executes during object creation, after setting all properties.

function edit3_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit3 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called
```

视觉算法

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double
% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double



% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called



% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a double


% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

视学算法

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end




% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Retrieve input data from the GUI
teta=get(handles.edit5);
teta=str2num(teta.String);
```

```
freq=get(handles.edit1);

freq=str2num(freq.String);



er=get(handles.edit2);

er=str2num(er.String);



mr=get(handles.edit3);

mr=str2num(mr.String);



sigma=get(handles.edit4);

sigma=str2num(sigma.String);



Zlay=get(handles.edit6);

Zlay=str2num(Zlay.String);

Zlay=[0 Zlay];
```

桃学算法

```
fH(1)=handles.axes1;

fH(2)=handles.axes4;

fH(3)=handles.axes5;

fH(4)=handles.axes6;



A0=get(handles.edit7);

A0=str2num(A0.String);



anim=get(handles.radiobutton1);

anim=anim.Value;



RealorAbs=get(handles.popupmenu1,'Value');

TETM=get(handles.popupmenu2,'Value');



zval=get(handles.edit9);

zval=str2num(zval.String);
```

```
nsam=get(handles.edit10);
nsam=str2num(nsam.String);


Calculations=get(handles.radiobutton2);
Calculations=Calculations.Value;


% Define an alias of the function
PlaneW=@PlaneWave_MultLayer;


% Check if input data are correct
ResInp=CheckInputs(teta,freq,er,mr,sigma,Zlay,3,0,fH,A0,zval,zval,nsam,anim,RealorAbs,TETM,Calculations);


% If check is passed, run the code
if ResInp==1
    PlaneWave_MultLayer(teta,freq,er,mr,sigma,Zlay,3,0,fH,A0,zval,zval,nsam,anim,RealorAbs,TETM,Calculations)
end
```

光学算法

```
%%%%%%%%%%%%%%%%%%%%%%%%% USER DEFINED
FUNCTIONS %%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%% Main function of the whole
GUI %%%%%%%%%%%%%%%%%%%%%%
function
PlaneWave_MultLayer(teta,freq,er,mr,sigma,Zlay,what,ev,fH,A0,yrange,zrange,ns,ani
m,RoA,TE_TM,Calc)
%
% function
PlaneWave_MultLayer(teta,freq,er,mr,sigma,Zlay,what,ev,fH,A0,yrange,zrange,ns,ani
m,RoA,TE_TM,Calc)
%
% The function implements a TE/TM plane wave with multiple layers in the yz
% plane (not dependent on x). Respectively the electric and the magnetic
% fields are calculated. Both ordinary materials and metamaterials can
% be introduced. The temporal convention implemented is exp(j*omega*t):
% imaginary parts of the permittivity and/or of the permeability has to be
% negative: eps=eps0.*(epsr_real-j*epsr_imag)-j*sigma./omega (sigma is
% obiviously positive).

%

% INPUT

%    teta = angle between the propagation direction and the normal to the

%            y-axis [癩

%    freq = frequency [Hz]

%    er = vector of relative electric permittivity

%    mr = vector of relative magnetic permeability

%    sigma = vector of conductivity [S/m]

%    Zlay = vector of the z-ccordinates of the interfaces between materials;

%            the first element must always be 0 m [m]. Interfaces must be

%            defined incrementally negative.

%    fH = handles of the GUI axes on which results are plotted

%    A0 = amplitude of the incidente wave [V/m] (TE) or [A/m] (TM)

%    yrange/zrange = limits of the y/z axis [m]

%    ns = number of samples of each axis

%    anim = 1 --> animate field

%            0 --> do not animate field

%    RoA = 1 --> plot real values

%          2 --> plot absolute values

%    TE_TM = 1 --> TE wave

%             2 --> TM wave
%    what = it must be set to 3

%    ev = it must be set to 0

%

% By: L.Luini

% Release: 28.VIII.2007



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%                              LAYERS

PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Additional option to be set inside the code

% last layer type: 1) indefinite dielectric or conductor --> 0

%                        2) PMC -->   1

%                        3) PEC --> -1

LastLay=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



global CalcDetails

global risp

global Lim1

global Lim2
```

```
ym=yrange(1);

yM=yrange(2);

zm=zrange(1);

zM=zrange(2);



pushbutton2_Callback



Nl=length(er);    % number of layers



% environment

delta=(max([zM yM])-min([zm ym]))/ns;    % spatial sampling [m]

x=zeros(1,ns);

y=[ym:delta:yM];

z=[zm:delta:zM];



% wave numbers and reflection coefficients
```

```
v0=3e8;
mu0=pi*4e-7;
eps0=1/(v0^2*mu0);
omega=2*pi*freq;
eps=eps0*er;
mu=mu0*mr;
kq=omega^2.*mu.*(eps-j.*sigma/omega);
k=sqrt(kq);
beta=real(k);
alfa=-imag(k);
wl=(2*pi)./real(k);


% define ky based on the input choice
if ev==0
    ky=real(k(1))*sind(teta);
elseif ev==1
    ky=teta*k(1);
end


for q=1:Nl   % for all layers
```

```
    if real(er(q))<0&real(mr(q))<0
        if sigma(q)==0&(ky.^2>kq(q))
            kz(q)=sqrt(kq(q)-ky.^2);
        else
            kz(q)=-sqrt(kq(q)-ky.^2);
        end
    else
        kz(q)=sqrt(kq(q)-ky.^2);
    end
    yTE(q)=kz(q)./(omega.*mu(q));
    zTE(q)=1./yTE(q);
    yTM(q)=(omega.*(eps(q)-j*sigma(q)/omega))./kz(q);
    zTM(q)=1./yTM(q);
end


if LastLay==0   % the last layer is an indefinite dielectric/conductor
    gammaTEdx(Nl-1)=(zTE(end)-zTE(end-1))./(zTE(end)+zTE(end-1));
    gammaTMdx(Nl-1)=-(zTM(end)-zTM(end-1))./(zTM(end)+zTM(end-1));
elseif LastLay==-1   % the last layer is a PEC (electric gamma=-1)
    gammaTEdx(Nl-1)=-1;
    gammaTMdx(Nl-1)=1;
```

视学算法

```
elseif LastLay==1    % the last layer is a PMC (electric gamma=1)
    gammaTEdx(Nl-1)=1;
    gammaTMdx(Nl-1)=-1;
end
for q=Nl-1:-1:2    % for all layers
    gammaTEsx(q)=gammaTEdx(q).*exp(-j.*2.*kz(q).*(-(Zlay(q+1)-Zlay(q))));
    gammaTMsx(q)=gammaTMdx(q).*exp(-j.*2.*kz(q).*(-(Zlay(q+1)-Zlay(q))));
    zTEL(q)=zTE(q).*(1+gammaTEsx(q))./(1-gammaTEsx(q));
    zTML(q)=zTM(q).*(1+gammaTMsx(q))./(1-gammaTMsx(q));
    gammaTEdx(q-1)=(zTEL(q)-zTE(q-1))./(zTEL(q)+zTE(q-1));
    gammaTMdx(q-1)=(zTML(q)-zTM(q-1))./(zTML(q)+zTM(q-1));
end


if TE_TM==1    % TE wave


    clear J;
    J=find(z>=0);
    % primary field (only on plane yz, with z >= 0)
    EpTE(J,:)=A0.*exp(-j*kz(1)*(-z(J)).')*exp(-j*ky*y);
    ApEsTEDx=A0.*exp(-j*ky*y);
```

视学算法

```
EsTE_progr_only(J,:)=EpTE(J,:);



clear J;

J=find(z<0);

EpTE(J,:)=zeros(length(J),length(y));



% reflected field for layer 1

clear J;

J=find(z>=0);

znow=z(J);



% TE case

EsTE(J,:)=exp(-j*kz(1)*(znow).')*ApEsTEDx*gammaTEdx(1);



% save only regressive field in layer 1

EsTE_regr_only(J,:)=EsTE(J,:);
```

视学算法

```
for q=2:Nl-1    % progressive and regressive fields for layers 2 --> N-1
    clear J;
    J=find(z<Zlay(q)&z>=Zlay(q+1));
    znow=z(J);


    % TE case
    ApEsTESx=ApEsTEDx.*(1+gammaTEdx(q-1))./(1+gammaTEsx(q));
    EsTE(J,:)=exp(-j*kz(q)*(-(znow.'-Zlay(q))))*ApEsTESx;


    % save only progressive field in all intermediate layers
    EsTE_progr_only(J,:)=EsTE(J,:);


    ApEsTEDx=ApEsTESx.*exp(-j*kz(q)*(-(Zlay(q+1)-Zlay(q))));
    AmEsTEDx=ApEsTEDx.*gammaTEdx(q);
    EsTE(J,:)=EsTE(J,:)+exp(-j*kz(q)*((znow.'-Zlay(q+1))))*AmEsTEDx;
```

```
        % save only regressive field in all intermediate layers

        EsTE_regr_only(J,:)=exp(-j*kz(q)*((znow.'-Zlay(q+1))))*AmEsTEDx;

    end




    % only progressive field in the last layer

    clear J;

    J=find(z<Zlay(end));

    znow=z(J);



    if LastLay==1|LastLay==-1

        EsTE(J,:)=0;

    else

        % TE case

         EsTE(J,:)=exp(-j*kz(end)*(-(znow.'-Zlay(end))))*ApEsTEDx*(1+gammaTEdx(end));



        % save only progressive field in the last layer

        EsTE_progr_only(J,:)=EsTE(J,:);

    end
```

润学算法

```
elseif TE_TM==2    % TM wave

    clear J;

    J=find(z>=0);

    % primary field (only on plane yz, with z >= 0)

    EpTE(J,:)=A0.*exp(-j*kz(1)*(-z(J)).')*exp(-j*ky*y);

    ApEsTEDx=A0.*exp(-j*ky*y);




    EsTE_progr_only(J,:)=EpTE(J,:);




    clear J;

    J=find(z<0);

    EpTE(J,:)=zeros(length(J),length(y));



    % reflected field for layer 1

    clear J;

    J=find(z>=0);

    znow=z(J);
```

桃学算法

```
% TE case
EsTE(J,:)=exp(-j*kz(1)*(znow).')*ApEsTEDx*gammaTMdx(1);


% save only regressive field in layer 1
EsTE_regr_only(J,:)=EsTE(J,:);


for q=2:Nl-1    % progressive and regressive fields for layers 2 --> N-1
    clear J;
    J=find(z<Zlay(q)&z>=Zlay(q+1));
    znow=z(J);

    % TE case
    ApEsTESx=ApEsTEDx.*(1+gammaTMdx(q-1))./(1+gammaTMsx(q));
    EsTE(J,:)=exp(-j*kz(q)*(-(znow.'-Zlay(q))))*ApEsTESx;


    % save only progressive field in all intermediate layers
    EsTE_progr_only(J,:)=EsTE(J,:);
    ApEsTEDx=ApEsTESx.*exp(-j*kz(q)*(-(Zlay(q+1)-Zlay(q))));

    AmEsTEDx=ApEsTEDx.*gammaTMdx(q);

    EsTE(J,:)=EsTE(J,:)+exp(-j*kz(q)*((znow.'-Zlay(q+1))))*AmEsTEDx;



    % save only regressive field in all intermediate layers

    EsTE_regr_only(J,:)=exp(-j*kz(q)*((znow.'-Zlay(q+1))))*AmEsTEDx;

end



% only progressive field in the last layer

clear J;

J=find(z<Zlay(end));

znow=z(J);



if LastLay==1|LastLay==-1

    EsTE(J,:)=0;

else

    % TE case

     EsTE(J,:)=exp(-j*kz(end)*(-(znow.'-Zlay(end))))*ApEsTEDx*(1+gammaTMdx(e
```

视学算法

```
nd));



        % save only progressive field in the last layer

        EsTE_progr_only(J,:)=EsTE(J,:);

    end

end



if what==3

    % sum fields

    EtTE=EpTE+EsTE;



    if RoA==1    % plot real values

        if anim==1    % animate fields (only vertical cut)



            FieldT=EtTE;

            FieldP=EsTE_progr_only;

            FieldR=EsTE_regr_only;
```

```
axes(fH(1))

imagesc(y,z,real(FieldT))

xlabel('y [m]')

ylabel('z [m]')

set(gca,'YDir','normal')

axis square

hold on;

for q=2:Nl

    h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);

    set(h,'Color','k','LineWidth',1)

end

colorbar

hold off;

if TE_TM==1

    title('Total field E_x')

elseif TE_TM==2

    title('Total field H_x')

end




axes(fH(2))
```

科学算法

```
imagesc(y,z,real(FieldP))
xlabel('y [m]')
ylabel('z [m]')
set(gca,'YDir','normal')
axis square
hold on;
for q=2:Nl
    h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);
    set(h,'Color','k','LineWidth',1)
end
colorbar
hold off;
if TE_TM==1
    title('Progressive field E_x')
elseif TE_TM==2
    title('Regressive field H_x')
end


axes(fH(3))
imagesc(y,z,real(FieldR))
xlabel('y [m]')
```

学算法

```
ylabel('z [m]')

set(gca,'YDir','normal')

axis square

hold on;

for q=2:Nl

    h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);

    set(h,'Color','k','LineWidth',1)

end

colorbar

hold off;

if TE_TM==1

    title('Regressive field E_x')

elseif TE_TM==2

    title('Regressive field H_x')

end



% movie

tv=0;

dT=1./(32.*freq);

risp=1;

while risp==1
```

```
tv=tv+dT;
axes(fH(4))
% plot the field for a fixed value of y (center of the axis) REAL
plot(z,real(EtTE(:,round(ns/2)).*exp(j.*omega.*tv)),'LineWidth',1.5)
axis([z(1) z(end) -2*A0 2*A0])
hold on;
for q=2:Nl
    h=line([Zlay(q) Zlay(q)],[-2*A0 2*A0]);
    set(h,'Color','k','LineWidth',1)
end
xlabel('z [m]')
if TE_TM==1
    ylabel('real(E) [V/m]')
    title('E_x field for y = 0 m')
elseif TE_TM==2
    ylabel('real(H) [A/m]')
    title('H_x field for y = 0 m')
end
set(gca,'XDir','Reverse')
hold off;
```

```
    end
else    % static fields
    FieldT=EtTE;
    FieldP=EsTE_progr_only;
    FieldR=EsTE_regr_only;


    axes(fH(1))
    imagesc(y,z,real(FieldT))
    xlabel('y [m]')
    ylabel('z [m]')
    set(gca,'YDir','normal')
    axis square
    hold on;
    for q=2:Nl
        h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);
        set(h,'Color','k','LineWidth',1)
    end
    colorbar
    hold off;
    if TE_TM==1
        title('Total field E_x')
```

```
elseif TE_TM==2
    title('Total field H_x')
end



axes(fH(2))
imagesc(y,z,real(FieldP))
xlabel('y [m]')
ylabel('z [m]')
set(gca,'YDir','normal')
axis square
hold on;
for q=2:Nl
    h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);
    set(h,'Color','k','LineWidth',1)
end
colorbar
hold off;
if TE_TM==1
    title('Progressive field E_x')
elseif TE_TM==2
    title('Regressive field H_x')
```

```
end



axes(fH(3))

imagesc(y,z,real(FieldR))

xlabel('y [m]')

ylabel('z [m]')

set(gca,'YDir','normal')

axis square

hold on;

for q=2:Nl

    h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);

    set(h,'Color','k','LineWidth',1)

end

colorbar

hold off;

if TE_TM==1

    title('Regressive field E_x')

elseif TE_TM==2

    title('Regressive field H_x')

end
```

```
        axes(fH(4))

        % plot the field for a fixed value of y (center of the axis) REAL

        plot(z,real(EtTE(:,round(ns/2))),'LineWidth',1.5)

        axis([z(1) z(end) -2*A0 2*A0])

        hold on;

        for q=2:Nl

            h=line([Zlay(q) Zlay(q)],[-2*A0 2*A0]);

            set(h,'Color','k','LineWidth',1)

        end

        xlabel('z [m]')

        if TE_TM==1

            ylabel('real(E) [V/m]')

            title('E_x field for y = 0 m')

        elseif TE_TM==2

            ylabel('real(H) [A/m]')

            title('H_x field for y = 0 m')

        end

        set(gca,'XDir','Reverse')

        hold off;

    end

else    % plot absolute values
```

```
if anim==1    % animate fields (only vertical cut)
    FieldT=EtTE;
    FieldP=EsTE_progr_only;
    FieldR=EsTE_regr_only;

    axes(fH(1))
    imagesc(y,z,abs(FieldT))
    xlabel('y [m]')
    ylabel('z [m]')
    set(gca,'YDir','normal')
    axis square
    hold on;
    for q=2:Nl
        h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);
        set(h,'Color','k','LineWidth',1)
    end
    colorbar
    hold off;
    if TE_TM==1
        title('Total field E_x')
    elseif TE_TM==2
```

数学算法

```
    title('Total field H_x')
end


axes(fH(2))
imagesc(y,z,abs(FieldP))
xlabel('y [m]')
ylabel('z [m]')
set(gca,'YDir','normal')
axis square
hold on;
for q=2:Nl
    h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);
    set(h,'Color','k','LineWidth',1)
end
colorbar
hold off;
if TE_TM==1
    title('Progressive field E_x')
elseif TE_TM==2
    title('Progressive field H_x')
end
```

光学算法

```
axes(fH(3))

imagesc(y,z,abs(FieldR))

xlabel('y [m]')

ylabel('z [m]')

set(gca,'YDir','normal')

axis square

hold on;

for q=2:NI

    h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);

    set(h,'Color','k','LineWidth',1)

end

colorbar

hold off;

if TE_TM==1

    title('Regressive field E_x')

elseif TE_TM==2

    title('Regressive field H_x')

end
```

学算法

```
% movie
tv=0;
dT=1./(32.*freq);
risp=1;
while risp==1
    tv=tv+dT;
    axes(fH(4))
    % plot the field for a fixed value of y (center of the
    % axis) ABS
    plot(z,abs(EtTE(:,round(ns/2)).*exp(j.*omega.*tv)),'LineWidth',1.5)
    axis([z(1) z(end) 0 2*A0])
    hold on;
    for q=2:Nl
        h=line([Zlay(q) Zlay(q)],[-2*A0 2*A0]);
        set(h,'Color','k','LineWidth',1)
    end
    xlabel('z [m]')
    if TE_TM==1
        ylabel('abs(E) [V/m]')
        title('E_x field for y = 0 m')
    elseif TE_TM==2
        ylabel('abs(H) [A/m]')
```

```
            title('H_x field for y = 0 m')
        end
        set(gca,'XDir','Reverse')
        hold off;
    end
else    % static fields
    FieldT=EtTE;
    FieldP=EsTE_progr_only;
    FieldR=EsTE_regr_only;


    axes(fH(1))
    imagesc(y,z,abs(FieldT))
    xlabel('y [m]')
    ylabel('z [m]')
    set(gca,'YDir','normal')
    axis square
    hold on;
    for q=2:Nl
        h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);
        set(h,'Color','k','LineWidth',1)
    end
```

```
colorbar
hold off;
if TE_TM==1
    title('Total field E_x')
elseif TE_TM==2
    title('Total field H_x')
end


axes(fH(2))
imagesc(y,z,abs(FieldP))
xlabel('y [m]')
ylabel('z [m]')
set(gca,'YDir','normal')
axis square
hold on;
for q=2:Nl
    h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);
    set(h,'Color','k','LineWidth',1)
end
colorbar
hold off;
```

```
if TE_TM==1

    title('Progressive field E_x')

elseif TE_TM==2

    title('Progressive field H_x')

end




axes(fH(3))

imagesc(y,z,abs(FieldR))

xlabel('y [m]')

ylabel('z [m]')

set(gca,'YDir','normal')

axis square

hold on;

for q=2:Nl

    h=line([y(1) y(end)],[Zlay(q) Zlay(q)]);

    set(h,'Color','k','LineWidth',1)

end

colorbar

hold off;

if TE_TM==1

    title('Regressive field E_x')
```

学算法

```
elseif TE_TM==2
    title('Regressive field H_x')
end


axes(fH(4))
% plot the field for a fixed value of y (center of the axis)
% ABS
plot(z,abs(EtTE(:,round(ns/2))),'LineWidth',1.5)
axis([z(1) z(end) 0 2*A0])
hold on;
for q=2:Nl
    h=line([Zlay(q) Zlay(q)],[-2*A0 2*A0]);
    set(h,'Color','k','LineWidth',1)
end
xlabel('z [m]')
if TE_TM==1
    ylabel('abs(E) [V/m]')
    title('E_x field for y = 0 m')
elseif TE_TM==2
    ylabel('abs(H) [A/m]')
    title('H_x field for y = 0 m')
```

```
            end
            set(gca,'XDir','Reverse')
            hold off;
        end
    end
end



% plot calculation details if requested
if Calc==1
    % plot info about the layers (beta, alfa, wavelength, gamma)
    %%%%%%%% PLOT PARAMETERS (TO MOVE BOXES EASILY) %%%%%%%%
    deltaXfig=0.233;
    deltaYfig=-0.42;
    Line1_x=[0.035 0.27];
    Line1_y=[0.84 0.84];
    Line2_x=[0.035 0.27];
    Line2_y=[0.73 0.73];
    Ell1_x=0.266;
    Ell1_y=0.836;
    Ell2_x=0.266;
    Ell2_y=0.726;
```

```
Text1_x=0.045;

Text1_y=0.64;

Text2_x=0.045;

Text2_y=0.525;


% the first variable indicates the maximum number of line sections

% plotted on the same line; the second one indicates the maximum number

% of line sections plotted on the same figure

Lim1=3;

Lim2=6;


% colors of the line sections

colors=['b';'k';'r';'b';'k';'r';'b';'k';'r';'b';'k';'r'];

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%




% plot 'Working' message

hdlg=helpdlg('Working...please wait','Information');




% set the figure according to the screen resolution
```

```
scrsz=get(0,'ScreenSize');



 % Old vector for 'Position' option of the figure opened below, with 'Units','pixels'  --> [120 1 1000 650]
CalcDetails1=figure('Name','Calculation details','Units','normalized','Position',[0.05 0.1 0.95 0.8]);



% begin plot on the first figure
CalcDetails=CalcDetails1;

% title
annotation(CalcDetails1,'textbox',...
    'String',{'Calculation details: transmission line model'},...
    'HorizontalAlignment','center',...
    'FontSize',16,...
    'FontName','Verdana',...
    'FitHeightToText','off',...
    'LineStyle','none',...
    'Position',[0.2116 0.895 0.5915 0.09662]);
```

数学算法

```
% Create line

annotation(CalcDetails1,'line',Line1_x,Line1_y,'Color',colors(1,:),'LineWidth',2);



% Create line

annotation(CalcDetails1,'line',Line2_x,Line2_y,'Color',colors(1,:),'LineWidth',2);



% Create line (layers discriminator)

annotation(CalcDetails1,'line',[Line1_x(2) Line1_x(2)],[Line1_y(2) Line2_y(2)],'Color',colors(1,:),'LineWidth',2,'LineStyle','--');



% Create line (relative to gamma -)

annotation(CalcDetails1,'line',[Line1_x(2) Line1_x(2)]-0.01,[Line1_y(2)-0.05 Line2_y(2)]+0.1,'Color',colors(1,:),'LineWidth',2);



% Create line (relative to gamma +)

annotation(CalcDetails1,'line',[Line1_x(2) Line1_x(2)]+0.01,[Line1_y(2)-0.08 Line2_y(2)]+0.1,'Color',colors(2,:),'LineWidth',2);
```

```
% Create textbox (relative to gamma -)

            annotation(CalcDetails1,'textbox','String',{['\Gamma_-          =
',num2str(gammaTEdx(1),'%2.3g')]},...

    'FontSize',11,...

    'FontWeight','demi',...

    'Color',colors(1,:),...

    'FontName','Arial',...

    'FitHeightToText','off',...

    'EdgeColor','none',...

    'Position',[Text1_x+0.203 Text1_y+0.244 0.25 0.05]);



% Create ellipse

annotation(CalcDetails1,'ellipse','FaceColor',colors(1,:),...

    'Position',[Ell1_x Ell1_y 0.008 0.008],'Color',colors(1,:));



% Create ellipse

annotation(CalcDetails1,'ellipse','FaceColor',colors(1,:),...

    'Position',[Ell2_x Ell2_y 0.008 0.008],'Color',colors(1,:));
    if TE_TM==1    % TE wave
        % Create textbox
                                   annotation(CalcDetails1,'textbox','String',{['\epsilon_{r1}       =
',num2str(er(1),'%2.3g')],['\mu_{r1}     =     ',num2str(mr(1),'%2.3g')],['\sigma_{1}     =
',num2str(sigma(1),'%2.3g'),' S/m']},...
            'FontSize',11,...
            'FontWeight','demi',...
            'Color',colors(1,:),...
            'FontName','Arial',...
            'FitHeightToText','off',...
            'EdgeColor','none',...
            'Position',[Text1_x Text1_y 0.2 0.2]);
    else
        % Create textbox
                                   annotation(CalcDetails1,'textbox','String',{['\epsilon_{r1}       =
',num2str(er(1),'%2.3g')],['\mu_{r1}     =     ',num2str(mr(1),'%2.3g')],['\sigma_{1}     =
',num2str(sigma(1),'%2.3g'),' S/m']},...
            'FontSize',11,...
            'FontWeight','demi',...
            'Color',colors(1,:),...
            'FontName','Arial',...
            'FitHeightToText','off',...
            'EdgeColor','none',...
            'Position',[Text1_x Text1_y 0.2 0.2]);
    end



    if TE_TM==1    % TE wave
        % Create textbox
        annotation(CalcDetails1,'textbox',...
              'String',{['\eta_{TE} = ',num2str(zTE(1),'%2.3g'),' \Omega'],['\lambda = ',num2str(wl(1),'%2.3g'),' m'],['\beta = ',num2str(beta(1),'%2.3g'),' rad/m'],['\alpha = ',num2str(alfa(1),'%2.3g'),' Np/m']},...
            'FontWeight','demi',...
            'FontSize',11,...
            'Color',colors(1,:),...
            'FontName','Arial',...
            'FitHeightToText','on',...
            'EdgeColor','none',...
            'LineWidth',2,...
            'Position',[Text2_x Text2_y 0.2 0.2]);
    else
```

```
        % Create textbox
        annotation(CalcDetails1,'textbox',...
            'String',{['\eta_{TM} = ',num2str(zTM(1),'%2.3g'),' \Omega'],['\lambda = ',num2str(wl(1),'%2.3g'),' m'],['\beta = ',num2str(beta(1),'%2.3g'),' rad/m'],['\alpha = ',num2str(alfa(1),'%2.3g'),' Np/m']},...
            'FontWeight','demi',...
            'FontSize',11,...
            'Color',colors(1,:),...
            'FontName','Arial',...
            'FitHeightToText','on',...
            'EdgeColor','none',...
            'LineWidth',2,...
            'Position',[Text2_x Text2_y 0.2 0.2]);
    end


    deltaX=0;
    for q=2:Nl-1
        if q<=Lim2    % in the first figure
            CalcDetails=CalcDetails1;
            if q<=Lim1    % first line in the figure
                deltaY=0;
```

视学算法

```
                if q==1;
                    deltaX=0;
                else
                    deltaX=deltaX+deltaXfig;
                end
            else    % second line in the figure
                if q==Lim1+1;
                    deltaX=0;
                else
                    deltaX=deltaX+deltaXfig;
                end
                deltaY=deltaYfig;
            end
        else    % second figure
            if ~exist('CalcDetails2','var')
                CalcDetails2=figure('Name','Calculation details','Units','normalized','Position',[0.05 0.1 0.95 0.8]);
            end
            CalcDetails=CalcDetails2;
            if q<=Lim2+Lim1    % first line in the figure
                deltaY=0;
                if q==Lim2+1;
```

```
                    deltaX=0;

                else

                    deltaX=deltaX+deltaXfig;

                end

            else    % second line in the figure

                if q==Lim2+Lim1+1;

                    deltaX=0;

                else

                    deltaX=deltaX+deltaXfig;

                end

                deltaY=deltaYfig;

            end

        end



        % Create line

         annotation(CalcDetails,'line',Line1_x+deltaX,Line1_y+deltaY,'Color',colors(q,:
),'LineWidth',2);



        % Create line

         annotation(CalcDetails,'line',Line2_x+deltaX,Line2_y+deltaY,'Color',colors(q,:
```

```
),'LineWidth',2);



    % Create line

        annotation(CalcDetails,'line',[Line1_x(2)    Line1_x(2)]+deltaX,[Line1_y(2)
Line2_y(2)]+deltaY,'Color',colors(q,:),'LineWidth',2,'LineStyle','--');



    % Create line (relative to gamma -)

                                                        annotation(CalcDetails,'line',[Line1_x(2)
Line1_x(2)]+deltaX-0.01,[Line1_y(2)-0.05
Line2_y(2)]+deltaY+0.1,'Color',colors(q,:),'LineWidth',2);



    % Create line (relative to gamma +)

    %       if q==Lim1

    %           deltaXdiff=0-(Line1_x(2)-Line1_x(1));

    %           deltaYdiff=deltaYfig;

    %       else

    %           deltaXdiff=deltaX;

    %           deltaYdiff=deltaY;

    %       end
    if q<Lim2    % first figure
        CalcDetailsDiff=CalcDetails1;
        if q==Lim1
            deltaXdiff=0-(Line1_x(2)-Line1_x(1));
            deltaYdiff=deltaYfig;
        else
            deltaXdiff=deltaX;
            deltaYdiff=deltaY;
        end
    elseif q==Lim2
        if ~exist('CalcDetails2','var')
            CalcDetails2=figure('Name','Calculation details','Units','normalized','Position',[0.05 0.1 0.95 0.8]);
        end
        CalcDetailsDiff=CalcDetails2;
        deltaXdiff=0-(Line1_x(2)-Line1_x(1));
        deltaYdiff=0;
    else    % second figure
        CalcDetailsDiff=CalcDetails2;
        if q==Lim1+Lim2
```

```
            deltaXdiff=0-(Line1_x(2)-Line1_x(1));

            deltaYdiff=deltaYfig;

        else

            deltaXdiff=deltaX;

            deltaYdiff=deltaY;

        end

    end



                                                  annotation(CalcDetailsDiff,'line',[Line1_x(2)

Line1_x(2)]+deltaXdiff+0.01,[Line1_y(2)-0.08

Line2_y(2)]+deltaYdiff+0.1,'Color',colors(q+1,:),'LineWidth',2);



    % Create textbox (relative to gamma -)

                                annotation(CalcDetails,'textbox','String',{['\Gamma_-        =

',num2str(gammaTEdx(q),'%2.3g')]},...

        'FontSize',11,...

        'FontWeight','demi',...

        'Color',colors(q,:),...

        'FontName','Arial',...

        'FitHeightToText','off',...
        'EdgeColor','none',...

        'Position',[Text1_x+0.203+deltaX Text1_y+0.244+deltaY 0.25 0.05]);



    % Create textbox (relative to gamma +)

                        annotation(CalcDetails,'textbox','String',{['\Gamma_+        =
',num2str(gammaTEsx(q),'%2.3g')]},...

        'FontSize',11,...

        'FontWeight','demi',...

        'Color',colors(q,:),...

        'FontName','Arial',...

        'FitHeightToText','off',...

        'EdgeColor','none',...

        'Position',[Text1_x-0.01+deltaX Text1_y+0.215+deltaY 0.25 0.05]);



    % Create ellipse

    annotation(CalcDetails,'ellipse','FaceColor',colors(q,:),...

        'Position',[Ell1_x+deltaX Ell1_y+deltaY 0.008 0.008],'Color',colors(q,:));



    % Create ellipse
```

```
        annotation(CalcDetails,'ellipse','FaceColor',colors(q,:),...
            'Position',[Ell2_x+deltaX Ell2_y+deltaY 0.008 0.008],'Color',colors(q,:));



        if TE_TM==1    % TE wave
            % Create textbox
                annotation(CalcDetails,'textbox','String',{['\epsilon_{r',num2str(q),'} = ',num2str(er(q),'%2.3g')],['\mu_{r',num2str(q),'} = ',num2str(mr(q),'%2.3g')],['\sigma_{',num2str(q),'} = ',num2str(sigma(q),'%2.3g'),' S/m']},...
                'FontSize',11,...
                'FontWeight','demi',...
                'Color',colors(q,:),...
                'FontName','Arial',...
                'FitHeightToText','off',...
                'EdgeColor','none',...
                'Position',[Text1_x+deltaX Text1_y+deltaY 0.2 0.2]);
        else
            % Create textbox
                annotation(CalcDetails,'textbox','String',{['\epsilon_{r',num2str(q),'} = ',num2str(er(q),'%2.3g')],['\mu_{r',num2str(q),'} = ',num2str(mr(q),'%2.3g')],['\sigma_{',num2str(q),'} = ',num2str(sigma(q),'%2.3g'),'
S/m']},...
                'FontSize',11,...
                'FontWeight','demi',...
                'Color',colors(q,:),...
                'FontName','Arial',...
                'FitHeightToText','off',...
                'EdgeColor','none',...
                'Position',[Text1_x+deltaX Text1_y+deltaY 0.2 0.2]);
        end



        if TE_TM==1    % TE wave
            % Create textbox
            annotation(CalcDetails,'textbox',...
                'String',{['\eta_{TE} = ',num2str(zTE(q),'%2.3g'),' \Omega'],['\lambda = ',num2str(wl(q),'%2.3g'),' m'],['\beta = ',num2str(beta(q),'%2.3g'),' rad/m'],['\alpha = ',num2str(alfa(q),'%2.3g'),' Np/m']},...
                'FontWeight','demi',...
                'FontSize',11,...
                'Color',colors(q,:),...
                'FontName','Arial',...
                'FitHeightToText','on',...
                'EdgeColor','none',...
                'LineWidth',2,...
                'Position',[Text2_x+deltaX Text2_y+deltaY 0.2 0.2]);
        else
            % Create textbox
            annotation(CalcDetails,'textbox',...
                'String',{['\eta_{TM} = ',num2str(zTM(q),'%2.3g'),' \Omega'],['\lambda = ',num2str(wl(q),'%2.3g'),' m'],['\beta = ',num2str(beta(q),'%2.3g'),' rad/m'],['\alpha = ',num2str(alfa(q),'%2.3g'),' Np/m']},...
                'FontWeight','demi',...
                'FontSize',11,...
                'Color',colors(q,:),...
                'FontName','Arial',...
                'FitHeightToText','on',...
                'EdgeColor','none',...
                'LineWidth',2,...
                'Position',[Text2_x+deltaX Text2_y+deltaY 0.2 0.2]);
        end


    end
```

```
% plot last layer
if Nl<=Lim1    % change line in the figure
    deltaX=deltaX+deltaXfig;
    deltaY=0;
elseif Nl<=Lim2
    if Nl==Lim1+1;
        deltaX=0;
    else
        deltaX=deltaX+deltaXfig;
    end
    deltaY=deltaYfig;
elseif Nl<=Lim2+Lim1
    deltaX=deltaX+deltaXfig;
    deltaY=0;
elseif Nl<=Lim2+Lim2
    if Nl==Lim1+1;
        deltaX=0;
    else
        deltaX=deltaX+deltaXfig;
    end
    deltaY=deltaYfig;
```

数学算法

```
    end



    % Create line

    annotation(CalcDetails,'line',Line1_x+deltaX,Line1_y+deltaY,'Color',colors(Nl,:),'LineWidth',2);



    % Create line

    annotation(CalcDetails,'line',Line2_x+deltaX,Line2_y+deltaY,'Color',colors(Nl,:),'LineWidth',2);



    % Create textbox (relative to gamma +)

    annotation(CalcDetails,'textbox','String',{['\Gamma_+ = 0']},...

        'FontSize',11,...

        'FontWeight','demi',...

        'Color',colors(Nl,:),...

        'FontName','Arial',...

        'FitHeightToText','off',...

        'EdgeColor','none',...

        'Position',[Text1_x-0.01+deltaX Text1_y+0.215+deltaY 0.25 0.05]);
```

```
%     % Create ellipse
%     annotation(CalcDetails,'ellipse','FaceColor',colors(Nl,:),...
%         'Position',[Ell1_x+deltaX Ell1_y+deltaY 0.008 0.008],'Color',colors(Nl,:));
%
%     % Create ellipse
%     annotation(CalcDetails,'ellipse','FaceColor',colors(Nl,:),...
%         'Position',[Ell2_x+deltaX Ell2_y+deltaY 0.008 0.008],'Color',colors(Nl,:));


    if TE_TM==1    % TE wave
        % Create textbox
            annotation(CalcDetails,'textbox','String',{['\epsilon_{r',num2str(Nl),'} = ',num2str(er(Nl),'%2.3g')],['\mu_{r',num2str(Nl),'} = ',num2str(mr(Nl),'%2.3g')],['\sigma_{',num2str(Nl),'} = ',num2str(sigma(Nl),'%2.3g'),' S/m']},...
            'FontSize',11,...
            'FontWeight','demi',...
            'Color',colors(Nl,:),...
            'FontName','Arial',...
            'FitHeightToText','off',...
            'EdgeColor','none',...
            'Position',[Text1_x+deltaX Text1_y+deltaY 0.2 0.2]);
    else
        % Create textbox
              annotation(CalcDetails,'textbox','String',{['\epsilon_{r',num2str(Nl),'}    =
',num2str(er(Nl),'%2.3g')],['\mu_{r',num2str(Nl),'}                                          =
',num2str(mr(Nl),'%2.3g')],['\sigma_{',num2str(Nl),'} = ',num2str(sigma(Nl),'%2.3g'),'
S/m']},...
            'FontSize',11,...
            'FontWeight','demi',...
            'Color',colors(Nl,:),...
            'FontName','Arial',...
            'FitHeightToText','off',...
            'EdgeColor','none',...
            'Position',[Text1_x+deltaX Text1_y+deltaY 0.2 0.2]);
    end

    if TE_TM==1    % TE wave
        % Create textbox
        annotation(CalcDetails,'textbox',...
             'String',{['\eta_{TE}  =  ',num2str(zTE(Nl),'%2.3g'),'  \Omega'],['\lambda  =
',num2str(wl(Nl),'%2.3g'),' m'],['\beta = ',num2str(beta(Nl),'%2.3g'),' rad/m'],['\alpha
= ',num2str(alfa(Nl),'%2.3g'),' Np/m']},...
            'FontWeight','demi',...
            'FontSize',11,...
            'Color',colors(Nl,:),...
            'FontName','Arial',...
            'FitHeightToText','on',...
            'EdgeColor','none',...
            'LineWidth',2,...
            'Position',[Text2_x+deltaX Text2_y+deltaY 0.2 0.2]);
    else
        % Create textbox
        annotation(CalcDetails,'textbox',...
            'String',{['\eta_{TM} = ',num2str(zTM(Nl),'%2.3g'),' \Omega'],['\lambda =
',num2str(wl(Nl),'%2.3g'),' m'],['\beta = ',num2str(beta(Nl),'%2.3g'),' rad/m'],['\alpha
= ',num2str(alfa(Nl),'%2.3g'),' Np/m']},...
            'FontWeight','demi',...
            'FontSize',11,...
            'Color',colors(Nl,:),...
            'FontName','Arial',...
            'FitHeightToText','on',...
            'EdgeColor','none',...
            'LineWidth',2,...
            'Position',[Text2_x+deltaX Text2_y+deltaY 0.2 0.2]);
    end
    close(hdlg)    % close 'Working...' message
end
return




function [z,deltaz]=linIntegr(a,b,N);


% a = starting point
% b = ending point
% N = number of pooints


dz=(b-a)/N;



zp=a+dz*[0:N];    % get delta z
```

```
z=(zp(1:N)+zp(2:(N+1)))/2;   % get mean points where the function has to be evaluated

deltaz=diff(zp);
return

function out=sqrt2(in,what)

% The script accounts for the branch cut crossing

% in: complex number in input
% what: 0 --> ordinary material (real(in) & imag(in) must be > 0)
%         1 --> ordinary material (real(in) & imag(in) must be < 0)

if what==0
```

光学算法

```
    I=find(imag(in)<0);

    out=in;

    out(I)=-out(I);

elseif what==1

    I=find(imag(in)>0);

    out=in;

    out(I)=-out(I);

end

return




% --- Executes on button press in pushbutton2.

function pushbutton2_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton2 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)



global risp

risp=0;
```

学算法

```
function edit7_Callback(hObject, eventdata, handles)

% hObject    handle to edit7 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)



% Hints: get(hObject,'String') returns contents of edit7 as text

%        str2double(get(hObject,'String')) returns contents of edit7 as a double



% --- Executes during object creation, after setting all properties.

function edit7_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit7 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called
```

视学算法

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end




% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hint: get(hObject,'Value') returns toggle state of radiobutton1
% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1


% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



RealORAbs{1,1}='Real values';
RealORAbs{2,1}='Absolute values';
set(hObject,'String',RealORAbs)




% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

测学算法

```
% Hints: contents = get(hObject,'String') returns popupmenu2 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu2




% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called



% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
TE_TM{1,1}='TE wave';

TE_TM{2,1}='TM wave';

set(hObject,'String',TE_TM)




function edit9_Callback(hObject, eventdata, handles)

% hObject    handle to edit9 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)



% Hints: get(hObject,'String') returns contents of edit9 as text

%        str2double(get(hObject,'String')) returns contents of edit9 as a double




% --- Executes during object creation, after setting all properties.

function edit9_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit9 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called



% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end




function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



% Hints: get(hObject,'String') returns contents of edit10 as text
%        str2double(get(hObject,'String')) returns contents of edit10 as a double
```

学算法

```
% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in radiobutton2.
```

```
function radiobutton2_Callback(hObject, eventdata, handles)

% hObject    handle to radiobutton2 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)



% Hint: get(hObject,'Value') returns toggle state of radiobutton2



function

res=CheckInputs(teta,freq,er,mr,sigma,Zlay,what,ev,fH,A0,yrange,zrange,ns,anim,Ro

A,TE_TM,Calc)



% The script receives all inputs from the GUI and checks if they are ok in

% order to allow the script to run properly



% if checks are ok, output 1

res=1;
```

视学算法

```
% check if any of the fields is empty
if
isempty(teta)|isempty(freq)|isempty(er)|isempty(mr)|isempty(sigma)|isempty(Zlay)|isempty(A0)|isempty(yrange)|isempty(zrange)|isempty(ns)
    errordlg('Please check that none of the input fields is empty','Input error');
    res=0;
    return
end


% limited angle of incidence
if teta>90||teta<-90
    errordlg('Please check field "Angle of incidence": the angle of incidence must be comprised between -90? and 90?','Input error');
    res=0;
    return
end


% positive frequence
```

```
if freq<=0
    errordlg('Please check the field "Frequency": the frequency must be greater than 0 Hz','Input error');
    res=0;
    return
end



% positive conductivity
if sum(sigma<0)>0
    errordlg('Please check the field "Conductivity": the conductivity of each material must be equal to or greater than 0 S/m','Input error');
    res=0;
    return
end



% at least two layers to be defined
if length(er)<2|length(mr)<2|length(sigma)<2
    errordlg('Please check the field(s) "Electric permittivity", "Magnetic permeability" or "Conductivity": at least 2 media must be defined','Input error');
    res=0;
```

视学算法

```
    return
end



% same dimension of the er mr and sigma vectors
if (length(er)==length(mr))&(length(er)==length(sigma))&(length(sigma)==length(mr))
    res=1;
else
       errordlg('Please check the field(s) "Electric permittivity", "Magnetic permeability" and "Conductivity": the vectors specifying such values must be of the same length','Input error');
    res=0;
    return
end



% first interface in 0
if length(Zlay)==1
    errordlg('Please check the field "Z-coordinate of the interfaces": at least 1 interface must be defined in 0 m','Input error');
```

视学算法

```
    res=0;
    return
else
    if Zlay(2)~=0
        errordlg('Please check the field "Z-coordinate of the interfaces": the first interface must be in 0 m','Input error');
        res=0;
        return
    end
end



% match between interfaces and media
if length(Zlay)==(length(er))
    res=1;
elseif length(Zlay)<(length(er))
    errordlg(['Please check the field "Z-coordinate of the interfaces": the number of interfaces does not match the number of the media defined; please add ',num2str(abs(length(Zlay)-length(er))),' interface(s) in the field "Z-coordinates of the interfaces"'],'Input error');
    res=0;
    return
```

```
elseif length(Zlay)>(length(er))
    errordlg(['Please check the field "Z-coordinate of the interfaces": the number of interfaces does not match the number of the media defined; please remove ',num2str(abs(length(Zlay)-length(er))),' interface(s) in the field "Z-coordinates of the interfaces"'],'Input error');
    res=0;
    return
end



% iterfaces coordinates incrementally negative
Val=Zlay(2);
Monot=1;
for i=2:length(Zlay)-1
    if Zlay(i+1)>=Val
        Monot=0;
        break
    else
        Val=Zlay(i+1);
    end
end
if Monot==0
```

洸学算法

```
    errordlg('Please check the field "Z-coordinate of the interfaces": the coordinate of the interfaces must be incrementally negative','Input error');
    res=0;
    return
end


% last interface beyond axis range
if Zlay(end)<=yrange(1)
    errordlg('Please check the fields "Z-coordinate of the interfaces" and "Axes range": at least one of the interfaces coordinates is exceeding the axes range','Input error');
    res=0;
    return
end


% positive conductivity
if sum(sigma<0)>0
    errordlg('Please check the field "Conductivity": the conductivity of each material must be equal to or greater than 0 S/m','Input error');
    res=0;
    return
end


% positive incident field
if A0<0
    errordlg('Please check the field "Incidente field": the absolute value of the incidente field must be positive','Input error');
    res=0;
    return
end


% axes values
if yrange(1)>=yrange(2)
    errordlg('Please check the field "Axes range": the second value must be greater than the first one','Input error');
    res=0;
    return
end
```

```
% axis range must include the interface at 0 m
if yrange(2)<=0
    errordlg('Please check the field "Axes range": it must include the coordinate z = 0 m','Input error');
    res=0;
    return
end


% number of samples
if ns<=0
    errordlg('Please check the field "Number of samples of each axis": such value must be greater than 0','Input error');
    res=0;
    return
end


% warning if both the calculation details and the animation are selected
if anim==1&&Calc==1
    hdlg=helpdlg('Calculation details will be shown after the field animation will have been stopped','Information');
```

视觉算法

```
    uiwait(hdlg)
    return
end



% max number of layers
global Lim1
global Lim2
if (length(er)>2*(Lim2))&Calc==1
    errordlg(['When "Calculation deltails" is selected, the maximum number of materials is ',num2str(2*Lim2),': please remove ',num2str(length(er)-2*Lim2),' material(s) or deselect the field "Calculation deltails"'],'Input error');
    res=0;
    return
end




% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
```

视学算法

```
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)


web(['file:///' which('Help.html')])


% --- Creates and returns a handle to the GUI figure.
function h1 = PlaneWaveGUI_LayoutFcn(policy)
% policy - create a new figure or use a singleton. 'new' or 'reuse'.

persistent hsingleton;
if strcmpi(policy, 'reuse') & ishandle(hsingleton)
    h1 = hsingleton;
    return;
end


appdata = [];
```

```
appdata.GUIDEOptions = struct(...
    'active_h', [], ...
    'taginfo', struct(...
    'figure', 2, ...
    'uipanel', 2, ...
    'axes', 7, ...
    'edit', 11, ...
    'text', 11, ...
    'pushbutton', 4, ...
    'togglebutton', 2, ...
    'radiobutton', 3, ...
    'listbox', 2, ...
    'popupmenu', 3), ...
    'override', 1, ...
    'release', 13, ...
    'resize', 'simple', ...
    'accessibility', 'callback', ...
    'mfile', 1, ...
    'callbacks', 1, ...
    'singleton', 1, ...
    'syscolorfig', 1, ...
    'blocking', 0, ...
```

视学算法

```
    'lastSavedFile', 'K:\PlaneWaveGUI.m');
appdata.lastValidTag = 'figure1';
appdata.GUIDELayoutEditor = [];



h1 = figure(...
'Units','normalized',...
'PaperUnits',get(0,'defaultfigurePaperUnits'),...
'Color',[0.925490196078431 0.913725490196078 0.847058823529412],...
'Colormap',[0 0 0.5625;0 0 0.625;0 0 0.6875;0 0 0.75;0 0 0.8125;0 0 0.875;0 0
0.9375;0 0 1;0 0.0625 1;0 0.125 1;0 0.1875 1;0 0.25 1;0 0.3125 1;0 0.375 1;0 0.4375
1;0 0.5 1;0 0.5625 1;0 0.625 1;0 0.6875 1;0 0.75 1;0 0.8125 1;0 0.875 1;0 0.9375 1;0 1
1;0.0625 1 1;0.125 1 0.9375;0.1875 1 0.875;0.25 1 0.8125;0.3125 1 0.75;0.375 1
0.6875;0.4375 1 0.625;0.5 1 0.5625;0.5625 1 0.5;0.625 1 0.4375;0.6875 1 0.375;0.75 1
0.3125;0.8125 1 0.25;0.875 1 0.1875;0.9375 1 0.125;1 1 0.0625;1 1 0;1 0.9375 0;1
0.875 0;1 0.8125 0;1 0.75 0;1 0.6875 0;1 0.625 0;1 0.5625 0;1 0.5 0;1 0.4375 0;1 0.375
0;1 0.3125 0;1 0.25 0;1 0.1875 0;1 0.125 0;1 0.0625 0;1 0 0;0.9375 0 0;0.875 0
0;0.8125 0 0;0.75 0 0;0.6875 0 0;0.625 0 0;0.5625 0 0],...
'IntegerHandle','off',...
'InvertHardcopy',get(0,'defaultfigureInvertHardcopy'),...
'MenuBar','none',...
'Name','TE/TM plane wave propagation through multilayered structures (by
Lorenzo Luini)',...
'NumberTitle','off',...
'PaperPosition',get(0,'defaultfigurePaperPosition'),...
'PaperSize',[20.98404194812 29.67743169791],...
'PaperType',get(0,'defaultfigurePaperType'),...
'Position',[0.05 0.1 0.95 0.8],...
'HandleVisibility','callback',...
'Tag','figure1',...
'UserData',[],...
'Visible','on',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );


appdata = [];
appdata.lastValidTag = 'uipanel1';


h2 = uipanel(...
'Parent',h1,...
'Title','Parameters',...
'Tag','uipanel1',...
'Clipping','on',...
```

视学算法

```
'Position',[0.751284686536486 0.0716612377850163 0.236382322713258
0.887622149837134],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );



appdata = [];
appdata.lastValidTag = 'edit1';



h3 = uicontrol(...
'Parent',h2,...
'Units','normalized',...
'BackgroundColor',[1 1 1],...
'Callback','PlaneWaveGUI(''edit1_Callback'',gcbo,[],guidata(gcbo))',...
'Position',[0.0530973451327433 0.926559008500539 0.429203539823009
0.036144578313253],...
'String','1e9',...
'Style','edit',...
'CreateFcn', {@local_CreateFcn,
'PlaneWaveGUI(''edit1_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...
'Tag','edit1');
```

```
appdata = [];

appdata.lastValidTag = 'edit2';



h4 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'BackgroundColor',[1 1 1],...

'Callback','PlaneWaveGUI(''edit2_Callback'',gcbo,[],guidata(gcbo))',...

'Position',[0.513274336283186 0.926559008500539 0.429203539823009
0.036144578313253],...

'String','[1 2 1]',...

'Style','edit',...

'CreateFcn', {@local_CreateFcn,
'PlaneWaveGUI(''edit2_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...

'Tag','edit2');



appdata = [];

appdata.lastValidTag = 'edit3';
```

```
h5 = uicontrol(...
'Parent',h2,...
'Units','normalized',...
'BackgroundColor',[1 1 1],...
'Callback','PlaneWaveGUI(''edit3_Callback'',gcbo,[],guidata(gcbo))',...
'Position',[0.517699115044248 0.842981870119094 0.429203539823009
0.036144578313253],...
'String','[1 1 1]',...
'Style','edit',...
'CreateFcn', {@local_CreateFcn,
'PlaneWaveGUI(''edit3_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...
'Tag','edit3');


appdata = [];
appdata.lastValidTag = 'edit4';


h6 = uicontrol(...
'Parent',h2,...
'Units','normalized',...
```

```
'BackgroundColor',[1 1 1],...

'Callback','PlaneWaveGUI(''edit4_Callback'',gcbo,[],guidata(gcbo))',...

'Position',[0.0530973451327433 0.842981870119094 0.429203539823009
0.036144578313253],...

'String','[0 0 0]',...

'Style','edit',...

'CreateFcn', {@local_CreateFcn,
'PlaneWaveGUI(''edit4_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...

'Tag','edit4');



appdata = [];

appdata.lastValidTag = 'edit5';



h7 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'BackgroundColor',[1 1 1],...

'Callback','PlaneWaveGUI(''edit5_Callback'',gcbo,[],guidata(gcbo))',...

'Position',[0.292035398230089 0.548655150511905 0.429203539823009
0.036144578313253],...
```

```
'String','30',...

'Style','edit',...

'CreateFcn',                                              {@local_CreateFcn,

'PlaneWaveGUI(''edit5_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...

'Tag','edit5');




appdata = [];

appdata.lastValidTag = 'text1';




h8 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'Position',[0.0442477876106195      0.96442475720966      0.446902654867256

0.0292598967297762],...

'String','Frequency [Hz]',...

'Style','text',...

'Tag','text1',...

'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );
```

```
appdata = [];

appdata.lastValidTag = 'text2';




h9 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'Position',[0.504424778761062 0.96442475720966 0.446902654867256

0.0292598967297762],...

'String','Electric permittivity',...

'Style','text',...

'Tag','text2',...

'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );




appdata = [];

appdata.lastValidTag = 'text3';




h10 = uicontrol(...

'Parent',h2,...

'Units','normalized',...
```

视学算法

```
'Position',[0.486725663716814 0.880847618828217 0.486725663716814
0.0292598967297762],...
'String','Magnetic permeability',...
'Style','text',...
'Tag','text3',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'text4';

h11 = uicontrol(...
'Parent',h2,...
'Units','normalized',...
'Position',[0.0353982300884956 0.880847618828217 0.464601769911504
0.0292598967297762],...
'String','Conductivity [S/m]',...
'Style','text',...
'Tag','text4',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );
```

润学算法

```
appdata = [];

appdata.lastValidTag = 'text5';



h12 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'Position',[0.256637168141593 0.586520899221028 0.495575221238938 0.0292598967297762],...

'String','Angle of incidence [癞',...

'Style','text',...

'Tag','text5',...

'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );



appdata = [];

appdata.lastValidTag = 'edit6';



h13 = uicontrol(...

'Parent',h2,...
```

光学算法

```
'Units','normalized',...

'BackgroundColor',[1 1 1],...

'Callback','PlaneWaveGUI(''edit6_Callback'',gcbo,[],guidata(gcbo))',...

'Position',[0.293577981651376 0.718929254302098 0.431192660550459 0.0363288718929254],...

'String','[0 -0.5]',...

'Style','edit',...

'CreateFcn', {@local_CreateFcn, 'PlaneWaveGUI(''edit6_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...

'Tag','edit6');



appdata = [];

appdata.lastValidTag = 'text6';



h14 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'Position',[0.0707964601769911 0.757499267762106 0.871681415929203 0.0292598967297762],...

'String','Z-coordinate of the interfaces [m]',...
```

```
'Style','text',...

'Tag','text6',...

'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );




appdata = [];

appdata.lastValidTag = 'edit7';




h15 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'BackgroundColor',[1 1 1],...

'Callback','PlaneWaveGUI(''edit7_Callback'',gcbo,[],guidata(gcbo))',...

'Position',[0.293577981651376 0.634799235181637 0.431192660550459

0.0363288718929254],...

'String','1',...

'Style','edit',...

'CreateFcn', {@local_CreateFcn,

'PlaneWaveGUI(''edit7_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...

'Tag','edit7');
```

```
appdata = [];

appdata.lastValidTag = 'text7';



h16 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'Position',[0.188073394495413 0.673040152963664 0.63302752293578

0.0286806883365201],...

'String','Incident field ([V/m] or [A/m])',...

'Style','text',...

'Tag','text7',...

'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );



appdata = [];

appdata.lastValidTag = 'radiobutton1';



h17 = uicontrol(...

'Parent',h2,...
```

```
'Units','normalized',...

'Callback','PlaneWaveGUI(''radiobutton1_Callback'',gcbo,[],guidata(gcbo))',...

'Position',[0.265486725663717 0.163877497847859 0.482 0.026],...

'String','Animate fields',...

'Style','radiobutton',...

'Tag','radiobutton1',...

'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );




appdata = [];

appdata.lastValidTag = 'popupmenu1';




h18 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'BackgroundColor',[1 1 1],...

'Callback','PlaneWaveGUI(''popupmenu1_Callback'',gcbo,[],guidata(gcbo))',...

'CData',[],...

'Position',[0.2715 0.241902792141668 0.47 0.036144578313253],...

'String',{   'Real values'; 'Absolute values' },...

'Style','popupmenu',...
```

光学算法

```
'Value',1,...
'CreateFcn',                                                                    {@local_CreateFcn,
'PlaneWaveGUI(''popupmenu1_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...
'Tag','popupmenu1',...
'UserData',[]);



appdata = [];
appdata.lastValidTag = 'popupmenu2';



h19 = uicontrol(...
'Parent',h2,...
'Units','normalized',...
'BackgroundColor',[1 1 1],...
'Callback','PlaneWaveGUI(''popupmenu2_Callback'',gcbo,[],guidata(gcbo))',...
'Position',[0.272 0.27976854085079 0.47 0.036144578313253],...
'String',{   'TE wave'; 'TM wave' },...
'Style','popupmenu',...
'Value',1,...
'CreateFcn',                                                                    {@local_CreateFcn,
'PlaneWaveGUI(''popupmenu2_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...
```

科学算法

```
'Tag','popupmenu2');



appdata = [];

appdata.lastValidTag = 'edit9';



h20 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'BackgroundColor',[1 1 1],...

'Callback','PlaneWaveGUI(''edit9_Callback'',gcbo,[],guidata(gcbo))',...

'Position',[0.293577981651376 0.460889806156856 0.431192660550459
0.0363288718929254],...

'String','[-1 0.5]',...

'Style','edit',...

'CreateFcn', {@local_CreateFcn,
'PlaneWaveGUI(''edit9_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...

'Tag','edit9');



appdata = [];
```

```
appdata.lastValidTag = 'text9';


h21 = uicontrol(...
'Parent',h2,...
'Units','normalized',...
'Position',[0.256880733944954 0.499130723938883 0.495412844036697 0.0286806883365201],...
'String','Axes range   [m]',...
'Style','text',...
'Tag','text9',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );


appdata = [];
appdata.lastValidTag = 'edit10';


h22 = uicontrol(...
'Parent',h2,...
'Units','normalized',...
'BackgroundColor',[1 1 1],...
```

润学算法

```
'Callback','PlaneWaveGUI(''edit10_Callback'',gcbo,[],guidata(gcbo))',...
'Position',[0.292035398230089 0.373450266523014 0.429203539823009
0.036144578313253],...
'String','300',...
'Style','edit',...
'CreateFcn', {@local_CreateFcn,
'PlaneWaveGUI(''edit10_CreateFcn'',gcbo,[],guidata(gcbo))', appdata} ,...
'Tag','edit10');


appdata = [];
appdata.lastValidTag = 'text10';


h23 = uicontrol(...
'Parent',h2,...
'Units','normalized',...
'Position',[0.146788990825688 0.411263359923657 0.747706422018349
0.0286806883365201],...
'String','Number of samples of each axis',...
'Style','text',...
'Tag','text10',...
```

```
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );



appdata = [];

appdata.lastValidTag = 'radiobutton2';



h24 = uicontrol(...

'Parent',h2,...

'Units','normalized',...

'Callback','PlaneWaveGUI(''radiobutton2_Callback'',gcbo,[],guidata(gcbo))',...

'Position',[0.265486725663717 0.136338771513953 0.482300884955752 0.0258175559380378],...

'String','Calculation details',...

'Style','radiobutton',...

'Tag','radiobutton2',...

'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );



appdata = [];

appdata.lastValidTag = 'axes1';
```

```
h25 = axes(...
'Parent',h1,...
'Position',[0.0626242544731609 0.569117647058824 0.298210735586481
0.382352941176471],...
'Box','on',...
'CameraPosition',[0.5 0.5 9.16025403784439],...
'CameraPositionMode',get(0,'defaultaxesCameraPositionMode'),...
'Color',get(0,'defaultaxesColor'),...
'ColorOrder',get(0,'defaultaxesColorOrder'),...
'LooseInset',[0.240815533980582 0.189107611548556 0.175980582524271
0.128937007874016],...
'XColor',get(0,'defaultaxesXColor'),...
'YColor',get(0,'defaultaxesYColor'),...
'ZColor',get(0,'defaultaxesZColor'),...
'Tag','axes1',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );


h26 = get(h25,'title');
```

```
set(h26,...
'Parent',h25,...
'Units','data',...
'FontUnits','points',...
'BackgroundColor','none',...
'Color',[0 0 0],...
'EdgeColor','none',...
'EraseMode','normal',...
'DVIMode','auto',...
'FontAngle','normal',...
'FontName','Helvetica',...
'FontSize',10,...
'FontWeight','normal',...
'HorizontalAlignment','center',...
'LineStyle','-',...
'LineWidth',0.5,...
'Margin',2,...
'Position',[0.498275862068966 1.02765957446809 1.00005459937205],...
'Rotation',0,...
'String',blanks(0),...
'Interpreter','tex',...
'VerticalAlignment','bottom',...
'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','on',...

'XLimInclude','on',...

'YLimInclude','on',...

'ZLimInclude','on',...

'CLimInclude','on',...

'ALimInclude','on',...

'IncludeRenderer','on',...

'Clipping','off');
```

```
h27 = get(h25,'xlabel');



set(h27,...

'Parent',h25,...

'Units','data',...

'FontUnits','points',...

'BackgroundColor','none',...

'Color',[0 0 0],...

'EdgeColor','none',...

'EraseMode','normal',...

'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...

'FontWeight','normal',...

'HorizontalAlignment','center',...

'LineStyle','-',...

'LineWidth',0.5,...

'Margin',2,...

'Position',[0.498275862068966 -0.1 1.00005459937205],...

'Rotation',0,...
```

```
'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','cap',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','on',...

'XLimInclude','on',...

'YLimInclude','on',...

'ZLimInclude','on',...

'CLimInclude','on',...

'ALimInclude','on',...

'IncludeRenderer','on',...
```

```
'Clipping','off');

h28 = get(h25,'ylabel');

set(h28,...
'Parent',h25,...
'Units','data',...
'FontUnits','points',...
'BackgroundColor','none',...
'Color',[0 0 0],...
'EdgeColor','none',...
'EraseMode','normal',...
'DVIMode','auto',...
'FontAngle','normal',...
'FontName','Helvetica',...
'FontSize',10,...
'FontWeight','normal',...
'HorizontalAlignment','center',...
'LineStyle','-',...
'LineWidth',0.5,...
```

数学算法

```
'Margin',2,...

'Position',[-0.0982758620689655 0.495744680851064 1.00005459937205],...

'Rotation',90,...

'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','bottom',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','on',...

'XLimInclude','on',...

'YLimInclude','on',...

'ZLimInclude','on',...
```

```
'CLimInclude','on',...

'ALimInclude','on',...

'IncludeRenderer','on',...

'Clipping','off');



h29 = get(h25,'zlabel');



set(h29,...

'Parent',h25,...

'Units','data',...

'FontUnits','points',...

'BackgroundColor','none',...

'Color',[0 0 0],...

'EdgeColor','none',...

'EraseMode','normal',...

'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...

'FontWeight','normal',...
```

```
'HorizontalAlignment','right',...
'LineStyle','-',...
'LineWidth',0.5,...
'Margin',2,...
'Position',[-0.212068965517241 1.12127659574468 1.00005459937205],...
'Rotation',0,...
'String',blanks(0),...
'Interpreter','tex',...
'VerticalAlignment','middle',...
'ButtonDownFcn',[],...
'CreateFcn', {@local_CreateFcn, [], ''} ,...
'DeleteFcn',[],...
'BusyAction','queue',...
'HandleVisibility','off',...
'HelpTopicKey',blanks(0),...
'HitTest','on',...
'Interruptible','on',...
'SelectionHighlight','on',...
'Serializable','on',...
'Tag',blanks(0),...
'UserData',[],...
'Visible','off',...
```

```
'XLimInclude','on',...
'YLimInclude','on',...
'ZLimInclude','on',...
'CLimInclude','on',...
'ALimInclude','on',...
'IncludeRenderer','on',...
'Clipping','off');


appdata = [];
appdata.lastValidTag = 'pushbutton2';


h30 = uicontrol(...
'Parent',h1,...
'Units','normalized',...
'Callback','PlaneWaveGUI(''pushbutton2_Callback'',gcbo,[],guidata(gcbo))',...
'Position',[0.882703777335979 0.107271507951715 0.0914512922465206 0.0426470588235294],...
'String','Stop animation',...
'Tag','pushbutton2',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );
```

沉学算法

```
appdata = [];

appdata.lastValidTag = 'axes4';



h31 = axes(...

'Parent',h1,...

'Position',[0.43041749502982 0.569117647058824 0.298210735586481 0.382352941176471],...

'Box','on',...

'CameraPosition',[0.5 0.5 9.16025403784439],...

'CameraPositionMode',get(0,'defaultaxesCameraPositionMode'),...

'Color',get(0,'defaultaxesColor'),...

'ColorOrder',get(0,'defaultaxesColorOrder'),...

'LooseInset',[0.240815533980582 0.189107611548556 0.175980582524271 0.128937007874016],...

'XColor',get(0,'defaultaxesXColor'),...

'YColor',get(0,'defaultaxesYColor'),...

'ZColor',get(0,'defaultaxesZColor'),...

'Tag','axes4',...

'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );
```

```
h32 = get(h31,'title');

set(h32,...
'Parent',h31,...
'Units','data',...
'FontUnits','points',...
'BackgroundColor','none',...
'Color',[0 0 0],...
'EdgeColor','none',...
'EraseMode','normal',...
'DVIMode','auto',...
'FontAngle','normal',...
'FontName','Helvetica',...
'FontSize',10,...
'FontWeight','normal',...
'HorizontalAlignment','center',...
'LineStyle','-',...
'LineWidth',0.5,...
'Margin',2,...
```

```
'Position',[0.498275862068965 1.02765957446809 1.00005459937205],...

'Rotation',0,...

'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','bottom',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','on',...

'XLimInclude','on',...

'YLimInclude','on',...

'ZLimInclude','on',...

'CLimInclude','on',...
```

统学算法

```
'ALimInclude','on',...

'IncludeRenderer','on',...

'Clipping','off');



h33 = get(h31,'xlabel');



set(h33,...

'Parent',h31,...

'Units','data',...

'FontUnits','points',...

'BackgroundColor','none',...

'Color',[0 0 0],...

'EdgeColor','none',...

'EraseMode','normal',...

'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...

'FontWeight','normal',...

'HorizontalAlignment','center',...
```

数学算法

```
'LineStyle','-',...
'LineWidth',0.5,...
'Margin',2,...
'Position',[0.498275862068965 -0.1 1.00005459937205],...
'Rotation',0,...
'String',blanks(0),...
'Interpreter','tex',...
'VerticalAlignment','cap',...
'ButtonDownFcn',[],...
'CreateFcn', {@local_CreateFcn, [], ''} ,...
'DeleteFcn',[],...
'BusyAction','queue',...
'HandleVisibility','off',...
'HelpTopicKey',blanks(0),...
'HitTest','on',...
'Interruptible','on',...
'SelectionHighlight','on',...
'Serializable','on',...
'Tag',blanks(0),...
'UserData',[],...
'Visible','on',...
'XLimInclude','on',...
```

学算法

```
'YLimInclude','on',...

'ZLimInclude','on',...

'CLimInclude','on',...

'ALimInclude','on',...

'IncludeRenderer','on',...

'Clipping','off');



h34 = get(h31,'ylabel');



set(h34,...

'Parent',h31,...

'Units','data',...

'FontUnits','points',...

'BackgroundColor','none',...

'Color',[0 0 0],...

'EdgeColor','none',...

'EraseMode','normal',...

'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...
```

刘学算法

```
'FontSize',10,...
'FontWeight','normal',...
'HorizontalAlignment','center',...
'LineStyle','-',...
'LineWidth',0.5,...
'Margin',2,...
'Position',[-0.0982758620689659 0.495744680851064 1.00005459937205],...
'Rotation',90,...
'String',blanks(0),...
'Interpreter','tex',...
'VerticalAlignment','bottom',...
'ButtonDownFcn',[],...
'CreateFcn', {@local_CreateFcn, [], ''} ,...
'DeleteFcn',[],...
'BusyAction','queue',...
'HandleVisibility','off',...
'HelpTopicKey',blanks(0),...
'HitTest','on',...
'Interruptible','on',...
'SelectionHighlight','on',...
'Serializable','on',...
'Tag',blanks(0),...
```

数学算法

```
'UserData',[],...
'Visible','on',...
'XLimInclude','on',...
'YLimInclude','on',...
'ZLimInclude','on',...
'CLimInclude','on',...
'ALimInclude','on',...
'IncludeRenderer','on',...
'Clipping','off');


h35 = get(h31,'zlabel');


set(h35,...
'Parent',h31,...
'Units','data',...
'FontUnits','points',...
'BackgroundColor','none',...
'Color',[0 0 0],...
'EdgeColor','none',...
'EraseMode','normal',...
```

视学算法

```
'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...

'FontWeight','normal',...

'HorizontalAlignment','right',...

'LineStyle','-',...

'LineWidth',0.5,...

'Margin',2,...

'Position',[-1.44655172413793 1.12127659574468 1.00005459937205],...

'Rotation',0,...

'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','middle',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...
```

数学算法

```
'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','off',...

'XLimInclude','on',...

'YLimInclude','on',...

'ZLimInclude','on',...

'CLimInclude','on',...

'ALimInclude','on',...

'IncludeRenderer','on',...

'Clipping','off');



appdata = [];

appdata.lastValidTag = 'axes5';



h36 = axes(...

'Parent',h1,...

'Position',[0.0626242544731609      0.0735294117647059      0.298210735586481

0.382352941176471],...
```

深学算法

```
'Box','on',...
'CameraPosition',[0.5 0.5 9.16025403784439],...
'CameraPositionMode',get(0,'defaultaxesCameraPositionMode'),...
'Color',get(0,'defaultaxesColor'),...
'ColorOrder',get(0,'defaultaxesColorOrder'),...
'LooseInset',[0.240815533980582 0.189107611548556 0.175980582524271 0.128937007874016],...
'XColor',get(0,'defaultaxesXColor'),...
'YColor',get(0,'defaultaxesYColor'),...
'ZColor',get(0,'defaultaxesZColor'),...
'Tag','axes5',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );


h37 = get(h36,'title');


set(h37,...
'Parent',h36,...
'Units','data',...
'FontUnits','points',...
'BackgroundColor','none',...
```

视学算法

```
'Color',[0 0 0],...

'EdgeColor','none',...

'EraseMode','normal',...

'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...

'FontWeight','normal',...

'HorizontalAlignment','center',...

'LineStyle','-',...

'LineWidth',0.5,...

'Margin',2,...

'Position',[0.498275862068966 1.02765957446809 1.00005459937205],...

'Rotation',0,...

'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','bottom',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...
```

```
'HelpTopicKey',blanks(0),...
'HitTest','on',...
'Interruptible','on',...
'SelectionHighlight','on',...
'Serializable','on',...
'Tag',blanks(0),...
'UserData',[],...
'Visible','on',...
'XLimInclude','on',...
'YLimInclude','on',...
'ZLimInclude','on',...
'CLimInclude','on',...
'ALimInclude','on',...
'IncludeRenderer','on',...
'Clipping','off');


h38 = get(h36,'xlabel');


set(h38,...
'Parent',h36,...
```

优学算法

```
'Units','data',...

'FontUnits','points',...

'BackgroundColor','none',...

'Color',[0 0 0],...

'EdgeColor','none',...

'EraseMode','normal',...

'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...

'FontWeight','normal',...

'HorizontalAlignment','center',...

'LineStyle','-',...

'LineWidth',0.5,...

'Margin',2,...

'Position',[0.498275862068966 -0.1 1.00005459937205],...

'Rotation',0,...

'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','cap',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...
```

训学算法

```
'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','on',...

'XLimInclude','on',...

'YLimInclude','on',...

'ZLimInclude','on',...

'CLimInclude','on',...

'ALimInclude','on',...

'IncludeRenderer','on',...

'Clipping','off');




h39 = get(h36,'ylabel');
```

```
set(h39,...

'Parent',h36,...

'Units','data',...

'FontUnits','points',...

'BackgroundColor','none',...

'Color',[0 0 0],...

'EdgeColor','none',...

'EraseMode','normal',...

'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...

'FontWeight','normal',...

'HorizontalAlignment','center',...

'LineStyle','-',...

'LineWidth',0.5,...

'Margin',2,...

'Position',[-0.0982758620689655 0.495744680851064 1.00005459937205],...

'Rotation',90,...

'String',blanks(0),...

'Interpreter','tex',...
```

数学算法

```
'VerticalAlignment','bottom',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','on',...

'XLimInclude','on',...

'YLimInclude','on',...

'ZLimInclude','on',...

'CLimInclude','on',...

'ALimInclude','on',...

'IncludeRenderer','on',...

'Clipping','off');
```

视学算法

```
h40 = get(h36,'zlabel');

set(h40,...
'Parent',h36,...
'Units','data',...
'FontUnits','points',...
'BackgroundColor','none',...
'Color',[0 0 0],...
'EdgeColor','none',...
'EraseMode','normal',...
'DVIMode','auto',...
'FontAngle','normal',...
'FontName','Helvetica',...
'FontSize',10,...
'FontWeight','normal',...
'HorizontalAlignment','right',...
'LineStyle','-',...
'LineWidth',0.5,...
'Margin',2,...
'Position',[-0.212068965517241 2.41489361702128 1.00005459937205],...
```

数学算法

```
'Rotation',0,...

'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','middle',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','off',...

'XLimInclude','on',...

'YLimInclude','on',...

'ZLimInclude','on',...

'CLimInclude','on',...

'ALimInclude','on',...
```

沉学算法

```
'IncludeRenderer','on',...
'Clipping','off');


appdata = [];
appdata.lastValidTag = 'axes6';


h41 = axes(...
'Parent',h1,...
'Position',[0.43041749502982    0.0735294117647059    0.298210735586481
0.382352941176471],...
'Box','on',...
'CameraPosition',[0.5 0.5 9.16025403784439],...
'CameraPositionMode',get(0,'defaultaxesCameraPositionMode'),...
'Color',get(0,'defaultaxesColor'),...
'ColorOrder',get(0,'defaultaxesColorOrder'),...
'LooseInset',[0.240815533980582    0.189107611548556    0.175980582524271
0.128937007874016],...
'XColor',get(0,'defaultaxesXColor'),...
'YColor',get(0,'defaultaxesYColor'),...
'ZColor',get(0,'defaultaxesZColor'),...
```

学算法

```
'Tag','axes6',...

'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );



h42 = get(h41,'title');



set(h42,...

'Parent',h41,...

'Units','data',...

'FontUnits','points',...

'BackgroundColor','none',...

'Color',[0 0 0],...

'EdgeColor','none',...

'EraseMode','normal',...

'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...

'FontWeight','normal',...

'HorizontalAlignment','center',...

'LineStyle','-',...
```

统学算法

```
'LineWidth',0.5,...

'Margin',2,...

'Position',[0.498275862068965 1.02765957446809 1.00005459937205],...

'Rotation',0,...

'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','bottom',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','on',...

'XLimInclude','on',...

'YLimInclude','on',...
```

```
'ZLimInclude','on',...

'CLimInclude','on',...

'ALimInclude','on',...

'IncludeRenderer','on',...

'Clipping','off');



h43 = get(h41,'xlabel');



set(h43,...

'Parent',h41,...

'Units','data',...

'FontUnits','points',...

'BackgroundColor','none',...

'Color',[0 0 0],...

'EdgeColor','none',...

'EraseMode','normal',...

'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...
```

学算法

```
'FontWeight','normal',...

'HorizontalAlignment','center',...

'LineStyle','-',...

'LineWidth',0.5,...

'Margin',2,...

'Position',[0.498275862068965 -0.1 1.00005459937205],...

'Rotation',0,...

'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','cap',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...
```

数学算法

```
'Visible','on',...
'XLimInclude','on',...
'YLimInclude','on',...
'ZLimInclude','on',...
'CLimInclude','on',...
'ALimInclude','on',...
'IncludeRenderer','on',...
'Clipping','off');


h44 = get(h41,'ylabel');


set(h44,...
'Parent',h41,...
'Units','data',...
'FontUnits','points',...
'BackgroundColor','none',...
'Color',[0 0 0],...
'EdgeColor','none',...
'EraseMode','normal',...
'DVIMode','auto',...
```

洞见学算法

```
'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...

'FontWeight','normal',...

'HorizontalAlignment','center',...

'LineStyle','-',...

'LineWidth',0.5,...

'Margin',2,...

'Position',[-0.0982758620689659 0.495744680851064 1.00005459937205],...

'Rotation',90,...

'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','bottom',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...

'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...
```

数学算法

```
'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','on',...

'XLimInclude','on',...

'YLimInclude','on',...

'ZLimInclude','on',...

'CLimInclude','on',...

'ALimInclude','on',...

'IncludeRenderer','on',...

'Clipping','off');



h45 = get(h41,'zlabel');



set(h45,...

'Parent',h41,...

'Units','data',...

'FontUnits','points',...

'BackgroundColor','none',...

'Color',[0 0 0],...
```

```
'EdgeColor','none',...

'EraseMode','normal',...

'DVIMode','auto',...

'FontAngle','normal',...

'FontName','Helvetica',...

'FontSize',10,...

'FontWeight','normal',...

'HorizontalAlignment','right',...

'LineStyle','-',...

'LineWidth',0.5,...

'Margin',2,...

'Position',[-1.44655172413793 2.41489361702128 1.00005459937205],...

'Rotation',0,...

'String',blanks(0),...

'Interpreter','tex',...

'VerticalAlignment','middle',...

'ButtonDownFcn',[],...

'CreateFcn', {@local_CreateFcn, [], ''} ,...

'DeleteFcn',[],...

'BusyAction','queue',...

'HandleVisibility','off',...

'HelpTopicKey',blanks(0),...
```

科学算法

```
'HitTest','on',...

'Interruptible','on',...

'SelectionHighlight','on',...

'Serializable','on',...

'Tag',blanks(0),...

'UserData',[],...

'Visible','off',...

'XLimInclude','on',...

'YLimInclude','on',...

'ZLimInclude','on',...

'CLimInclude','on',...

'ALimInclude','on',...

'IncludeRenderer','on',...

'Clipping','off');



appdata = [];

appdata.lastValidTag = 'pushbutton1';




h46 = uicontrol(...

'Parent',h1,...
```

视学算法

```
'Units','normalized',...
'Callback','PlaneWaveGUI(''pushbutton1_Callback'',gcbo,[],guidata(gcbo))',...
'Position',[0.772867420349435 0.107491856677524 0.091469681397739
0.0423452768729642],...
'String','Calculate',...
'Tag','pushbutton1',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );


appdata = [];
appdata.lastValidTag = 'pushbutton3';


h47 = uicontrol(...
'Parent',h1,...
'Units','normalized',...
'Callback','PlaneWaveGUI(''pushbutton3_Callback'',gcbo,[],guidata(gcbo))',...
'Position',[0.827338129496403 0.0185667752442998 0.091469681397739
0.0423452768729642],...
'String','Help',...
'Tag','pushbutton3',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );
```

光学算法

```
hsingleton = h1;

% --- Set application data first then calling the CreateFcn.

function local_CreateFcn(hObject, eventdata, createfcn, appdata)

if ~isempty(appdata)

   names = fieldnames(appdata);

   for i=1:length(names)

       name = char(names(i));

       setappdata(hObject, name, getfield(appdata,name));

   end

end
```

```
if ~isempty(createfcn)

    eval(createfcn);

end





% --- Handles default GUIDE GUI creation and callback dispatch

function varargout = gui_mainfcn(gui_State, varargin)





%   GUI_MAINFCN provides these command line APIs for dealing with GUIs

%

%       PLANEWAVEGUI, by itself, creates a new PLANEWAVEGUI or raises the existing

%       singleton*.

%

%       H = PLANEWAVEGUI returns the handle to a new PLANEWAVEGUI or the handle to

%       the existing singleton*.
```

```
%
%       PLANEWAVEGUI('CALLBACK',hObject,eventData,handles,...) calls the local
%       function named CALLBACK in PLANEWAVEGUI.M with the given input arguments.
%
%       PLANEWAVEGUI('Property','Value',...) creates a new PLANEWAVEGUI or raises the
%       existing singleton*.  Starting from the left, property value pairs are
%       applied to the GUI before untitled_OpeningFunction gets called.  An
%       unrecognized property name or invalid value makes property application
%       stop.  All inputs are passed to untitled_OpeningFcn via varargin.
%
%       *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%       instance to run (singleton)".


%   Copyright 1984-2006 The MathWorks, Inc.
%   $Revision: 1.1.6.3 $ $Date: 2006/10/10 02:22:41 $



gui_StateFields =  {'gui_Name'
    'gui_Singleton'
    'gui_OpeningFcn'

    'gui_OutputFcn'

    'gui_LayoutFcn'

    'gui_Callback'};

gui_Mfile = '';

for i=1:length(gui_StateFields)

    if ~isfield(gui_State, gui_StateFields{i})

        error('MATLAB:gui_mainfcn:FieldNotFound', 'Could not find field %s in the
gui_State struct in GUI M-file %s', gui_StateFields{i}, gui_Mfile);

    elseif isequal(gui_StateFields{i}, 'gui_Name')

        gui_Mfile = [gui_State.(gui_StateFields{i}), '.m'];

    end

end



numargin = length(varargin);



if numargin == 0

    % PLANEWAVEGUI

    % create the GUI only if we are not in the process of loading it

    % already
    gui_Create = true;
elseif local_isInvokeActiveXCallback(gui_State, varargin{:})
    % PLANEWAVEGUI(ACTIVEX,...)
    vin{1} = gui_State.gui_Name;
    vin{2} = [get(varargin{1}.Peer, 'Tag'), '_', varargin{end}];
    vin{3} = varargin{1};
    vin{4} = varargin{end-1};
    vin{5} = guidata(varargin{1}.Peer);
    feval(vin{:});
    return;
elseif local_isInvokeHGCallbak(gui_State, varargin{:})
    % PLANEWAVEGUI('CALLBACK',hObject,eventData,handles,...)
    gui_Create = false;
else
    % PLANEWAVEGUI(...)
    % create the GUI and hand varargin to the openingfcn
    gui_Create = true;
end


if ~gui_Create
    % In design time, we need to mark all components possibly created in
    % the coming callback evaluation as non-serializable. This way, they
    % will not be brought into GUIDE and not be saved in the figure file
    % when running/saving the GUI from GUIDE.
    designEval = false;
    if (numargin>1 && ishghandle(varargin{2}))
        fig = varargin{2};
        while ~isempty(fig) && ~isa(handle(fig),'figure')
            fig = get(fig,'parent');
        end

        designEval = isappdata(0,'CreatingGUIDEFigure') || isprop(fig,'__GUIDEFigure');
    end

    if designEval
        beforeChildren = findall(fig);
    end

    % evaluate the callback now
    varargin{1} = gui_State.gui_Callback;
    if nargout
        [varargout{1:nargout}] = feval(varargin{:});
    else
        feval(varargin{:});
    end

    % Set serializable of objects created in the above callback to off in
    % design time. Need to check whether figure handle is still valid in
    % case the figure is deleted during the callback dispatching.
    if designEval && ishandle(fig)
        set(setdiff(findall(fig),beforeChildren), 'Serializable','off');
    end
else
    if gui_State.gui_Singleton
        gui_SingletonOpt = 'reuse';
    else
        gui_SingletonOpt = 'new';
    end

    % Check user passing 'visible' P/V pair first so that its value can be
    % used by oepnfig to prevent flickering
    gui_Visible = 'auto';
    gui_VisibleInput = '';
```

视学算法

```
for index=1:2:length(varargin)

    if length(varargin) == index || ~ischar(varargin{index})

        break;

    end




    % Recognize 'visible' P/V pair

    len1 = min(length('visible'),length(varargin{index}));

    len2 = min(length('off'),length(varargin{index+1}));

    if  ischar(varargin{index+1})  &&  strncmpi(varargin{index},'visible',len1)  && len2 > 1

        if strncmpi(varargin{index+1},'off',len2)

            gui_Visible = 'invisible';

            gui_VisibleInput = 'off';

        elseif strncmpi(varargin{index+1},'on',len2)

            gui_Visible = 'visible';

            gui_VisibleInput = 'on';

        end

    end

end


% Open fig file with stored settings.   Note: This executes all component
% specific CreateFunctions with an empty HANDLES structure.



% Do feval on layout code in m-file if it exists
gui_Exported = ~isempty(gui_State.gui_LayoutFcn);
% this application data is used to indicate the running mode of a GUIDE
% GUI to distinguish it from the design mode of the GUI in GUIDE. it is
% only used by actxproxy at this time.
setappdata(0,genvarname(['OpenGuiWhenRunning_', gui_State.gui_Name]),1);
if gui_Exported
    gui_hFigure = feval(gui_State.gui_LayoutFcn, gui_SingletonOpt);
    % openfig (called by local_openfig below) does this for guis without
    % the LayoutFcn. Be sure to do it here so guis show up on screen.
    movegui(gui_hFigure,'onscreen');
else
    gui_hFigure = local_openfig(gui_State.gui_Name, gui_SingletonOpt, gui_Visible);
    % If the figure has InGUIInitialization it was not completely created
    % on the last pass.  Delete this handle and try again.
    if isappdata(gui_hFigure, 'InGUIInitialization')
        delete(gui_hFigure);
```

```
            gui_hFigure = local_openfig(gui_State.gui_Name, gui_SingletonOpt, gui_Visible);
        end
    end
    rmappdata(0,genvarname(['OpenGuiWhenRunning_', gui_State.gui_Name]));



    % Set flag to indicate starting GUI initialization
    setappdata(gui_hFigure,'InGUIInitialization',1);



    % Fetch GUIDE Application options
    gui_Options = getappdata(gui_hFigure,'GUIDEOptions');
    % Singleton setting in the GUI M-file takes priority if different
    gui_Options.singleton = gui_State.gui_Singleton;



    if ~isappdata(gui_hFigure,'GUIOnScreen')
        % Adjust background color
        if gui_Options.syscolorfig
            set(gui_hFigure,'Color', get(0,'DefaultUicontrolBackgroundColor'));
        end
        % Generate HANDLES structure and store with GUIDATA. If there is
        % user set GUI data already, keep that also.
        data = guidata(gui_hFigure);
        handles = guihandles(gui_hFigure);
        if ~isempty(handles)
            if isempty(data)
                data = handles;
            else
                names = fieldnames(handles);
                for k=1:length(names)
                    data.(char(names(k)))=handles.(char(names(k)));
                end
            end
        end
        guidata(gui_hFigure, data);
    end


    % Apply input P/V pairs other than 'visible'
    for index=1:2:length(varargin)
```

```
    if length(varargin) == index || ~ischar(varargin{index})
        break;
    end


    len1 = min(length('visible'),length(varargin{index}));
    if ~strncmpi(varargin{index},'visible',len1)
        try set(gui_hFigure, varargin{index}, varargin{index+1}), catch break, end
    end
end


% If handle visibility is set to 'callback', turn it on until finished
% with OpeningFcn
gui_HandleVisibility = get(gui_hFigure,'HandleVisibility');
if strcmp(gui_HandleVisibility, 'callback')
    set(gui_hFigure,'HandleVisibility', 'on');
end


    feval(gui_State.gui_OpeningFcn, gui_hFigure, [], guidata(gui_hFigure), varargin{:});
```

```
if isscalar(gui_hFigure) && ishandle(gui_hFigure)
    % Update handle visibility
    set(gui_hFigure,'HandleVisibility', gui_HandleVisibility);


    % Call openfig again to pick up the saved visibility or apply the
    % one passed in from the P/V pairs
    if ~gui_Exported
        gui_hFigure = local_openfig(gui_State.gui_Name, 'reuse',gui_Visible);
    elseif ~isempty(gui_VisibleInput)
        set(gui_hFigure,'Visible',gui_VisibleInput);
    end
    if strcmpi(get(gui_hFigure, 'Visible'), 'on')
        figure(gui_hFigure);

        if gui_Options.singleton
            setappdata(gui_hFigure,'GUIOnScreen', 1);
        end
    end
    % Done with GUI initialization

    rmappdata(gui_hFigure,'InGUIInitialization');

end




% If handle visibility is set to 'callback', turn it on until finished with

% OutputFcn

if isscalar(gui_hFigure) && ishandle(gui_hFigure)

    gui_HandleVisibility = get(gui_hFigure,'HandleVisibility');

    if strcmp(gui_HandleVisibility, 'callback')

        set(gui_hFigure,'HandleVisibility', 'on');

    end

    gui_Handles = guidata(gui_hFigure);

else

    gui_Handles = [];

end




if nargout

    [varargout{1:nargout}] = feval(gui_State.gui_OutputFcn, gui_hFigure, [], gui_Handles);
    else
        feval(gui_State.gui_OutputFcn, gui_hFigure, [], gui_Handles);
    end



    if isscalar(gui_hFigure) && ishandle(gui_hFigure)
        set(gui_hFigure,'HandleVisibility', gui_HandleVisibility);
    end
end



function gui_hFigure = local_openfig(name, singleton, visible)



% openfig with three arguments was new from R13. Try to call that first, if
% failed, try the old openfig.
if nargin('openfig') == 2
    % OPENFIG did not accept 3rd input argument until R13,
    % toggle default figure visible to prevent the figure
    % from showing up too soon.
    gui_OldDefaultVisible = get(0,'defaultFigureVisible');
    set(0,'defaultFigureVisible','off');
```

学算法

```
    gui_hFigure = openfig(name, singleton);
    set(0,'defaultFigureVisible',gui_OldDefaultVisible);
else
    gui_hFigure = openfig(name, singleton, visible);
end


function result = local_isInvokeActiveXCallback(gui_State, varargin)


try
    result = ispc && iscom(varargin{1}) ...
             && isequal(varargin{1},gcbo);
catch
    result = false;
end

function result = local_isInvokeHGCallbak(gui_State, varargin)


try
    fhandle = functions(gui_State.gui_Callback);
```

视觉算法

```
    result = ~isempty(findstr(gui_State.gui_Name,fhandle.file)) || ...
             (ischar(varargin{1}) ...
             && isequal(ishandle(varargin{2}), 1) ...
             && ~isempty(strfind(varargin{1},[get(varargin{2}, 'Tag'), '_'])));
catch
    result = false;
end
```

专注保研|考研公众号：视学算法

视学算法