```
%---------initial condition-------------
global g0 E_sat beta2 beta2_ngvd gamma q0 q_sat lambda f0 z_step z_step_ngvd f_axis delta_om
% g0=100/10^3;  %per mm
% E_sat=10; % nJ
% beta2=1/(2*1.83*10^3); %fs^2/mm
% delta_omega= 270; %THz
% gamma=0.4; % 1/(W*m)
% q0=5/10^3;    %1/mm, 1/m=1/(mm*10^3)
% q_sat=0.3;  %MW

g0=100/10^3;  %per mm
E_sat=10; % nJ


delta_omega= 270; %THz
gamma=0.4; % 1/(W*m)
q0=5/10^3;    %1/mm, 1/m=1/(mm*10^3)
q_sat=0.3;  %MW
oc_eff=0.22; % output coupler transmission
lambda= 800; %nm, central frequency
f0=0; %THz, c/lambda
crystal_length=7; %mm
cavity_length=1820; %mm
ngvd_length=cavity_length-crystal_length; %mm
z_step=0.05;% mm
z_step_ngvd=5; %mm
pgvd_step=round(crystal_length/z_step);
ngvd_step=round(cavity_length/z_step_ngvd);

%---------GVD parameter-------------------
GDD_total=18; %fs^2
beta2=58; %fs^2/mm
GDD_beta2=beta2*crystal_length*2; %fs^2
GDD_beta2_ngvd=GDD_total-GDD_beta2; %fs^2
beta2_ngvd=GDD_beta2_ngvd/cavity_length; %fs^2/mm

t_step=0.2 ; %fs
t_stop=2000; %fs
t_index=-t_stop:t_step:t_stop;
%a0=1*q0*exp(-t_index.^2/100);
%a0=0.1*q0*sech(t_index/10); %sqrt(MW)
a0=0.01*q0*sech(t_index/10); %sqrt(MW)
%a0=(5*10^-4*rand(1,size(t_index,2))+10^-3*sech(t_index/10))*q0;
```

```
%--------fft setting--------

f_step=(1/t_step)*10^3/size(t_index,2); % THz
f_axis=zeros(1,size(a0,2));
left_half=round(size(f_axis,2)/2);
right_half=size(f_axis,2)-left_half;
for i=1:left_half,
    f_axis(1,i)=f0+i*f_step;
end
f_middle=f0-f_step*left_half;
for i=1:right_half,
    f_axis(1,left_half+i)=f_middle+i*f_step;
end

a1(1,:)=a0;
%a1(1,:)=a1(10,:);

t=cputime;
for num=1:10,
    for frame=1:4,
    for loop1=1:ngvd_step,
        a1(num+1,:)=split_step_fft_ngvd(a1(num,:));
        a1(num,:)=a1(num+1,:);
    end
    for loop2=1:2*pgvd_step,
        a1(num+1,:)=split_step_fft(a1(num,:));
        a1(num+1,:)=nonlinear(a1(num+1,:));
        a1(num,:)=a1(num+1,:);
    end
    for loop3=1:ngvd_step,
        a1(num+1,:)=split_step_fft_ngvd(a1(num,:));
        a1(num,:)=a1(num+1,:);
    end
    a1(num+1,:)=(1-oc_eff)*a1(num,:);
    a1(num,:)=oc_eff*a1(num,:);
ma(1,num)=max(abs(a1(num,:)));
    end
%energy check point
% t_step*sum(a1(num,:).*conj(a1(num,:)))
%subplot(211)

hold on
plot(t_index,abs(a1(num,:)));
% subplot(212)
% hold on
% plot(t_index,angle(a1(num,:)));
```

```
end
cputime-t

a1=abs(a1);
%plot(t_index,a1(1,:),'b',t_index,a1(2,:),t_index,a1(3,:),t_index,a1(4,:),t_index,a1(5,:))
%plot(t1,a1)


Error using ==> evalin
Subscripted assignment dimension mismatch.
```