Study of the reflection and transmission between the three material by 1D-FDTD PML Final Report

呂彥穎 R07941023

摘要

本研究是利用 1D-FDTD (Finite-Difference Time-Domain)來模擬並計算三個材料之間的反射係數和穿透係數，邊界的部分則是利用 PLM (Perfectly Matched Layer)當作吸收，最後使用 Fresnel equation 來計算並驗證因為不同的折射係數發生的反射率和穿透率。

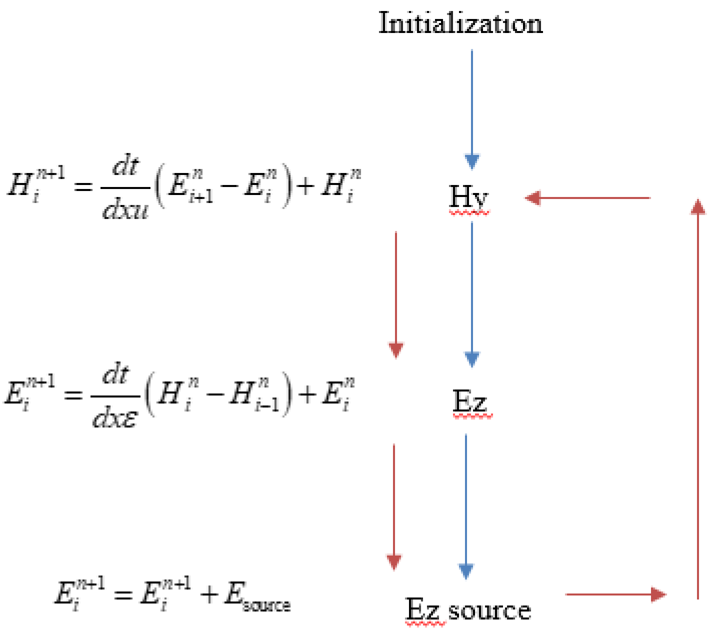
方法

本程式使用的 1D-FDTD 是由 Maxwell's equations 推倒後的電磁場結果

$$H_i^{n+\frac{\Delta t}{2}} = -\frac{\Delta t}{\Delta xu}\left(E_i^n - E_{i-1}^n\right) + H_i^{n-\frac{\Delta t}{2}}$$

$$E_i^{n+\frac{\Delta t}{2}} = -\frac{\Delta t}{\Delta x\varepsilon}\left(H_{i+1}^{n+\frac{\Delta t}{2}} - H_i^{n+\frac{\Delta t}{2}}\right) + E_i^n$$

為了模擬同時計算，程式的計算步驟如下:



模擬反射率的計算方法為路徑積分。

$$E_z \text{ simulation value} = \int_{x=\text{material i}}^{x=\text{material f}} \left|E_z\left(1,\mathrm{x}\right)\right| dx$$

$$H_y \text{ simulation value} = \int_{x=\text{material i}}^{x=\text{material f}} \left|H_y\left(1,\mathrm{x}\right)\right| dx$$

為了有更準確的測量，邊界使用 PML 來吸收，使波可以在兩個材料間傳播而不受到反射波的影響，方程式必須做以下修改:

$$H_i^{n+1} = \frac{u - \dfrac{0.5dt\sigma u}{\varepsilon}}{u + \dfrac{0.5dt\sigma u}{\varepsilon}} H_i^n + \frac{dt}{dx\left(u + \dfrac{0.5dt\sigma u}{\varepsilon}\right)}\left(E_{i+1}^n - E_i^n\right)$$

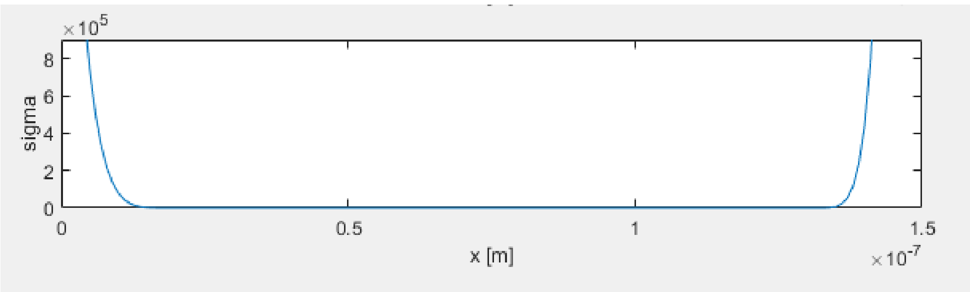
$$E_i^{n+1} = \frac{\varepsilon - 0.5dt\sigma}{\varepsilon + 0.5dt\sigma} E_i^n + \frac{dt}{dx\left(\varepsilon + 0.5dt\sigma\right)}\left(H_i^n - H_{i-1}^n\right)$$

而電導率的設置如下:

$$\sigma = \begin{cases} \dfrac{\varepsilon_r - \varepsilon_0 c \log\left(R\right)\left(\gamma+1\right)}{2dxw^{\gamma+1}\left(\gamma+1\right)}\left(x+0.5^{\gamma+1}\right) - \left(x-0.5\right)^{\gamma+1}, & \text{x at PML} \\ 1, & \text{otherelse} \end{cases}$$

其中γ為導電率的變化程度，R 為 PML 邊界的吸收係數，w 為 PML 的寬度，c 為光速。

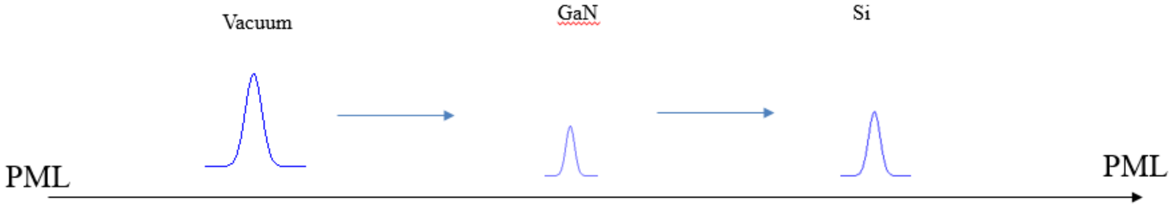
電導率的分布結果圖。



實驗

下圖是當模擬高斯波的行徑方向以及材料(Vacuum, GaN 以及 Si)位置的參數配置。

Source $E_z = -e^{\frac{-(x-20)^2}{50}}$

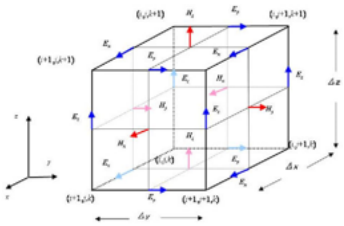


Dielectric constant
- Vacuum=ε_0
- GaN $= 11.8\varepsilon_0$
- Si $= 10\varepsilon_0$

magnetic permeability
- Vacuum=μ_0
- GaN $= \mu_0$
- Si $= \mu_0$

Position
- Vacuum=1-50 nm
- GaN = 51-100 nm
- Si = 101−150 nm

$S = E \times B$



驗證

為了驗證模擬值，這裡使用 Fresnel equation 來驗證模擬。

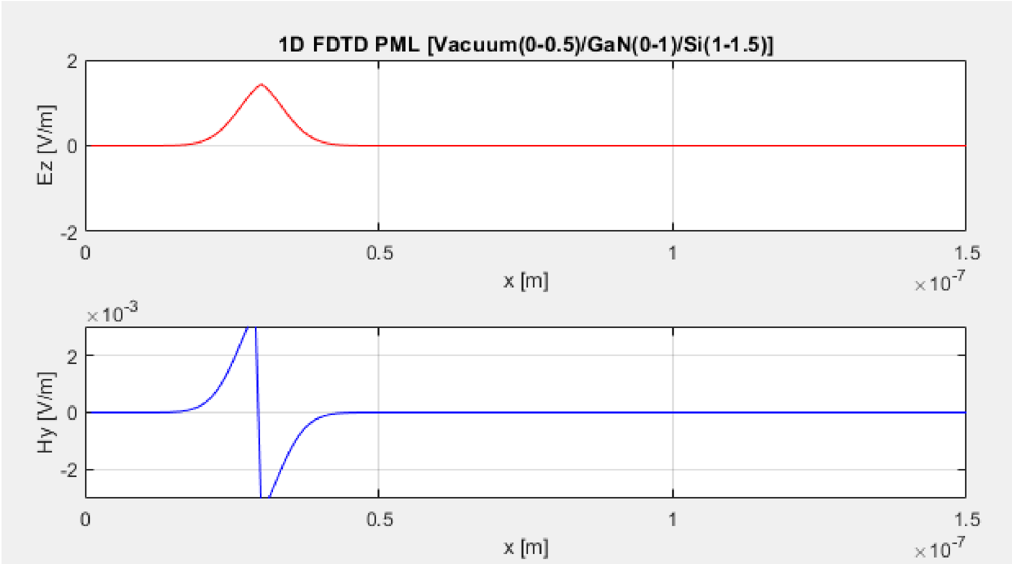
$$R = \left(\frac{n_1 - n_2}{n_1 + n_2}\right)^2,\ \mathrm{n}=\sqrt{\varepsilon_r \mu_r}$$

$$T = 1 - R$$

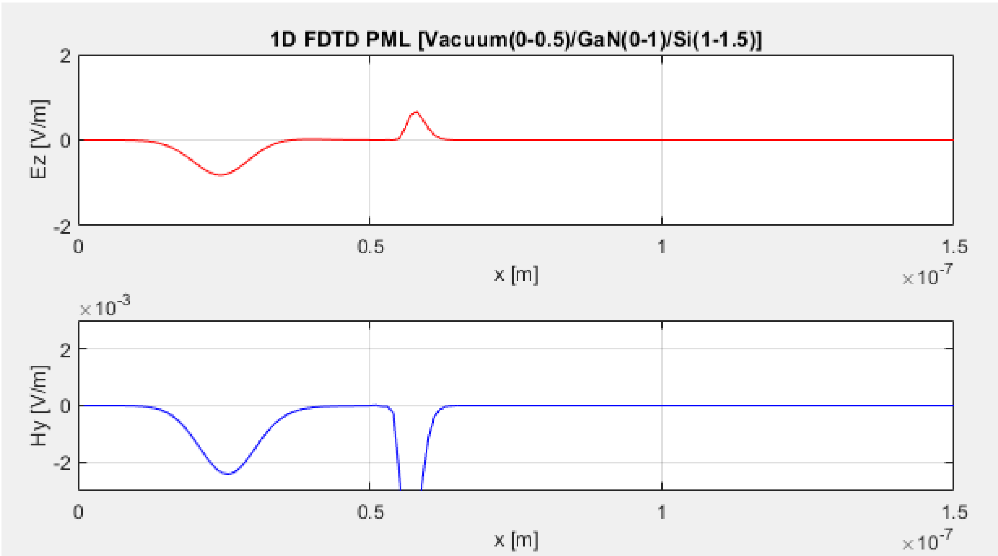
R 違反射係數，n 為折射率，ε_r 為介電係數，μ_r 為導磁係數，T 為穿透率。

結果

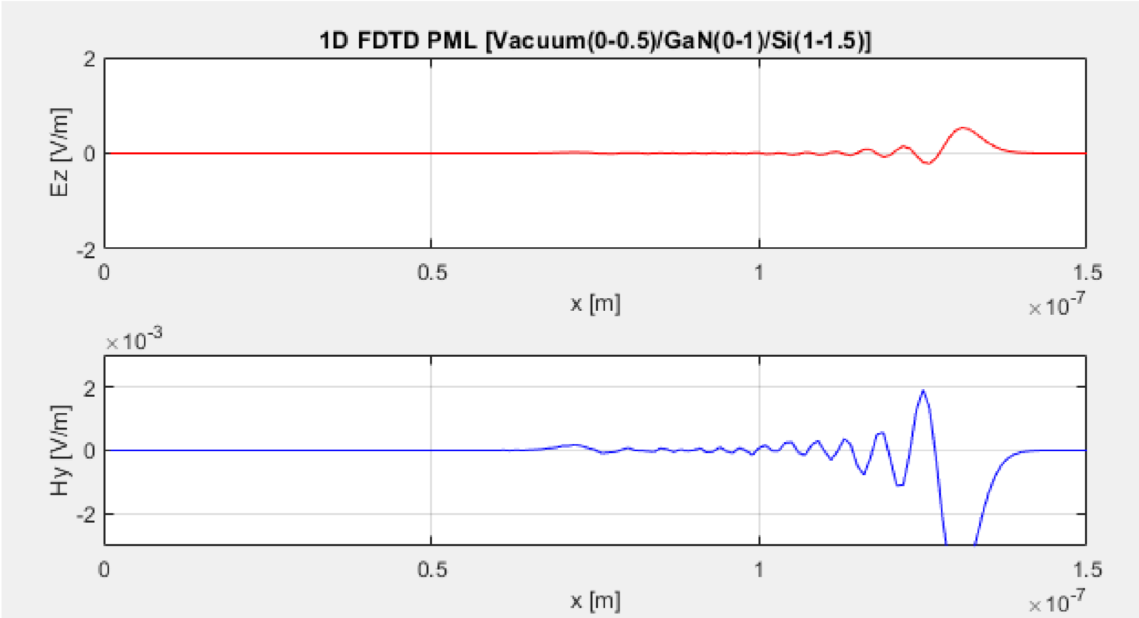
高斯波源位於真空，並開始傳播。



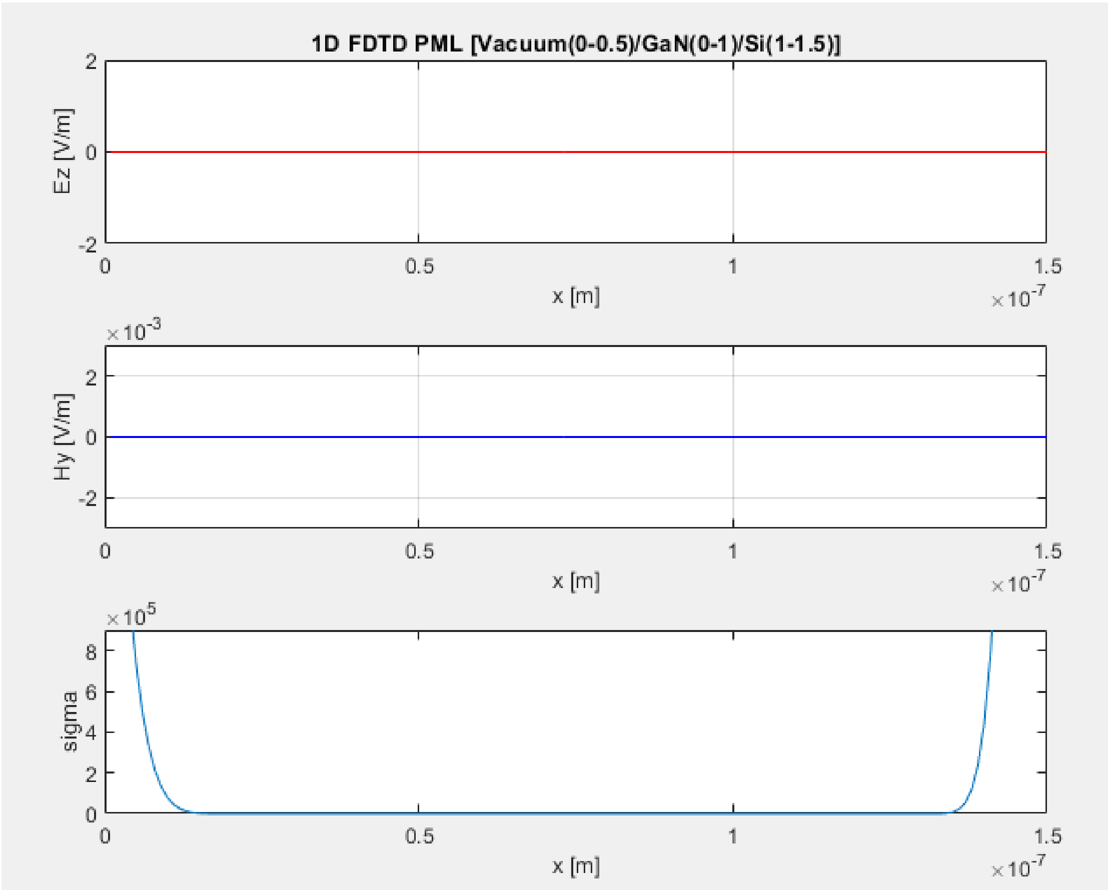
當第一個波從真空傳播到 GaN 時，因為折射率差異很大，所以有很明顯的反射



而 GaN 和 Si 折射率相近，所以反射不明顯。



由於觸碰到 PML 邊界，所以最後所有的波均被吸收。



利用 1D-FDTD 的模擬值與 Fresnel equation 理論值做比較。

材料	*R* 模擬值%	*R* 理論值%	誤差%
Vacuum - GaN	80.20%	29.65%	170.46%

GaN - Si	10.53%	0.12%	8537.8532%
n=1 – n=1	100%	100%	0%
n=1 – n=999	100%	88%	13.6%

結論

由於在非常極端的狀況下(n=1 – n=999)就存在著 13.6%的誤差，導致很難在一般的情況下有很準確的模擬值，如果使用較高精度的模擬，也許會對結果會有很大的改善。

MATLAB codes:

```
 clear all
clc

%% dimenstion parameter
xdim=150;
dx=1e-9;  % [m]
Steps=1000;
PML_w=20;
PML_n=6;
PML_R=1e-6;  % reflection coefficient

%% Souce
source=30;
intensity = 3;
wide = 1;
const = 50;

%%
epsilon0=8.85e-12;
u0=1.2566e-6;
c=3e8;
dt=dx/c;
Ez=zeros(1,xdim);
Hy=zeros(1,xdim);
Ez_g=zeros(1,xdim);
Hy_g=zeros(1,xdim);
```

```
globel_Hy = zeros(Steps,xdim);
globel_Ez = zeros(Steps,xdim);

%% three materials
epsilon=epsilon0*ones(1,xdim);
u=u0*ones(1,xdim);
a_meterial = [1,50];
b_meterial = [51,100];
c_meterial = [101,150];
epsilon(1,a_meterial(1):a_meterial(2))=1*epsilon0;
epsilon(1,b_meterial(1):b_meterial(2))=11.5*epsilon0;
epsilon(1,c_meterial(1):c_meterial(2))=10*epsilon0;
u(1,a_meterial(1):a_meterial(2))=1*u0;
u(1,b_meterial(1):b_meterial(2))=1*u0;
u(1,c_meterial(1):c_meterial(2))=1*u0;
record = [80,300]; %caculation reflection (simulaiton) [80,330]

%% PML
% Ez conductivity
PML_maxsigma=(-log10(PML_R)*(PML_n+1)*epsilon0*c)/(2*PML_w*dx);
PML_boundary_l=((epsilon(1,PML_w)/epsilon0)*PML_maxsigma)/((PML_w^PML
_n)*(PML_n+1));
PML_boundary_r=((epsilon(1,xdim-PML_w)/epsilon0)*PML_maxsigma)/((PML_
w^PML_n)*(PML_n+1));
sigma=zeros(1,xdim);
x=0:PML_w;
for i=1:xdim

sigma(1,PML_w+1:-1:1)=PML_boundary_l*((x+0.5*ones(1,PML_w+1)).^(PML_n
+1)-(x-0.5*[0 ones(1,PML_w)]).^(PML_n+1));

sigma(1,xdim-PML_w:xdim)=PML_boundary_r*((x+0.5*ones(1,PML_w+1)).^(PM
L_n+1)-(x-0.5*[0 ones(1,PML_w)]).^(PML_n+1));
end
% Hy conductivity
sigma_s=(sigma.*u)./epsilon;

% Hy coefficient
```

```
A=((u-0.5*dt*sigma_s)./(u+0.5*dt*sigma_s));
B=(dt/dx)./(u+0.5*dt*sigma_s);

% Ez coefficient
C=((epsilon-0.5*dt*sigma)./(epsilon+0.5*dt*sigma));
D=(dt/dx)./(epsilon+0.5*dt*sigma);


%% 1D-FDTD

for time=1:1:Steps
    time;
    % time boundary
    if time < source-2  % left side
        xi=source-time-1;
    else
        xi=1;
    end
    if time < xdim-1-source % right side
        xf=source+time;
    else
        xf=xdim-1;
    end

    % Update Hy from Ez
    for i = xi:xf
        Hy(i)=A(i).*Hy(i)+B(i).*(Ez(i+1)-Ez(i));
    end

    % Hy source
    Hy_g(time) = intensity*exp(-((time-source)/wide)^2/const);

    % Update Ez from Hy
    for i = xi:xf
        Ez(i+1)=C(i+1).*Ez(i+1)+D(i+1).*(Hy(i+1)-Hy(i));
    end
```

```
    % Ez source
    Ez_g(time) = intensity*exp(-((time-source)/wide)^2/const);
    Ez(source) = Ez(source) + Ez_g(time);

    globel_Ez(time,:) = Ez(1,:);
    globel_Hy(time,:) = Hy(1,:);

    %Reflection
    totalE(1,time) = sum(abs(globel_Ez(time,:)));
    totalH(1,time) = sum(abs(globel_Hy(time,:)));
    if time == record(1)
        %caculate the simulation value
        rE1=sum(abs(globel_Ez(time,a_meterial(1):a_meterial(2))));
        RE1 = rE1/totalE(1,time);
        tE1=sum(abs(globel_Ez(time,b_meterial(1):b_meterial(2))));
        TE1 = tE1/totalE(1,time);
        rH1=sum(abs(globel_Hy(time,a_meterial(1):a_meterial(2))));
        RH1 = rH1/totalH(1,time);
        tH1=sum(abs(globel_Hy(time,b_meterial(1):b_meterial(2))));
        TH1 = tH1/totalH(1,time);
        %caculation the fresnel value
        n1 = sqrt(epsilon(1,a_meterial(1))*u(1,a_meterial(1)));
        n2 = sqrt(epsilon(1,b_meterial(1))*u(1,b_meterial(1)));
        f_R1 = ((n1-n2)/(n1+n2))^2;
        %error (fresnel-simulation)/fresnel
        error_RE1 = abs(f_R1-RE1)/f_R1*100;
        error_RH1 = abs(f_R1-RH1)/f_R1*100;
        string = [' '];
        disp(string)
        string = ['In first reflection'];
        disp(string)
        string = ['Ez Simulation value is ', num2str(RE1*100), '%, fresnel value is ', num2str(f_R1*100), '%, error is ', num2str(error_RE1), '%'];
        disp(string)
        % string = ['Hy Simulation value is ', num2str(RH1*100), '%, fresnel value is ', num2str(f_R1*100), '%, error is ', num2str(error_RH1), '%'];
        % disp(string)
    end
```

```
    if time == record(2)
        %caculate the simulation value
        rE2=sum(abs(globel_Ez(time,b_meterial(1):b_meterial(2))));
        RE2 = rE2/tE1;
        tE2=sum(abs(globel_Ez(time,c_meterial(1):c_meterial(2))));
        TE2 = tE2/tE1;
        rH2=sum(abs(globel_Hy(time,b_meterial(1):b_meterial(2))));
        RH2 = rH2/tH1;
        tH2=sum(abs(globel_Hy(time,c_meterial(1):c_meterial(2))));
        TH2 = tH2/tH1;
        %caculation the fresnel value
        n2 = sqrt(epsilon(1,b_meterial(1))*u(1,b_meterial(1)));
        n3 = sqrt(epsilon(1,c_meterial(1))*u(1,c_meterial(1)));
        f_R2 = ((n2-n3)/(n2+n3))^2;
        %error (fresnel-simulation)/fresnel
        error_RE2 = abs(f_R2-RE2)/f_R2*100;
        error_RH2 = abs(f_R2-RH2)/f_R2*100;
        string = [' '];
        disp(string)
        string = ['In second reflection'];
        disp(string)
        string = ['Ez Simulation value is ', num2str(RE2*100), '%, fresnel
value is ', num2str(f_R2*100), '%, error is ', num2str(error_RE2), '%'];
        disp(string)
        % string = ['Hy Simulation value is ', num2str(RH2*100), '%, fresnel
value is ', num2str(f_R2*100), '%, error is ', num2str(error_RH2), '%'];
        % disp(string)
    end

    % plot
    subplot(3,1,1)
    plot((1:xdim)*dx,Ez,'color','r');
    titlestring=['1D FDTD PML [Vacuum(0-0.5)/GaN(0-1)/Si(1-1.5)]'];
    title(titlestring,'color','k');
    xlabel('x [m]');
    ylabel('Ez [V/m]');
    axis([0 xdim*dx -2 2]);
    grid on
```

```
    subplot(3,1,2)
    plot((1:xdim)*dx,Hy,'color','b');
    xlabel('x [m]');
    ylabel('Hy [V/m]');
    axis([0 xdim*dx -3e-3 3e-3]);
    grid on

    subplot(3,1,3)
    plot((1:xdim)*dx,sigma)
    xlabel('x [m]');
    ylabel('sigma');
    axis([0 xdim*dx 0 900000]);
    getframe;
end
```