Physic 640
Sept 23, 2011

Project 1 Summary and Presentations

| | |
|---|---|
| Greg | # points per period; why SNR=1 is harder to recover vs SNR=2 |
| Joe | Matlab code debugging |
| Chris | Amplitude filter in frequency |
| Zhiqin | Frequency axis scale; pull each frequency out of a bi-frequency signal; use apriori knowledge of frequency and a bandpass filter to work with SNR=1 |
| Gwen | Location of low and high frequencies in the FFT spectrum plot |
| Pinze | Applying low-pass and high-pass filter on a .wav file; signal compression using amplitude filter in time |
| Ryan | Plot(fft(f)) v. plot(t,fft(f)) |
| Tom | Sampling frequency in 'wavplay' |
| Peyling | Array declaration in f77: fixed maximum. |

## Project 2
## Weeks 3 & 4

-Introduction to numerical algorithms
-Finite-Difference in the Time-Domain (FDTD) and the numerical solution of wave equations in 1D
-Finite-Difference in the Time-Domain (FDTD) and the numerical solution of Maxwell's equations in 2D
-Implementation of a FDTD code in Fortran:

Project 2:

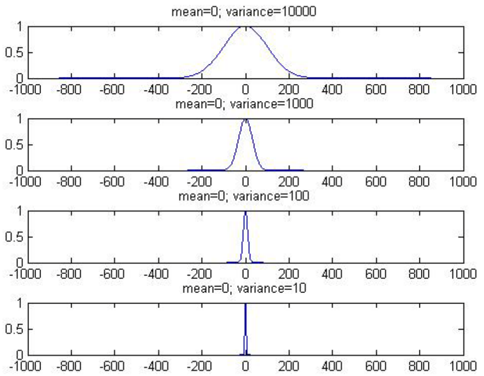
a) Develop a Fortran code to propagate a sinusoid or a Gaussian pulse in 1D, save outputs into a file
b) Show the animations using Matlab

*How do I create a movies in Matlab?
By plotting results for successive time iterations, and grabbing frames and store in a movies file (.avi) that can be played later on in windows.
*What is a numerical algorithm?
To develop a numerical simulation, we need to go through the following 5 steps:
a) Define the problem and find appropriate equation to describe it (e.g. wave propagation in a string → wave equation in 1D
y(x,t)= A cos(kx –wt)

$\frac{\partial^2 y}{\partial x^2} = \frac{1}{v^2}\frac{\partial^2 y}{\partial t^2}$ v is the wave speed)

b) Use a numerical scheme to obtain an algorithm**, appropriate discretization of the equation** to be used with a computer (in this Project #2 we will use the FDTD numerical scheme)
c) Implement the algorithm into a code using a computer language (for example we will use Fortran in this project)
d) Debug and plot results
e) Testing and validation against theoretical results, experimental values, or results from other establish numerical methods

*What is a Gaussian wave packet? Examples of Gaussians of different variances (widths) are shown below with the Matlab code.



```
%Gaussian distribution
clear all
close all
for i=1:4
var=10^(5-i);%1000;%variance
sigma=sqrt(var);%standard deviation
xbar=0;%%mean
x=-1000:1000;
fx=1/(sqrt(2*pi)*sigma)*exp(-(x-
xbar).^2/(2*var));
fx=fx/max(fx);
figure(1), subplot(4,1,i),plot(x,fx),
title(strcat('mean=',num2str(xbar),';
variance=',num2str(var)))
end
```

Bilbiography:

**1**.- A. Taflove and S. C. Hagness, Computational Electrodynamics: The Finite-Difference Time-Domain Method, 3rd ed. Norwood, MA: Artech House, 2005.
**2**.- D.M. Sullivan, Electromagnetic Simulation Using the FDTD Method , IEEE Press
**3**.- K. S. Yee, ``Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propagat.*, vol. AP-14, no. 4, pp. 302--307, 1966.
**4**.- A. Taflove and M. E. Brodwin, "Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations," IEEE Trans. Microwave Theory and Techniques, vol. 23, pp. 623-630, Aug. 1975. (Download Paper2.pdf )
**5**.- A. Taflove and K. R. Umashankar, "Solution of Complex Electromagnetic Penetration and Scattering Problems in Unbounded Regions," pp. 83-113 in Computational Methods for Infinite Domain Media-Structure Interactions, American Society of Mechanical Engineers, AMD vol. 46 (1981).
**6**.-V. Sathiaseelan, B. B. Mittal, A. J. Fenn, and A. Taflove, "Recent Advances in External Electromagnetic Hyperthermia," Chap. 10 in Advances in Radiation

Therapy, B. B. Mittal, J. A. Purdy, and K. K. Ang, eds. (Cancer Treatment and Research, S. T. Rosen, series ed.) Dordrecht, Netherlands: Kluwer (1998).

## FDTD Method and the 1D Maxwell Equation:

Finite difference is an approximation to derivatives using Taylor's expansion of different orders. The Maxwell equations (governing behavior of electric and magnetic fields) can be written in differential form:

$$\vec{\nabla} \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t}$$

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

This is a set of 6 equations since there are 3 components to each vector

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y}\right)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z}\right)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}\right)$$

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma E_x\right)$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - \sigma E_y\right)$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z\right)$$

σ is the electrical conductivity

In 1D: suppose we work in x; variations in y and z are zero

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x}\right)$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x} - \sigma E_z\right)$$

In empty space (no materials) σ=0

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu_0}\left(\frac{\partial E_z}{\partial x}\right) \rightarrow \frac{\partial}{\partial x}\frac{\partial H_y}{\partial t} = \frac{1}{\mu_0}\left(\frac{\partial^2 E_z}{\partial x^2}\right)$$

$$\frac{\partial E_z}{\partial t}=\frac{1}{\varepsilon_0}\left(\frac{\partial H_y}{\partial x}\right)\rightarrow\frac{\partial}{\partial t}\frac{\partial E_z}{\partial t}=\frac{1}{\varepsilon_0}\left(\frac{\partial^2 H_y}{\partial t\partial x}\right)\rightarrow\frac{\partial^2 E_z}{\partial t^2}=\frac{1}{\varepsilon_0\mu_0}\frac{\partial^2 E_z}{\partial x^2}=c^2\frac{\partial^2 E_z}{\partial x^2}$$

Using Taylor's expansion up to second order, derivatives can be approximated using finite differences: we need to discretize space and time derivatives.

$$\frac{\partial F^n(i)}{\partial x}\cong\frac{F^n\left(i+{}^1\!/_2\right)-F^n\left(i-{}^1\!/_2\right)}{\delta}$$

$$\frac{\partial F^n(i)}{\partial t}\cong\frac{F^{n+{}^1\!/_2}(i)-F^{n-{}^1\!/_2}(i)}{\delta t}$$

n is the time index; i is the space index; δt is the time increment; δ is the space increment.

Next step is to apply this numerical scheme (FDTD) to write the algorithm for the wave equation and proceed to its implementation using Fortran

Hints for Project 2:

1) Follow the steps below to see how to derive the algorithm based on the FDTD method for wave propagation:

$$\frac{\partial^2 E_z}{\partial t^2}=c^2\frac{\partial^2 E_z}{\partial x^2}$$

$$\frac{\partial F^n(i)}{\partial x}\cong\frac{F^n\left(i+{}^1\!/_2\right)-F^n\left(i-{}^1\!/_2\right)}{\delta}$$

$$\frac{\partial F^n(i)}{\partial t}\cong\frac{F^{n+{}^1\!/_2}(i)-F^{n-{}^1\!/_2}(i)}{\delta t}$$

$$\frac{F^{n+1}(i)+F^{n-1}(i)-2F^n(i)}{(\delta t)^2}=c^2\frac{F^n(i+1)+F^n(i-1)-2F^n(i)}{\delta^2}$$

$$\rightarrow F^{n+1}(i)=2F^n(i)-F^{n-1}(i)+\left(\frac{c\delta t}{\delta}\right)^2\left[F^n(i+1)+F^n(i-1)-2F^n(i)\right]$$

2) While implementing the code in Fortran, you should always start with variable declaration and parameter definitions, then for this Project define a Gaussian pulse initialization, followed by the algorithm for propagation.

3) In one dimension, FDTD algorithm stability requires $\Delta t\leq\frac{\Delta x}{c}$

4) Explain in your report: where should the center of the Gaussian be placed to initialize the propagation algorithm, where it should not be placed. How many points in time should the Gaussian initialization be done for? If more than one points in time what would the center of subsequent Gaussians for the algorithm to work?

5) Explain in your report: how many points to use for the spatial domain; and how many points for the time evolution?

More hints for Project 2:

1) Implement a Fortran code to a) Initialize the propagation with the Gaussian pulse b) The 1D FDTD algorithm c)Writing results out into a file

*How to write the Fortran code?

Look at the example we did for DFT, use Notepad, start writing in column #7, start with "program"

then variable declaration (real fx(100,101); integer nn, nx, ibar) and definition (dt=0.1; pi=3.141592654); then do initialization; then propagation. The code should end with the statement "end"

```
*     This code propagates a Gaussian pulse in 1D using FDTD
*     Physics 640, Sept 24, 2007
      program prop

*     Variable declaration
      real fx(100,101), var, sigma, dx, dt, c
      integer nx, nn, ibar
*     parameter definition
      pi=3.141592654
      .....
      do 20 i=1,nx
        do 10 k=1,nn+1
          fx(i,k)=0.
10    continue
20    continue

*     initialization (Gaussian pulse)
      .....


*     propagation (FDTD in 1D)

      .....

*     output
      open(10, file='prop.prb')
      do 70 n=1,nn+1
      do 60 i=1,nx
        write(10, *) fx(i,n)
60    continue
70    continue
```

```
       close(10)
       end
```

Initialization: for fx at a fixed time instant (second index of fx); need one “do” loop over spatial indices i

Propagation: for fx; need two “do” loops over spatial and time indices i and k

2) Write a Matlab code to read in the output file; then plot; then grab frames to make a movies file

```
%Code to read in output file from prop.f and visualize the wave
propagation
%in 1D
%Physics 640 Sept 25, 2007

clear all;
close all;

load prop.prb;
. . . .

% Make file reading into 2D array


    ......

%Movies

frname='prop1D.avi';
avi=avifile(frname,'compression','none');
for j = 1:nn+1
plot(fx(:,j)), axis([0 100 0 10])
        Fb = getframe;
        avi = addframe(avi,Fb);
        clear Fb;
end;
avi=close(avi);
```

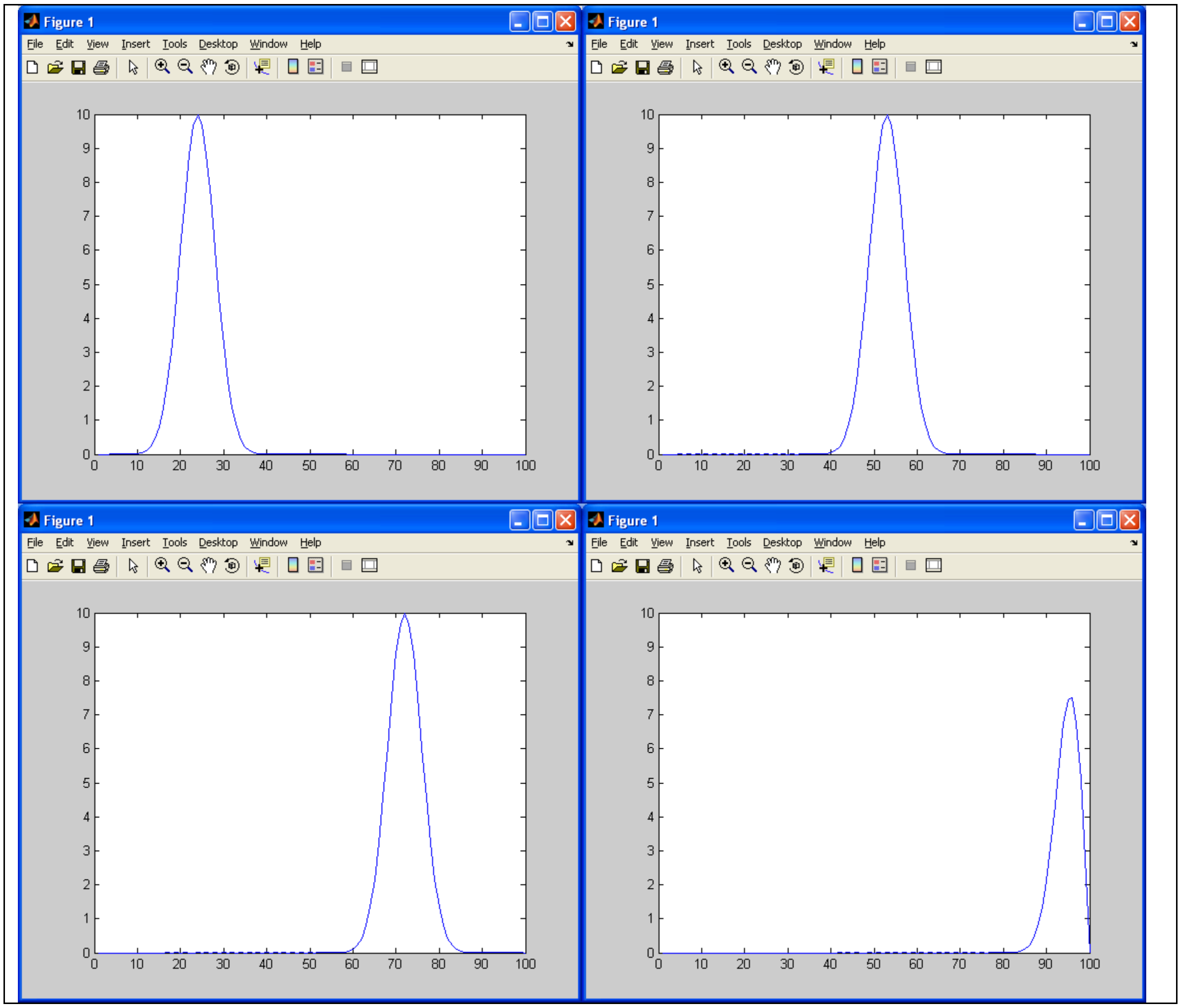


Figure 1
File Edit View Insert Tools Desktop Window Help
10
9
8
7
6
5
4
3
2
1
0
0 10 20 30 40 50 60 70 80 90 100
Figure 1
File Edit View Insert Tools Desktop Window Help
10
9
8
7
6
5
4
3
2
1
0
0 10 20 30 40 50 60 70 80 90 100
Figure 1
File Edit View Insert Tools Desktop Window Help
10
9
8
7
6
5
4
3
2
1
0
0 10 20 30 40 50 60 70 80 90 100
Figure 1
File Edit View Insert Tools Desktop Window Help
10
9
8
7
6
5
4
3
2
1
0
0 10 20 30 40 50 60 70 80 90 100