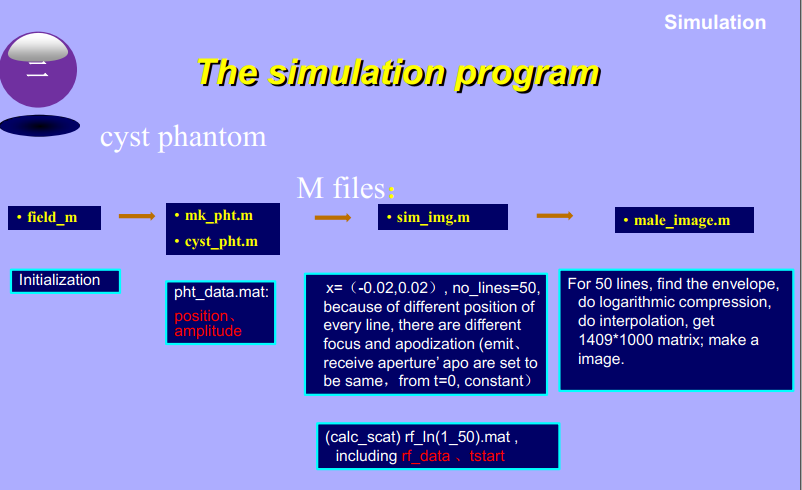
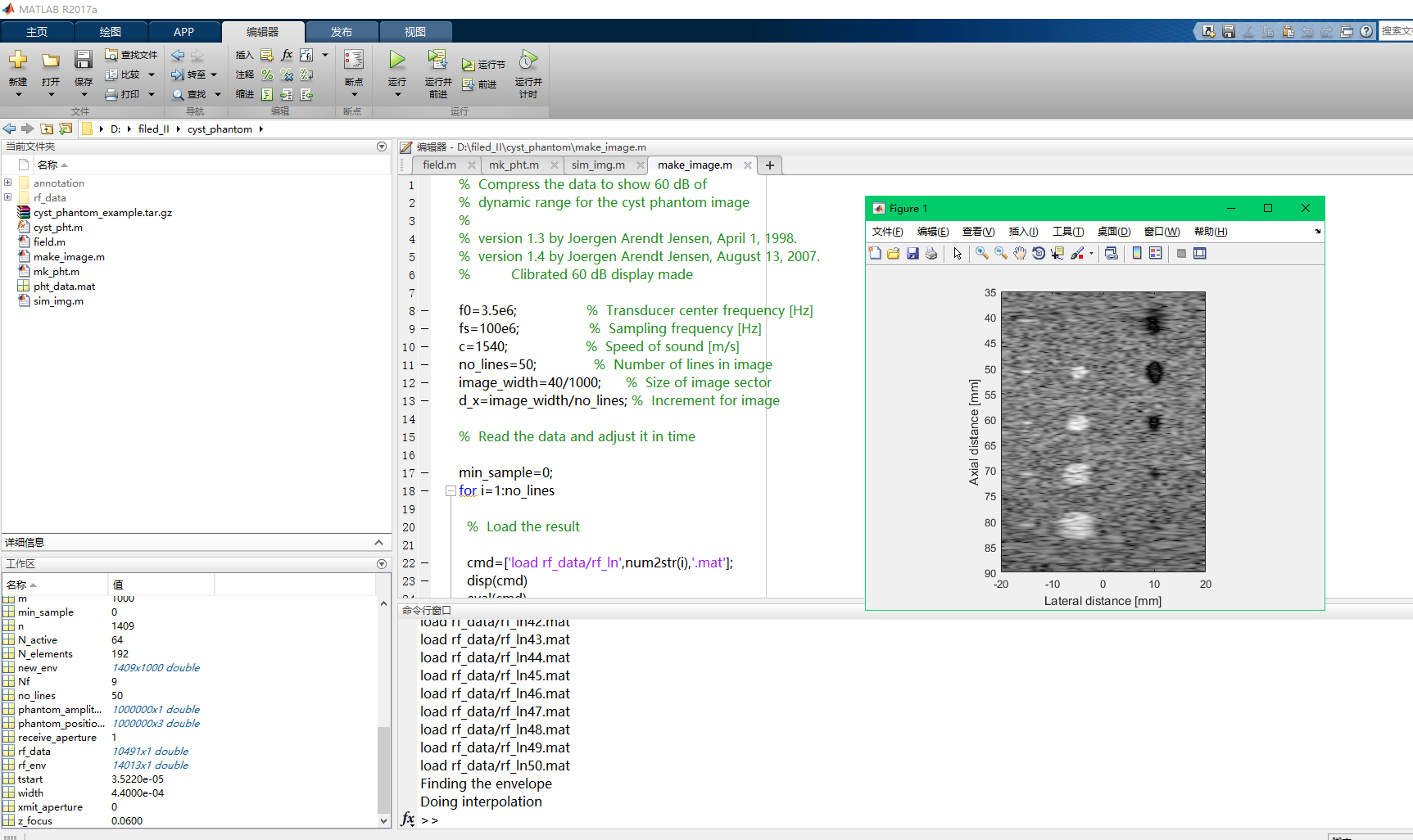
超声仿真软件使用过程简介

## 一 看实例囊肿超声程序使用



**注：cyst\_pht.m不需要运行**

生成图像

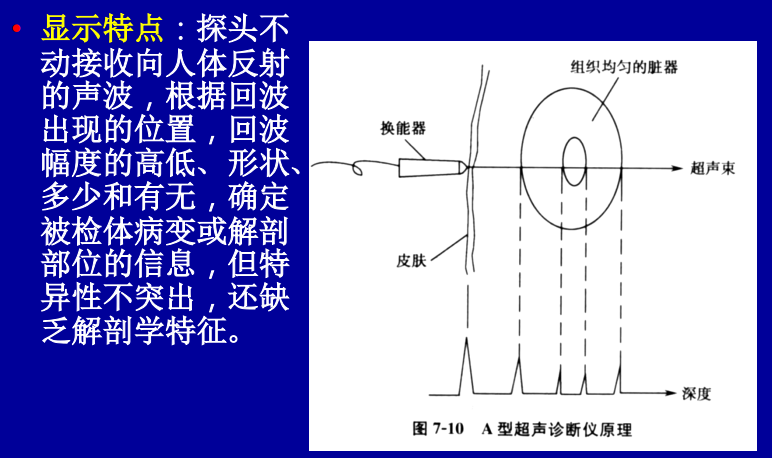


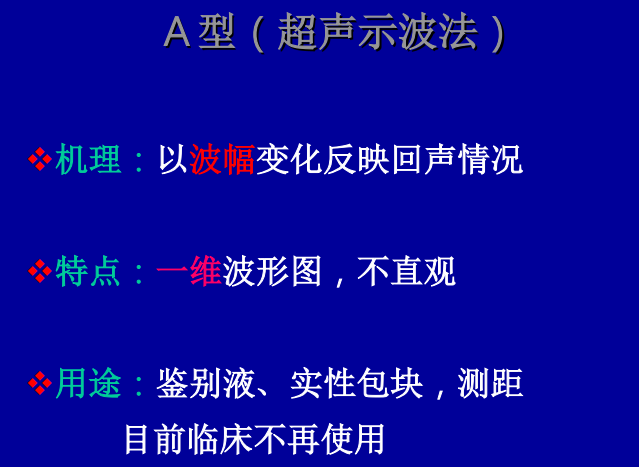
## 一 超声原理

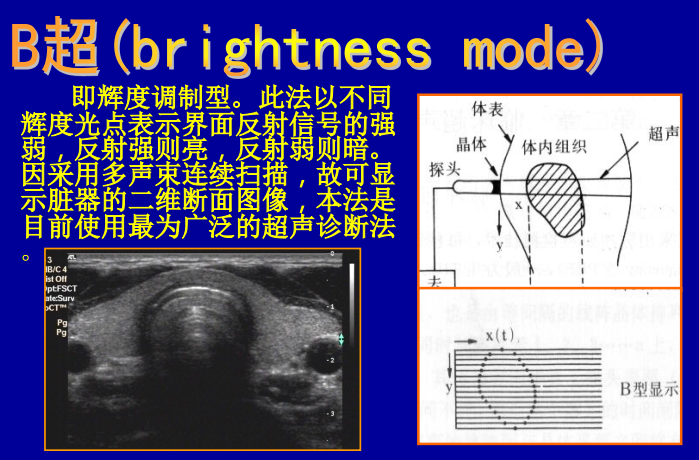
### 1.1 超声分类



### 1.2 超声原理



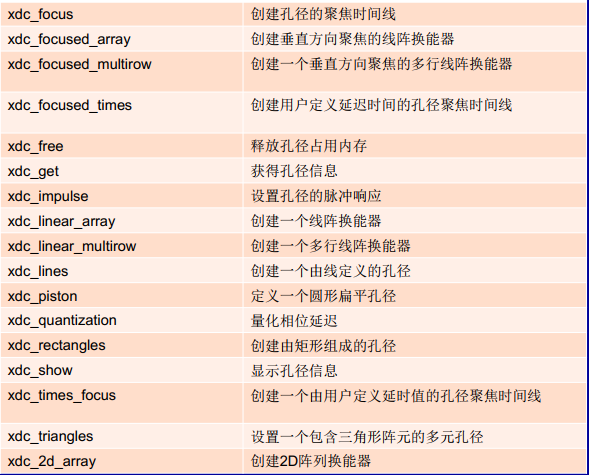




## 二 field\_II基本命令











## 二 囊肿程序一步步注解

### field.m

path(path, 'D:\研一上学期课件\医学超声\Field\_II\_PC7')

field\_init

### cyst\_pht.m囊肿函数

% Create a computer model of a cyst phantom. The phantom contains

% fiven point targets and 6, 5, 4, 3, 2 mm diameter waterfilled cysts,

% and 6, 5, 4, 3, 2 mm diameter high scattering regions. All scatterers

% are situated in a box of (x,y,z)=(50,10,60) mm and the box starts

% 30 mm from the transducer surface.

%构建一个计算机囊肿仿真模型 在一个（x,y,z）=（50，10，60）mm矩形框内，分别放置5个点目标

%直径为6，5，4，3, 2mm液性囊肿 以及直径为6,5,4,3,2mm高回声囊肿 探头表面从30mm开始

% Calling: [positions, amp] = cyst\_phantom (N);

% Parameters: N - Number of scatterers in the phantom

% Output: positions - Positions of the scatterers.

% amp- amplitude of the scatterers.

% Version 2.2, April 2, 1998 by Joergen Arendt Jensen

function [positions, amp] = cyst\_phantom (N)

x\_size = 50/1000; % Width of phantom [mm]

y\_size = 10/1000; % Transverse width of phantom [mm]

z\_size = 60/1000; % Height of phantom [mm]

z\_start = 30/1000; % Start of phantom surface [mm];

% Create the general scatterers

x = (rand (N,1)-0.5)\*x\_size;

y = (rand (N,1)-0.5)\*y\_size;

z = rand (N,1)\*z\_size + z\_start;

% Generate the amplitudes with a Gaussian distribution

amp=randn(N,1);

% Make the cyst and set the amplitudes to zero inside

% 6 mm cyst

r=6/2/1000; % Radius of cyst [mm]

xc=10/1000; % Place of cyst [mm]

zc=10/1000+z\_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2);

%之所以这么设置参数（不用设置yc）和B超扫描方式是一一对应的

%amp = amp .\* (1-inside);

% 5 mm cyst

r=5/2/1000; % Radius of cyst [mm]

zc=20/1000+z\_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2);

amp = amp .\* (1-inside);

% 4 mm cyst

r=4/2/1000; % Radius of cyst [mm]

zc=30/1000+z\_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2);

amp = amp .\* (1-inside);

% 3 mm cyst

r=3/2/1000; % Radius of cyst [mm]

zc=40/1000+z\_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2);

amp = amp .\* (1-inside);

% 2 mm cyst

r=2/2/1000; % Radius of cyst [mm]

zc=50/1000+z\_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2);

amp = amp .\* (1-inside);

% Make the high scattering region and set the amplitudes to 10 times inside

% 6 mm region

r=5/2/1000; % Radius of cyst [mm]

xc=-5/1000; % Place of cyst [mm]

zc=50/1000+z\_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2) ;

amp = amp .\* (1-inside) + 10\*amp .\* inside;

%可以理解的是为什么amp加上10\*amp\*inside（因为是实性囊肿 所以反射弧度更大） 但是值得好奇的是

%为什么是乘inside

% 5 mm region

r=4/2/1000; % Radius of cyst [mm]

zc=40/1000+z\_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2) ;

amp = amp .\* (1-inside) + 10\*amp .\* inside;

% 4 mm region

r=3/2/1000; % Radius of cyst [mm]

zc=30/1000+z\_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2) ;

amp = amp .\* (1-inside) + 10\*amp .\* inside;

% 3 mm region

r=2/2/1000; % Radius of cyst [mm]

zc=20/1000+z\_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2) ;

amp = amp .\* (1-inside) + 10\*amp .\* inside;

% 2 mm region

r=1/2/1000; % Radius of cyst [mm]

zc=10/1000+z\_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2) ;

amp = amp .\* (1-inside) + 10\*amp .\* inside;

% Place the point scatterers in the phantom

for i=N-5:N

x(i) = -15/1000;

y(i) = 0;

z(i) = z\_start + (10+5\*10)/1000 + (i-N)\*10/1000;

amp(i) = 20;

%就是不知道这个是什么意思 为什么是这样写

end

% Return the variables

positions=[x y z];

### mk\_pht.m

% Make the scatteres for a simulation and store

% it in a file for later simulation use

% Joergen Arendt Jensen, Feb. 26, 1997

[phantom\_positions, phantom\_amplitudes] = cyst\_pht(100000);

save pht\_data.mat phantom\_positions phantom\_amplitudes

### sim\_img.m

% Example of use of the new Field II program running under

% Matlab.

%

% This example shows how a linear array B-mode system scans an image

%B超扫描图像

% This script assumes that the field\_init procedure has been called

% Here the field simulation is performed and the data is stored

% in rf-files; one for each rf-line done. The data must then

% subsequently be processed to **yield** the image. The data for the

% scatteres are read from the file pht\_data.mat, so that the procedure

% can be started again or run for a number of workstations.

%

% Example by Joergen Arendt Jensen and Peter Munk,

% Version 1.2, August 14, 1998, JAJ.

% Ver. 1.1: 1/4-98: Procedure xdc\_focus\_center inserted to use the new

% focusing scheme for the Field II program

% Ver. 2.0: 13/8 2007: Parallel version that checks whether the simulation

% of a line has been made before, which makes it possible

% to run the code in parallel on multiple workstations.

% Generate the transducer apertures for send and receive

f0=3.5e6; % Transducer center frequency [Hz]

fs=100e6; % Sampling frequency [Hz]

%接收信号的采样

c=1540; % Speed of sound [m/s]

lambda=c/f0; % Wavelength [m]

%发射波长

width=lambda; % Width of element

element\_height=5/1000; % Height of element [m]

%为什么单位高是这个值 还有就是为什么要设置单位高

kerf=0.05/1000; % Kerf [m]

%Distance in x-direction between elements

focus=[0 0 70]/1000; % Fixed focal point [m]

N\_elements=192; % Number of physical elements

N\_active=64; % Number of active elements

% Set the sampling frequency

set\_sampling(fs);

% Generate aperture for emission

%创建一个线阵列换能器

xmit\_aperture = xdc\_linear\_array (N\_elements, width, element\_height, kerf, 1, 10,focus);

% Set the impulse response and excitation of the xmit aperture

impulse\_response=sin(2\*pi\*f0\*(0:1/fs:2/f0));

impulse\_response=impulse\_response.\*hanning(max(size(impulse\_response)))';

%设置孔径脉冲响应

xdc\_impulse (xmit\_aperture, impulse\_response);

excitation=sin(2\*pi\*f0\*(0:1/fs:2/f0));

%设置孔径的激励脉冲

xdc\_excitation (xmit\_aperture, excitation);

% Generate aperture for reception

receive\_aperture = xdc\_linear\_array (N\_elements, width, element\_height, kerf, 1, 10,focus);

% Set the impulse response for the receive aperture

xdc\_impulse (receive\_aperture, impulse\_response);

% Load the computer phantom

if ~exist('pht\_data.mat')

disp('Scatterer positions should be made by the script mk\_pht')

disp('before this script can be run')

return

else

load pht\_data

end

% Set the different focal zones for reception

focal\_zones=[30:20:200]'/1000;

Nf=max(size(focal\_zones));

focus\_times=(focal\_zones-10/1000)/1540;

z\_focus=60/1000; % Transmit focus

% Set the apodization

apo=hanning(N\_active)';

% Do linear array imaging

no\_lines=50; % Number of lines in image

image\_width=40/1000; % Size of image sector

d\_x=image\_width/no\_lines; % Increment for image

% Do imaging line by line

for i=[1:no\_lines]

% Test if the file for the line exist.

% Skip the simulation, if the line exits and

% go the next line. Else make the simulation

file\_name=['rf\_data/rf\_ln',num2str(i),'.mat'];

if ~exist(file\_name)

% Save a file to reserve the calculation

cmd=['save rf\_data/rf\_ln',num2str(i),'.mat i'];

eval(cmd);

disp(['Now making line ',num2str(i)])

% The the imaging direction

x= -image\_width/2 +(i-1)\*d\_x;

% Set the focus for this direction with the proper reference point

%设置动态聚焦线原点

xdc\_center\_focus (xmit\_aperture, [x 0 0]);

%创建孔径聚焦时间线

xdc\_focus (xmit\_aperture, 0, [x 0 z\_focus]);

%设置动态聚焦线原点

xdc\_center\_focus (receive\_aperture, [x 0 0]);

xdc\_focus (receive\_aperture, focus\_times, [x\*ones(Nf,1), zeros(Nf,1), focal\_zones]);

% Calculate the apodization

N\_pre = round(x/(width+kerf) + N\_elements/2 - N\_active/2);

N\_post = N\_elements - N\_pre - N\_active;

apo\_vector=[zeros(1,N\_pre) apo zeros(1,N\_post)];

xdc\_apodization (xmit\_aperture, 0, apo\_vector);

xdc\_apodization (receive\_aperture, 0, apo\_vector);

% Calculate the received response

[rf\_data, tstart]=calc\_scat(xmit\_aperture, receive\_aperture, phantom\_positions, phantom\_amplitudes);

% Store the result

cmd=['save rf\_data/rf\_ln',num2str(i),'.mat rf\_data tstart'];

disp(cmd)

eval(cmd);

else

disp(['Line ',num2str(i),' is being made by another machine.'])

end

end

% Free space for apertures

xdc\_free (xmit\_aperture)

xdc\_free (receive\_aperture)

disp('You should now run make\_image to display the image')

### make\_image

% Compress the data to show 60 dB of

% dynamic range for the cyst phantom image

%

% version 1.3 by Joergen Arendt Jensen, April 1, 1998.

% version 1.4 by Joergen Arendt Jensen, August 13, 2007.

% Clibrated 60 dB display made

f0=3.5e6; % Transducer center frequency [Hz]

fs=100e6; % Sampling frequency [Hz]

c=1540; % Speed of sound [m/s]

no\_lines=50; % Number of lines in image

image\_width=40/1000; % Size of image sector

d\_x=image\_width/no\_lines; % Increment for image

% Read the data and adjust it in time

min\_sample=0;

for i=1:no\_lines

% Load the result

cmd=['load rf\_data/rf\_ln',num2str(i),'.mat'];

disp(cmd)

eval(cmd)

% Find the envelope

rf\_env=abs(hilbert([zeros(round(tstart\*fs-min\_sample),1); rf\_data]));

env(1:max(size(rf\_env)),i)=rf\_env;

end

% Do logarithmic compression

D=10; % Sampling frequency decimation factor

disp('Finding the envelope')

log\_env=env(1:D:max(size(env)),:)/max(max(env));

log\_env=20\*log10(log\_env);

log\_env=127/60\*(log\_env+60);

% Make an interpolated image

disp('Doing interpolation')

ID=20;

[n,m]=size(log\_env);

new\_env=zeros(n,m\*ID);

for i=1:n

new\_env(i,:)=abs(interp(log\_env(i,:),ID));

end

[n,m]=size(new\_env);

fn=fs/D;

clf

image(((1:(ID\*no\_lines-1))\*d\_x/ID-no\_lines\*d\_x/2)\*1000,((1:n)/fn+min\_sample/fs)\*1540/2\*1000,new\_env)

xlabel('Lateral distance [mm]')

ylabel('Axial distance [mm]')

colormap(gray(127))

axis('image')

axis([-20 20 35 90])

## 出现问题及解决方案

### 1 中文显示方块

