

MATLAB|**绘图辅助工具箱**tools使用说明

[1. 引言](#)

[2. 功能简介](#)

[3. 标准参数对话框](#)

[3.1 问题描述](#)

[3.2 功能要点](#)

[3.3 基本调用方法](#)

[3.4 高级调用功能](#)

[4. 文件读取模块](#)

[4.1 文件路径读取](#)

[函数说明](#)

[运行示例](#)

[4.2 获取完整文件名](#)

[函数说明](#)

[运行示例](#)

[4.3 读取CSV数据文件](#)

[函数说明](#)

[CSV数据说明](#)

[运行示例](#)

[4.4 读取TXT数据文件](#)

[函数说明](#)

[数据格式](#)

[运行示例](#)

[4.5 读取mat文件](#)

[函数说明](#)

[数据格式](#)

[运行示例](#)

[5. 绘图辅助](#)

[5.1 坐标轴与标题便捷标注](#)

[函数说明](#)

[运行示例](#)

[5.2 图像刷白](#)

[函数功能](#)

[运行示例](#)

[5.3 图像中贯通直线绘制](#)

[函数说明](#)

[问题描述](#)

[运行示例](#)

[5.4 网格线增加绘制](#)

[函数说明](#)

[运行示例](#)

[5.5 当前图像保存为图片文件](#)

[函数说明](#)

[运行示例](#)
[快捷使用技巧](#)

[5.6 默认颜色序列](#)

[函数说明](#)
[运行示例](#)

[5.7 典型快捷绘图](#)

[函数说明](#)
[运行示例](#)

[6. 数据操作](#)

[6.1 数字与字符串序列粘贴](#)

[函数说明](#)
[运行示例](#)

[6.2 向量数值区间确定](#)

[函数说明](#)
[运行示例](#)

[6.3 行形式的数据转换列矩阵](#)

[函数说明](#)
[运行示例](#)

[6.4 复数矩阵的归一化](#)

[函数说明](#)
[运行示例](#)

[7. 信号处理](#)

[7.1 信号去趋势](#)

[函数说明](#)
[运行示例](#)

[7.2 低通滤波器设计](#)

[函数说明](#)
[运行示例](#)

[7.3 多波峰正弦信号生成](#)

[函数说明](#)
[运行示例](#)

[7.4 激励信号带宽计算](#)

[函数说明](#)
[运行示例](#)

[8. 其他函数](#)

[8.1 工具箱版本显示](#)

[函数说明](#)
[运行示例](#)

[9. 使用技巧](#)

[9.1 工具箱的路径设置](#)

[9.2 函数帮助信息显示](#)

[10. 更新与下载](#)

1. 引言

针对特定的研究目标相应的MATLAB程序中存在大量的重复模块，为了提高工作效率和程序的复用性，将本人经常使用的MATLAB程序模块重组设计成绘图辅助工具箱。

2. 功能简介

3. 标准参数对话框

3.1 问题描述

MATLAB除了数值计算代码的编写，我们往往修改**输入参数**，作为研究变量。如果直接在代码脚本中修改有两个问题：

- 修改不便，必须进入代码内部，找出相关变量；
- 不熟悉程序的人有可能将程序修改错误，得到错误的结果。

解决这一问题的一种方案是：采用**参数对话框**输入。但是，MATLAB自带的 `inputdlg` 函数功能有限，调用较为麻烦，因此在此基础上订制开发了 `paradlg`。

3.2 功能要点

该函数的功能要点为：

- 创建标准化对话框
- 支持多种数据格式：标量、向量、字符串，自动判别
- 记忆上次输入参数

函数说明如下：

```
function para = paradlg(prompt0,dlg0,isShow)
% 题目:标准化对话框创建程序
% 输入:
%     prompt0    -- 提示语以及默认参数, n*2
%     dlg0       -- 可选定制化参数
%     .width     -- 对话框宽度
%     .title     -- 标题
%     .save      -- data_dlg后缀名
%     isShow     -- 是否弹出对话框,
% 输出:
%     para       -- 对话框输入参数, 默认弹出, 若isShow=0, 则不弹出, 数值取上次默认值
% 功能:
%     创建标准化参数输入对话框
%     支持 标量、向量、字符串
%     导出输入参数
%     记忆上次输入
```

3.3 基本调用方法

- 构造 `prompt0` 数据，第一列为输入提示，第二列为默认参数；
- 支持三类数据输入，标量、向量、字符串；
- 调用 `paradlg` 函数，返回输入的数据集 `para`；
- 将 `para` 中的元素依次赋值给目标的变量 `f_center`, `filename_pub`, `dt`, `f_range`；

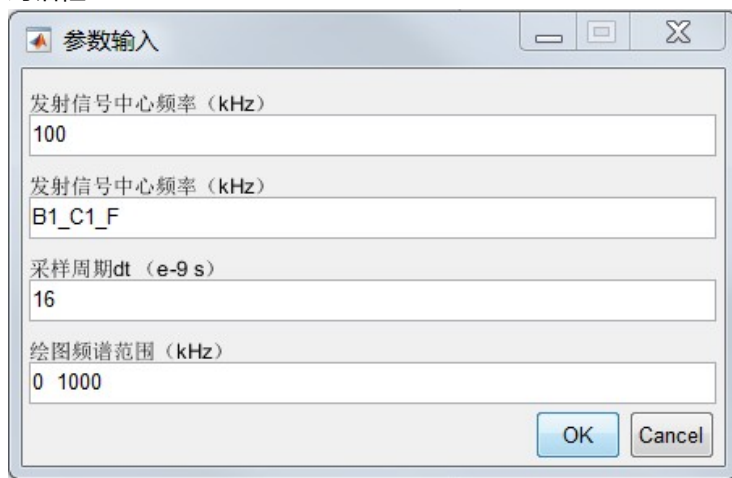
- 调用代码示例

```
prompt0 = { % 对话框参数
    '发射信号中心频率 ( kHz ) ', 100
    '发射信号中心频率 ( kHz ) ', 'B1_C1_F'
    '采样周期dt ( e-9 s ) ', 16
    '绘图频谱范围 ( kHz ) ', [0 20]
};

para = paradlg(prompt0);

f_center = para{1}; % 发射信号中心频率 ( kHz )
filename_pub = para{2}; % 输出报告文件名
dt = para{3}; % 采样周期
f_range = para{4};
```

- 对话框



- 参数读取结果

```
para =
    [    100]
    'B1_C1_F'
    [    16]
    [1x2 double]
```

dt	16
f_center	100
f_range	[0, 20]
filename_pub	'B1_C1_F'

3.4 高级调用功能

如需控制对话框的**宽度**、**标题**，则在函数 `paradlg` 中输入第二个变量 `dlg0`。

```
%      dlg0.width -- 对话框宽度
%      dlg0.title -- 对话框标题
%      dlg0.save  -- data_dlg后缀名
```

- 调用代码

```
prompt0 = { % 对话框参数
```

```

    '发射信号中心频率 ( kHz ) ', []
    '发射信号中心频率 ( kHz ) ', 'B1_C1_F'
    '采样周期dt ( e-9 s ) ', 16
    '绘图频谱范围 ( kHz ) ', [0 20]
};

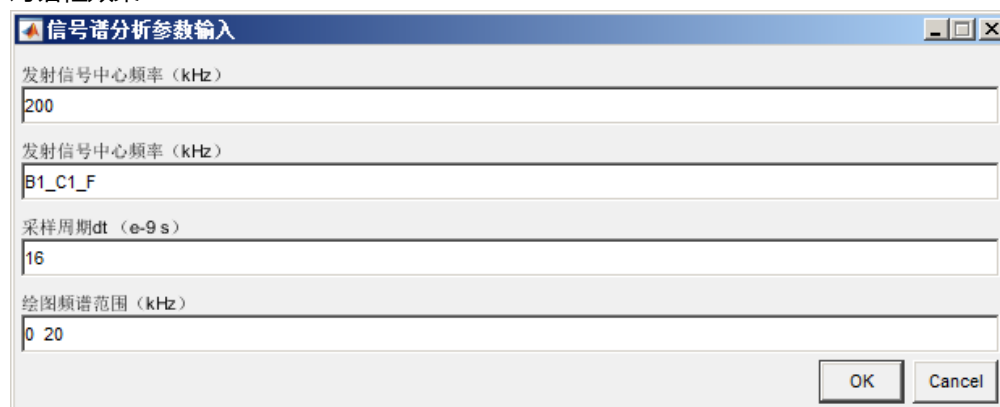
dlg0.width = 100;
dlg0.title = '信号谱分析参数输入';
dlg0.save = 'freqpara';

para = tools.paradlg(prompt0,dlg0);

f_center = para{1}; % 发射信号中心频率 ( kHz )
filename_pub = para{2}; % 输出报告文件名
dt = para{3}; % 采样周期
f_range = para{4};

```

- 对话框效果



对于同一个程序，如果多次参数没有改变，可以让对话框不弹出：

```
para = tools.paradlg(prompt0,dlg0,0);
```

在不弹出对话框的情况下，直接载入上次设定的参数。

4. 文件读取模块

笔者自定义的文件读取模块，通过对系统默认函数改造得到，主要的改进是增加了**路径记忆**。

4.1 文件路径读取

函数说明

```

function [dir_name] = getdir()
% 题目：获取文件夹名称
% 输出：
%       dir_name -- 文件夹路径

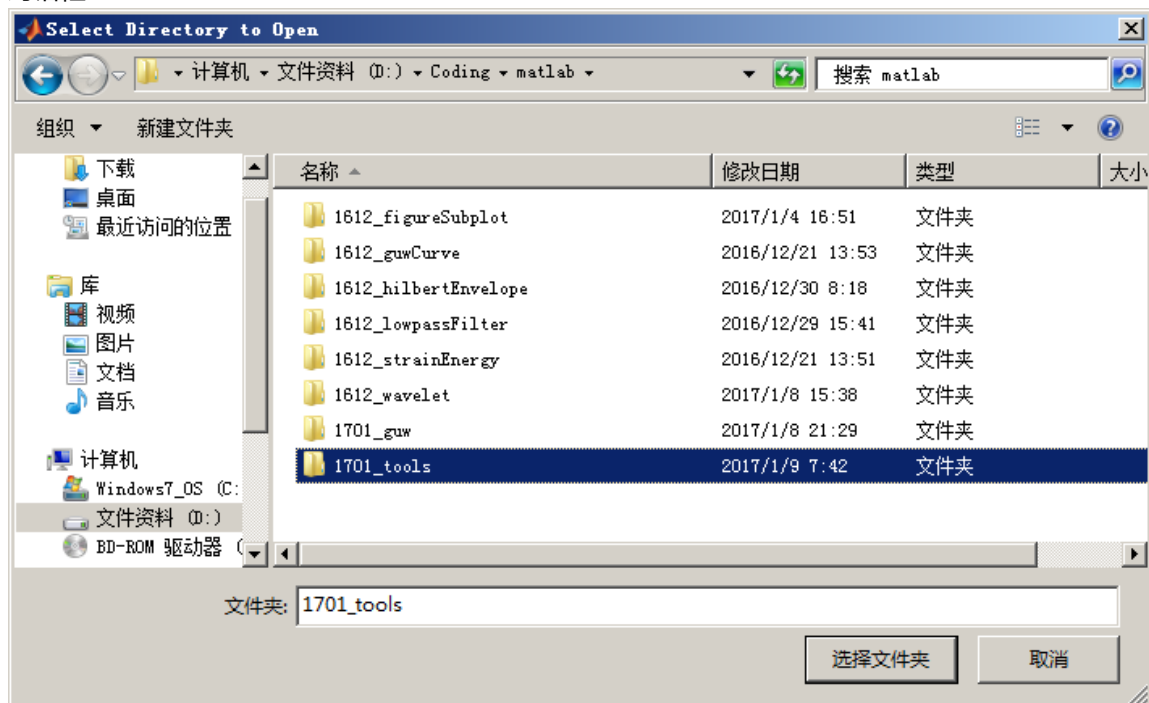
```

运行示例

- 运行函数

```
mydir = tools.getdir;
```

- 对话框



- 获取路径

```
mydir =  
D:\Coding\matlab\1701_tools
```

系统默认函数为：`uigetdir`。

4.2 获取完整文件名

函数说明

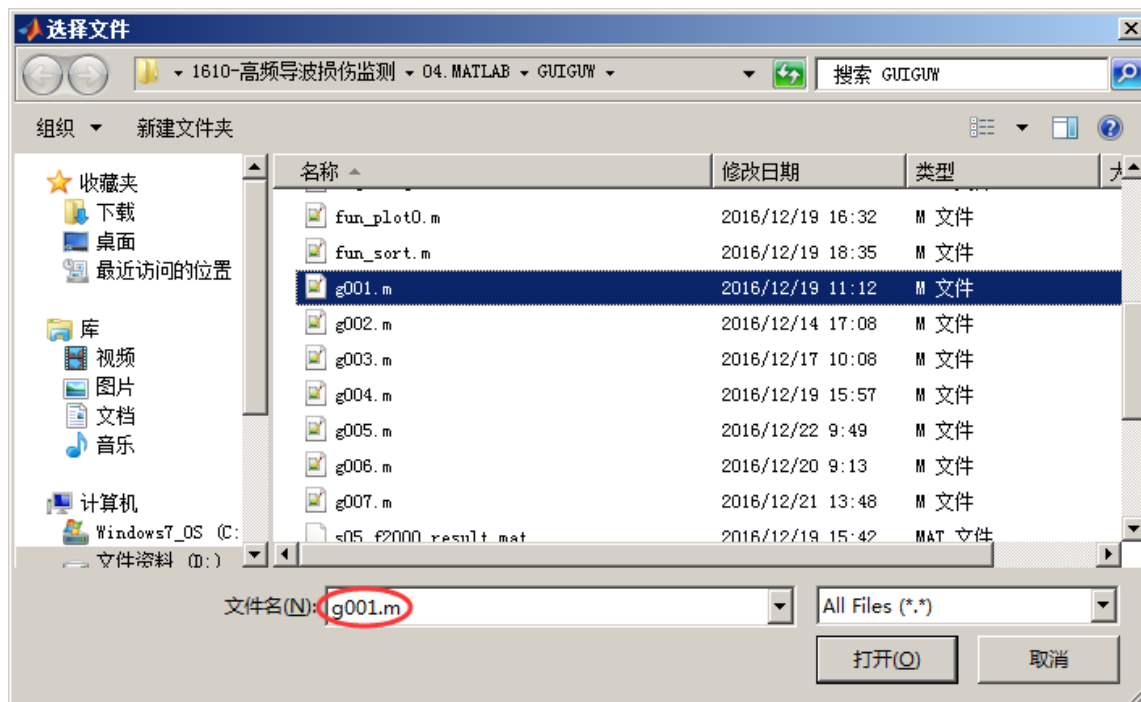
```
function [fullname,pathname,filename] = getfile(type,ext)  
% 题目：读取文件全名、路径、文件名  
% 输入：  
%     type      -- 类型标识  
% 输出：  
%     fullname  -- 全名  
%     pathname  -- 路径  
%     filename  -- 文件名
```

运行示例

- 运行函数

```
[fullname,pathname,filename] = tools.getfile;
```

- 选取文件



- 获取文件名与完整路径

```
fullname =
D:\Projects\1610-高频导波损伤监测\04.MATLAB\GUIGUW\g001.m

pathname =
D:\Projects\1610-高频导波损伤监测\04.MATLAB\GUIGUW\

filename =
g001.m
```

4.3 读取CSV数据文件

函数说明

```
function [data,para0] = getcsv(flag)
% 题目：示波器输出csv数据标准读取
% 输入：
%     flag    -- flag==1，则弹出对话框，否则默认标准参数
%     R0      -- 读取信号数据起始行数
%     R1、C1  -- 特定单元格的位置参数
% 输出：
%     data    -- 信号数据
%     para0   -- 特定单元格参数
% 功能：
%     从R0行开始读取信号数据，直到末尾
%     读取(R1,C1)单元格的参数
```

CSV数据说明

本程序特定针对DPO2024仪器输出的CSV信号数据，其格式为：

```
Model,DPO2024
Firmware Version,1.25

Point Format,Y,
```

```

Horizontal Units,S,
Horizontal Scale,0.0001,
Sample Interval,8e-09,
Filter Frequency,2e+08,
Record Length,125000,
Gating,0.0% to 100.0%,0.0% to 100.0%
Probe Attenuation,10,10
Vertical Units,V,V
Vertical Offset,0,0
Vertical Scale,20,1
Label,,
TIME,CH1,CH2
-1.28000e-04,-0.45,0.0198438
-1.27992e-04,-0.4,0.02
-1.27984e-04,-0.4,0.025
-1.27976e-04,-0.5,0.0151562
...

```

CSV数据在Excel中打开的视图如下：

1	Model	DP02024		
2	Firmware Versi	1.25		
3				
4	Point Format	Y		
5	Horizontal Uni	S		
6	Horizontal Sca	0.0001		
7	Sample Interva	8.00E-09		
8	Filter Frequen	2.00E+08		
9	Record Length	125000		
10	Gating	0.0% to 10.0% to 100.0%		
11	Probe Attenuat	10	10	
12	Vertical Units	V	V	
13	Vertical Offse	0	0	
14	Vertical Scale	20	1	
15	Label			
16	TIME	CH1	CH2	
17	-1.28E-04	-0.45	0.019844	
18	-1.28E-04	-0.4	0.02	
19	-1.28E-04	-0.4	0.025	
20	-1.28E-04	-0.5	0.015156	
21	-1.28E-04	-0.4	0.0175	

数据读取的要点在于：

- 需要跳过文件头的说明信息，以矩阵形式读取序列数据
- 需要读取特定单元格的参数，如Sample Interval

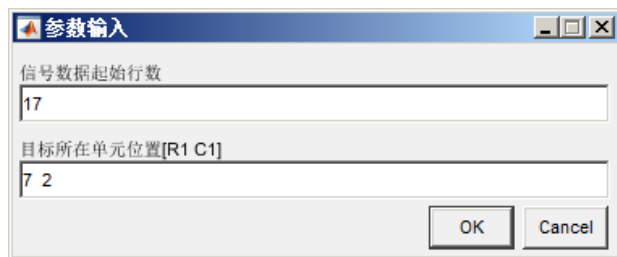
运行示例

- 运行代码

```
[data,para0] = tools.getcsv();
```

- 参数输入

如上所示，有效数据序列从17行开始；要读取7行2列的Sample Interval数据，在对话框输入7 2。



- 数据结果

data	125000x4 double
para0	8.0000e-09

- 不弹出参数对话框运行

第一次设置好读取参数后，后续不需要重复设置，即可记忆以前的设定及文件路径。

```
[data,para0] = tools.getcsv(0);
```

4.4 读取TXT数据文件

函数说明

```
function data = gettxt(nrow_start)
% 题目：读取txt数据文件，跳跃文件头说明行
% 输入：
%     nrow_start  -- 起始行
% 输出：
%     data        -- 信号数据
```

数据格式

一般仪器输出的数据为CSV、raw等格式，而数值模拟输出的数据往往是标准的txt格式。

VA12_S1600_F200_D8_L500.TXT	2016/11/26 16:31	TXT 文件	71 KB
VA12_S1600_F200_D9_L500.TXT	2016/11/26 17:30	TXT 文件	71 KB
VA12_S1600_F200_D10_L250.TXT	2016/11/26 21:24	TXT 文件	71 KB
VA12_S1600_F200_D10_L500.TXT	2016/11/26 18:30	TXT 文件	71 KB

运行示例

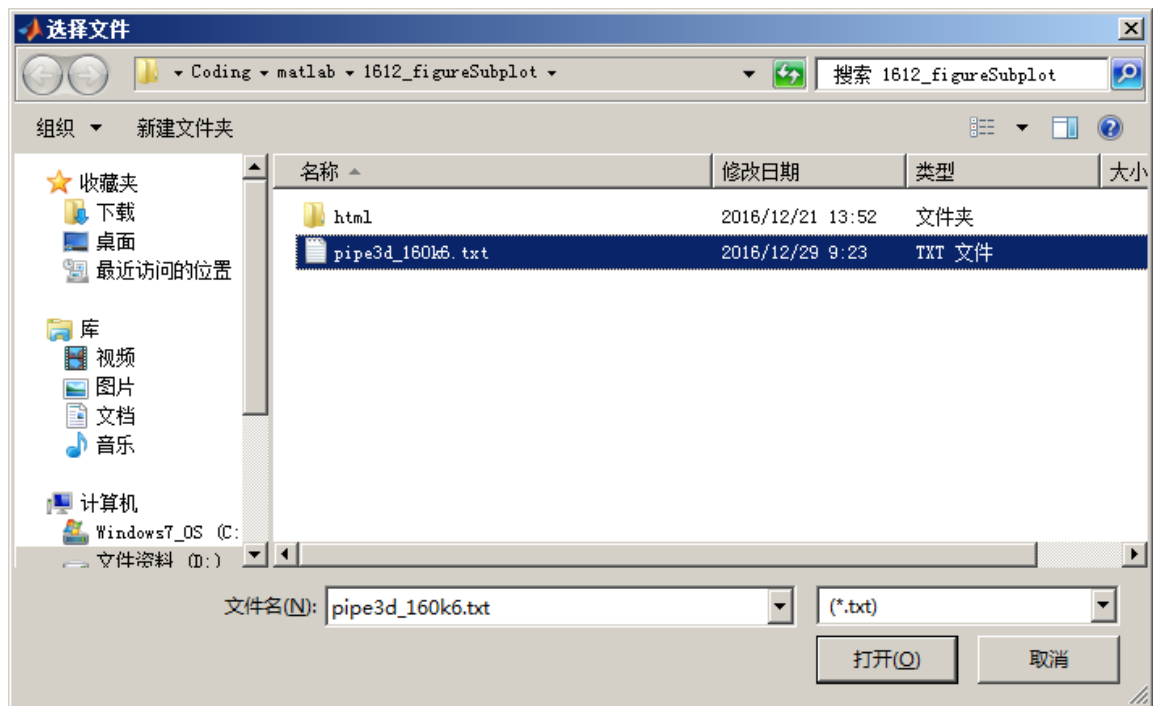
- 直接读取数据

```
data = gettxt()
```

- 跳过10行读取数据

```
data = gettxt(10)
```

- 对话框



`gettxt` 函数尚存一些问题，曾经发生读数出错，如果无法正常使用，可以采用以下方法：

```
filename = tools.getfile();  
data = load(filename);
```

4.5 读取mat文件

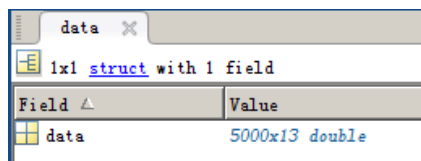
函数说明

```
function data = getmat()  
% 题目：读取只有一个变量的矩阵数据mat文件
```

数据格式

.mat是MATLAB默认的数据格式，一般直接通过命令导入即可，但如果以下面的方式导入：

```
data = load('matlab.mat')
```



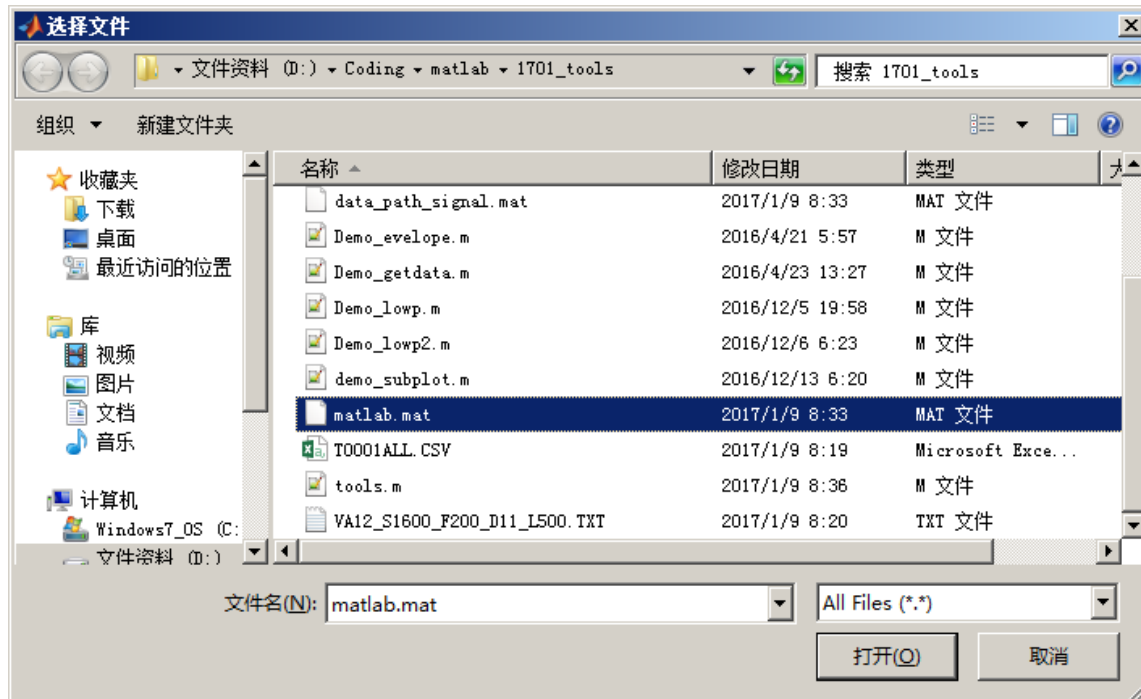
读入的数据会形成一个结构体，不便后续操作。本函数通过数据格式转换，保证读取的数据依然为矩阵形式。

运行示例

- 命令

```
data = tools.getmat;
```

- 对话框



- 结果

Workspace		
Name	Value	
data	5000x13 double	

5. 绘图辅助

5.1 坐标轴与标题便捷标注

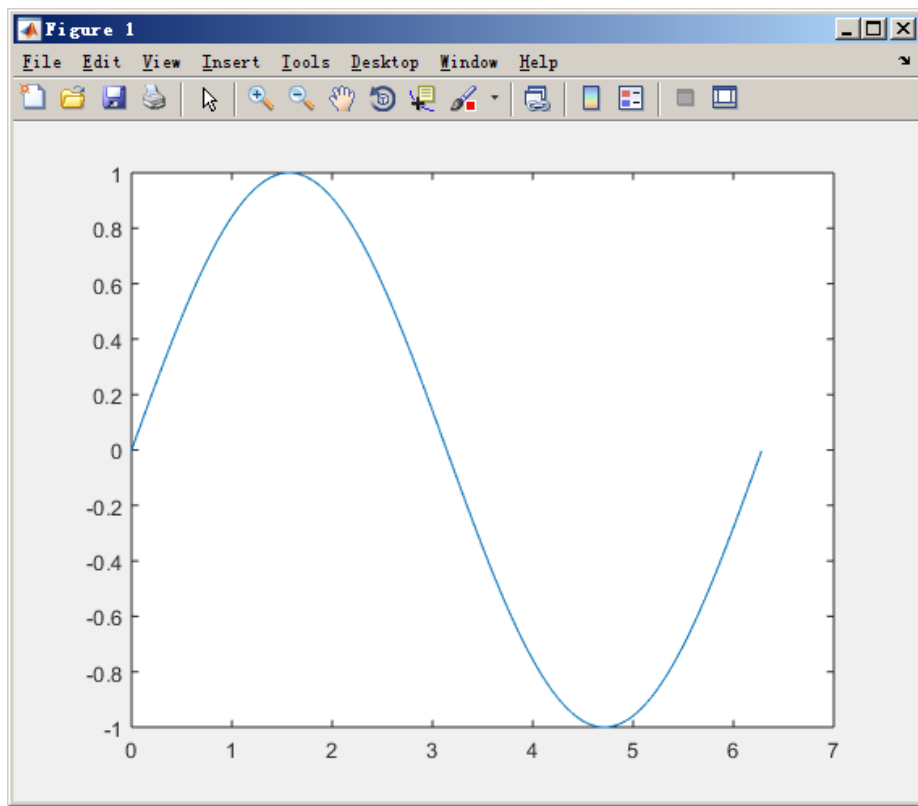
函数说明

```
function xyt(str_xyt)
% 题目：生成xlabel,ylabel,title
```

运行示例

- 绘制典型sine图像

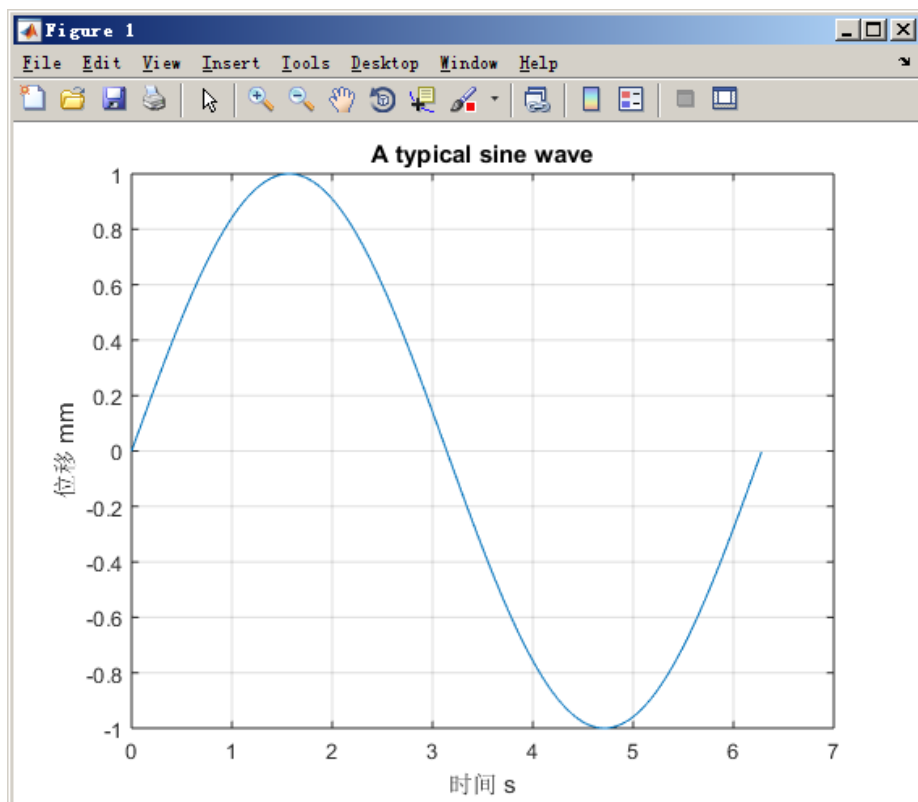
```
figure
tools.plot0; % 绘制一个基
本的sine
```



- 添加坐标轴及标题标注

```
tools.xyt({'时间 s','位移 mm','A typical sine wave'})
```

% 图像标注



以上代码相当于：

```
xlabel('时间 s')  
ylabel('位移 mm')  
title('A typical sine wave')
```

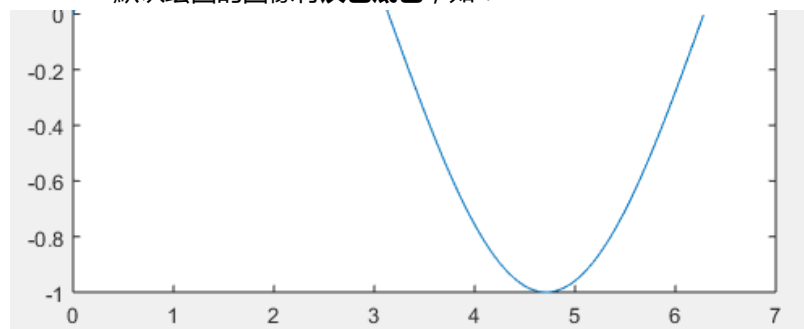
```
grid on
set(gcf,'color','white')
```

代码的集成效率大大提高。

5.2 图像刷白

函数功能

MATLAB默认绘图图像有灰色底色，如：



本函数的功能是将figure底色改为白色，便于截图处理。

```
function white()
% 题目:图像刷白
    set(gcf,'color','white');
    grid on;
end % white
```

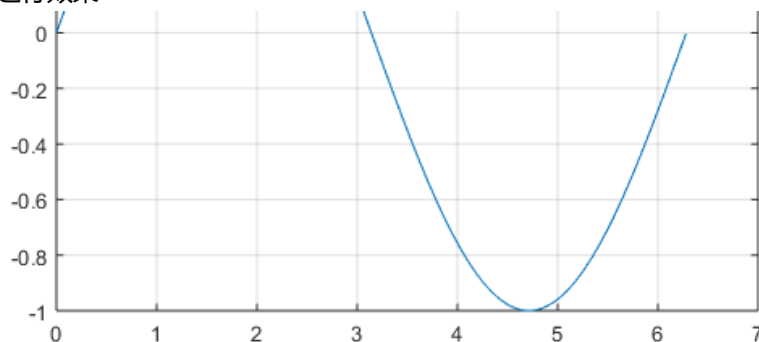
运行示例

- 调用代码

```
figure
tools.plot0;

tools.white;
```

- 运行效果



5.3 图像中贯通直线绘制

函数说明

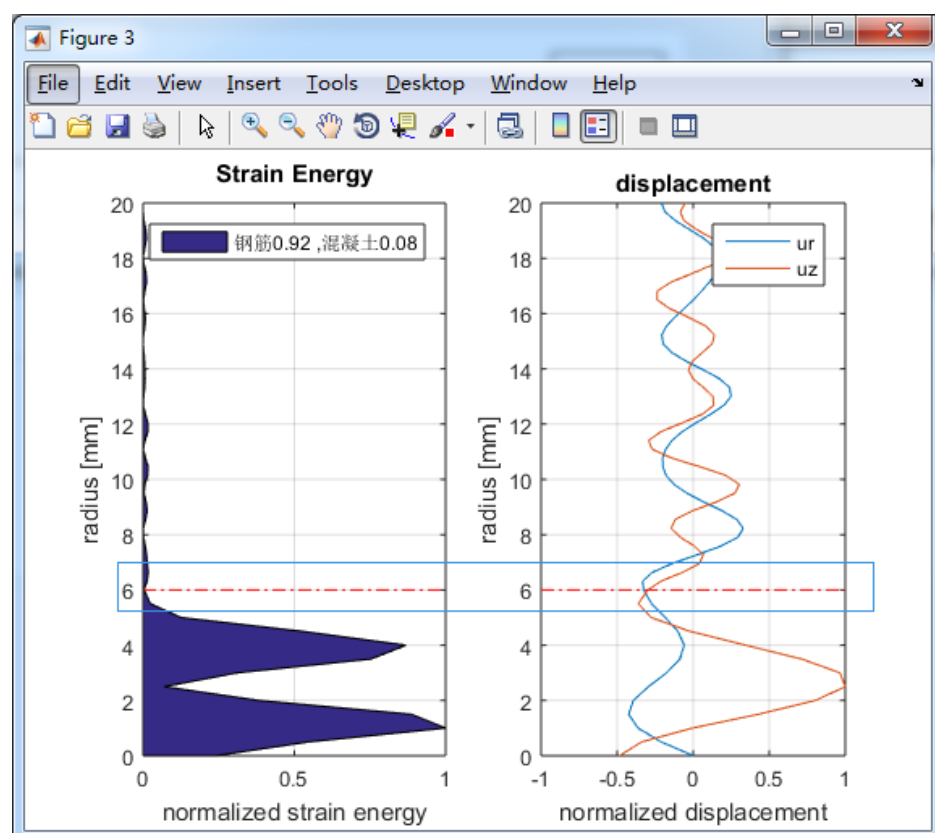
```
function xline(position,lineSpec)
```

```
% 题目：输入MATLAB默认颜色向量
% 输入：
%     position    -- [x y], [1 0]在x=1处绘制竖线，[0 1]在y=1处绘制横线
%     lineSpec    -- 'r-*'
% 输出：辅助直线
% 示例：
%     xline([0 6], 'r-')
```

问题描述

这是一个没有难度，但经常出现的问题。

在MATLAB绘图中，除了基本的数据绘图，我们往往需要绘制辅助性直线。



MATLAB中绘制这样的直线很简单，只需确定两个点即可。但是要确定贯通方向的坐标范围，较为繁琐，还需要根据主体数据进行调整。

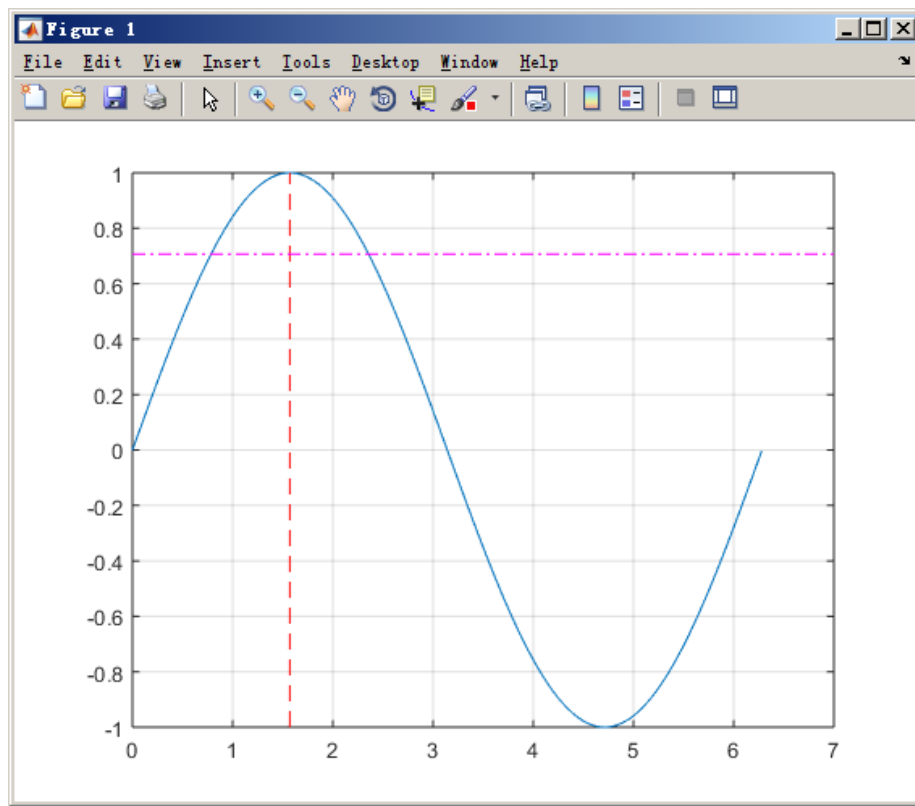
运行示例

- 测试代码

```
figure
tools.plot0;

tools.xline([pi/2,0], 'r--') % 竖向直线
tools.xline([0, sin(pi/4)], 'm-.') % 横向直线
```

- 绘图效果



这一函数的不足之处在于，没有显示出直线所在的坐标刻度。

5.4 网格线增加绘制

函数说明

网格线增加绘制函数有两个：

- xGrid
- yGrid

```
function xGrid(x0,angle)
% 题目：图中增加x网格线
% 时间：2017.01.08
```

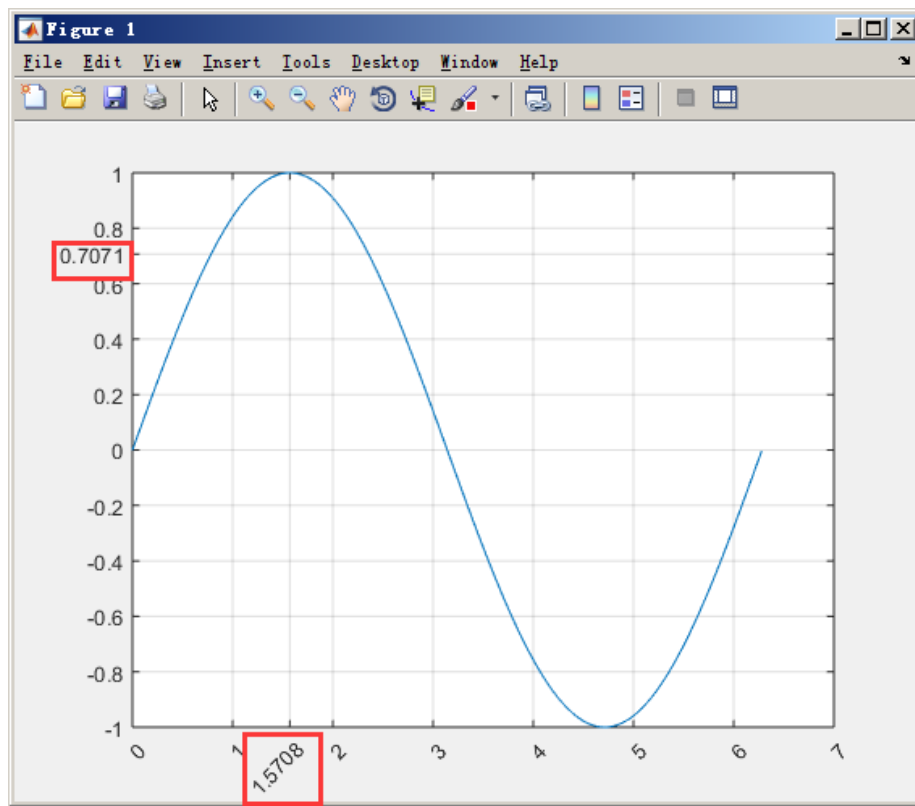
运行示例

- 测试代码

```
figure
tools.plot0;

tools.xGrid(pi/2,45)
tools.yGrid(sin(pi/4))
```

- 运行效果



当坐标轴标签密集，显示不清楚，可以调整标签的角度。

5.5 当前图像保存为图片文件

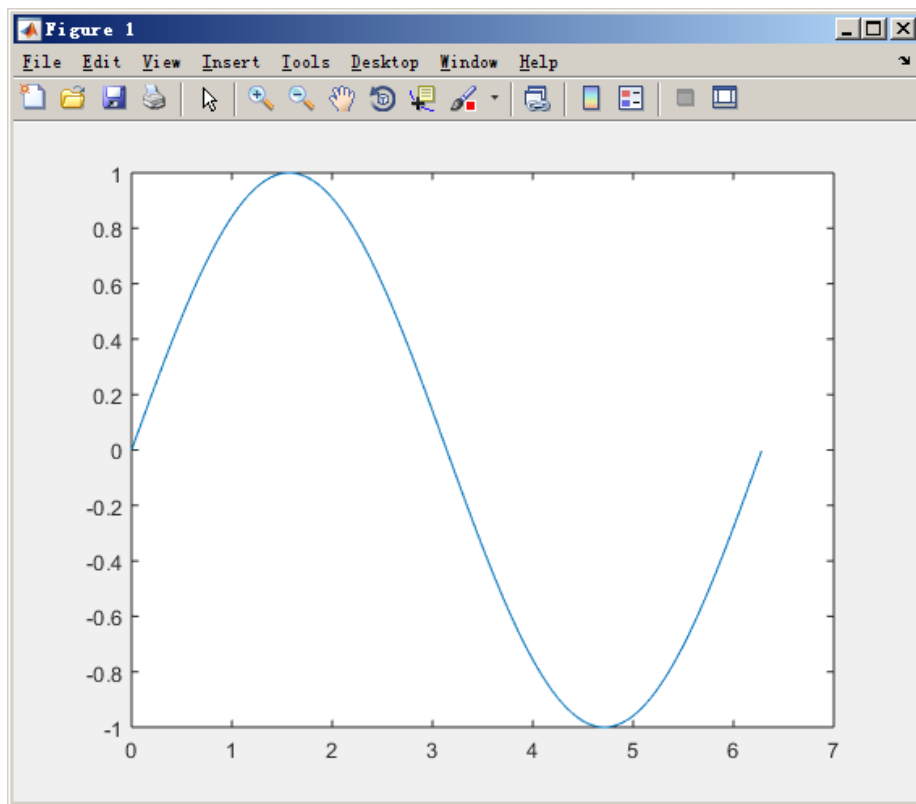
函数说明

```
function saveGraph()  
% 题目：保存gcf图像  
% 功能：  
%     自定义图片格式  
%     自定义图片文件名编号  
% 时间：2017.01.05
```

运行示例

- 基本图像绘制

```
figure  
tools.plot0;
```

– 图像保存对话框

此处可以输入文件名，批量存储图片，可以设置固定的前缀和后缀，以便查阅。

Image Save dialog box fields:

- 编号: 01
- 前缀: test
- 后缀: sine
- 图片格式 (png/jpg): png

– 批量存储图片效果

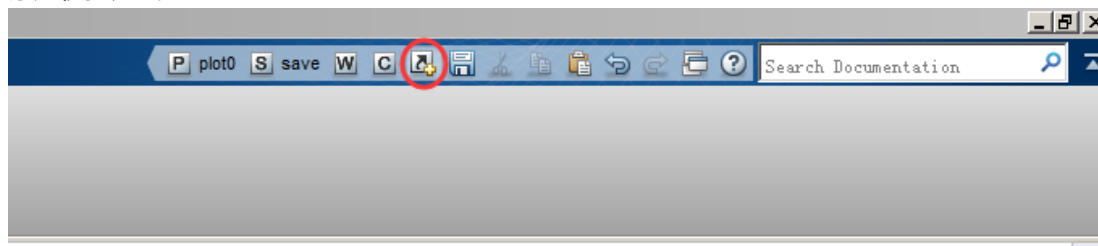
E005a-Day-1-Tone5-cwt. png
E005a-Day-2-Tone5-cwt. png
E005a-Day-3-Tone5-cwt. png
E005a-Day-4-Tone5-cwt. png
E005a-Day-5-Tone5-cwt. png
E005a-Day-6-Tone5-cwt. png
E005a-Day-7-Tone5-cwt. png
E005a-Day-8-Tone5-cwt. png
E005a-Day-9-Tone5-cwt. png
E005a-Day-10-Tone5-cwt. png
E005a-Day-11-Tone5-cwt. png
E005a-Day-12-Tone5-cwt. png
E005a-Day-13-Tone5-cwt. png
E005a-Day-14-Tone5-cwt. png

快捷使用技巧

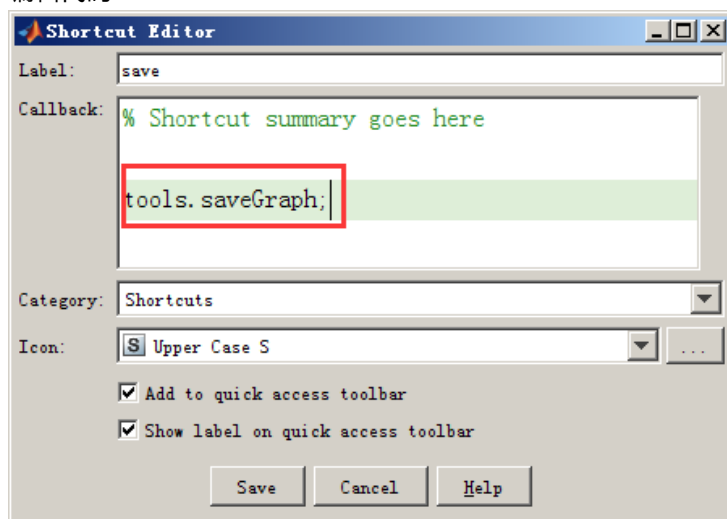
本函数仅支持存储当前figure中的图像，gcf。

此函数需要经常调用，可以设置为**快捷方式**

- 添加快捷方式



- 编辑代码



- 点击快捷方式



5.6 默认颜色序列

函数说明

运行示例

5.7 典型快捷绘图

函数说明

运行示例

6. 数据操作

6.1 数字与字符串序列粘贴

函数说明

```
function str_x = paste(x,prefix,suffix)
% 题目：对数值序列粘贴前后缀，构成字符串
% 输入：
%     x      -- 数值序列
%     prefix -- 前缀
%     suffix  -- 后缀
```

```
% 输出 :
%      str_x  -- 合并后字符串
% 作者: 马骋
% 2016.04.17 @HIT
```

运行示例

- 测试代码

```
x = [1:10]';
str = tools.paste(x,'通道','原始信号');
```

- 运行效果

```
str =

    '通道1原始信号'
    '通道2原始信号'
    '通道3原始信号'
    '通道4原始信号'
    '通道5原始信号'
    '通道6原始信号'
    '通道7原始信号'
    '通道8原始信号'
    '通道9原始信号'
    '通道10原始信号'
```

这一功能主要用于多组数据绘图的legend字符串构造。

6.2 向量数值区间确定

函数说明

```
function rg = range(data)
% 题目：给出一个向量/矩阵的数值范围
```

本函数结合了max和min函数，主要用于绘图的坐标限确定。

运行示例

- 测试代码

```
x = 1:10;
x_range = tools.range(x)
```

- 运行效果

```
x_range =

     1
    10
```

6.3 行形式的数据转换列矩阵

函数说明

```
function mat = row2mat(row)
% 题目：将行向量或者行向量组成的矩阵转换为列向量形式
% 时间：2017.01.11
```

此函数用于矩阵形式的标准化，避免大量数据处理中矩阵维度不匹配的情况。

运行示例

- 调用代码

```
a = rand(2,4)
a2 = tools.row2mat(a)
```

- 运行结果

```
a =
    0.6787    0.7431    0.6555    0.7060
    0.7577    0.3922    0.1712    0.0318

a2 =
    0.6787    0.7577
    0.7431    0.3922
    0.6555    0.1712
    0.7060    0.0318
```

6.4 复数矩阵的归一化

函数说明

此函数用于信号处理中，复数结果的归一化计算。

```
function xNorm = norm(x)
% 题目:复数向量归一化
% 输入:
%     x          -- 复数向量
% 输出:
%     xNorm       -- 归一化后的向量
% 作者: 马骋
% 2016.04.17 @HIT
```

运行示例

- 测试代码

```
a = 2*rand(2,4)+rand(2,4)*i
a_norm= tools.norm(a);
a_norm_abs = abs(a_norm)
```

- 运行效果

```
a =
    1.1376 + 0.1656i    0.0238 + 0.2630i    0.3244 + 0.6892i    0.6224 + 0.4505i
    0.9388 + 0.6020i    0.6742 + 0.6541i    1.5886 + 0.7482i    1.0571 + 0.0838i

a_norm =
    0.9896 - 0.1441i    0.5346 - 0.3428i
    0.0207 - 0.2287i    0.3840 - 0.3725i
    0.2821 - 0.5995i    0.9047 - 0.4261i
    0.5414 - 0.3919i    0.6020 - 0.0477i

a_norm_abs =
    1.0000    0.6351
    0.2297    0.5350
    0.6626    1.0000
    0.6684    0.6039
```

注意，归一化以列向量为单位。

7. 信号处理

7.1 信号去趋势

函数说明

```
function data = clean(data0,tol)
% 题目：信号去除环境噪声（矩阵运算）
% 功能：去趋势项，对阈值一下的信号归零
% 输入：
%       data0 -- 原始激励信号
%       tol   -- 阈值
```

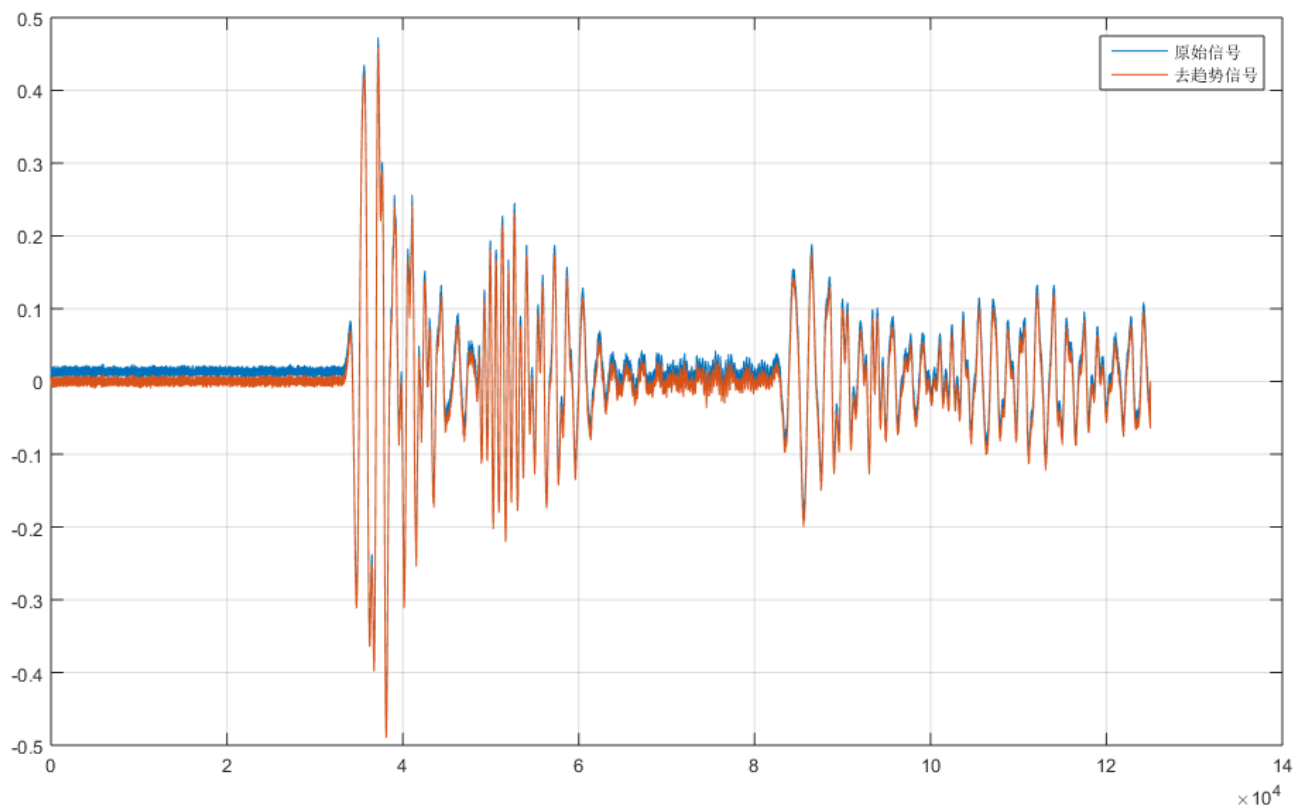
运行示例

- 测试代码

```
M = tools.getcsv(0); % 读取csv
文件
s = M(:,3); % 提取典型信号
号
s2 = tools.clean(s); % 信号去除环境噪声

figure
plot(s),hold on
plot(s2)
legend({'原始信号','去趋势信号'})
grid on
```

- 运行效果



7.2 低通滤波器设计

此处简要介绍，低通滤波器设计的详细说明见：

- [MATLAB|切比雪夫低通滤波器设计与滤波实现](#)
- [MATLAB|低通滤波器参数设置问题处理方法](#)

函数说明

```
function y=lowp(x,para,isFreqz)
% 题目：低通滤波器
% 输入：
%     x      -- 原始信号序列
%     para.
%         f1  -- 通带截止频率
%         f3  -- 阻带截止频率
%         rp  -- 边带区衰减DB数设置
%         rs  -- 截止区衰减DB数设置
%         fs  -- 序列x的采样频率
%         type-- 滤波器类型
%     isFreqz -- 是否绘制滤波器曲线
% 输出：
%     y      -- 滤波后的信号
% 功能：
%     低通滤波，滤除高频噪音
%     Cheby1
%     Butterworth
% 注意：
%     通带或阻带的截止频率的选取范围是不能超过采样率的一半
%     f1,f3的值都要小于fs/2
%     rp=0.1;rs=30;%通带边衰减DB值和阻带边衰减DB值
```

运行示例

- 测试代码

```
[M,dt] = tools.getcsv(0); % 读取csv文件
s = M(:,3); % 提取典型信号

% 参数对话框
prompt0 = {
    '低通滤波 fp-fs kHz', [500 700]
    '低通滤波 Rp',0.1
    '是否显示滤波器频谱',1
};

dlg0.save = 'myfilter';
para0 = tools.paradlg(prompt0,dlg0); % 对话框参数

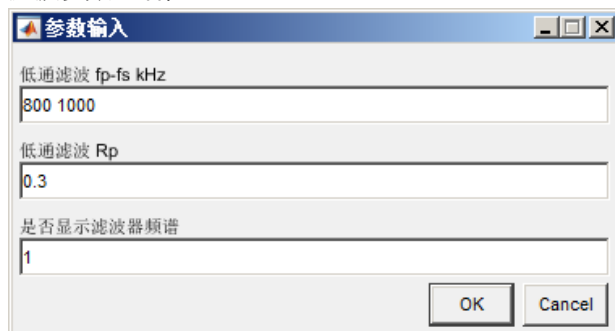
para_lp.f1 = para0{1}(1)*1e3; % 滤波器 fp
para_lp.f3 = para0{1}(2)*1e3; % 滤波器 fs
para_lp.rp = para0{2}; % 滤波器 rp
para_lp.rs = 30; % 滤波器 rs
para_lp.fs = 1/dt; % 信号采样频率

para_lp.type = 1; % 滤波器类
型：切比雪夫-1
flag = para0{3}; % 是否绘制滤波器频域曲线

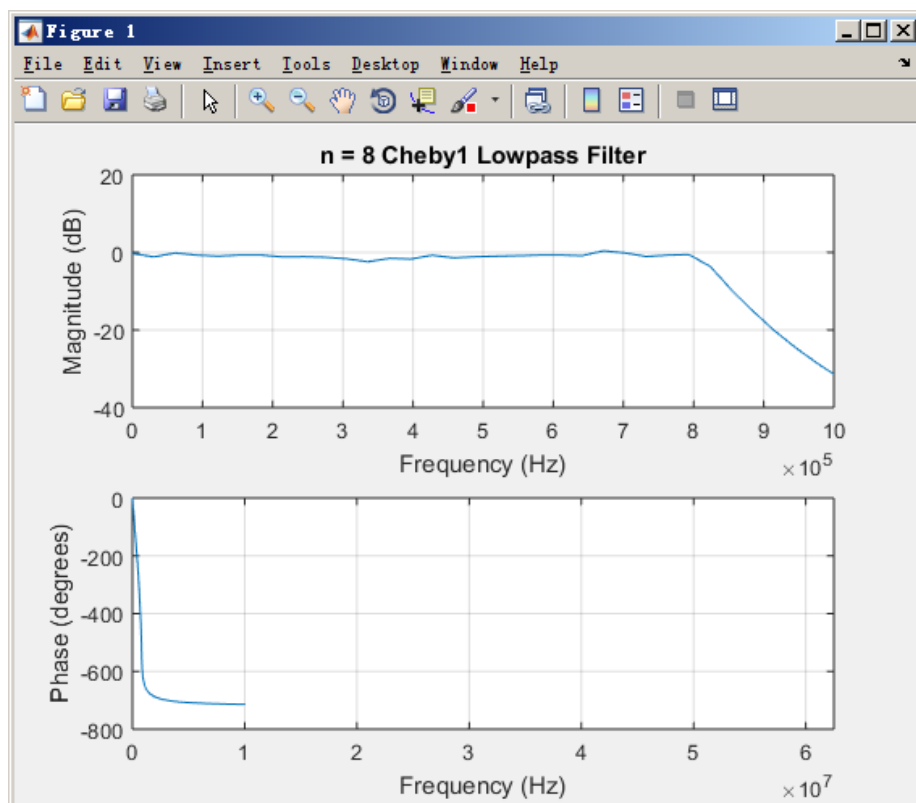
s_lp = tools.lowp(s,para_lp,flag); % 输入信号-滤波

figure % 滤波前后对比
plot(s),hold on
plot(s_lp)
legend({'原始信号','滤波后信号'})
tools.white;
```

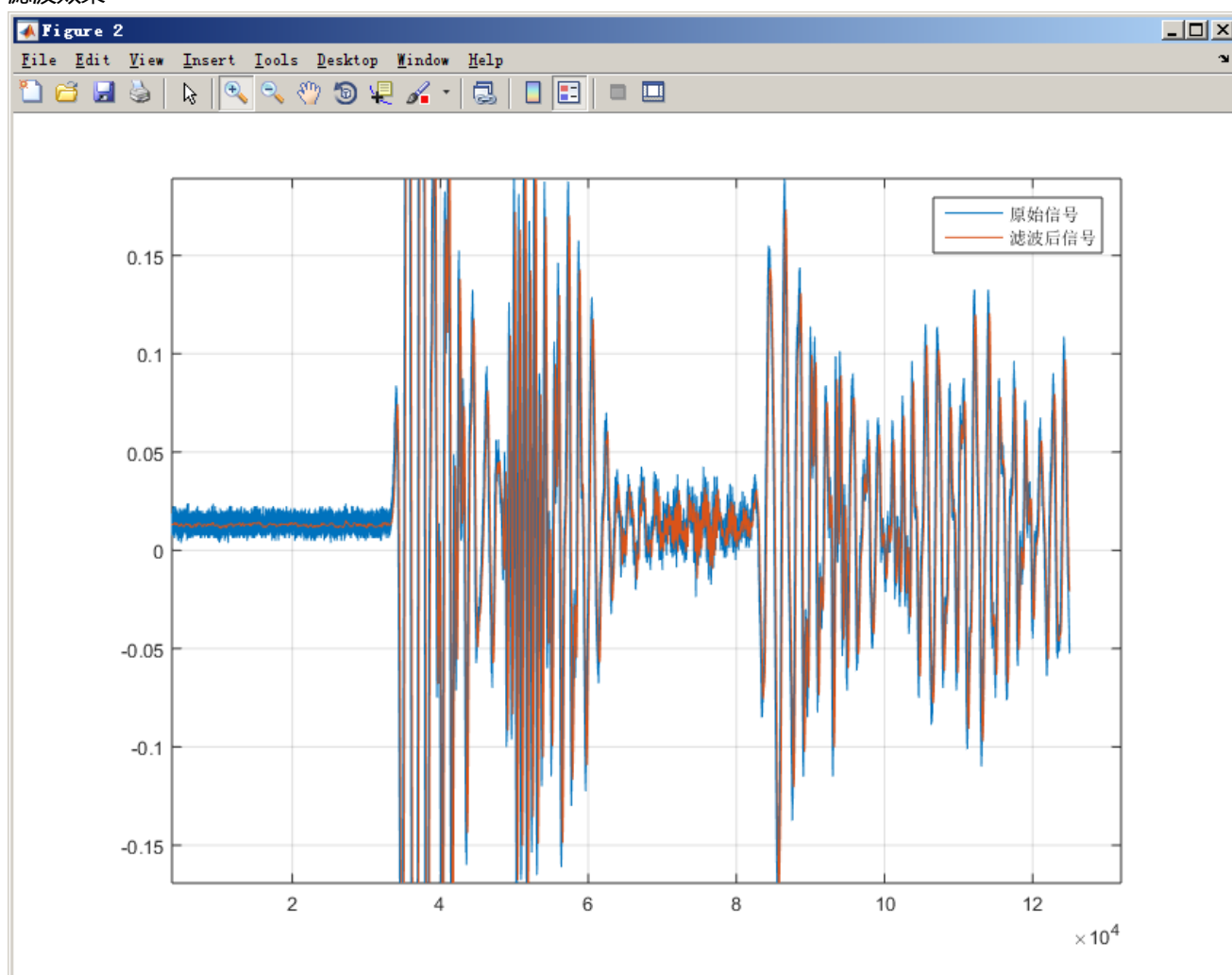
- 滤波参数对话框



- 滤波器频域特性曲线



- 滤波效果



7.3 多波峰正弦信号生成

这一问题的详细描述见：[T301|超声导波激励信号的生成与频谱分析](#)

函数说明


```
function toneburst()  
% 题目：超声导波激励信号的生成与频谱分析  
% 参数：  
%     N - cycle数，即激励信号波峰数  
%     fc - 激励信号中心频率  
% 功能：  
%     生成激励信号序列  
%     绘制时域图和频域图  
%     对比不同cycle数信号的特征  
%     输出txt文件  
% 作者：马骋  
% 2016.03.18 @HIT
```

运行示例

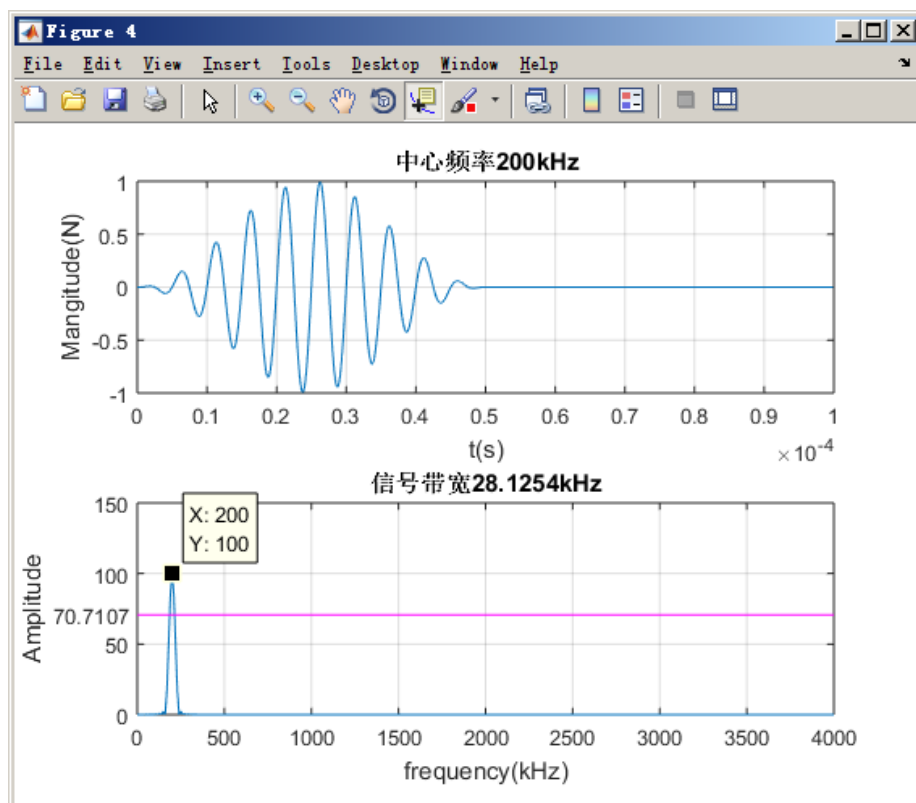
- 调用代码

```
[s,fs] = tools.toneburst;
```

- 参数对话框



- 信号时程与频谱



- 生成信号数据文件

名称	修改日期	类型	大小	
V200K_C10_100US_S800.TXT	2017/1/11 8:27	TXT 文件	15 KB	0.0000000000 0.0000096495 0.0000762406 0.0002519931 0.0005799300 0.0010898871 0.0017952230 0.0026904140 0.0037496823 0.0049267585 0.0061558297 0.0073536725 0.0084229128 0.0092563056 0.0097418744 0.0097687083 0.0092331773 0.0080452977

7.4 激励信号带宽计算

函数说明

此函数用于计算给定激励信号的频域带宽，确定信号宽带、窄带的指标，采用3Db带宽。

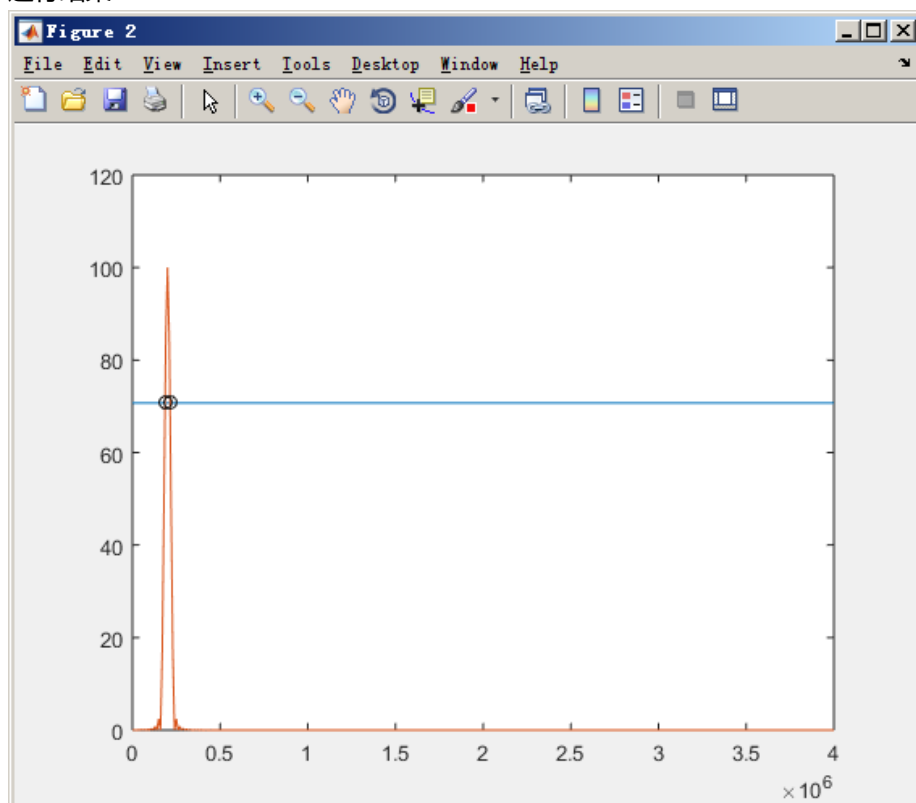
```
% 题目：计算给定信号的3db带宽
% 输入：
%     fs  -- 采样频率
%     S   -- 信号时程
%     flag-- 是否绘制图像
% 时间：2017.01.10
```

运行示例

- 测试代码

```
[s,fs] = tools.toneburst;
[band3db,x0] = tools.getband3db(fs,s);
band3db_fk = band3db/1000;
```

- 运行结果



- 带宽数值z，单位为kHz

```
band3db_fk  
=  
28.1254
```

8. 其他函数

8.1 工具箱版本显示

函数说明

此函数用于记录程序的更新历程和版本信息。

```
function version()  
% 题目：版本自动说明
```

运行示例

- 调用代码

```
tools.version
```

- 运行效果

版本说明：数据处理与信号绘图辅助工具箱
马骋,创建于2016.04.29

更新日志：

2016.04.29, 增加xyt函数；
2016.04.30, 增加getmat函数；
2016.04.30, 更新paradlg冲突bug；
2016.05.03, 更新paradlg空格bug；
2016.05.03, 增加range函数；
2016.05.04, 增加row2mat函数；
2016.05.06, 增加html函数；
2016.05.06, 增加clean函数；
2016.12.13, 增加colorOrder函数；
2016.12.21, 增加xline函数；
2017.01.05, 修改paradlg函数，智能弹出；
2017.01.05, 增加saveGraph函数；
2017.01.08, 增加xGrid,yGrid网格线
2017.01.08, 增加plot0函数；
2017.01.10, 增加intersection函数；
2017.01.10, 增加get3band3db函数；
2017.01.10, 增加toneburst函数；

9. 使用技巧

9.1 工具箱的路径设置

自编MATLAB函数（如信号处理函数库tools.m），如果不做进一步的设置，往往只能在函数文件下调用，如果工作路径在其他文件夹下则不能使用。

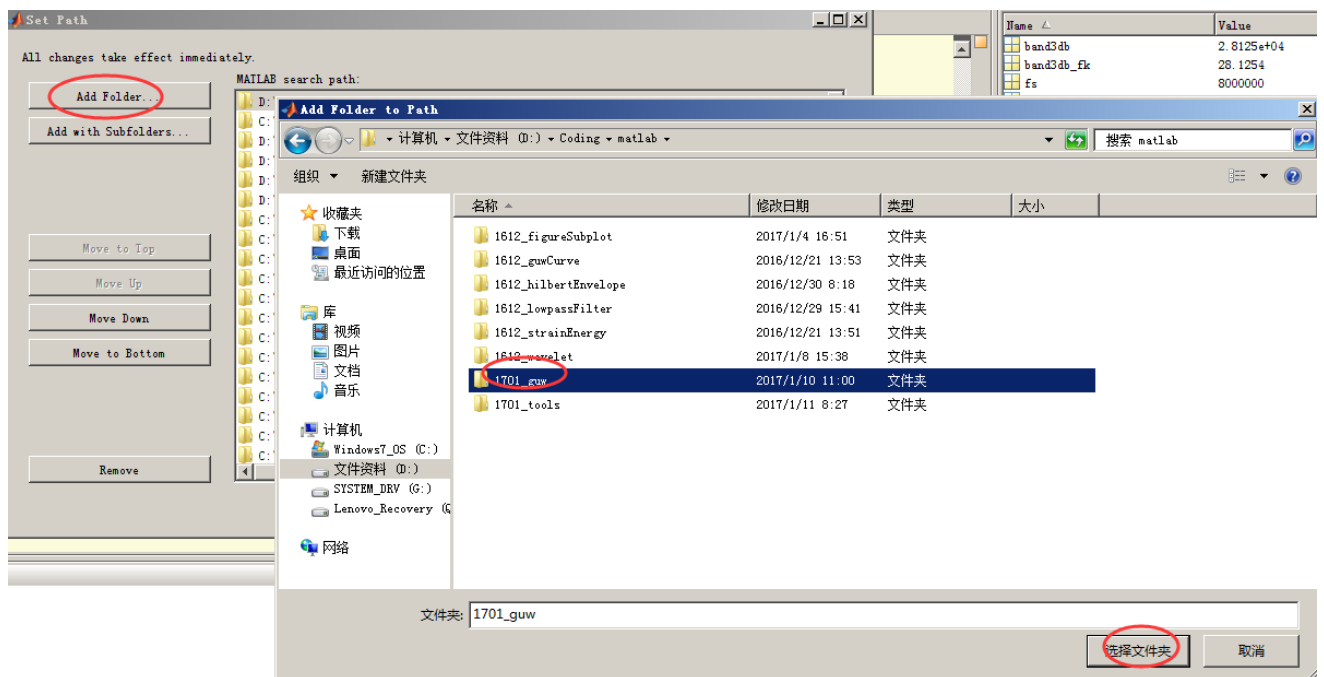
```
Command Window
>> tools.xyt('Frequency/kHz','Energy Velocity/(m/s)','能量速度'))
Undefined variable "tools" or class "tools.xyt".
```

基本的解决方法是：将 **tools.m** 添加到MATLAB可以搜索的路径下：

- 设置添加路径对话框

pathtool

- 添加 **tools** 工具箱所在的路径



9.2 函数帮助信息显示

输入函数名，按 **F1** 键，即可显示函数的帮助信息。



与MATLAB内置函数的帮助显示一样。

10. 更新与下载

本工具箱功能持续更新，代码托管于coding.net，下载网址为：

[coding.net-frank0449https://coding.net/u/frank0449/p/matlab/git](https://coding.net/u/frank0449/p/matlab/git)

frank0449 tools		142a7f1b5a 提交于 4分钟前	
1612_figureSubplot	3天前	frank0449	update guw toolbox
1612_guwCurve	21天前	frank0449	add strainEnergy analysis, and pulished other projects to html
1612_hilbertEnvelope	12天前	frank0449	1612_hilbertEnvelope
1612_lowpassFilter	13天前	frank0449	debug lowpassFilter
1612_strainEnergy	21天前	frank0449	add strainEnergy analysis, and pulished other projects to html
1612_wavelet	3天前	frank0449	update guw toolbox
1701_guw	5分钟前	frank0449	tools
1701_tools	5分钟前	frank0449	tools
README.md	12天前	frank0449	1612_hilbertEnvelope