MATLAB 技术论坛

电子期刊

编辑：xiezhh
制作：MATLAB 技术论坛

网址：http://www.matlabsky.com

第 7 期 | 2011. 03 | No.7

中国权威 MATLAB 论坛核心期刊

MATLAB 技术论坛简介

Matlab 技术论坛 | Simulink 仿真论坛 | ——打造优秀、专业和权威的 Matlab 技术交流平台!

- http://www.matlabsky.com /cn/net/org

论坛拥有 40 多个专业版块，内容涉及资料下载、视频教学、数学建模、科学计算、程序设计、GUI 开发、simulink 仿真、统计概率、拟合优化、扩展编程、算法研究、控制系统、信号通信、图像处理、经济金融、生物化学、航空航天、人工智能、汽车设计、机械自动化、毕业设计等几十个方面!

论坛特色：

1. 拥有强大的技术人员、热情严谨的管理团队
2. 丰富多彩的Matlab电子资源免费共享
3. 免费提供技术交流和在线解答
4. 首个推出MATLAB函数百科中文帮助系统
5. 国内唯一提供Matlab汉化包的团队

联系方式：

客服 QQ： 1341692017

服务邮箱： matlabsky@gmail.com

支付宝账号： yuthreestone@163.com 郁磊

互动QQ群： http://www.matlabsky.com/thread-3-1-1.html

开通业务：

技术论坛： http://www.matlabsky.com

函数百科： http://wiki.matlabsky.com

官方博客： http://blog.matlabsky.com

读书频道： http://book.matlabsky.com

视频教程： http://video.matlabsky.com

有偿编程： http://paid.matlabsky.com

目　录

在此向本期内容涉及代码的原作者们表示最衷心的感谢！同时祝福版友们身体健康！万事如意！

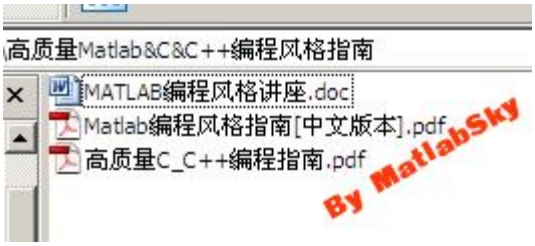
MATLAB 编程技巧版块

1 高质量 Matlab&C&C++编程风格指南

贴子地址：http://www.matlabsky.com/thread-9344-1-1.html

作者：faruto

和大家分享一些经典的关于编程风格的电子书，对于养成一个好的编程习惯和风格有一定的启示作用。



MATLAB 编程风格指南 Richard Johnson 著 Genial 译

MATLAB 编程风格指南

Richard Johnson

Version 1.5，Oct. 2002

版权：Datatool 所有

翻译：Genial @ USTC

"Language is like a cracked kettle on which we beat tuned to dance to, while all the time we long to move the stars to pity."

— Gustave Flaubert, in *Madame Bovary*

目 录：

高质量 C++/C 编程指南，v 1.0

目 录

==========外一篇： faruto 浅谈编程风格 ==========

我要说的关于编程的第一个主要思想：先整体，后具体；分块把握大的算法再实现。其实这也比较符合一些生活常识，要解决一个大问题，首先我们需要从整体上把握这个问题，然后再从细节上处理相关问题。编程也是一样的，当有一个大的算法需要实现，你需要做的事整体上把握一下算法脉络，然后再分块实现，这样是好的习惯，因为每一个人对于代码的掌控能力是有限的，也就是说当一个函数的代码长度多长的话，最终你会失去掌控能力的，到后面你会连一些变量名字都会弄混的。我个人的对于单个函数的代码的掌控能力为 1000 行左右，所以当一个函数过于太大，超过我个人的掌控能力的时候，我就会把这个函数再进行分块，进而每个字块函数的代码行数让其在我的掌控能力之内，这样做的好处是一则对于单个子函数调试找 bug 方便，二则对于整体的大函数项目，最后整体做一个接口把各个子函数

链接到一起就行了，大函数出错的话，很容易找到是哪个子函数出的问题，若都放到一起来实现，调试起来会让人疯掉的。所以说，编程的时候需要：先整体，后具体；分块把握大的算法再实现。这个说起来做起来难，需要个中体会。

第二个我想谈的是：编程风格问题。无论学习什么编程语言，最开始的基础是很重要的，一个好的编程习惯和代码风格会使你往后的编程道路很顺利，否则的话往后进展会出现瓶颈。从代码的质量也就能看出你本身的编程能力了，如果你平时的编程习惯好，在最终代码成型时出现的 bug 就会少很多，所以说平时的编程习惯是很很重要的，少量的代码无所谓的(200 行以内)，但当代码量多起来，如果你平时编程习惯不好，到时你连前后的变量名称定义自己都会搞混的。说了这么多，到底怎么是一个好的编程风格呢？这个没有十分统一的标准，以前看过一本书《C++编程高手》里面详细讲了一些同认的好的编程习惯风格，包括变量的合理命名，恰当使用空格和换行符使得代码看起来更舒服等等。

比如下面这段普通简单的 A+B Problem 的代码（MATLAB）：

```
function solve_AplusB_problem
disp('This is a programme for solving A+B problem');
A=input('Please input the fist number A:');
B=input('Please input the second number B:');
C=A+B;
str=['The result of A+B is:',num2str(C)];
disp(str);
```

这段代码再简单不过，当年的我可能也差不多类似于上面那样写代码，但现在的我就会这样来写：

```
function solve_AplusB_problem
% by faruto
% a test programme for solving A+B problem

disp('This is a programme for solving A+B problem');

A = input('Please input the fist number A:');

B = input('Please input the second number B:');

C = A + B;

str = ['The result of A+B is:', num2str( C )];

disp( str );
```

两段代码几乎没有差别，实现内容也一样，后者的代码段就像“神经病”一样多了很多空格和空白行，但我敢说后者的代码风格习惯更好，代码和别人交流起来也更容易，这就是一个非常非常简单的例子，如果代码很长很复杂，你会发现好的代码风格超级重要的！个中感觉还望看官您自己体会。O(∩_∩)O~

回想自己学习 MATLAB 的过程（或者说学习其他编程语言的过程），感受最深的一点就是一定要自己多多动手，遇到不会的东西先要查看帮助文件或者通过网络搜索来找到相应的答案，也就是说需要有自我帮助能力，这样才能不断提高自己的能力。当然有些非常棘手的问题是需要向高人请教的。

========== rocwoods 的回复 ==========

嗯，好的代码风格非常重要，各个公司对需要编码的人员都非常强调好的编程风格习惯的培养。我再补充一点，变量命名很重要，对于重要的变量应该“望名生义”就是看到变量名，大致猜到什么意思。推荐使用大小写混合形式命名，譬如：SegmentLength,ColorOrder 等命名方式，让人一目了然。

经常见到初学者用汉语拼音命名变量，这是非常不好的一个习惯，别的不说，让你的代码非常“山

寨”，而且代码复杂后会让人莫名其妙的，应该摒弃。

2 Writing Fast Matlab Code[如何优化你的程序,让其跑的更快]

贴子地址：http://www.matlabsky.com/thread-9457-1-1.html

作者：faruto

很不错的一个 Ebook 交易如何优化 matlab 代码提升其速度。当然里面也有一些速度提升理念是落后的，去糟取精即可~O(∩_∩)O~

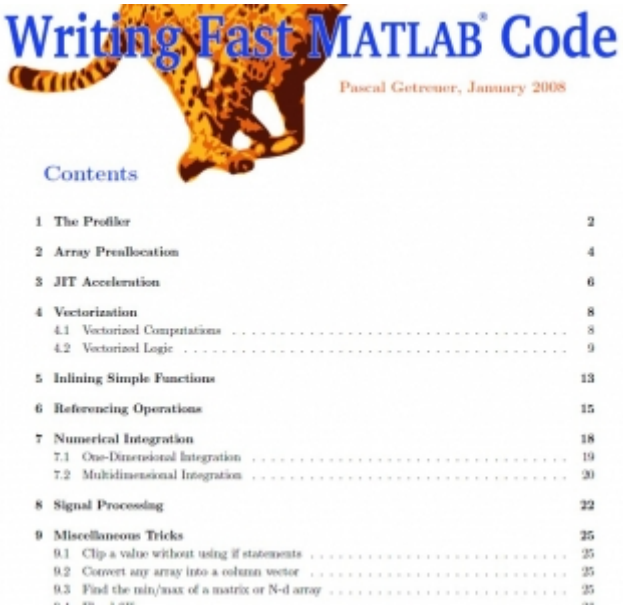
matlab 以其强大的功能在科学领域有不可替代的位置,用了 matlab 多年,唯一感觉的遗憾就是在速度上 matlab 有时是拼不过 C(C++)的,前一阵子在我们院的 FTP 上看到下面这个电子书(Writing Fast Matlab Code),挺有感想,对如何提升 matlab 的速度上有了一定认识,如果你对 matlab 的速度方面也有要求,不妨看看,它会对你如何写出更高效的 matlab 代码有一点的启示!

看过这个电子书我个人的核心理解就是 preallocation,预先定义,虽说 matlab 中的变量都不用预先定义,但当你的程序中有维数很大的矩阵时,预先定义会使你的程序跑起来的速度比原来快两倍多!!

因为在 Matlab 中如果你不对某一矩阵或向量预先定义,则在跑循环时 Matlab 会重复对这个矩阵或向量扩容,直到满足大小为止,所以你如果预先定义了,这样会使得程序在跑循环时不再对矩阵或向量重复进行扩容,从而省去很多时间...

个中详细原由你可以自己细看,电子书中做了测试,时间可省去一半之多.我自己也也过相关测试,的确能提升 Matlab 代码的速度!

[当然小问题没有预先定义的必要]


Writing Fast MATLAB Code

Pascal Getreuer, January 2008

Contents

Ebook下载： 请自行去原帖下载。

3 在高版本 MATLAB 下重新认识循环

贴子地址：http://www.matlabsky.com/thread-9249-1-1.html

作者：rocwoods

高版本 MATLAB 对循环结构的优化

从 MATLAB6.5 版开始，MATLAB 引入了 JIT（just in time）技术和加速器（accelerator），并在后续版本中不断优化。到了 R2009a，很多情况下，循环体本身已经不是程序性能提高的瓶颈了，瓶颈更多的来源于循环体内部的代码实现方式，以及使用循环的方式。循环就是多次重复做一件事，如果这件事本身代码写得不优化，放在循环体内多次实现后必造成运行时间过长。

老版本的 MATLAB 对循环机制的支持不好，所以我们提倡避免循环，而高版本的 MATLAB 对循环机制的支持大大提高了，我们就不能像过去那样谈“循环”色变，非得千方百计避免循环。当使用了循环，造成程序运行时间过长时，我们不要武断地将代码运行效率低归结到使用了循环。有的时候我们千方百计的把一段代码矢量化了，凭自己编程经验（很多是从老版本遗留下来的经验）和常识（从老的教科书得到的）觉得程序很标准，很优化，殊不知实际测量时会发现性能不比采用很自然的想法实现的程序效率高多少，甚至还会降低。我们看下面这个例子：

【例 1】运行下面的测试 JIT/ accelerator 的代码，体会高版本的 MATLAB 对循环的加速。

```
function JITAcceleratorTest
u = rand(1e6,1);%随机生成一个 1*1000000 的向量
v = zeros(1e6,1);
tic
u1 = u + 1;
time = toc;
disp(['用向量化方法的时间是：',num2str(time),'秒!']);
tic
for ii = 1:1000000
    v(ii) = u(ii)+1;
end
time = toc;
disp(['循环的时间是：',num2str(time),'秒!']);

feature jit off;
tic
for ii = 1:1000000
    v(ii) = u(ii)+1;
end
time = toc;
disp(['只关闭 jit 的时间是：',num2str(time),'秒!']);

feature accel off;

tic
for ii = 1:1000000
    v(ii) = u(ii)+1;
end
time = toc;
disp(['关闭 accel 和 JIT 的时间是：',num2str(time),'秒!']);
```

```
feature accel on;%测试完毕重新打开 accelerator 和 JIT
feature jit on;
end
```

我的电脑上结果如下：

```
用向量化方法的时间是：0.0095308 秒!
循环的时间是：0.010176 秒!
只关闭 JIT 的时间是：0.084027 秒!
关闭 accel 和 JIT 的时间是：1.2673 秒!
```

自从引入 JIT 和 accelerator 后，MATLAB 对这两项功能默认都是打开的，这也是为什么高版本的 MATLAB 对循环的支持好。关闭 JIT 和 accelerator 需要用到 MATLAB 一个未公开（undocumented）的函数：feature。feature accel on/off 即为打开/关闭 accelerator，类似的打开/关闭 JIT 是 feature jit on/off。

上面代码算一个长向量与一个标量的和，我们会发现，JIT 和 accelerator 都打开的状态下，循环和矢量化运算所需要的时间从统计意义上来讲，已经没有显著差别了。关闭 JIT 后，运行时间变为原来的 8 倍左右。而再关闭 accelerator，运行时间立刻变为原来的 100 多倍。

我们再来看一个例子：

【例 2】 由一个 m*n 的矩阵构造一个 m*(m+n-1)的矩阵，构造方式如下：以 4*4 矩阵 A 为例，目的构造 B：

```
A =
   1   2   3   4
   3   4   5   6
   2   3   4   5
   1   3   4   6
B =
1  2  3  4  0  0  0
0  3  4  5  6  0  0
0  0  2  3  4  5  0
0  0  0  1  3  4  6
```

这个例子用循环解决非常容易实现，我们这里想要讨论的就是用向量化的方法解决该问题和用循环解决该问题之间的时间对比（还不考虑写出矢量化的代码比用循环实现多花的时间）。

向量化思路：我们观察 B，发现如果按行数的话，B 的元素排列顺序是原来 A 中的每一行和下一行之间以 4 个 0 相隔。这样我们可以计算出 A 中的元素在 B 中相应的索引值（当然是按行）I。这样我们可以生成一个和 A 大小一样的全 0 矩阵 B，然后 B(I) = a(:);最后注意到 MATLAB 是按列的顺序遍历元素的，所以最后再转置一下 B。写成代码就是：

```
function B=rowmove(A)
[m,n]=size(A);
I=repmat(1:n,m,1)+repmat((0:m-1)'*(m+n),1,n);
B=zeros(m+n-1,m);
B(I(:))=A(:);
B=B';
```

循环的代码如下：

```
function C = LoopRowMove(A)
[m,n] = size(A);
C = zeros(m,m+n-1);
```

```
for k = 1:m
C(k,k:k+n-1) = A(k,:);
end
```

可以看出，循环的思路以及操作非常简单，就是循环赋值操作。下面我们随机生成一个 1000*1000 的矩阵，来用上述两种方法来比较下速度：

```
A = rand(1000);
tic;B = rowmove(A);time = toc;
disp(['向量化求解时间是：',num2str(time),'秒!'])
tic;C = LoopRowMove(A);time = toc;
disp(['用循环求解时间是：',num2str(time),'秒!'])
if isequal(B,C)
    disp('两种方法结果完全一样')
end
```

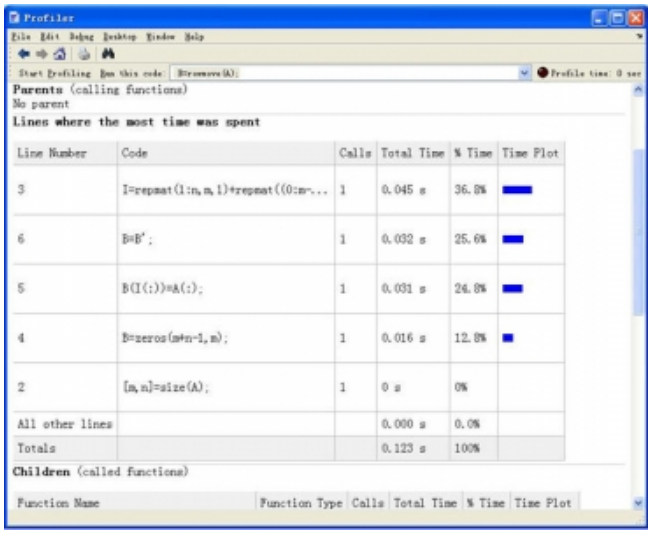
在我的电脑上运行结果如下：

```
向量化求解时间是：0.11389 秒!
用循环求解时间是：0.054231 秒!
```

两种方法结果完全一样。循环的时间反而不到向量化时间的一半，而且循环的代码要比向量化的容易写得多，算上“开发”时间，本例用循环比用向量化要高效的多。这是为什么呢？因为本例循环结构内部的代码仅仅为“赋值”这样简单的操作，不存在函数调用。目前在高版本的 MATLAB 中，循环本身往往不是程序瓶颈，反而函数调用产生的额外开销在很多情况下是构成程序瓶颈的因素之一，尤其是采用大量低效率的函数结构时。（譬如 inline 对象，这个将在后续章节详细讨论）。

反观向量化操作，调用了两次 repmat 函数，至于 repmat 内部又做了些什么，有兴趣的话我们可以运行 edit repmat 命令，查看 repmat 的源代码，我们会发现，里面有很多行代码都是我们这个问题没必要执行的。事实上，为了保证每个函数具有尽可能广的适应面，Mathworks 开发人员在编写函数时一般都在函数入口做很多层判断，确保不同的输入能正确地执行相应的语句。

下图是用 MATLAB 的 Profiler 工具剖析“B=rowmove(A);”这条语句的结果。我们可以看出，两次调用 repmat 函数所占时间比例最高，其次是转置操作，这两项一共占去了整个程序的运行时间的 62%。而这两项对于本例来说都完全都可以避免。而我们用循环实现时，只需要赋值 m 次即可，所用的系统开销相对要少。



========== gipfel 的回复 ==========

照我的理解，向量化或者其他优化方式，只要运用得当，仍然会比循环快。依赖 matlab 来优化，不如自己优化。

2010b 下面，我的电脑运行结果显示向量化有明显的优势：

用向量化方法的时间是：0.0038615 秒!
循环的时间是：0.046797 秒!
只关闭 jit 的时间是：0.044876 秒!
关闭 accel 和 JIT 的时间是：0.86756 秒!

向量化比循环有 10 倍以上的速度优势，比关闭 accel 和 JIT 之后的快几百倍。
差距拉大，可能跟新的处理器并行度提高或者对向量运算的支持提高有关系，我用的是 i7 的 CPU。

4 在 MATLAB 中发送电子邮件（支持附件）

贴子地址：http://www.matlabsky.com/thread-12663-1-2.html
作者：dynamic，rocwoods

通过 MATLAB 发送邮件的意义在于，假如您编写了一个程序，希望客户对您的程序有一个反馈或者 bug 提交等互动交流！另外假如在运行一个大型程序，我们希望将运行过程和结果发送到我们指定的邮箱，方便了解运行进展情况。

MATLAB 自带有 sendmail 函数，可以用来发送电子邮件，但是 sendmail 只能发送不需要 authentication 的邮箱地址，但是互联网上公共邮箱几乎都需要 authentication。不用担心，由于 Matlab 支持 smtp 邮件，故 163，qq，gmail 都可以在 MATLAB 中利用 Java 的方式发送邮件。

MATLAB中的Demo中有详细介绍。但是不同的MATLAB版本会遇到不同的问题。另外这里有一篇论文，大家可以下载参考（http://www.matlabsky.com/thread-12663-1-2.html）。

设定 smtp 服务器，然后使用 sendmail 发送邮件：

```
        % 设置发送邮件的配置，我们用 gmail 举例
        mail = 'my.email.address@gmail.com'; %gmail 地址，qq 或者 163 也可以的
        password = 'matlabsky';  %邮箱密码

        %下面是 gmail 的标准配置，其他邮箱，可以相应修改
        setpref('Internet','E_mail',mail);
        setpref('Internet','SMTP_Server','smtp.gmail.com');
        setpref('Internet','SMTP_Username',mail);
        setpref('Internet','SMTP_Password',password);
        props = java.lang.System.getProperties;
        props.setProperty('mail.smtp.auth','true');
        props.setProperty('mail.smtp.socketFactory.class',
'javax.net.ssl.SSLSocketFactory');
        props.setProperty('mail.smtp.socketFactory.port','465');

        % 发送邮件
```

```
sendmail('收件人 email','邮件标题','邮件内容!')
```

写成函数的形式（支持附件发送）：

```
function MySendMail
a = rand(100);
DataPath = [matlabroot,filesep,'mydata.mat'];
save(DataPath,'a');
MailAddress = '我的邮箱地址';
password = '我的密码';
setpref('Internet','E_mail',MailAddress);
setpref('Internet','SMTP_Server','smtp.gmail.com');
setpref('Internet','SMTP_Username',MailAddress);
setpref('Internet','SMTP_Password',password);
props = java.lang.System.getProperties;
props.setProperty('mail.smtp.auth','true');
props.setProperty('mail.smtp.socketFactory.class',
'javax.net.ssl.SSLSocketFactory');
props.setProperty('mail.smtp.socketFactory.port','465');
subject = 'MATLAB 发的测试邮件';
content = '你好，这份邮件是我用 MATLAB 发的，数据见附件';
sendmail('收件人地址',subject,content,DataPath);
```

如果在 7.0 以下的版本中 sendmail 的一些 smtp 语法跟不上当前的时代，需要修改语句。在 Matlab7.0 版 sendmail.m 的 140 到 150 行左右：

```
sendSMTP(out, in, ['MAIL FROM: ' from], 1);
改为：sendSMTP(out, in, ['MAIL FROM: <' from '>'], 1);
sendSMTP(out, in, ['RCPT TO: ' to{i}], 1);
改为：sendSMTP(out, in, ['RCPT TO: <' to{i} '>'], 1);
sendSMTP(out, in, ['RCPT TO: ' to], 1);
改为：sendSMTP(out, in, ['RCPT TO: <' to '>'], 1);在第 210~220 行附近：
sendSMTP(out, in, '.',1); 改为：sendSMTP(out, in, sprintf('\n.\n'), 1);
```

在 7.0 版本以上不需要修改即可用，输入以下代码：

```
props = java.lang.System.getProperties;
props.setProperty('mail.smtp.auth','true');
props.setProperty('mail.smtp.socketFactory.class',
'javax.net.ssl.SSLSocketFactory');
props.setProperty('mail.smtp.socketFactory.port','465');
```

然而，现行的 smtp 服务器都需要身份验证，Matlab7.0 版的 sendmail.m 不支持身份验证，向 m 文件中添加验证语句。

打开 Matlab7.0 版的 sendmail.m 找到 130 多行的语句：

```
sendSMTP(out, in, 'HELO mailman', 1);
```

在此行之后添加以下三行：

```
sendSMTP(out, in, 'auth login', 1);
sendSMTP(out, in, '加密的用户名', 1);
sendSMTP(out, in, '加密的密码', 1);
```

然后输入：

```
setpref('Internet','SMTP_Server','smtp.email.com');
setpref('Internet', 'E_mail', '用户名');
```

```
setpref('Internet', 'E_mail', '密码');
```

最后，可以向别人发邮件了：

```
sendmail('收件人地址','邮件标题',{'邮件内容','内容','内容'},'附件地址')
```

注意： Matlab发送邮件是通过Java接口，速度比较慢，所以不推荐添加太大的附件，以免发送失败。并且MATLAB不支持发送中文信息，会出现乱码，当然假如十分特别有这种想法的，可以先用函数double将中文转化为数字矩阵保存，然后用setstr再转为中文即可。

MATLAB GUI 开发版块

5 GUI 新手之教你读懂 GUI 的 M 文件

贴子地址：http://www.matlabsky.com/thread-3565-1-1.html

作者：zhenghui

GUIDE 生成的 GUI 的 M 文件控制了你编制的 GUI 界面的所有属性和行为，或者说外观和对用户操作的响应。比如说按下一个按钮或者选择了一个菜单项之类。M 文件包括了运行你整个界面程序所需要的全部代码，包括所有 GUI 组件的 CALLBACKS 函数。其实这些 callbacks 函数算是 M 文件里的子程序，callback 里面就填写你所期望程序做的动作，比如画一个图或者算一个算式。

5.1 什么是子程序

这里解释一下什么是子程序，懂的人跳过^_^

```
function [avg, med] = newstats(u) % Primary function
% NEWSTATS Find mean and median with internal functions.
n = length(u);
avg = mean(u, n);
med = median(u, n);
function a = mean(v, n)            % Subfunction
% Calculate average.
a = sum(v)/n;
function m = median(v, n)          % Subfunction
% Calculate median.
w = sort(v);
if rem(n, 2) == 1
   m = w((n+1) / 2);
else
   m = (w(n/2) + w(n/2+1)) / 2;
end
```

以上就是一个大的程序 function nestats，它下面另外包含了两个小的 function mean 和 median，这样在大程序的里面就可以以如上的方式调用它们了。子程序的好处在于如果你总是要重复用到一组计算方式的时候，那你就把这组重复计算方式类似以上的方法编写成一个子程序，避免大量重复代码。在 M 文件里面，会看到最外层，也就是最上面那一一行：

```
function varargout = setfire(varargin)
```

（setfire 是我 m 文件存的名字）就是那个大程序框，它下面有很多小 function 比如什么 creatFcn 或者什么 callback 之类。看上去那个复杂，其实就跟上面这个一样的道理。只不过是 M 文件的类似

```
avg = mean(u, n);
med = median(u, n);
```

这两句话系统隐藏（就当它是隐藏好了）起来了，它会在你点击鼠标（或者响应操作）时候自动去调用执行一次 callback 函数。所以你只用管把代码写入响应的函数名下就行了。至于系统为什么会自动调用，我们不用管，我们只用知道，我点击鼠标，我拖动滑竿时，系统会执行哪里的代码。就够了。

5.2 M 文件的数据管理模式

用 handles 这个东西共享数据（Sharing Data with the Handles Structure）

在你运行你的 GUI 的时候，M 文件会自动生成一个叫做 handles 的东西（准确的说它属于 handles 类型的结构体，且取的名字也叫做 handles），不用管那么复杂，只用知道你可以从它这里找到 GUI 的所有数据，比如说控件的信息，菜单信息，axes 信息。想象 handles 就是一个缸了，它里面装载了所有的信息，而且这个缸在各个控件的 callback 之间传来传去，理所当然那每个控件的 callback 都可以放入一些想放入的数据，也可以从里面取出任何想要的数据包括别的控件的信息（比如滑竿的当前值，edit text 的当前值）和别的控件放进去的数据。

所以，用 handles 可以达到的目的有两个：

1．各个控件的 callback 的信息交换

在某控件下的 callback 写入下面这一句，就表示你把这个数据放缸里了。这里的 current_data 是随便设置的变量名。

```
handles.current_data = X;
```

接着别忘了保存！

```
guidata(hObject,handles);
```

然后在你需要的地方把它从缸里捞出来

```
X1 = handles.current_data;
```

2．读取 GUI 控件的信息，自然也可以设置 GUI 控件的信息（比如说背景色随着按钮点击而变换之类，或者你想让按钮 A 点一下，字符 B 跳一下，也行）。

例如：

```
all_choices = get(handles.my_menu, 'String');
current_choice = all_choices{get(handles.my_menu, 'Value')};
```

这里 all_choices 是随便取的变量名，my_menu 是你那个菜单项的 TAG 名字，这样 current_choice 就得到了用户界面操作中，目录或者菜单的选择结果。

所以，要什么信息，直接用 handles.你的对象就行了。存什么信息也直接 handles.你的对象 就行了。如果是自己的数据，就.变量名；如果是控件信息，就用 get 和 set 函数。

5.3 M-File 里的各个函数代表什么意思

在设计面板设计排列好自己需要的各种按钮或者编辑框之后，下一步任务便是在自动生成的 M 文件中添加自己的响应代码。

Opening function 添加在它名下的代码，在 GUI 开始运行但是还不可见的时候执行。这里的代码一般都是做一些初始化工作的。

Output function 如果有需要，可以向命令行输出数据。（这个函数我没用过，不多说了^_^）

Callbacks 每一次点击按钮或者向输入框输入数据或者拖动滑竿，这些控件名下的 callback 就会执行一次。

函数的输入参数

M-File 名下的全部 function 都会有这两个输入参数

hObject 它代表的是当前的这个控件（也就是你点哪一个按钮或者拖的哪一个滑竿）

handles 它代表的是现在这整一个 GUI 界面

对这两个变量进行修改后

guidata(hObject, handles); 进行保存，否则修改无效;

5.3.1 Opening Function

```
function my_gui_OpeningFcn(hObject, eventdata, handles, varargin)
```

在 GUI 开始运行但是还不可见的时候执行。这里的代码一般都是做一些初始化工作的。这个函数名下的代码在界面可见之前执行。其实你也可以在这个函数名下用 handles.什么 tag 来获得组件的信息。因为在 Opening 函数之前，所有的组件就已经生成了，只不过 opening 函数是把这些组件‘打开’，让它们显示出来。所以你可以在这个函数下面，添加代码，对界面做一些初始化工作。比如，计算一些数据，显示一幅图或者别的什么工作。

前两个输入参数 hObject 和 eventdata 是 matlab 的保留参数，为以后开发准备的，我们不用管它。用户可通过 varargin 传递自己的参数。

GUI 也是函数，它只不过是有个界面的函数。它的调用，同样是函数名（输入参数）。例如：

```
my_gui('Position', [71.8 44.9 74.8 19.7])
```

这里就表示 GUI 在打开时，位置这个属性被设置成了右边那个值。也就是在这个位置打开 GUI。Position 是你 GUI 界面的一个属性。（要想知道各个控件有什么属性，在它上面双击就看到了。）所以同样，也可以用这种方式输入其他的初始化命令。例如

```
my_gui('路人甲','年十八')
```

但如果你输入别的，左边那个根本就不是界面的属性名称。这是输入的参数就保存在 varargin 里面。也就是 vararging{1}='路人甲' vararging{2}='年十八'。这样也可以达到向调用的 GUI 传入数据的目的。

5.3.2 Output Function

有输入自然就有输出，顾名思义，这个函数就是用来输出的。

```
function varargout = my_gui_OutputFcn(hObject, eventdata, handles)
% Get default command line output from handles structure
  varargout{1} = handles.output;
```

这个函数不是我们自己编写的，也不是我们负责调用。我们只用知道要传出去的数据，其实是放在 vararout 这个变量里面的。但是我们在别的控件的 callback 是叫不到 vararout 这个变量的，因为你去看这些 callback 的输入参数里并没有 vararout 这个变量给它用。所以我们只能间接修改 handles.output 这个变量。（当然在后面别忘了添加 guidata(hObject, handles);保存修改）。因为默认的 output 函数里面 varargout{1}=handles.output，也就是修改了 vararout。

（所以知道了原理，output 只是一个名字而已，你也可以用任意别的名字，或者添加更多的输出变量，只要在 outputFcn 下面添加 varargout{2}=handles.第二个变量名 。。。。类似）

guidata(hObject, handles)之后不要立即 delete 窗口命令，因为这时候 outputfcn 没有再执行一次，也就是你刚刚修改的 output 并没有更新到 varargin 里面去。所以要么单独设计一个关闭按钮；要么跟 uiwait(handles.figure1);uiresume 合用。

5.3.3 Callbacks（回调函数）

当你对组件做点击或者别的动作，则自动调用相应的 callback。callback 的名字取决于你的控件的 tag 和控件类型以及响应类型。例如

```
function print_button_Callback(hObject, eventdata, handles)
```

通用callback 写法请参考：http://www.matlabsky.com/thread-3566-1-1.html

6 MATLAB 在 GUI 中实现多标签页的方法

贴子地址：http://www.matlabsky.com/thread-5187-1-1.html

作者：dynamic

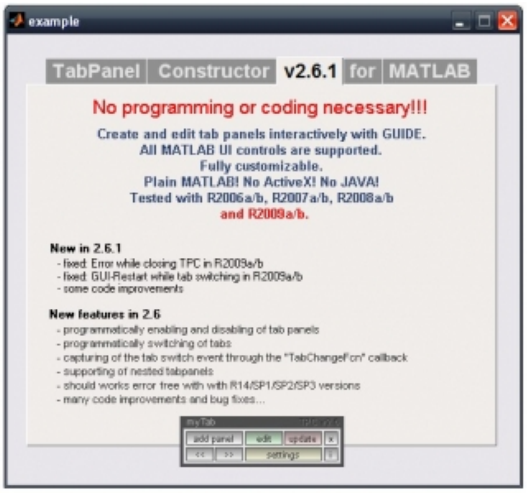
MATLAB 的 GUI 中没有直接提供想 VB 中多标签页的控件，我在 ActiveX 找了好久也没有找到！

但是仔细想一想，其实这个实现起来也不是很困难，只是有时需要使用一些障眼法而已！！好下面我们就列出一些直接的例子给大家参考下！

6.1 TabPanel Constructor v2.6.1 (2009)

本地下载：http://www.matlabsky.com/thread-5187-1-1.html

官方下载：http://www.mathworks.com/matlabcentral/fileexchange/6996



这里压缩包中直接提供了的多标签面板构造器，我们可以在 guide 中直接很容易的编辑修改。

(1)不需要任何编程；

(2)可以在 GUIDE 中直接编辑；

(3)支持所有 Matlab 的 GUI 函数；

(4)没有使用任何 ActiveX 控件和 Java 组件。

6.2 Multiple Tab GUI

本地下载：http://www.matlabsky.com/thread-5187-1-1.html

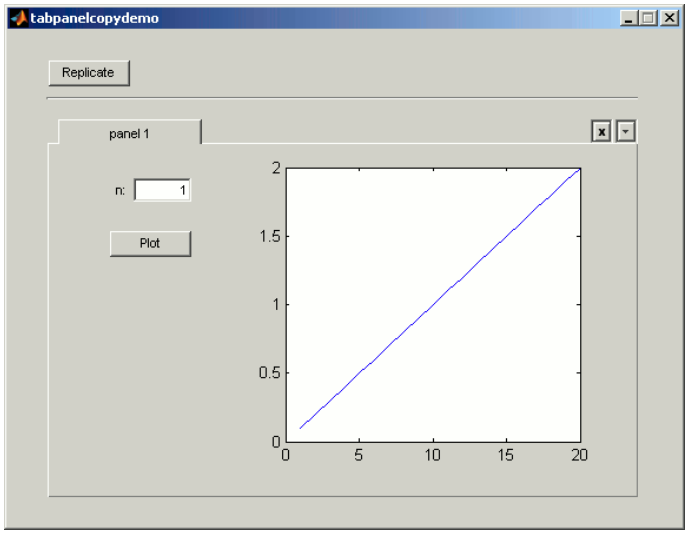
官方下载：http://www.mathworks.com/matlabcentral/fileexchange/25938

创建一个拥有多标签页的 figure，并演示如何使用。标签页的数量可以改变，程序自动读取屏幕尺寸来调整 figure 的大小。需要注意的是，本程序基于 Matlab2009b，如果您不是，请将 210-234 行中“~”改为 tmp1 和 tmp2，因为 Matlab2009b 使用“~”表示不使用的参数，而在之前版本是不支持的！

6.3 Tab panel Copy

本地下载：http://www.matlabsky.com/thread-5187-1-1.html

官方下载：http://www.mathworks.com/matlabcentral/fileexchange/24346



运行压缩包中 tabpanelcopydemo.m 程序，点击 replicas 按钮，此时会自动添加标签页，其核心程序是 tabpanelcopyfcn.m，换句话说你可以在其它 GUI 中直接调用这个函数，实现多标签页。

6.4 GUI 简单标签页程序

这几个代码都是相当简单障眼法，不同按钮时显示不同的东西，将不显示的东西隐藏，或者改变它们的 Postion 属性，不过为了视觉上的效果，需要设置选中和未选标签的颜色和外形，这个我们可以标签页放在一个 panel 面板，如果设置 BorderType 和 color 属性就好了！

6.4.1 Simple Tab with GUIDE

本地下载：http://www.matlabsky.com/thread-5187-1-1.html

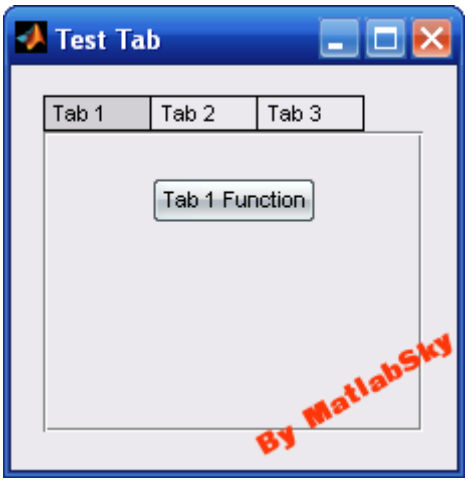
官方下载：http://www.mathworks.com/matlabcentral/fileexchange/15238

主要使用了 visible 属性，适当照顾视觉效果，比如 color 还有 bordertype 等。

6.4.2 Simple tab panel

本地下载：http://www.matlabsky.com/thread-5187-1-1.html

官方下载：http://www.mathworks.com/matlabcentral/fileexchange/15193

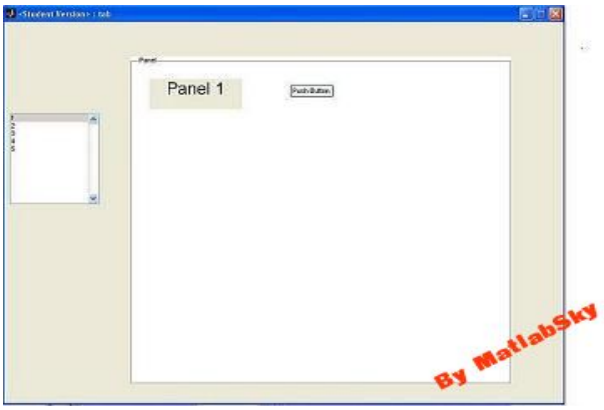


主要是修改 postion 属性和 visible 属性来达到障眼的目的，不过程序在欲行时中创建 3 个 axes，覆盖于那 3 个标签页上，此时点击鼠标时激活的是坐标系的 ButtonDownFcn 回调函数，这个步骤我感觉有点多余。

6.4.3 a new way to tab for GUIDE

本地下载：http://www.matlabsky.com/thread-5187-1-1.html

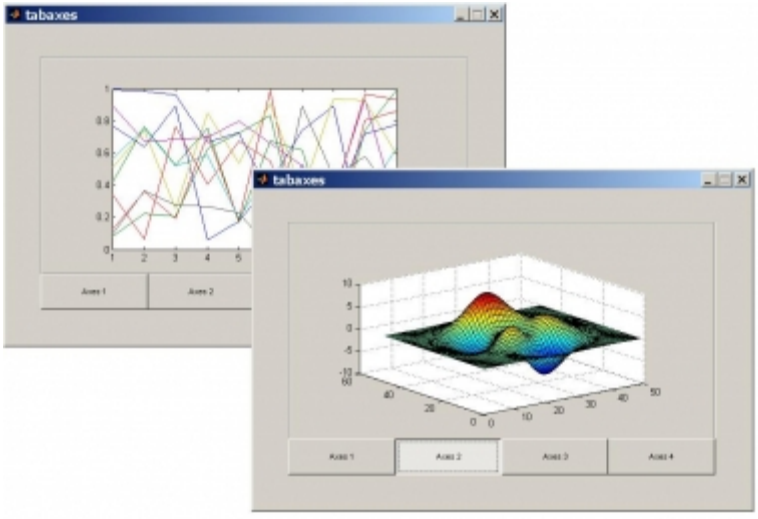
官方下载：http://www.mathworks.com/matlabcentral/fileexchange/14599



6.4.4 Tab Axes Gui

本地下载：http://www.matlabsky.com/thread-5187-1-1.html

官方下载：http://www.mathworks.com/matlabcentral/fileexchange/17815

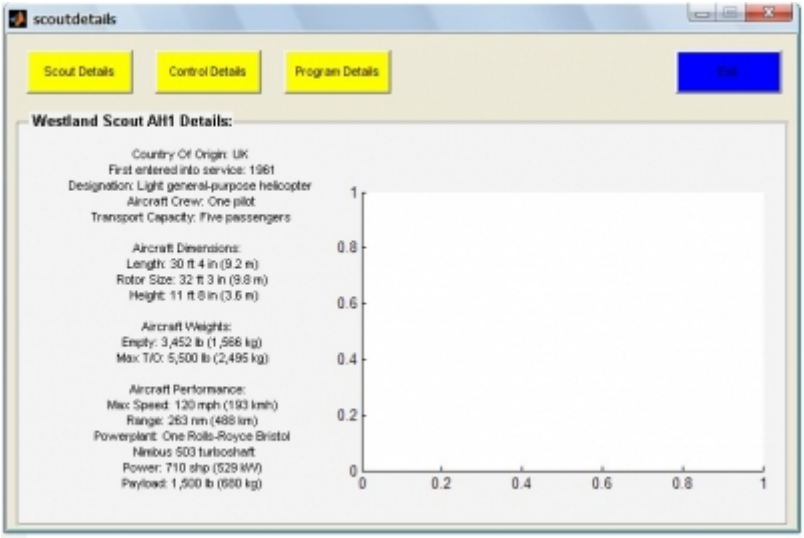


这个就更加简单，直接使用 togglebutton 控件来作为标签页，多个 axes 重合，需要时就显示那个！

6.4.5 Simple Tab Example

本地下载：http://www.matlabsky.com/thread-5187-1-1.html

官方下载：http://www.mathworks.com/matlabcentral/fileexchange/7361



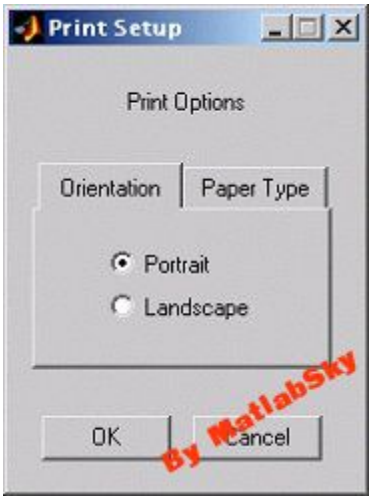
6.5 自己编写相关 M 函数，然后调用

以下程序是调用了自己编写的多标签页 M 函数，和上面直接的障眼法有些区别，我们可以在其它 GUI 中直接调用这个函数！

6.5.1 Tab panel example

本地下载：http://www.matlabsky.com/thread-5187-1-1.html

官方下载：http://www.mathworks.com/matlabcentral/fileexchange/1741



6.5.2 tabpanel，This tool allows the creation of tab panels within MATLAB

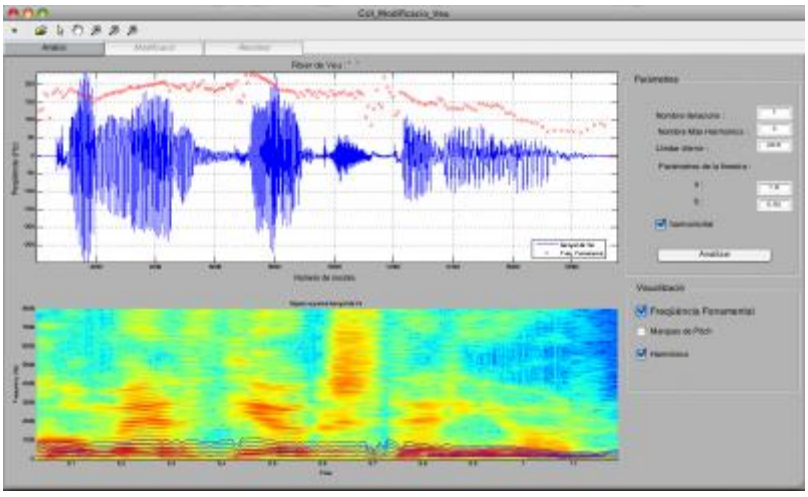
本地下载：http://www.matlabsky.com/thread-5187-1-1.html

官方下载：http://www.mathworks.com/matlabcentral/fileexchange/2852

6.6 高亮显示多页标签(Highlight Tab objects)

本地下载：http://www.matlabsky.com/thread-5187-1-1.html

官方下载：http://www.mathworks.com/matlabcentral/fileexchange/22488



本来这代码没有是什么新颖的，同样使用障眼法实现多标签页，但是它的标签使用的是 pushbutton，然后使用自己编写的 highlight.m 函数，修改 pushbutton 的颜色、字体外观，值得学习。

7 Excel、listbox、edit 和 inputdlg 之间的交互操作

贴子地址：http://www.matlabsky.com/thread-10771-1-1.html

作者：makesure5

这次的原创是关于 Excel、listbox、edit 和 inputdlg 之间的交互操作，Matlab GUI 的常规控件相比于 Delphi，C#少得可怜，所以会感觉到用来用去无非就是那几个控件，不过，说实话，能把这几个控件掌握好，做起 Project 还是很容易的，无非就是界面的美观程度，操作简易的区别，功能还是可以实现的。

作为人机界面，要求的就是能够很好的互相交流，纯粹的输入或者纯粹的输出显得不够友好，只有交互的操作才能使界面使用起来更加方便。事实上，GUI 中的控件往往具有双重性，即既可用于显示数据，也可以用与输入数据，例如：edit 框用 get 就是输入数据，用 set 就是显示数据。

在输入方面，我们往往需要输入一系列参数，这种情况下，建议不要老是使用 edit 输入，我们可以用 inputdlg 代替，因为它上面就有很多“edit”，而且可以是动态变化的，需要输入 3 个参数，它上面就显示 3 个 edit，输入 4 个它就显示 4 个，非常方便。

```
answer = inputdlg(prompt,dlg_title,num_lines,defAns)
```

虽然从字面上理解，inputdlg 是用于输入数据的对话框，其实如果能够设置好 defAns，它也可以用于显示数据（即默认数据）。

而在输出方面，Listbox 是很常用的控件，虽然无法在它上面直接编辑，但是可以通过配合其他控件达到目的。除了控件，往往还会用数据文件来记载或者显示数据，其中常见的就是 Excel，xlsread 和 xlswrite 2 个函数可以说在 GUI 中是经常用到。

下面我将介绍我的一个小作品，来配合上面所阐述的个人看法。这个作品实际是为了设置产品的某些参数，涉及到添加、修改和删除等操作！作品用脚本文件编写，理论上支持所有版本，前提是版本中包含那些控件和函数。作品的后台数据源是一个名为“位置信息”的 Excel 文件，记录的是各个产品的参数信息，如图所示：

	A	B	C	D	E
1	方案名称	电堆电压	电堆电流	氢气流量	辅助功率
2	同济大学	7	8	20	27
3	上海大众	1	2	3	4
4	神力公司	1	3	5	7
5	大连化工	2	4	6	8
6					

运行程序后，主界面如下图：

通过点击 listbox 中的名称，可以在 edit 中观察到各个产品的具体参数。如果需要修改具体的参数，修改完后，点击“保存方案”即可。如果需要添加新的方案，点击“添加方案”，会出现 inputdlg 输入参数：

注意，必须每个参数都得输入，否则会提示错误！

如果要删除 listbox 中的参数，可以点击删除方案，实际上就是删除 Excel 中的一行数据，这个精彩见到会员提起。我采取的方法是将要删除的那一行后面的数据各自往上移一行，然后再将最后一行写空白，因为最后一行和倒数第二行是重复的。这样就达到了删除的效果。这里应注意删除的是中间一行或者是最后一行是不同的方法。

注意： 完整程序需要到原帖地址（http://www.matlabsky.com/thread-10771-1-1.html）下载。

8 在 GUI 中查询并选择 Matlab 工作空间(workspace)中的变量

这是一个集成帖，集合了 3 个优秀的内容相近的帖子。

8.1 在 GUI 中查询并选择 Matlab 工作空间(workspace)中的变量

贴子地址：http://www.matlabsky.com/thread-13249-1-1.html

作者：xiezhh

大家做界面的时候可能遇到过这样的问题：如何查询并选择 Matlab 工作空间（workspace）中的变量？

关键是我们可能事先不知道 workspace 中都有什么变量，就想通过一个界面查看 workspace 中的所有变量，并选择所需变量。为此，我做了一个 GUI 界面，用来通过界面操作查询并选择工作空间里的变量。GUI 界面中带有下拉菜单，用来显示工作空间里的所有变量的相关信息，包括变量名，变量类型和大小。

界面效果如下图：

程序中用到的图标文件如右：

程序代码如下：

```
function xie_vardata = VarSelect
%选择工作空间里的变量
%   vardata = VarSelect，通过 GUI 界面操作选择工作空间里的变量. GUI 界面中带有下拉
%   菜单，用来显示工作空间里的所有变量的相关信息，包括变量名，变量类型和大小.
%
%   Copyright  xiezhh.
%   $Revision: 1.0.0.1 $  $Date: 2010/06/11 09:43:00 $
OldHandle = findobj( 'Type', 'figure', 'Tag', 'VarSelect' ) ;
if ishandle( OldHandle )
    close( OldHandle ) ;
end
scnsize = get(0,'screensize');
fs = 12*sqrt(scnsize(3)*scnsize(4)/(1024*768));
fig = figure('units','normalized','position',[0.35 0.35 0.3 0.2],...
    'menubar','none','name','选择工作窗口变量','resize','off',...
    'numbertitle','off','color',[0.925 0.914 0.847],'tag','VarSelect');
icon_VarSelect = imread('tubiao.jpg');
uicontrol(fig,'style','radiobutton','units','normalized',...
    'pos',[0.025 0.3 0.265 0.48],'string','',...
    'cdata',icon_VarSelect,'bac',[0.925 0.914 0.847])
uicontrol(fig,'style','text','units','normalized',...
    'pos',[0.475 0.73 0.35 0.15],'string','选择变量',...
    'fontsize',fs,'fontweight','bold','fontunits','normalized')

s = evalin('base','whos');
```

```
VarNames = char(s.name);
if isempty(VarNames)
    VarNames = '当前窗口没有变量';
    popstring = VarNames;
else
    for i = 1:size(VarNames,1)
        popstring{i,:} = [VarNames(i,:),',  类型: ',s(i).class,',   大小: ',...
            num2str(s(i).size(1)),'X',num2str(s(i).size(2))];
    end
end
setappdata(gcf,'VarNames',VarNames);

uicontrol(fig,'style','popupmenu','units','normalized',...
    'pos',[0.35 0.33 0.59 0.337],'string',popstring,...
    'fontsize',fs,'fontunits','normalized',...
    'backgroundcolor',[1 1 1],'tag','VarShow','value',1)
uicontrol(fig,'style','push','units','normalized',...
    'pos',[0.4 0.2 0.18 0.21],'string','确  定',...
    'fontsize',fs,'fontweight','bold',...
    'fontunits','normalized','callback',...
    ['handles = guidata(gcf);',...
    'value = get(handles.VarShow,''value'');'...
    'VarNames = getappdata(gcf,''VarNames'');'...
    'if ~strcmp(VarNames,''当前窗口没有变量'');'...
        'bs = VarNames(value,:);'...
        'xie_vardata = eval(bs);'...
        'set(gcf,''userdata'',xie_vardata);'...
    'end;'...
    'evalin(''base'',''clear value VarNames handles bs xie_vardata'');'...
    'uiresume(gcbf);']);
uicontrol(fig,'style','push','units','normalized',...
    'pos',[0.7 0.2 0.18 0.21],'string','取  消',...
    'fontsize',fs,'fontweight','bold',...
    'fontunits','normalized','callback',...
    ['set(gcf,''userdata'',[]);'...
    'delete(gcbf);']);
handles = guihandles(gcf);
guidata(gcf,handles);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
uiwait(gcf);
if nargout>0
    xie_vardata = get(gcf,'userdata');
end
if ishandle(fig)
    delete(fig);
end
```

8.2 将保存在 workspace 里的数据导入到 GUI 中绘图

贴子地址：http://www.matlabsky.com/thread-4067-1-2.html

作者：f1v2b35

最近一直在学习怎么将 workspace 里面已经有的数据导入到 GUI 中绘图。刚也是参考了一“院长”的杰作才完成的。很简单。但是传来大家指点下。呵呵，里面有很多不足的地方，希望知道改进的给点建议以及方法。

运行方法：

1．在 Command Window 中输入 load mine.mat

2．运行 mine.fig 文件，点击 plot 按钮

3．自动调用 workspace 中的变量 x 和 y 进行绘图

核心代码如下：

```
function pushbutton1_Callback(hObject, eventdata, handles)
x=evalin('base','x');
y=evalin('base','y');
plot(x,y);
xlabel('t');
ylabel('love');
title('love no fade');
grid on;
```

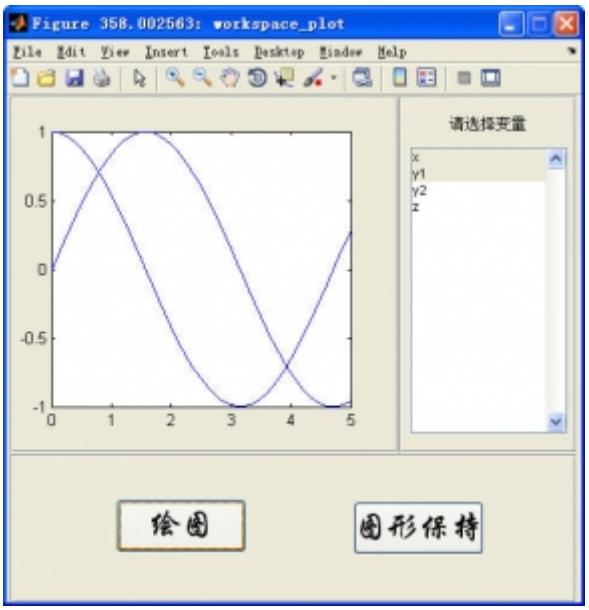
注意： 完整程序请到原帖地址下载。

8.3 将保存在 workspace 里的数据导入到 GUI 中绘图

贴子地址：http://www.matlabsky.com/thread-6377-1-2.html

作者：海上飞鱼

为了响应论坛的号召，这里利用 GUIDE 写了个在 GUI 中获取工作空间变量并作图的程序，效果图如图所示。



注意： 完整程序请到原帖地址下载。

类似的帖子还有：

1．把 workspace 中的数据导入 gui 中小程序

帖子地址：http://www.matlabsky.com/thread-2881-1-1.html

作者：tubehu

2．如何把 GUI 里的运算结果存入 Workspace

帖子地址：http://www.matlabsky.com/thread-4938-1-1.html

提问者：siegfried

回复者：dynamic

MATLAB 基础应用版块

9 绘制两圆公切线的 MATLAB 代码

贴子地址：http://www.matlabsky.com/thread-13241-1-1.html

作者：xiezhh

补充回复者：dynamic

已知两圆的圆心坐标和两圆半径，可以调用本程序绘制两圆公切线（若有公切线的话）。废话不说了，上程序：

```
function common_tangent(x1,y1,r1,x2,y2,r2)
% 画两圆的公切线
% x1  -- 第一个圆的圆心横坐标
% y1  -- 第一个圆的圆心纵坐标
% r1  -- 第一个圆的半径
% x2  -- 第二个圆的圆心横坐标
% y2  -- 第二个圆的圆心纵坐标
% r2  -- 第二个圆的半径
% 用到了坐标平移和旋转
% by：xiezhh（谢中华）
% 2011.2.27

d = sqrt((x1-x2)^2+(y1-y2)^2);
d1 = abs(r1-r2);
d2 = r1+r2;

if d < d1
    error('大圆包含小圆，没有公切线')
end
if isequal([x1 y1 r1],[x2 y2 r2])
    error('两圆重合，有无穷多条公切线')
end

t = linspace(0,2*pi,100);
x = x1+r1*cos(t);
y = y1+r1*sin(t);
plot(x,y,'r')
hold on
x = x2+r2*cos(t);
y = y2+r2*sin(t);
plot(x,y,'r')
```

```
xmin = min([x1 x2]-[r1 r2])-min([r1 r2])/5;
xmax = max([x1 x2]+[r1 r2])+min([r1 r2])/5;
ymin = min([y1 y2]-[r1 r2])-min([r1 r2])/5;
ymax = max([y1 y2]+[r1 r2])+min([r1 r2])/5;

XYR = [r1 x1 y1;r2 x2 y2];
XYR = sortrows(XYR);
r = XYR(1,1);
R = XYR(2,1);
xy = XYR(1,2:3)';
axis([-1.2*r d+R+0.2*r -R-0.2*r R+0.2*r]);

fun = @(beta)[abs(beta(2))/sqrt(1+beta(1)^2)-r;...
abs(beta(1)*d+beta(2))/sqrt(1+beta(1)^2)-R];

theta = atan((XYR(2,3)-XYR(1,3))/(XYR(2,2)-XYR(1,2))) + pi*(XYR(1,2) > XYR(2,2));
A = [cos(theta) -sin(theta);sin(theta) cos(theta)];

if d == d1
    h = line([-r;-r],[-R;R]);
    reline(h,A,xy);
else
    [k,jieju] = SlopeIntercept(0,-r,d,-R);
    ab = fsolve(fun,[k jieju]);
    h = refline(ab);
    reline(h,A,xy);

    [k,jieju] = SlopeIntercept(0,r,d,R);
    ab = fsolve(fun,[k jieju]);
    h = refline(ab);
    reline(h,A,xy);

    if d == d2
        h = line([r;r],[-R;R]);
        reline(h,A,xy);
    elseif d > d2
        [k,jieju] = SlopeIntercept(0,-r,d,R);
        ab = fsolve(fun,[k jieju]);
        h = refline(ab);
        reline(h,A,xy);

        [k,jieju] = SlopeIntercept(0,r,d,-R);
        ab = fsolve(fun,[k jieju]);
        h = refline(ab);
        reline(h,A,xy);
    end
end

axis equal
axis([xmin xmax ymin ymax])

function [Slope,Intercept] = SlopeIntercept(x1,y1,x2,y2)
Slope = (y2-y1)/(x2-x1);
Intercept = (y1*x2-x1*y2)/(x2-x1);
```

```
function reline(handle,A,xy)
xydata = A*[get(handle,'Xdata');get(handle,'Ydata')]+[xy xy];
set(handle,'Xdata',xydata(1,:),'Ydata',xydata(2,:))
```

调用举例：

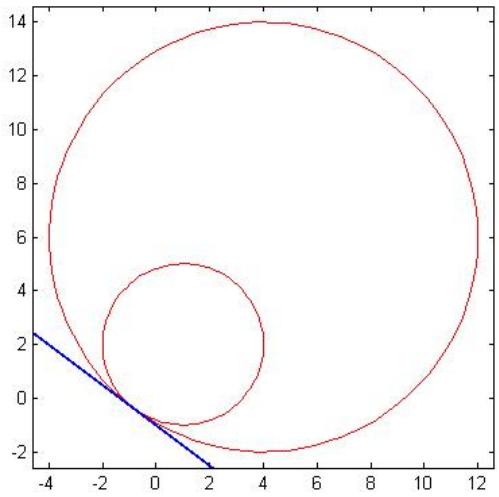
【例 1】两圆相套

```
>> common_tangent(0,0,10,0,1,3)
??? Error using ==> common_tangent at 18
大圆包含小圆，没有公切线
```

【例 2】两圆内切

```
>> common_tangent(1,2,3,4,6,8)
```

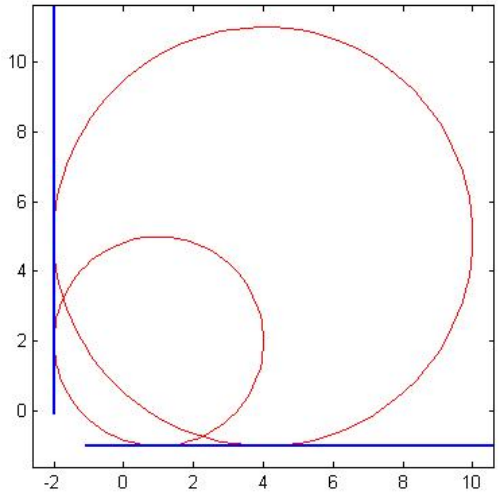
效果图：



【例 3】两圆相交

```
>> common_tangent(1,2,3,4,5,6)
```

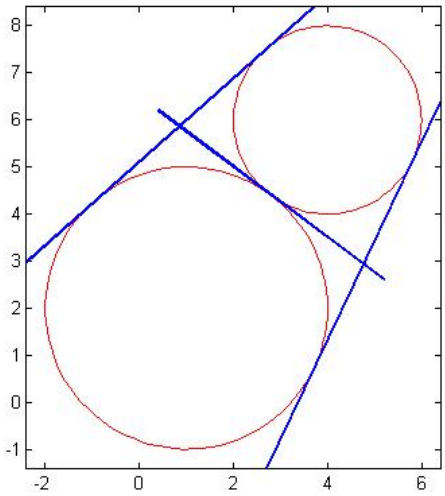
效果图：



【例 4】两圆外切

```
>> common_tangent(1,2,3,4,6,2)
```

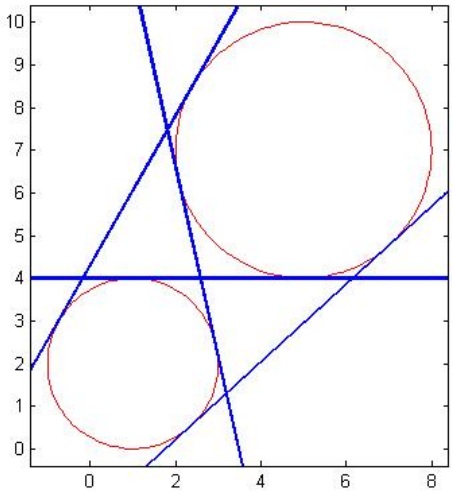
效果图：



【例 5】两圆相离

```
>> common_tangent(1,2,2,5,7,3)
```

效果图：



========== mttc 关于程序原理的提问 ==========

谢老师能否讲解下您求解公切线的原理，这样方便大家理解程序的内容。辛苦谢老师了

========== dynamic 关于程序原理的回复 ==========

哈哈，谢老师这个原理是这样的：

1．判断是否两圆是否存在公切线

2．以第一个圆心为原点，将两圆圆心连线旋转成 X 轴（也就是相当于在 X 轴上绘制了两个圆，第一个圆圆心正好在原点），

此时第一个圆圆坐标为（0,0），第二个圆心坐标为（d,0），其中 d=r+R 为两圆圆心距离

3．计算 X 轴上两圆两个最低点（或一高一低两点）连线的斜率和截距

4．计算 X 轴上两圆的公切线，用第三步中的数据作为初值，解方程组：

假设公切线方程为 y=kx+b，即 kx-y+b=0，那么圆心到公切线的距离必须等于半径，也就是

切线到第一个圆的距离：r=abs(b)/sqrt(1+k^2)，注意第一个圆圆坐标为（0,0）

切线到第二个园的距离：R=abs(kd-b)/sqrt(1+k^2)，第二个圆心坐标为（d,0）

联立上面两个方程，即可求解出 k 和 b

5．将第 4 步得到公切线方程进行坐标旋转和平移成原始坐标系下的切线。