

<https://www.zhihu.com/question/24499729>

谢不邀

提问都说了是命令，大家回答那么多函数干什么... 我来给一个超级大杀器
在命令行敲入 `dbstop if error`

如果运行出现错误，matlab 会自动停在出错的那行，并且保存所有相关变量。再也不用设断点

了有没有!!!

`bsxfun`

强大的、万能的、不同维数的矩阵扩展混合运算，从此告别矩阵运算中的 for 循环

另，matlab 里所有以 fun 为后缀的命令都很好用，`arrayfun`，`cellfun`，`structfun`，等等

@() 匿名函数

使用函数式编程，在编写以数学公式为主的程序中比传统编程方法好得多

`set(gca,...)`等

所有与画图相关的命令都包含了千奇百怪的属性和参数，非常实用，如著名的

`set(gca,...)`

\$ \$

画图中的标题、注释等文字内容是可以直接使用 latex 格式的，只须在包含

latex 命令的部分前后用\$框起来即可

slice

高维切片，要在三维空间中表现三个以上变量的函数关系，必备此命令

Matlab 里面应该多用向量运算，把循环语句转变为向量运算会省很多时间，程序也更简洁易读。

比方说，一个名叫 array 数组里面，你要将里面大于 1 的都变成 0，就不必用到循环：

```
array(array>1)=0;
```

把大于 1 小于 3 的变成 0。

```
array(and(array>1,array<3))=0;
```

当然，还可以使用 find，这个也很好用。

另一个重要的技巧是 delete/clf-plot-pause

用 plot 可以画图（注意记录句柄），然后用 delete 删掉特定图象，或用 clf 清图，再绘制，这可以在 figure 窗口产生动画。但是如果只 plot，往往只会在全部程序执行结束时显示，这时候需要用 pause 让 figure 完成图像的更新。

drawnow 貌似也可以，但是我比较喜欢用 pause，能够简单地控制动画的速度。

这会方便调试和展示。这个技巧尤其适合使用 matlab 的图形用户界面设计功能时构造一个显示运行状态等信息的 figure。

mathworks 有一个 fig 函数（不是系统自带的，是别人编写的），可以很容易地调整字体、尺寸以及绘图是否有边框等等，不必画出来再自己手动调整。这对于写论文的人来说会很方便。

善用 eval，可以让你的程序的灵活度大大加强。尤其是在变量名的问题上。当然，这可能会对代码维护和调试带来麻烦。很多情况可以用其他方法代替。@王备 指出了[一个链接](#)，值得一读 [Alternatives to the eval Function](#)

save、load 可以将部分或全部变量、结构体等存入 mat 文件或从 mat 导入 workspace

global 可以将变量变为全局变量，在各函数之间共享。不过这不太好用，尽量慎用吧。

exist 可以检测某目标（如变量）是否存在，减少一些麻烦。

surf、mesh 都很漂亮，不过 surf 之后记得用 shading interp，看起来更漂亮。

对于一些重复性的矩阵赋值，比如：1、2、3、4、1、2、3、4
可以使用 repmat，将一个矩阵重复扩展为更大的矩阵。

很多函数都有高级的用法，当使用到了，但又觉得有点麻烦的时候，不妨 help 一下，看看其他的用法。

总而言之，还是多上网搜索，一般问题总有很好的答案。

1. 转置但不共轭

```
b = a.'
```

2. 得到列向量

```
b = a(:)
```

相应的，得到行向量就是

```
b = a(:).'
```

3. 清除并更新有 class 的定义

```
clear classes
```

当你修改了某些 class 的定义时，需要调用此命令——仅仅 clear all 是不够的。

4. 忽略某些返回参数

```
[~, idx] = max(a)
```

忽略最大值，只返回最大值的索引。

【注】

这种语法是在 Matlab 7.9 (2009b) 中引入的，在此之前的版本不支持这种用法。如果你使用早期版本，可以使用如下写法：

```
[tmp, idx] = max(a); clear tmp
```

或者一句话搞定

```
[idx, idx] = max(a)
```

不过第二种写法不见得每个人都会喜欢。

5. 去除长度为 1 的维度 (常用于 3D 矩阵转为 2D)

```
b = squeeze(a)
```

Simulink 的 save to workspace 模块经常得到一个 $[m \times 1 \times n]$ 的三维矩阵，直接使用非常不便。使用 squeeze 后，会将其转换为 $[m \times n]$ 的二维矩阵。

6. 代码运行耗时显示

```
tic; some_code_to_run(); toc
```

tic/toc 会在 Matlab 命令行中打印出两条命令之间的代码的运行耗时。更详细的代码效率分析，可以使用 profile 命令。

7. 电脑跑分

```
bench
```

只能用于相同版本的 Matlab 下电脑之间的比较。

8. 记录命令行命令

```
diary on  
% Lots of my Matlab commands here.  
diary off
```

使用 diary 好处是运行结果也会被记录下来，并且以纯文本方式存储，方便编辑。更正式的报告，则推荐使用 publish。

9. 反转向量

```
b = fliplr(a)    % For row-vector a.  
b = flipud(a)   % For column-vector a.  
b = wrev(a)      % For any vector a.  
b = a(end:-1:1); % This is the implementation of function wrev.
```

10. 去除数据中的 NaN

```
a(isnan(a)) = []
```

对于 inf 和 -inf, 使用 isinf 命令。

答个冷门的, 未必人人都相见恨晚, 但对于合适的项目还是有点用。

其实也很简单, 就是 sound 函数, 能发出声音, 可以在程序跑完的时候来两下。因为我的一个项目需要反复跑程序调参数, 程序跑一次从 10 分钟到几小时不等都有可能。很多时候电脑开着跑程序, 我在旁边干别的事, 如果为了不浪费时间就要隔一会看看电脑跑完程序没, 觉得很烦。后来想到能不能让电脑跑完主动告诉我? 就搜到了这个函数。当然网上还有更高端的, 程序运行到指定位置可以给自己的邮箱发个邮件, 这样只要有网的地方就能通过手机之类随时远程获知了。。。

然后觉得自己之前真是 sb, 为什么早没想到。。。

另外再补充一条: shutdown 命令, 可以自动关机, 并且可以设置延时。很多时候为了赶时间, 我都在睡前最后一次跑程序之前在程序末尾加上 shutdown (当然之前还要加 save), 然后就可以呼呼睡大觉了, 程序跑完会自动关机的。另外 windows 的命令框里也可以 shutdown (可能就是从那里来的), 不过估计很少有人用。

1. 关于向量化

大家都知道，如果用一个函数对一个矩阵中的每一个元素进行求值，那么就要用到 `.*`, `./` 之类

```
function r = fmat(x)
    r = x.^2 + 1./x;
end

A = [1 2 3 4];
fmat(A)
```

上面这样就可以做到用 `fmat` 对 `A` 每一个元素分别求值。但不是所有的函数都像 `fmat` 这样，下面这种情况就不行：

```
function r = fmat2(x)
if x > 0
    r = x.^2;
else
    r = 1./x;
end

fmat2(A)    % 会出错
```

可以看到，因为上面 `fmat2` 里面有判断语句，就不能把整个矩阵作为输入参数。我知道的解决方法有 3 种：

```
% 法 1
for i = 1:length(A)
    r(i) = fmat2(A(i));
end

% 法 2 用 arrayfun
r = arrayfun(@fmat2, A)

% 法 3 用逻辑矩阵

r = zeros(1,length(A));
l1 = (A>0);
l2 = ~l1;
r(l1) = A(l1).^2;
r(l2) = 1./A(l2);
```

值得说一说的是第二种和第三种。**arrayfun** 只是比用 for 更加简洁，速度上没有太大差别。但是，用**逻辑矩阵**的话，就比前两种快 10 倍。逻辑矩阵是向量化的利器。比如 $A > 0$ 就会返回一个逻辑矩阵，里面全是 1, 0。并且，逻辑矩阵可以当作索引， $A(A > 0)$ 就把所有 $A > 0$ 的数返回。

2. 类型转换

matlab 中一共有 4 种类型：func(函数句柄), num(数值), sym(符号), str(字符)

- 在数值运算中，使用 func, num
- 需要输入输出，要用到 str
- 进行符号运算(符号微分，积分)时，用 sym

我们使用 matlab，一般有这么个过程：

- 先进行公式推导，这时要用 sym, symfun 类型
- 再进行数值计算，这里要用普通的 func, 和 num 类型
- 最后与 GUI 界面交互的时候，要用 str 类型

这里介绍几个用于转换的函数

```
str2func('@(x,y)sin(x*y)') % str -> @func 返回一个函数句柄

syms x y
fs(x,y) = x^2+sin(x*y);
fh=matlabFunction(fs); % symfun -> @func 返回一个函数句柄 !!!
强烈推荐

fh = @(x)x.^2+sin(x);
fun2str(fh) % @func ->str 函数句柄变为字符
```


3. 三维画图

大家都晓得，作图第一步，对 x, y 划分网格，用的是 `meshgrid`。这样画出来的图在一个立方体的范围内，但是有时候我们要显示一个**圆柱体内的三维图形**这时候画网格可以用 `cylinder`

```
[x, y] = cylinder(linspace(0,10,100),200) % linspace(0,10,100)指在半径[0, 10]上划分为100份，参数200指的是在圆周方向上200等分
z = sin(x) + cos(y);
mesh(x, y, z);
```

4. 学无止境

学 matlab，还是要去论坛，看看大神们的说法，大神们的做法。比如说 1 中的向量化

```
function r = fmat2(x)
if x > 0
    r = x.^2;
else
    r = 1./x;
end

fmat2(A) % 会出错
```

论坛中的大神的做法是这样的

```
fmat3 = @(x)x.^2.*(x>0) + 1./x.*(x<=0);
fmat3(A)
```

极其简洁!!! 我看到后非常佩服

先这些吧