



ARTICLE

第1篇 ●

MATLAB 基础篇

MATLAB程序设计基础

读者将在本章了解 MATLAB 的基本程序控制结构以及 MATLAB 中几种基本的数据类型,同时掌握 MATLAB 的脚本、函数两种 M 文件的编写方法,并能够掌握一些提高程序执行效率的编程方法。

1.1 MATLAB 数据类型

MATLAB 有 17 种基本的数据类型,每种类型的数据都以矩阵或数组的形式存在。其中常用的数据类型有数值型、字符和字符串、元胞数组和构架数组。下面具体介绍这 4 种数据类型。

1.1.1 数值型

数值型数据包括有符号和无符号整数、单精度和双精度的浮点数。在默认状态下, MATLAB 将所有数据都看成是双精度的浮点数。但从节省空间的角度看,整数和单精度数组更节省内存空间。

所有的数值型数据都支持数组操作,如下标操作和尺寸的修改,除了 int64 和 uint64,都可以用于数学运算。

1. 整型

MATLAB 提供了 4 种带符号数和 4 种无符号数整型数据类型,整数的数据类型及其取值范围见表 1.1。

表 1.1 整数的数据类型及其取值范围

数据类型	取值范围	转换函数
单精度 8 位整数	$-2^7 \sim 2^7 - 1$	Int8
单精度 16 位整数	$-2^{15} \sim 2^{15} - 1$	Int16
单精度 32 位整数	$-2^{31} \sim 2^{31} - 1$	Int32
单精度 64 位整数	$-2^{63} \sim 2^{63} - 1$	Int64
无符号 8 位整数	$0 \sim 2^8 - 1$	Uint8
无符号 16 位整数	$0 \sim 2^{16} - 1$	Uint16
无符号 32 位整数	$0 \sim 2^{32} - 1$	Uint32
无符号 64 位整数	$0 \sim 2^{64} - 1$	Uint64



整数可以参与的运算有下面两种：

(1) 具有相同数据类型的整数或整数数组,运算结果的数据类型与操作数相同。

```
如: >> a = uint8(4)
>> b = uint8(9)
>> c = a * b
c =
    36
```

(2) 整数或者整数数组与下标 double 型浮点数进行数值运算,在运算过程中保持浮点运算精度,但是最终结果为整数。

```
如: >> a = 0.3
>> b = uint8(9)
>> c = a * b
c =
     3
```

2. 浮点数

浮点数有单精度(4 字节)和双精度(8 字节)两种,默认是双精度。两种格式可以进行强制转换。

(1) single: 将其他类型的数据转换成单精度浮点数。

(2) double: 将其他类型的数据转换成双精度浮点数。

例如:

```
>> a = uint8(4)
>> b = double(a)
>> class(a)
ans =
    uint8
>> class(b)
ans =
    double
```

class 函数返回的是对象类型,从上例可以看到,变量 a 原来是无符号 8 位整数,通过 double 函数将其变换成双精度浮点型数据。表 1.2 列出了常用的浮点数操作函数。

表 1.2 常用的浮点数操作函数

函 数 名	函 数 功 能
isfloat	检查输入是否为浮点数
realmax	返回本计算机能够表示的最大浮点数
realmin	返回本计算机能够表示的最小浮点数
eps	浮点相对精度
isreal	检查是否数组的所有元素为实数

1.1.2 字符和字符串

字符和字符串是 MATLAB 的重要组成部分。MATLAB 具有强大的字符串处理功能,在 MATLAB 中,字符串和字符数组基本上是等价的。



1. 字符

在 MATLAB 中,字符是以其 ASCII 码表示的,这样可直接在屏幕上显示字符或者在打印机上打印字符。输入字符数据时应用单引号括起来,例如输入:

```
>> name = '王'
```

这时采用 class 命令可以检查其类型。

```
>> class(name)
ans =
    char
```

这说明变量 name 的类型为字符型。

```
>> whos
Name      Size      Bytes  Class
name      1x1          2    char
```

通过 whos 函数查询得到,变量 name 含有 1 个汉字,占用了 2 个字节,这说明每个字符都采用 16 位的 ASCII 码存储,这大大方便了在 MATLAB 中使用双字节内码字符集,如汉字系统。利用 double 和 char 函数可在字符与其 ASCII 码之间进行转换。

2. 字符串

把字符放在一对单引号中,就定义了一个一维的字符数组,一维字符数组也称为字符串或字符向量。例如:

```
>> university = '上海工程技术大学'
university =
    上海工程技术大学
>> whos university
Name      Size      Bytes  Class
university 1x8        16    char
```

由于 MATLAB 语言是采用 C 语言开发的,因此它的字符串操作与 C 语言的相应操作基本相同。表 1.3 给出了 MATLAB 常用的字符串操作函数及其功能说明。

表 1.3 常用的字符串操作函数及功能说明

函数名	函数功能	函数名	函数功能
size	查看字符数组的维数	abs	查看一个字符的 ASCII 码
char	把数字按照 ASCII 码转换为字符串	strcat	字符串连接
strcmp	比较字符串	strcmp	替换字符串
strcmpi	忽略大小写比较字符串	upper	转换为大写
strncmp	比较字符串前 N 个字符	lower	转换为小写
findstr	在一个字符串中查找另一个字符	strtok	返回字符串中第一个分隔符前的部分
strjust	对齐字符数组	blanks	产生空字符串
strmatch	查找匹配的字符串	deblank	删除字符串中的空格



下面通过一些实例介绍部分字符串常用函数的使用方法。

1) 函数 size()

返回字符串的类型和长度。

```
>> university = '上海工程技术大学'
>> size(university)
ans =
         1         8
% 返回这个字符串是一个一维数组,字符串长度是 8
```

2) 函数 strcat()

连接多个字符串数组,可使用字符串连接函数 `strcat` 或者 `strvcat`,或者连接运算符 `[]`,这里重点讲解 `strcat` 函数的用法。

`strcat` 函数为水平连接字符串,调用格式为

```
T = strcat (s1,s2,s3,...)
```

`s1,s2,s3` 等经过水平连接得到的是一个更长的字符串。如:

```
>> s1 = 'hello'
s2 = ' world!'
T = strcat(s1,s2)
T =
hello world!
```

特别注意: 当 `s1,s2,s3` 等为字符串数组时,所有字符串数组的行数必须相等,每行相连组成新的字符串数组;如果 `s1,s2,s3` 中包含单个字符串,则将单个字符串扩展成与其他字符串数组相同行数的字符串数组后进行连接。

3) 函数 strcmp()

调用格式:

```
T = strcmp(S1,S2)
```

比较字符串 `S1,S2` 是否相同,如果相同,则返回值 `T=1`;反之,`T=0`。如:

```
>> a = 'hello'
>> b = 'hello!'
>> T = strcmp(a,b)
T =
0
```

4) 函数 findstr()

调用格式:

```
T = findstr (S1,S2)
```

自动比较输入的字符串 `S1,S2` 的长度,然后在长字符串中查找短的字符串,返回其所在位置,如:

```
>> s = 'How much wood would a woodchuck chuck?';
T = findstr(s,'a')
```



```
T =
    21
>> s = 'How much wood would a woodchuck chuck?';
    T = findstr(s, 'H')
T =
    1
```

1.1.3 元胞数组

MATLAB 设计的这种元胞数组 (Cell array) 允许存放并操作各种不同类型和不同大小的数据, 但存放和操作复杂字符串是元胞数组的一个主要用途。元胞是元胞数组的基本组成部分, 元胞数组是用下标来区分的, 单元胞数组由元胞和元胞内容两个部分组成, 元胞和元胞内容是两个不同范畴的东西。用圆括号表示元胞元素, 用花括号表示元胞数组的内容。以二维元胞数组 A 为例, A(2,3) 是指元胞数组 A 中的第 2 行第 3 列元胞元素; 而 A{2,3} 是指元胞数组 A 中的第 2 行第 3 列所允许存或者取的内容。如:

```
>> A = cell(2)
>> A(1,1) = {'hello'}
>> A(1,2) = {'world'}
>> A(2,1) = {1 + 2i}
>> A(2,2) = {9}
A =
    'hello'          'world'
    [1.0000 + 2.0000i]    [    9]
```

1.1.4 构架数组

与元胞数组一样, 构架数组 (Structure array) 也能在一个数组里存放各类数据, 使用指针方式传递数值。构架数组由结构变量名和属性名组成, 用指针操作符“.”连接结构变量名和属性名。从一定意义上讲, 构架数组组织数据的能力比元胞数组更强、更富于变化。

构架 (Structure) 是构架数组的基本组成部分, 数组中的每个构架是平等的, 它们以下标区分。构架必须在划分“域”后才能使用。数据不能直接存放于构架, 而只能存放在域中。构架的域可以存放任何类型、任何大小的数组 (如任意维数值数组、字符串数组、符号对象等)。而且, 不同构架的同名域中存放的内容可以不同。

(1) 直接创建一个构架数组。例如创建保存电子信息系专业方向和学生人数的构架数组:

```
>> information.direction = 'Electronic and information engineering';
>> information.studentnumber = 60
information =
    direction: 'Electronic and information engineering'
    studentnumber: 60
```

(2) 利用 struct 函数创建一个构架数组。struct 函数的调用格式为

```
S = struct ('field1', VALUES1, 'field2', VALUES2, ...)
```

其中, field1、field2 等为字段名; VALUES1、VALUES2 等为对应的字段数据, 必须是



同样大小的单元数组或者标量；S 为生成的结构数组。

例如，创建保存电子信息系专业方向和学生人数的构架数组：

```
S = struct('direction',{'Electronic and information engineering'}, 'studentnumber',{60})
S =
    direction: 'Electronic and information engineering'
 studentnumber: 60
```

(3) 构架数组的扩充。上面创建的 information 是一个 1×1 的构架数组，如果要再添加一个专业信息，就将构架 information 扩展为 1×2 的构架数组：

```
information(2).direction = 'NGB engineering';
information(2).studentnumber = 29
information =
    1 × 2 struct array with fields:
    direction
 studentnumber
```

对于多维构架数组，当输入构架数组名字时，MATLAB 不会显示单个字段的内容，而只显示结构数组包含的各类信息概略。这些信息也可以通过 fieldnames 函数获取。

1.2 脚本和函数

在 MATLAB 中，如果遇到一段代码要重复多次使用，那么可以将这段代码写成 M 文件的形式并保存，这使得程序的可读性和可重用性大大提高。在 MATLAB 中，M 文件分为脚本式和函数式两大类。

1. M 脚本文件

对于一些比较简单的问题，从指令窗口中直接输入指令进行计算是十分轻松简单的事。但随指令数的增加或随控制流复杂度的增加，以及重复计算要求的提出，直接从指令窗进行计算就显得繁琐。此时用脚本文件最为适宜。“脚本”本身反映这样一个事实：MATLAB 只是按文件所写的指令执行。这种文件构成比较简单，其特点是：

(1) 它只是一串按用户意图排列而成的 MATLAB 指令集合。

(2) 脚本文件运行后，所产生的全部变量都保存在 MATLAB 基本工作空间中。只要用户不使用 clear 命令清除，且 MATLAB 软件没有被关闭，这些变量始终保存在工作空间中。

【例 1.1】 脚本式 M 文件实例。建立一个 M 文件，求 $1 \sim 100$ 的自然数的和。

在 M 文件编辑器中，输入如下代码：

```
sumnumber = 0;    % 初始化变量
% FOR 循环实现从 1 加到 100
for i = 1:100;
    sumnumber = sumnumber + i;
end
```

保存文件名为 math.m，在命令窗口里输入 math 将会执行该命令文件。



注意：用户创建的 M 文件名要避免与 MATLAB 的内置函数和工具箱中的函数重名。M 脚本文件中的变量都是全局变量,任何其他脚本文件和函数都可以访问这些变量。

2. M 函数文件

与脚本文件不同,函数文件犹如一个“黑箱”,从外界只能看到传给它的输入量和送出来的计算结果,而内部运作藏而未见。它的特点是:

(1) 从形式上看,与脚本文件不同,函数文件的第一行总是以“function”引导的“函数声明行”开始,该行还罗列出函数与外界联系的全部“标称”输入输出宗量,并且对标称数目没有限制。

(2) MATLAB 允许使用比“标称数目”少的输入输出宗量,实现对函数的调用,但不能用多于 M 函数文件规定的输入和输出宗量数目。如果输入和输出宗量数目多于 M 函数文件中 function 语句初始所规定的数目,则调用时自动返回一个错误。

(3) 函数空间随具体 M 函数文件的被调用而产生,随调用结束而删除。函数空间是相对基本空间独立的、临时的。在 MATLAB 整个运行期间,可以产生任意多个临时函数空间。

(4) 函数名和文件名必须相同。

M 函数文件的一般结构如下所述。

(1) 函数声明行:位于函数文件首行,以 MATLAB 关键字 function 开头,函数名以及函数的输入输出宗量都在这一行被定义。

(2) 接着就是一些注释行,主要是对函数功能的描述,以及输入和输出宗量的含义、调用格式的说明,当然这些不是必需的。

(3) 函数主体,这个部分是实现该函数功能的 MATLAB 指令的集合。

【例 1.2】 函数式 M 文件实例。建立一个 M 函数文件,求一个数组前 100 个数值的和。

```
function sumnumber = calculatesum(x)
% calculatesum: 本程序求输入数组的前 100 个数的和
% x: 待求和的输入数组
% sumnumber: 保存求和结果
sumnumber = 0;      % 初始化变量
for i = 1:100;
    sumnumber = sumnumber + x(i);
end
```

3. 私有函数

私有函数是具有限制性访问权限的函数,对应的 M 文件需要保存在名为“private”的文件夹下。私有函数的定义与普通函数的定义一样,但私有函数只能被 private 目录下的直接父目录的脚本式 M 文件或 M 文件主函数调用。因此,在其他目录中可以存在同名的私有函数。这样的好处在于,用户可以创建自己定义的函数,同时可以保留其他目录中的原函数。由于 MATLAB 检查私有函数先于与之同名的标准函数,并且在检查到私有函数时,不会再检查其他的同名函数,因此,在私有目录的父目录中执行私有函数,不会出现错误。



4. 嵌套函数

在一个函数内部,可以定义一个或多个函数,这种定义在其他函数内部被称为嵌套函数,嵌套可以多层发生,即一个函数内部可以嵌套多个函数,在嵌套函数内部又可以再嵌套其他函数。

嵌套函数的语法格式如下:

```
Function a = A(m1,m2)
...
    Function b = B(m3)
    ...
    End
End
```

一般函数代码结尾是不需要专门标明“End”的,但是使用嵌套函数时,无论嵌套函数还是嵌套函数的父函数,在函数结束处都要明确标出“End”。而且,嵌套函数的互相调用和嵌套的层次密切相关。其特点为:

- (1) 外层的函数可以调用其内一层直接嵌套的函数,而不能调用更深层的嵌套函数。
- (2) 嵌套函数可以调用与自己具有相同父函数的其他同层嵌套函数。
- (3) 嵌套函数也可以调用其父函数或与父函数具有相同父函数的其他嵌套函数,但不能调用与其父函数具有相同父函数的其他嵌套函数内深层嵌套的函数。

1.3 变量和语句

MATLAB 能识别一般常用的加、减、乘、除和幂等运算,对于简单的计算,在命令窗口中输入表达式后,按 Enter 键即可完成。MATLAB 会将运算结果存入默认变量 ans 中,并显示其结果(如果表达式后加上分号“;”,则不会显示),当然用户也可设定自己的变量。与 C 语言不同的是,MATLAB 中的变量是不需要事先定义的。MATLAB 的主要功能虽然是数值运算,但是它也是一个完整的程序语言,有各种语句格式和语法规则。下面就进行详细介绍。

1.3.1 变量

程序中,为了方便操作内存中的值,需要给内存中的值设定一个标签,这个标签称为变量。变量是 MATLAB 的基本元素之一,与其他常规程序设计语言不同的是,MATLAB 语言不要求对所使用的变量进行事先说明,而且它也不需要指定变量的类型,系统会根据该变量被赋予的值或是对变量所进行的操作来自动确定变量的类型。在赋值过程中,如果变量已经存在,MATLAB 会用新的变量类型代替旧的变量类型。

变量命名的规则如下:

- (1) 变量名必须以字母开头,变量名中可以包含字母、数字和下划线,不得包含空格和标点。如,wav_01 是合法的,而 1_wav 则为非法变量。
- (2) 变量名长度不超过 63 个字符。