

浏览历年数学建模竞赛赛题可以发现,这些赛题总是附有大量的数据,在用 MATLAB 进行编程计算时,不可避免地要涉及数据的导入导出问题。通常情况下,数据文件为记事本文件(TXT 文件)和 Excel 文件,本章通过具体案例介绍 MATLAB 与这两种类型的文件之间的数据交换。

3.1 利用数据导入向导导入数据

MATLAB 中提供了基于界面操作的数据导入向导,可以很方便地导入外部数据。不同版本下,数据导入向导的界面及使用方法不尽相同,这里以 MATLAB R2020a(即 MATLAB 9.8)版本为例,介绍数据导入向导的使用方法。

3.1.1 利用数据导入向导导入 TXT 文件

数据导入向导适合导入排列整齐、文字说明在数据前面的数据文件,见下例。

【例 3.1-1】 TXT 文件“2006 平均气温.txt”中包含以下内容:

```
1-15 主要城市平均气温 (2006 年)
单位:摄氏度
城市 1月 2月 3月 4月 5月 6月 7月 8月 9月 10月 11月 12月
北京 -1.9 -0.9 8.0 13.5 20.4 25.9 25.9 26.4 21.8 16.1 6.7 -1.0
天津 -2.7 -1.4 7.5 13.2 20.3 26.4 25.9 26.4 21.3 16.2
6.5 -1.7
石家庄 -0.9 1.6 10.3 15.1 21.3 27.4 27.0 25.9 21.8 17.8
8.0 0.4
太原 -3.6 -0.4 6.8 14.5 19.1 23.2 25.7 23.1 17.4 13.4
4.4 -2.5
... ..
```

如图 1.1-1 所示,单击 MATLAB 工作界面上的导入数据(Import Data)图标,弹出如图 3.1-1 所示界面。

选择数据文件“2006 平均气温.txt”,然后单击“打开”按钮,弹出如图 3.1-2 所示界面。该界面用来导入 TXT 数据文件,一切操作都是可视化的。当用户选择的数据文件中包含可读取的数据时,数据预览区会显示相应的数据。导入类型中有 5 个选项:表、列向量、数值矩阵、字符串数组和

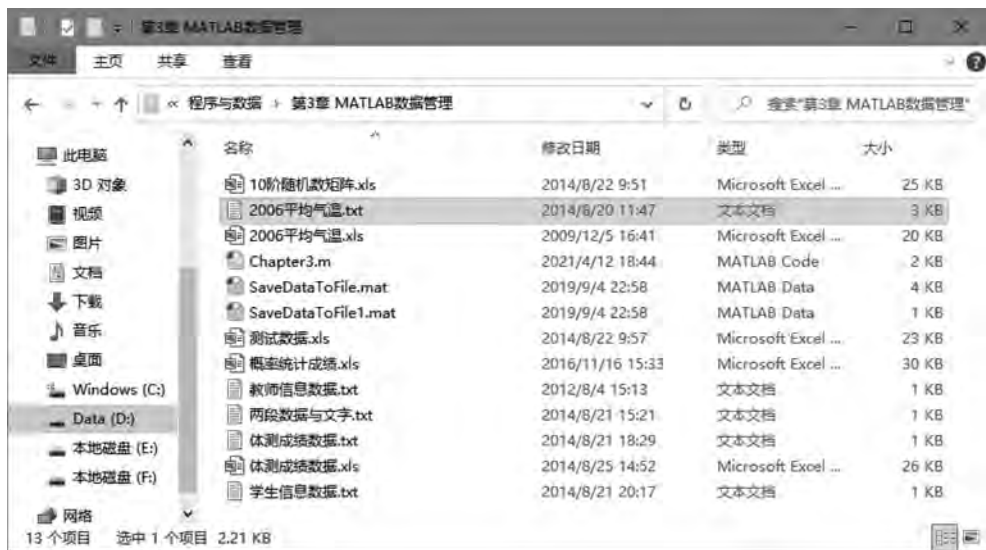


图 3.1-1 数据文件选择界面

元胞数组。在导入范围编辑框中,用户可以指定数据所在的单元格区域,形如“A4:M34”。界面的中上偏右区域是导入规则编辑区,通过单击规则条目后面的“+”或“-”,可增加或删除一个规则条目。图 3.1-2 中显示的默认规则条目为“用非数 NaN 替换无法导入的元胞”,表示在导入数据时将无法导入的数据用 NaN 代替。用户可通过“替换”和“无法导入的元胞”后面的下拉菜单查看可用的规则条目,并通过设置适当的规则条目来替换或过滤某些行和列。界面的左上角区域是列分隔符设置区,默认情况下,数据导入向导会自动识别数据文件中的列分隔符,不需要用户设定,当然用户也可以通过列分隔符设置区的下拉菜单和单选框自定义列分隔符,若改变列分隔符,导入结果可能会随之改变。当用户设置好输出类型、导入范围和导入规则后,只需单击导入按钮即可将数据导入 MATLAB 工作空间。如果选择导入列向量,则每列数据对应一个变量,默认变量名为 $\text{VarName}_i, i(i=1,2,\dots)$ 为列序号;如果选择导入表、数值矩阵、字符串数组或元胞数组,则将全部数据导入为一个变量,默认变量名为 untitled。用户可在变量名编辑区输入自定义变量名。

在用户通过图 3.1-2 所示界面导入数据的同时, MATLAB 执行一系列读取数据的代码,这些代码也是可视化的。单击“操作类型”下拉菜单,可以看到 4 个选项:导入数据、生成实时脚本、生成脚本和生成函数。其中“导入数据”选项用来导入数据,其功能相当于单击导入按钮;“生成脚本”选项用来生成与界面操作相关的脚本文件;“生成函数”选项用来生成与界面操作相关的函数文件,这些文件中均包含了与界面操作相对应的 MATLAB 代码,可作为标准函数使用。

3.1.2 利用数据导入向导导入 Excel 文件

【例 3.1-2】 把 Excel 文件“概率统计成绩.xls”中的数据导入 MATLAB 工作空间。文件中的数据格式如图 3.1-3 所示。

单击 MATLAB 工作界面上的导入数据(Import Data)图标 ,弹出如图 3.1-1 所示界面。选择数据文件“概率统计成绩.xls”,然后单击“打开”按钮,弹出如图 3.1-4 所示界面,该界

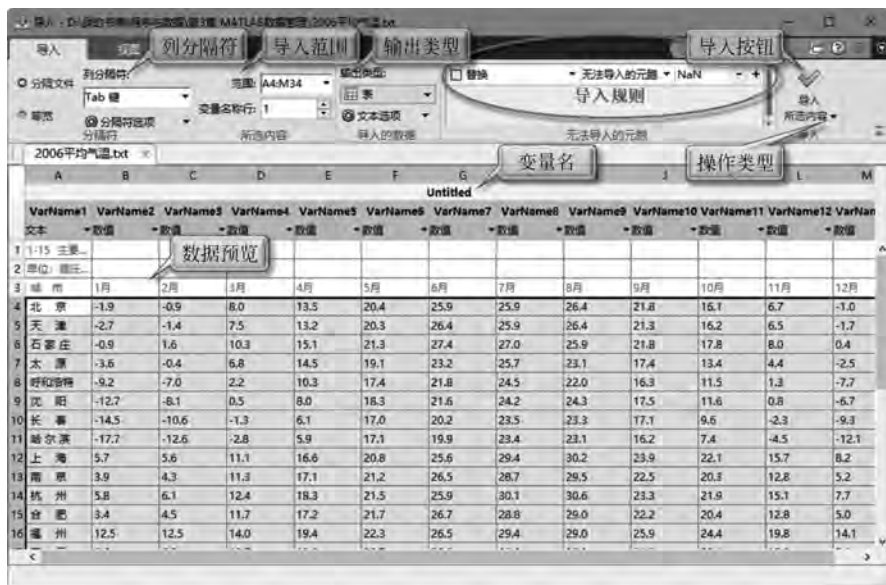


图 3.1-2 数据导入向导界面(TXT)

序号	班名	学号	姓名	平时成绩	期末成绩	总成绩	备注
1	60101	6010101	陈亮	21	42	63	
2	60101	6010102	李旭	25	48	73	
3	60101	6010103	刘鹏飞	0	0	0	缺考
4	60101	6010104	任时迁	27	55	82	
5	60101	6010105	苏宏宇	26	54	80	
6	60101	6010106	王海涛	21	49	70	
7	60101	6010107	王洋	27	61	88	
8	60101	6010108	徐靖鑫	26	54	80	
9	60101	6010109	阎世杰	30	62	92	
10	60101	6010110	姚前村	22	56	78	
11	60101	6010111	张金松	29	66	95	
12	60101	6010112	朱星宇	28	54	82	
13	60101	6010113	韩宏浩	25	50	75	
14	60101	6010114	刘非	23	48	71	
15	60101	6010115	苗锡红	23	47	70	
16	60101	6010116	宋佳艺	24	56	80	
17	60101	6010117	王峰琦	25	53	78	

图 3.1-3 Excel 数据表格



图 3.1-4 数据导入向导界面(Excel)

面用来导入 Excel 电子表格,一切操作也都是可视化的。当用户的 Excel 文件中有多个非空工作表时,可通过单击工作表标签来选择要导入的工作表,数据预览区会显示相应的数据。导入类型、导入范围、导入规则、导入按钮和操作类型的使用说明同上。

3.2 调用函数读取外部数据

MATLAB 中用于读取 TXT 和 Excel 数据文件的常用函数如表 3.2-1 所示。

表 3.2-1 MATLAB 中读取数据文件的常用函数

函 数 名	说 明	函 数 名	说 明
load	从文本文件导入数据到 MATLAB 工作空间	fopen	打开文件,获取打开文件的信息
importdata	从文本文件或特殊格式二进制文件(如图片、AVI 视频等)读取数据	fclose	关掉一个或多个打开的文件
readmatrix	从数据文件中读取数据	fgets	读取文件中的下一行,包括换行符
readtable	读取外部数据,创建表格型数组	fgetl	读取文件中的下一行,不包括换行符
sscanf	按指定格式从字符串中读取数据	fscanf	按指定格式从文本文件中读取数据
xlsread	从 Excel 文件中读取数据	textscan	按指定格式从文本文件或字符串中读取数据

3.2.1 调用 readmatrix 函数读取 TXT 数据文件

readmatrix 函数的调用格式如下:

```
A = readmatrix(filename)
A = readmatrix(filename,opts)
A = readmatrix(___,Name,Value)
```

filename 为字符串,用来指定文件名,若文件名中不包含文件完整路径,则数据文件一定得在当前目录或 MATLAB 搜索路径下才行。opts 为文件导入选项,是由 detectImportOptions 函数所创建的 SpreadsheetImportOptions、DelimitedTextImportOptions 或 FixedWidthImportOptions 对象,具体创建方法可参考如下示例:

```
>> opts = detectImportOptions('学生信息数据.txt') % 创建文件导入选项
opts =
    DelimitedTextImportOptions - 属性:
        格式 属性:
            Delimiter: {','}
            Whitespace: '\b\t'
            LineEnding: {'\n' '\r' '\r\n'}
            CommentStyle: {}
        ConsecutiveDelimitersRule: 'split'
        LeadingDelimitersRule: 'keep'
        EmptyLineRule: 'skip'
        Encoding: 'GB18030'
    替换 属性:
        MissingRule: 'fill'
```

```

ImportErrorRule: 'fill'
ExtraColumnsRule: 'addvars'
变量导入 属性:
    VariableNames: {'Name', 'Age', 'Height' ... and 1 more}
    VariableTypes: {'char', 'double', 'double' ... and 1 more}
    SelectedVariableNames: {'Name', 'Age', 'Height' ... and 1 more}
    VariableOptions: Show all 4 VariableOptions
    Access VariableOptions sub-properties using setvaropts/getvaropts
    PreserveVariableNames: false
位置 属性:
    DataLines: [2 Inf]
    VariableNamesLine: 1
    RowNamesColumn: 0
    VariableUnitsLine: 0
    VariableDescriptionsLine: 0
要显示该表的预览,请使用 preview

```

```

>> preview('学生信息数据.txt',opts)    % 预览文件中的数据
ans =

```

5 × 4 table

Name	Age	Height	Weight
{ '和平' }	18	170	65
{ '谢润和' }	16	160	52
{ '韩宇浩' }	15	160	50
{ '金志文' }	20	175	70
{ '邓泽楷' }	15	172	56

Name 和 Value 是可选的、成对出现的参数名和参数值,用来设置读取数据的选项。例如,将 Name 和 Value 分别设置为 'NumHeaderLines' 和 5,则表示表格数据的前五行是标题行。

【例 3.2-1】 TXT 文件“两段数据与文字.txt”中包含以下内容:

```

这是两行头文件,
你可以选择跳过,读取后面的数据.
1.096975, 0.635914, 4.045800, 4.483729, 3.658162, 7.635046
6.278964, 7.719804, 9.328536, 9.727409, 1.920283, 1.388742
6.962663, 0.938200, 5.254044, 5.303442, 8.611398, 4.848533
这里还有两行文字说明和两行数据,
看你还有没有办法!
5.472155, 1.386244, 1.492940
8.142848, 2.435250, 9.292636

```

本文件中有两段数据和两段文字说明,数据间用逗号作为分隔符,这是包含数值和文本混合类型的数据文件,默认情况下,readmatrix 将数据作为数值数组导入,数据间的文本及不等长部分将以 NaN(缺失数据)填充。

```

>> data = readmatrix('两段数据与文字.txt')    % 读取记事本文件中的数据矩阵
data =
    1.0970    0.6359    4.0458    4.4837    3.6582    7.6350
    6.2790    7.7198    9.3285    9.7274    1.9203    1.3887
    6.9627    0.9382    5.2540    5.3034    8.6114    4.8485

```

```

        NaN      NaN      NaN      NaN      NaN      NaN
        NaN      NaN      NaN      NaN      NaN      NaN
    5.4722  1.3862  1.4929      NaN      NaN      NaN
    8.1428  2.4352  9.2926      NaN      NaN      NaN

>> d1 = data(1:3,:) % 提取第一段数据
d1 =
    1.0970    0.6359    4.0458    4.4837    3.6582    7.6350
    6.2790    7.7198    9.3285    9.7274    1.9203    1.3887
    6.9627    0.9382    5.2540    5.3034    8.6114    4.8485
>> d2 = data(6:7,1:3) % 提取第二段数据
d2 =
    5.4722    1.3862    1.4929
    8.1428    2.4352    9.2926

```

3.2.2 调用 textscan 函数读取 TXT 数据文件

调用 textscan 函数读取数据的一般步骤是：调用 fopen 函数按指定格式打开文件，并获取文件标识符，调用 textscan 函数读取文件内容，然后调用 fclose 函数关闭文件。

1. 调用 fopen 函数打开文件

fopen 函数用于打开一个文件，也可用于获取已打开文件的信息。默认情况下，fopen 函数以读写二进制文件方式打开文件。fopen 函数用于打开文本文件的调用格式如下：

```
fid = fopen(filename, permission)
```

上述代码以指定方式打开一个文件。参数 filename 是一个字符串，用来指定文件名，可以包含完整路径，也可以只包含部分路径（在 MATLAB 搜索路径下）。permission 也是一个字符串，用来指定打开文件的方式，可用的方式如表 3.2-2 所示。

表 3.2-2 打开文本文件的方式列表

允许的打开方式 (permission)	说 明
'rt'	以只读方式打开文件，这是默认情况
'wt'	以写入方式打开文件，若文件不存在，则创建新文件并打开，原文件内容会被清除
'at'	以写入方式打开文件或创建新文件，在原文件内容后续写新内容
'r+t'	以同时支持读、写方式打开文件
'w+t'	以同时支持读、写方式打开文件或创建新文件，原文件内容会被清除
'a+t'	以同时支持读、写方式打开文件或创建新文件，在原文件内容后续写新内容
'At'	以续写方式打开文件或创建新文件，写入过程中不自动刷新文件内容，适合对磁带介质文件的操作
'Wt'	以写入方式打开文件或创建新文件，原文件内容会被清除，写入过程中不自动刷新文件内容，适合对磁带介质文件的操作

输出值 fid 是文件标识符(file identifier)，它作为其他低级 I/O(Input/Output)函数的输入参数。文件打开成功时，fid 为正整数，不成功时 fid 为 -1。

2. 调用 fclose 函数关闭文件

对 fopen 函数打开的文件操作结束后,应将其关闭,否则会影响其他操作。fclose 函数用于关闭文件,其调用格式如下:

```
status = fclose(fid)
status = fclose('all')
```

第 1 种调用用来关闭文件标识符 fid 指定的文件,第 2 种调用用来关闭所有被打开的文件。若操作成功,返回 status 为 0,否则为 -1。

3. 调用 textscan 函数读取数据

textscan 函数用来以指定格式从文本文件或字符串中读取数据。它提供了丰富的数据转换格式,能从文件的任何地方开始读取数据,能更好地处理大型数据。其调用格式如下。

(1) C = textscan(fid, 'format')。

输入参数 fid 为 fopen 函数返回的文件标识符。format 用来指定数据转换格式,它是一个字符串,包含一个或多个转换指示符。返回值 C 是一个元胞数组,format 中包含的转换指示符的个数决定了 C 中元胞的数目。

转换指示符是由“%”号引导的特殊字符串,基本的转换指示符如表 3.2-3 所示。用户还可以在“%”号和指示符之间插入数字,用来指定数据的位数或字符的长度。对于浮点数(%n, %f, %f32, %f64),还可以指定小数点右边的位数。textscan 函数支持的字段宽度设置如表 3.2-4 所示。

表 3.2-3 textscan 函数支持的基本转换指示符

字段类型	指示符	说明
有符号整型	%d	32 位
	%d8	8 位
	%d16	16 位
	%d32	32 位
	%d64	64 位
无符号整型	%u	32 位
	%u8	8 位
	%u16	16 位
	%u32	32 位
	%u64	64 位
浮点数	%f	64 位(双精度)
	%f32	32 位(单精度)
	%f64	64 位(双精度)
	%n	64 位(双精度)
字符串	%s	字符串
	%q	字符串,可能是由双引号括起来的字符串
	%c	任何单个字符,可以是分隔符

续表

字段类型	指示符	说明
模式匹配字符串	%[...]	读取和方括号中字符相匹配的字符,直到首次遇到不匹配的字符或空格时停止。若要包括“]”自身,可用“%[]...”。例如,%[mus] 会把 'summer ' 读作 'summ'
	%[^...]	读取和方括号中字符不匹配的字符,直到首次遇到匹配的字符或空格时停止。若要排除“]”自身,可用“%[]...”。例如,%[^xrg] 会把 'summer ' 读作 'summe'

表 3.2-4 textscan 函数支持的字段宽度设置

指示符		说明
%Nc		读取 N 个字符,包括分隔符。例如,%9c 会把 'Let's Go!' 读作 'Let's Go!'
%Ns	%Nn	读取 N 个字符或数字(小数点也算一个数字),直到遇到第 1 个分隔符,不论是什么分隔符。 例如,%5f32 会把 '473.238' 读作 473.2
%Nq	%Nd...	
%N[...]	%Nu...	
%N[^...]	%Nf...	
%N. Dn %N. Df...		读取 N 个数字(小数点也算一个数字),直到遇到第 1 个分隔符,不论是什么分隔符。返回的数字有 D 位小数。例如,%7.2f 会把 '473.238' 读作 473.23

对于每一个数值型转换指示符(例如%f),textscan 函数返回一个 $K \times 1$ 的数值型向量作为 C 的一个元胞,这里的 K 为读取指定文件时该转换指示符被使用的次数。对于每一个字符型转换指示符(例如%s),textscan 函数返回一个 $K \times 1$ 的字符元胞数组作为 C 的一个元胞。%Nc 会返回一个 $K \times N$ 的字符数组作为 C 的一个元胞。

通常情况下,textscan 函数按照用户设定的转换指示符来读取文件中的相应类型字段的全部内容。用户还可设置需要跳过的字段和部分字段,用来忽略某些类型的字段或字段的一部分,可用的设置如表 3.2-5 所示。

表 3.2-5 跳过某些字段或部分字段的转换指示符

指示符	说明
% * ...	跳过某些字段,不生成这些字段的输出。 例如,'%s % * s %s %s % * s % * s %s' 把字符串 'Blackbird singing in the dead of night' 转换成具有 4 个元胞的输出,元胞中字符串分别为 'Blackbird','in','the','night'
% * n...	忽略字段中的前 n 个字符,n 为整数,其值小于或等于字段中字符个数。例如,% * 4s 把 'summer' 读作 'er'
指定字符	忽略字段中指定的字符。例如,Level%u8 把 'Level1' 读作 1,或者%u8Step 把 '2Step' 读作 2

(2) $C = \text{textscan}(\text{fid}, \text{'format'}, N)$ 。

重复使用 N 次由 format 指定的转换指示符,从文件中读取数据。fid 和 format 的说明同上。N 为整数,当 N 为正整数时,表示重复次数,当 N 为 -1 时,表示读取全部文件。

(3) $C = \text{textscan}(\text{fid}, \text{'format'}, \text{param}, \text{value}, \dots)$ 。

利用可选的成对出现的参数名与参数值来控制读取文件的方式。fid 和 format 的说明同上。字符串 param 用来指定参数名,value 用来指定参数的取值。可用的参数名与参数值如

表 3.2-6 所示。

表 3.2-6 textscan 函数支持的参数名与参数值列表

参 数 名	参数值(设定值)	默 认 值
BufSize	最大字符串长度,单位是 byte	4095
CollectOutput	取值为整数,若不等于 0(即为真),则将具有相同数据类型的连续元胞连接成一个数组	0(假)
CommentStyle	忽略文本内容的标识符号,可以是单个字符串(如 '%'),也可以是由两个字符串构成的元胞数组(如 {'/*', '*/'})。若为单个字符串,则该字符串后面的在同一行上的内容会被忽略。若为元胞数组,则两个字符串中间的内容会被忽略	无
Delimiter	分隔符	空格
EmptyValue	空缺数字字段的填补值	NaN
EndOfLine	行结尾符号	从文件中识别: \n, \r, or \r\n
ExpChars	指数标记字符	'eEdD'
HeaderLines	跳过的行数(包括剩余的当前行)	0
MultipleDelimsAsOne	取值为整数,若不等于 0(即为真),则将连在一起的分隔符作为一个单一的分隔符。只有设定了 delimiter 选项,它才是有效的	0(假)
ReturnOnError	取值为整数,用来确定读取或转换失败时的行为。若非 0(即为真),则直接退出,不返回错误信息,输出读取的字段。若为 0(即为假),则退出并返回错误信息,此时没有输出	1(真)
TreatAsEmpty	在数据文件中被作为空值的字符串,可以是单个字符串或字符串元胞数组。只能用于数字字段	无
Whitespace	作为空格的字符	'\b\t'

(4) $C = \text{textscan}(\text{fid}, 'format', N, \text{param}, \text{value}, \dots)$ 。

结合了第 2 种和第 3 种调用格式,可以同时设定读取格式的重复使用次数和某些特定的参数。

(5) $C = \text{textscan}(\text{str}, \dots)$ 。

从字符串 str 中读取数据。第 1 个输入参数 str 是普通字符串,不再是文件标识符,除此之外,其余参数的用法与读取文本文件时相同。

(6) $[C, \text{position}] = \text{textscan}(\dots)$ 。

读取文件或字符串中的数据 C,并返回扫描到的最后位置 position。若读取的是文件,position 就是读取结束后文件指针的当前位置,等于 ftell(fid)的返回值。若读取的是字符串,position 就是已经扫描过的字符的个数。

【例 3.2-2】 TXT 文件“体测成绩数据.txt”中包含以下内容:

序号	身高	体重	肺活量	肥胖程度
1	168.4	74.2	4686	肥胖
2	162.3	50.3	3275	较低体重

3	177.1	63.8	3867	正常体重
4	169.8	48.7	3327	营养不良
5	174	71.5	2805	超重
6	161.9	52.1	3625	较低体重
7	178.3	53.8	3678	营养不良
8	159.9	55.2	3007	正常体重
9	162.1	57.7	2800	正常体重
10	171.2	72.2	1609	肥胖

调用 `textscan` 函数读取该文件的 MATLAB 命令如下：

```
>> fid = fopen('体测成绩数据.txt'); % 打开数据文件,返回文件标识符
% 按指定格式读取数据,并将相同类型的连续列进行合并
>> data1 = textscan(fid, '%d %f %f %d %s', 'HeaderLines', 1, 'CollectOutput', 1)
data1 =
    [10x1 int32]    [10x2 double]    [10x1 int32]    {10x1 cell}

>> data1{2} % 查看读到的数据
ans =
    168.4000    74.2000
    162.3000    50.3000
    177.1000    63.8000
    169.8000    48.7000
    174.0000    71.5000
    161.9000    52.1000
    178.3000    53.8000
    159.9000    55.2000
    162.1000    57.7000
    171.2000    72.2000

>> fclose(fid); % 关闭文件
```

上述命令中的数据转换格式为 `'%d %f %f %d %s'`, `textscan` 函数将第 1 和第 4 列数据读为 32 位整型数据,将第 2 和第 3 列数据读为 64 位双精度浮点型数据,将第 5 列读为字符型数据。读到的数据保存在元胞数组 `data1` 中。通过设置 `CollectOutput` 参数值为 1,可将读取到的具有相同数据类型的连续列(第 2 和第 3 列)合并输出。

【例 3.2-3】 TXT 文件“教师信息数据.txt”中包含以下内容：

```
Name: xiezh Age: 18 Height: 170 Weight: 65 kg
Name: molih Age: 16 Height: 160 Weight: 52 kg
Name: liaoj Age: 15 Height: 160 Weight: 50 kg
Name: lijun Age: 20 Height: 175 Weight: 70 kg
Name: xiagk Age: 15 Height: 172 Weight: 56 kg
```

调用 `textscan` 函数可以非常灵活地读取这种文字与数据交替出现的数据文件,参见下面的 MATLAB 命令。

```
>> fid = fopen('教师信息数据.txt', 'r'); % 以只读方式打开文件:教师信息数据.txt
% 调用 textscan 函数以指定格式从文件中读取数据,用空格(' ')作分隔符,并将相同类型的列合并
>> A = textscan(fid, '%s %s %s %s %d %s %d %s %d %s', ...
    'delimiter', ' ', 'CollectOutput', 1)
A =
```

```

{5x1 cell}    [5x3 int32]

>> A{1,1}                                % 查看 A 的第 1 行、第 1 列的元胞中的数据
ans =
    'xiezh'
    'moli'
    'liao'
    'lijun'
    'xiagk'

>> A{1,2}                                % 查看 A 的第 1 行、第 2 列的元胞中的数据
ans =
    18    170    65
    16    160    52
    15    160    50
    20    175    70
    15    172    56

>> fclose(fid);                          % 关闭文件

```

3.2.3 调用 fgetl 和 sscanf 函数读取 TXT 数据文件

fgetl 函数用来读取文件中的一行内容,并删除换行符。sscanf 函数用来从字符串中读取格式化数据。将 fgetl 和 sscanf 函数配合使用,可从 TXT 数据文件中逐行读取数据。

【例 3.2-4】 调用 fgetl 和 sscanf 函数逐行读取 TXT 文件“教师信息数据.txt”中的数值型数据。

```

>> fid = fopen('教师信息数据.txt','r'); % 以只读方式打开文件:教师信息数据.txt
>> k = 1;
>> while ~feof(fid)                      % 用 while 循环逐行读取数据
    str = fgetl(fid);                    % 读取一行内容
    % 忽略姓名字符串,读取该行中的整型数据(年龄、身高和体重)
    data(k,:) = sscanf(str,'Name: % * s Age: % d Height: % d Weight: % d kg');
    k = k + 1;
end
>> fclose(fid);                          % 关闭文件
>> data                                  % 查看读取的数据

data =
    18    170    65
    16    160    52
    15    160    50
    20    175    70
    15    172    56

```

【说明】 feof 函数用来判断是否到达文件末尾。调用格式如下:

```
eofstat = feof(fid)
```

输入参数 fid 为文件标识符。当到达文件末尾时,输出 eofstat = 1,否则 eofstat 为 0。

3.2.4 调用 xlsread 函数读取 Excel 数据文件

xlsread 函数用来读取 Excel 工作表中的数据。原理是这样的,当用户系统安装有 Excel 时,MATLAB 创建 Excel 服务器,通过服务器接口读取数据。当用户系统没有安装 Excel 或 MATLAB 不能访问 COM 服务器时,MATLAB 利用基本模式(basic mode)读取数据,即把 Excel 文件作为二进制映像文件读取进来,然后读取其中的数据。xlsread 函数的调用格式如下。

(1) num = xlsread(filename)。

读取由 filename 指定的 Excel 文件中第 1 个工作表中的数据,返回一个双精度矩阵 num。输入参数 filename 是由单引号括起来的字符串。

当 Excel 工作表的顶部或底部有一个或多个非数字行(如图 3.1-3 中的第 1 行),左边或右边有一个或多个非数字列(如图 3.1-3 中的第 H 列)时,在输出中不包括这些行和列。例如,xlsread 会忽略一个电子表格顶部的文字说明。

如图 3.1-3 中的第 D 列,它是一个处于内部的列。对于内部的行或列,即使它有部分非数字单元格,甚至全部都是非数字单元格,xlsread 也不会忽略这样的行或列。在读取的矩阵 num 中,非数字单元格位置用 NaN 代替。

(2) num = xlsread(filename, -1)。

在 Excel 界面中打开数据文件,允许用户交互式选取要读取的工作表以及工作表中需要导入的数据区域。这种调用会弹出一个提示界面,提示用户选择 Excel 工作表中的数据区域。在某个工作表上单击并拖动鼠标即可选择数据区域,然后单击提示界面上的“确定”按钮即可导入所选区域的数据。

(3) num = xlsread(filename, sheet)。

用参数 sheet 指定读取的工作表。sheet 可以是单引号括起来的字符串,也可以是正整数。当是字符串时,用来指定工作表的名字;当是正整数时,用来指定工作表的序号。

(4) num = xlsread(filename, range)。

用参数 range 指定读取的单元格区域。range 是字符串,为了区分 sheet 和 range 参数,range 参数必须包含冒号,形如 'C1:C2' 的表示区域的字符串。若 range 参数中没有冒号,xlsread 就会把它作为工作表的名字或序号,这有可能导致错误。

(5) num = xlsread(filename, sheet, range)。

同时指定工作表和工作表区域。此时 range 参数可以是 Excel 文件中定义的区域的名字。

【例 3.2-5】 用 xlsread 函数读取文件“概率统计成绩.xls”第 1 个工作表中区域 A2:H4 的数据。命令及结果如下:

```
% 读取文件"概率统计成绩.xls"第 1 个工作表中单元格 A2:H4 中的数据
% 第一种方式:
>> num1 = xlsread('概率统计成绩.xls', 'A2:H4')           % 返回读取的数据矩阵 num1
num1 =
     1    60101    6010101    NaN     21     42     63
     2    60101    6010102    NaN     25     48     73
     3    60101    6010103    NaN      0      0      0
```

% 第二种方式:

```
>> num2 = xlsread('概率统计成绩.xls', 1, 'A2:H4') % 返回读取的数据矩阵 num2
num2 =
```

1	60101	6010101	NaN	21	42	63
2	60101	6010102	NaN	25	48	73
3	60101	6010103	NaN	0	0	0

% 第三种方式:

```
>> num3 = xlsread('概率统计成绩.xls', 'Sheet1', 'A2:H4') % 返回读取的数据矩阵 num3
num3 =
```

1	60101	6010101	NaN	21	42	63
2	60101	6010102	NaN	25	48	73
3	60101	6010103	NaN	0	0	0

(6) num = xlsread(filename, sheet, range, 'basic')。

用基本模式(basic mode)读取数据。当用户系统没有安装 Excel 时,用这种模式导入数据导入功能会受限,range 参数的值会被忽略,可以设定 range 参数的值为空字符串(''),而 sheet 参数必须是字符串,此时读取的是整个工作表中的数据。

(7) [num, txt]= xlsread(filename, ...)。

返回数字矩阵 num 和文本数据 txt。txt 是一个元胞数组,txt 中与数字对应位置的元胞为空字符串('')。例如:

```
>> [num4, text4] = xlsread('概率统计成绩.xls', 'Sheet1', 'A2:H4')
```

```
num4 =
```

1	60101	6010101	NaN	21	42	63
2	60101	6010102	NaN	25	48	73
3	60101	6010103	NaN	0	0	0

```
text4 =
```

'陈亮'	''	''	''	''	''	''
'李旭'	''	''	''	''	''	''
'刘鹏飞'	''	''	''	''	''	'缺考'

3.2.5 调用 readtable 函数创建数据表

1.4.9 节介绍了用 table 函数把工作空间中的变量定义为表格型数组,这里介绍用 readtable 函数读取外部数据文件,创建表格型数组。readtable 函数是 MATLAB R2013b(8.2)版本才有的函数。

【例 3.2-6】 TXT 文件“学生信息数据.txt”中包含以下内容:

Name,	Age,	Height,	Weight
和平,	18,	170,	65
谢润和,	16,	160,	52
韩宇浩,	15,	160,	50
金志文,	20,	175,	70
邓泽楷,	15,	172,	56

用 readtable 函数读取该文件中的数据,创建表格型数组。

```
>> T = readtable('学生信息数据.txt','Delimiter',' ','ReadRowNames',true)
```

```
T =
```

	Age	Height	Weight
和平	18	170	65
谢润和	16	160	52
韩宇浩	15	160	50
金志文	20	175	70
邓泽楷	15	172	56

【例 3.2-7】 用 readtable 函数读取文件“概率统计成绩.xls”中的数据,创建表格型数组。

```
>> T = readtable('概率统计成绩.xls','ReadRowNames',true);
```

```
% 修改表格中变量名称
```

```
>> T.Properties.VariableNames = {'x1','x2','x3','x4','x5','x6','x7'}
```

```
T =
```

	x1	x2	x3	x4	x5	x6	x7
1	60101	6.0101e+06	'陈亮'	21	42	63	' '
2	60101	6.0101e+06	'李旭'	25	48	73	' '
3	60101	6.0101e+06	'刘鹏飞'	0	0	0	'缺考'
4	60101	6.0101e+06	'任时迁'	27	55	82	' '
5	60101	6.0101e+06	'苏宏宇'	26	54	80	' '
6	60101	6.0101e+06	'王海涛'	21	49	70	' '
... ..							

3.3 把数据写入文件

MATLAB 中用于写数据到文件的函数如表 3.3-1 所示。

表 3.3-1 MATLAB 中写数据到文件的常用函数

函 数 名	说 明
save	将工作空间中的变量写入文件
dlmwrite	按指定格式将数据写入文本文件
fprintf	按指定格式把数据写入文件
xlswrite	把数据写入 Excel 文件
writetable	把表格数据写入文件

在 MATLAB 7.x 版本中,选择 File 菜单下的 Save Workspace As 选项,在 MATLAB 8.x 及以后的版本中,单击工作界面上的保存工作区图标,均可将 MATLAB 工作空间里的所有变量导出到 MAT 文件(扩展名为“.mat”)。下面介绍 save 和 xlswrite 函数的用法。

3.3.1 调用 save 函数保存计算结果

save 函数用来将工作空间中的变量写入文件,调用格式如表 3.3-2 所示。

表 3.3-2 save 函数的调用格式

调 用 格 式	说 明
save(filename)	把工作空间中的所有变量数据写入二进制文件(MAT 文件)
save(filename,variables)	只把变量 variables 数据写入二进制文件

续表

调用格式	说 明
save (filename, variables, fmt)	把变量数据写入文件,用字符串 fmt 指定文件格式,fmt 的可用取值如下: '—mat' : 二进制 mat 文件(默认情形) '—ascii' : 8 位精度文本文件 '—ascii','—tabs' : 以制表符为分隔符的 8 位精度文本文件 '—ascii','—double' : 16 位精度文本文件 '—ascii','—double','—tabs' : 以制表符为分隔符的 16 位精度文本文件
save (filename, variables, version)	把变量数据写入二进制文件,用 version 指定文件版本,其可用字符串包括: '—v7.3','—v7','—v6','—v4'
save (filename, variables, '—append')	以续写方式把变量数据写入文件,不会覆盖文件中已有数据
save filename variables	不同于函数调用格式,这是命令行调用格式,也能把变量数据写入文件

表 3.3-2 中的输入参数 filename 为字符串变量,用来指定目标文件的文件名,可以包含路径,若不指定路径,则自动保存到 MATLAB 当前文件夹。variables 为变量名字符串,形如 'Var1','Var2',...

【例 3.3-1】 定义多个变量,调用 save 函数将它们写入二进制文件“SaveDataToFile.mat”。

```
>> x = 1:3; % 定义向量 x
>> y = [1 2 3;4 5 6;7 8 9]; % 定义矩阵 y
>> S = struct('Name',{ '谢中华', 'xzh'}, 'Age',{20,10}); % 定义结构体数组 S
>> ds = dataset('XLSFile','概率统计成绩.xls'); % 定义数据集数组 ds
>> save('SaveDataToFile.mat'); % 以函数调用格式保存所有变量
>> save SaveDataToFile.mat; % 以命令行调用格式保存所有变量
>> save('SaveDataToFile1.mat','y','S'); % 以函数调用格式保存变量 y 和 S
>> save SaveDataToFile1.mat y S % 以命令行调用格式保存变量 y 和 S
```

对于用 save 函数保存的数据文件,可用 load 函数重新加载,例如:

```
>> clear; % 清除工作空间中的所有变量
>> load SaveDataToFile.mat % 重新加载 SaveDataToFile.mat 中保存的变量数据
```

3.3.2 调用 xlswrite 函数把数据写入 Excel 文件

xlswrite 函数用来将数据矩阵 M 写入 Excel 文件,它有以下 7 种调用格式:

```
xlswrite(filename, M)
xlswrite(filename, M, sheet)
xlswrite(filename, M, range)
xlswrite(filename, M, sheet, range)
status = xlswrite(filename, ...)
[status, message] = xlswrite(filename, ...)
xlswrite filename M sheet range
```

其中前 6 种为函数调用格式,最后一种为命令行调用格式。参数 filename 为字符串变量,

用来指定文件名和文件路径。若 filename 指定的文件不存在,则创建一个新文件,文件的扩展名决定了 Excel 文件的格式。若扩展名为“.xls”,则创建一个 Excel 97-2003 下的文件,若扩展名为“.xlsx”““.xlsb”或“.xlsm”,则创建一个 Excel 2007 格式的文件。

M 可以是一个 $m \times n$ 的数值型矩阵或字符型矩阵,也可以是一个 $m \times n$ 的元胞数组,此时每一个元胞只包含一个元素。由于不同版本的 Excel 所能支持的最大行数和列数是不一样的,所以能写入的最大矩阵的大小取决于 Excel 的版本。

Sheet 用来指定工作表,可以是代表工作表序号的正整数,也可以是代表工作表名称的字符串。需要注意的是,Sheet 参数中不能有冒号。若由 Sheet 指定的工作表不存在,则在所有工作表的后面插入一个新的工作表。若 Sheet 为正整数,并且大于工作表的总数,则追加多个空的工作表直到工作表的总数等于 Sheet。这两种情况都会产生一个警告信息,表明增加了新的工作表。

range 用来指定单元格区域。对于 xlswrite 函数的第 3 种调用,range 参数必须包含冒号,形如 'C1:C2' 的表示单元格区域的字符串。当同时指定 sheet 和 range 参数时(如第 4 种调用),range 可以是形如 'A2' 的形式。range 指定的单元格区域的大小应与 **M** 的大小相匹配,若单元格区域超过了 **M** 的大小,则多余的单元格用“#N/A”填充,若单元格区域比 **M** 的大小还要小,则只写入与单元格区域相匹配的部分数据。

输出 status 反映了写操作完成的情况,若成功完成,则 status 等于 1(真);否则,status 等于 0(假)。只有在指定输出参数的情况下,xlswrite 函数才返回 status 的值。

输出 message 中包含了写操作过程中的警告和错误信息,它是一个结构体变量,有两个字段: message 和 identifier。其中 message 是包含警告和错误信息的字符串,identifier 也是字符串,包含了警告和错误信息的标识符。

【例 3.3-2】 生成一个 10×10 的随机数矩阵,将它写入 Excel 文件“10 阶随机数矩阵.xls”的第 2 个工作表的“D6:M15”区域。

```
>> x = rand(10); % 生成一个 10 行 10 列的随机矩阵
% 把矩阵 x 写入文件“10 阶随机数矩阵.xls”的第 2 个工作表中的单元格区域 D6:M15,并返回操作信息
>> [s,t] = xlswrite('10 阶随机数矩阵.xls', x, 2, 'D6:M15')

s =

    1

t =

    message: ''
 identifier: ''
```

上面返回的操作信息变量 $s = 1$,变量 t 的 message 字段为空,说明操作成功,没有出现任何警告,数据被写入文件“10 阶随机数矩阵.xls”的指定位置。

【例 3.3-3】 定义一个元胞数组,将它写入 Excel 文件“测试数据.xls”的自命名工作表的指定区域。

```
>> x = {1,60101,6010101,'陈亮',63,'';2,60101,6010102,'李旭',73,'';3,60101,...
        6010103,'刘鹏飞',0,'缺考'} % 定义一个元胞数组
```



```
x =  
    [1]    [60101]    [6010101]    '陈亮'    [63]    ''  
    [2]    [60101]    [6010102]    '李旭'    [73]    ''  
    [3]    [60101]    [6010103]    '刘鹏飞'    [ 0]    '缺考'  
  
% 把元胞数组 x 写入文件"测试数据.xls"的指定工作表(xiezhhh)中的单元格区域 A3:F5  
>> xlswrite('测试数据.xls', x, 'xiezhhh', 'A3:F5')  
Warning: Added specified worksheet.  
> In xlswrite> activate_sheet at 285  
    In xlswrite> ExecuteWrite at 249  
    In xlswrite at 207
```

上面写入数据的操作返回了一个警告信息：“Warning: Added specified worksheet”，说明文件“测试数据.xls”中指定的工作表 xiezhhh 不存在，此时新插入一个名称为 xiezhhh 的工作表。