

第 7 章 MATLAB 与 Excel 文件的数据交换

郑志勇 (www.ariszheng.com), 谢中华

Excel 是一款非常优秀的通用表格软件, 在学习、工作与科研中大量的数据可能都是以 Excel 表格的方式存储的。如何利用 matlab 强大的数值计算功能处理 Excel 中的数据, 首要解决的问题就是如何将 Excel 中的数据导入到 matlab 中或将 matlab 数值计算的结果转存入 Excel 中? 为此, 本章主要介绍以界面操作方式 (数据导入向导)、函数方式和 exlink 宏方式实现 matlab 与 Excel 的数据交互。

7.1 利用数据导入向导导入 Excel 文件

可以利用数据导入向导把 Excel 文件中的数据导入到 MATLAB 工作空间, 步骤与 6.2.1 节相同。

【例 7.1-1】把 Excel 文件 examp71_1.xls 中的数据导入到 MATLAB 工作空间。examp71_1.xls 中的数据格式如图 7.1-1 所示。

	A	B	C	D	E	F	G	H
1	序号	班名	学号	姓名	平时成绩	期末成绩	总成绩	备注
2	1	60101	6010101	陈亮	0	63	63	
3	2	60101	6010102	李旭	0	73	73	
4	3	60101	6010103	刘鹏飞	0	0	0	缺考
5	4	60101	6010104	任时迁	0	82	82	
6	5	60101	6010105	苏宏宇	0	80	80	
7	6	60101	6010106	王海涛	0	70	70	
8	7	60101	6010107	王洋	0	88	88	
9	8	60101	6010108	徐靖磊	0	80	80	
10	9	60101	6010109	阎世杰	0	92	92	
11	10	60101	6010110	姚前树	0	84	84	
12	11	60101	6010111	张金铭	0	95	95	
13	12	60101	6010112	朱星宇	0	82	82	
14	13	60101	6010113	韩宏洁	0	75	75	
15	14	60101	6010114	刘非	0	71	71	
16	15	60101	6010115	苗艳红	0	70	70	
17	16	60101	6010116	宋佳艺	0	80	80	
18	17	60101	6010117	王峰瑶	0	78	78	

图 7.1-1 Excel 数据表格

可以看出文件 examp71_1.xls 中包含了某两个班的某门课的考试成绩, 有序号、班级名称、学号、姓名、平时成绩、期末成绩、总成绩和备注等数据, 有数字也有文字说明。用数据导入向导会在 MATLAB 工作空间生成两个变量: data 和 textdata, data 为数值矩阵, textdata 为字符串元胞数组, 它们的数据格式如下 (部分数据):

```
>> data      % 查看导入的变量 data

data =

    1    60101    6010101    NaN    0    63    63
    2    60101    6010102    NaN    0    73    73
    3    60101    6010103    NaN    0     0     0
    4    60101    6010104    NaN    0    82    82
    5    60101    6010105    NaN    0    80    80
    ...

>> textdata  % 查看导入的变量 textdata

textdata =
```

'序号'	'班名'	'学号'	'姓名'	'平时成绩'	'期末成绩'	'总成绩'	'备注'
''	''	''	'陈亮'	''	''	''	''
''	''	''	'李旭'	''	''	''	''
''	''	''	'刘鹏飞'	''	''	''	'缺考'
''	''	''	'任时迁'	''	''	''	''
''	''	''	'苏宏宇'	''	''	''	''
...							

7.2 调用函数读写 Excel 文件

7.2.1 调用 xlsinfo 函数获取文件信息

在读取 Excel 目标数据文件前，可以通过 xlsinfo 函数获取该文件的相关信息，为后续操作获得有效信息（例如，文件类型、文件内部结构、相关的软件版本等）。

xlsinfo 函数的调用格式如下：

```
[typ, desc, fmt] = xlsinfo(filename)
```

其中输入参数 filename 为字符串变量，用来指定目标文件的文件名和文件路径。若目标文件在 matlab 搜索路径下，filename 为文件名字符串即可，例如：'abc.xls'；若目标文件不在 matlab 搜索路径下，filename 中还应包含文件的完整路径，例如：'E:\other\基础 matlab 案例书籍\abc.xls'。

输出参数的含义如下：

typ: 目标文件类型

desc: 目标文件内部表名称（sheetname）

fmt: 支持目标文件的软件版本

【例 7.2-1】调用 xlsinfo 函数读取 Excel 文件。以下代码保存在 m 文件 CaseXlsRead.m 中。

```
%code by ariszheng@gmail.com
%2010-6-22
%%
%文件名称“excel.xls”
[typ, desc, fmt] = xlsinfo('excel.xls')
%文件在当前工作目录下，直接输入文件名称即可。
system('taskkill /F /IM EXCEL.EXE');
```

注释：在用 matlab2009a 与 Excel2007 进行数据交互时，每次使用 xls 类函数，都会重新开启一个 Excel 进程，若反复使用 xls 类函数会导致系统中多个 Excel 进程并存，消耗系统资源，导致系统运行速度下降，故作者使用 system('taskkill /F /IM EXCEL.EXE')调用 windows 的 taskkill 函数关闭刚使用的 Excel 进程。

本例输出结果如下：

```
typ =
Microsoft Excel Spreadsheet
%文件类别为 excel 文件
desc =
    'Sheet1'    'Sheet2'    'Sheet3'
%文件中数据表为 'Sheet1'    'Sheet2'    'Sheet3'
fmt =
xlExcel8
%文件版本为 xlExcel8 版本 对应的为 excel 97~2003 版本
成功：已终止进程 "EXCEL.EXE"，其 PID 为 5508。
```

7.2.2 调用 xlsread 函数读取数据

数据导入向导在导入 Excel 文件时调用了 xlsread 函数，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，用来指定目标文件的文件名和文件路径。

当 Excel 工作表的顶部或底部有一个或多个非数字行（如图 7.1-1 中的第 1 行），左边或右边有一个或多个非数字列（如图 7.1-1 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 7.1-1 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 参数必需是包含冒号，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，这就可能导致错误。

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

同时指定工作表和工作表区域。

【例 7.2-2】调用 xlsread 函数读取文件 examp71_1.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 examp71_1.xls 第 1 个工作表中单元格 A2:H4 中的数据
% 第一种方式:
>> num = xlsread('examp71_1.xls','A2:H4')    % 返回读取的数据矩阵 num

num =

         1    60101    6010101    NaN         0         63         63
         2    60101    6010102    NaN         0         73         73
         3    60101    6010103    NaN         0          0          0

% 第二种方式:
>> num = xlsread('examp71_1.xls',1,'A2:H4')    % 返回读取的数据矩阵 num

num =
```

```

1      60101      6010101      NaN      0      63      63
2      60101      6010102      NaN      0      73      73
3      60101      6010103      NaN      0      0      0

% 第三种方式:
>> num = xlsread('examp71_1.xls','Sheet1','A2:H4') % 返回读取的数据矩阵 num

num =

1      60101      6010101      NaN      0      63      63
2      60101      6010102      NaN      0      73      73
3      60101      6010103      NaN      0      0      0

```

可以看出上述命令中用到的三种调用方式的作用是相同的，读取到了相同的数据。

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，此时导入功能受限，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 = xlsread(filename, ..., functionhandle)

在读取电子表格里的数据之前，先调用由函数句柄 functionhandle 指定的函数。它允许用户在读取数据之前对数据进行一些操作，例如在读取之前变换数据类型。

用户可以编写自己的函数，把函数句柄传递给 xlsread 函数。当调用 xlsread 函数时，它从电子表格读取数据，把用户函数作用在这些数据上，然后返回最终结果。xlsread 函数在调用用户函数时，它通过 Excel 服务器 Range 对象的接口访问电子表格的数据，所以用户函数必须包括作为输入输出的接口。

【例 7.2-3】将文件 examp71_1.xls 第 1 个工作表中 A2 至 C3 单元格中的数据加 1，并读取变换后的数据。

首先编写用户函数如下：

```

function DataRange = setplusone1(DataRange)
for k = 1:DataRange.Count
    DataRange.Value{k} = DataRange.Value{k}+1; % 将单元格取值加 1
end

```

用户函数中的输入和输出均为 DataRange，其实它就是一个变量名，用户可以随便指定。当 xlsread 函数调用用户函数时，会通过 DataRange 参数传递 Range 对象的接口，默认情况下传递的是第 1 个工作表对象的 UsedRange 接口，用户函数通过这个接口访问工作表中的数据。

把用户函数句柄作为 xlsread 函数的最后一个输入，可以如下调用：

```

% 读取文件 examp71_1.xls 第 1 个工作表中单元格 A2:C3 中的数据，将数据分别加 1 后返回
>> convertdata = xlsread('examp71_1.xls', '', 'A2:C3', '', @setplusone1)

convertdata =

2      60102      6010102
3      60102      6010103

```

8. [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，如同例 7.1-1 中的 textdata，txt 中与数字对应位置的元胞为空字符串（""）。

9. [num, txt, raw]= xlsread(filename, ...)

num 和 txt 的解释同上，返回的 raw 为未经处理的元胞数组，既包含数字，又包含文本数据。

10 . [num, txt, raw, X] = xlsread(filename, ..., functionhandle)

返回用户函数的额外的输出 **X**。此时的用户函数应有两个输出，第 1 个输出为 **Range** 对象的接口，第 2 个输出为这里的 **X**。

例如，可以将例 7.2-3 中的用户函数增加一个输出，变为如下形式：

```
function [DataRange, customdata] = setplusone2(DataRange)
for k = 1:DataRange.Count
    DataRange.Value{k} = DataRange.Value{k}+1;    % 将单元格取值加 1
    customdata(k) = DataRange.Value{k};    % 把单元格取值赋给变量 customdata
end
% 按照所选区域中单元格行数和列数把向量 customdata 变为矩阵
customdata = reshape(customdata, DataRange.Rows.Count, DataRange.Columns.Count);
```

把函数句柄作为 **xlsread** 函数的最后一个输入，读取文件 **examp71_1.xls** 第 1 个工作表中 **A2** 至 **H2** 单元格中的数据，命令如下：

```
% 读取文件 examp71_1.xls 第 1 个工作表中单元格 A2:H2 中的数据，将读取到的数据分别加 1
% 返回数值矩阵 num，文本矩阵 txt，元胞数组 raw，变换后数值矩阵 X
>> [num, txt, raw, X] = xlsread('examp71_1.xls', '', 'A2:H2', '', @setplusone2)

num =

     2     60102     6010102     NaN         1         64         64

txt =

     {}

raw =

     [2]     [60102]     [6010102]     [NaN]     [1]     [64]     [64]     [NaN]

X =

     2     60102     6010102     NaN         1         64         64     NaN
```

11 . xlsread filename sheet range basic

xlsread 函数的命令行调用方式。此时 **sheet** 参数必须是字符串（例如 **Sheet3**），当 **sheet** 参数中有空格时，必须用单引号括起来（例如 **'Income 2002'**）。

7.2.3 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，其主要调用方式如下：

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

其中输入参数 **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` 指定名称的工作表不存在，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` 参数必需是包含冒号，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。`xlswrite` 函数不能识别已命名区域的名称。`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。

例如：

```
message =
    message: [1x117 char]
    identifier: 'MATLAB:xlswrite:LockedFile'
```

表示目标文件被锁定从而导致无法写入。当目标文件被其他程序占用时，系统会锁定目标文件，就会出现这种情况。

【例 7.2-4】 生成一个 10×10 的随机数矩阵，将它写入 Excel 文件 `excel.xls` 的第 2 个工作表的默认区域。以下代码保存在 `m` 文件 `CaseXlsWrite.m` 中。

```
%code by ariszhen@gmail.com
%2010-6-22
%%
%产生随机数据
X = rand(10,10);
%将随机数据 X 写入 Excel 文件 excel.xls 的第 2 个工作表的默认区域
[status, message] = xlswrite('excel.xls', X, 'sheet2')
system('taskkill /F /IM EXCEL.EXE')
```

结果输出

```
status =

    1 %表示写入成功
message =

    message: ''
    identifier: ''

成功：已终止进程 "EXCEL.EXE"，其 PID 为 368。
```

7.3 Excel-Link 宏

Matlab 提供使其能与 Excel 互动操作的 Excel-link 宏。Excel-link 使得数据在 matlab 与 Excel 之间随意交换，以及在 Excel 下调用 matlab 的函数，它将 matlab 强大的数值计算功能、数据可视化功能与 Excel 的数据 Sheet 功能结合在一起，其功能原理如图 7.3-1 所示。下面就简单介绍 Excel-link 的基本操作。

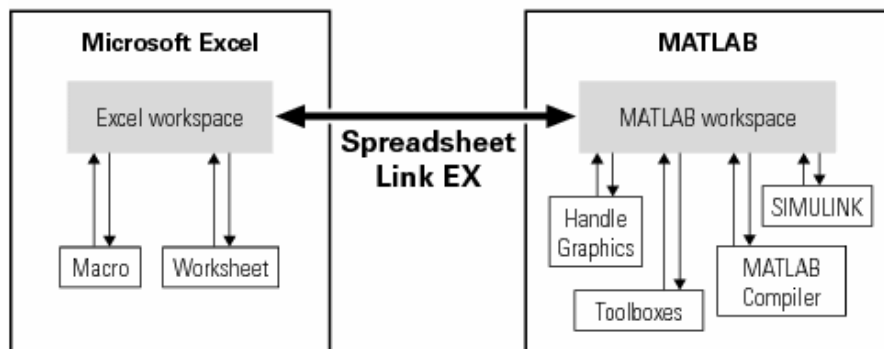


图 7.3-1: Excel-link 功能原理图

7.3.1 加载 Excel-link 宏

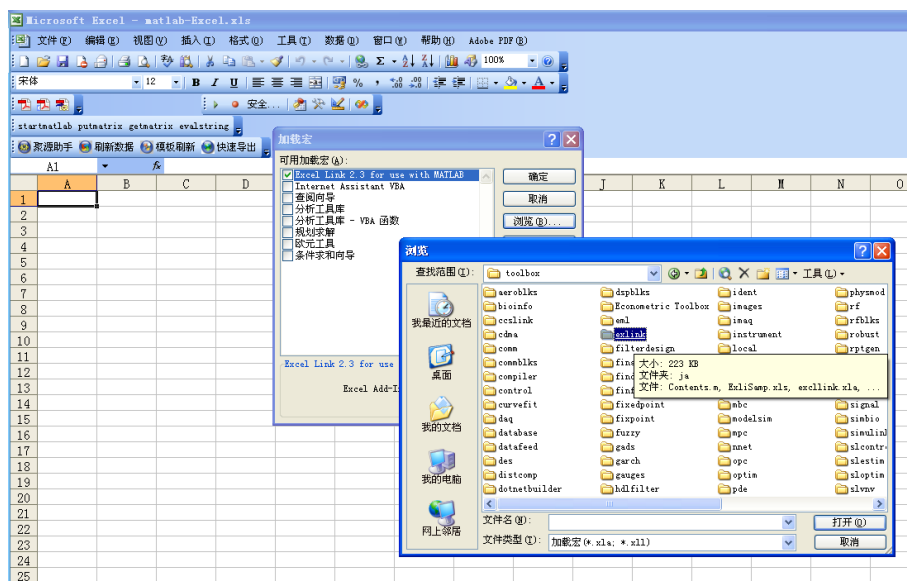


图 7.3-2 exlink 加载方法示意图 1

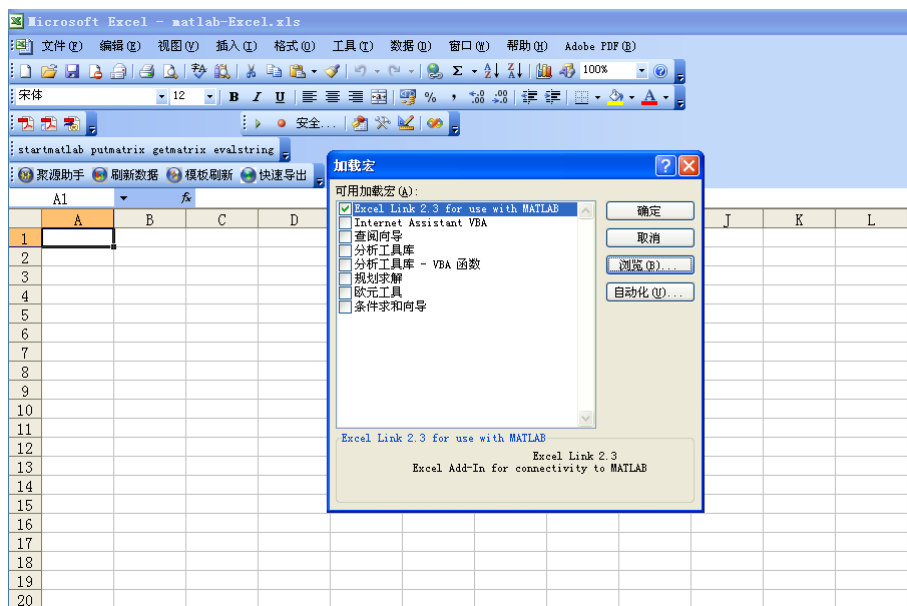


图 7.3-3 exlink 加载方法示意图 2

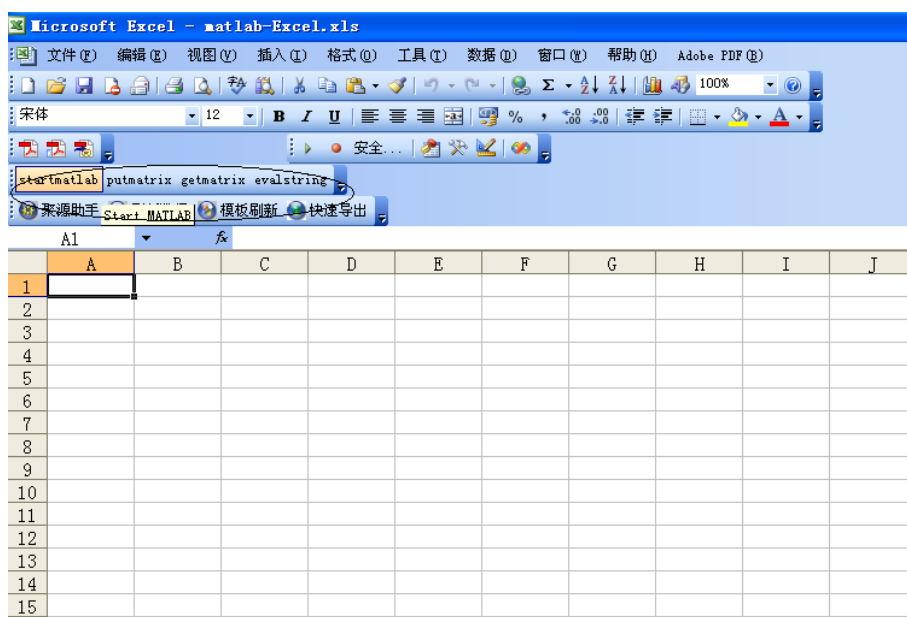


图 7.3-4 exlink 加载方法示意图 3

加载 Excel-link 宏的过程如图 7.3-2 至图 7.3-4 所示。单击“工具”菜单，选择“加载宏”选项，在弹出的加载宏界面中单击“浏览”按钮，通过浏览界面在路径“matlab 的安装路径\toolbox\exlink\”下找到 exclink.xla 文件，双击打开此文件则回到加载宏界面，在 Excel link2.3 for use with matlab 选项前打勾，点击“确定”按钮即可完成加载。加载 Excel-link 宏成功后会在 Excel 工具栏的下方出现“startmatlab、putmatrix、getmatrix、evalstring”等选项，通过这些选项可以实现 Matlab 与 Excel 之间的数据交互。下面介绍 Excel-link 宏的具体用法。

7.3.2 使用 Excel-link 宏

1. startmatlab 选项

单击 startmatlab 选项可以启动 matlab

Putmatrix: 将 Excel 的数据传输到 matlab 中

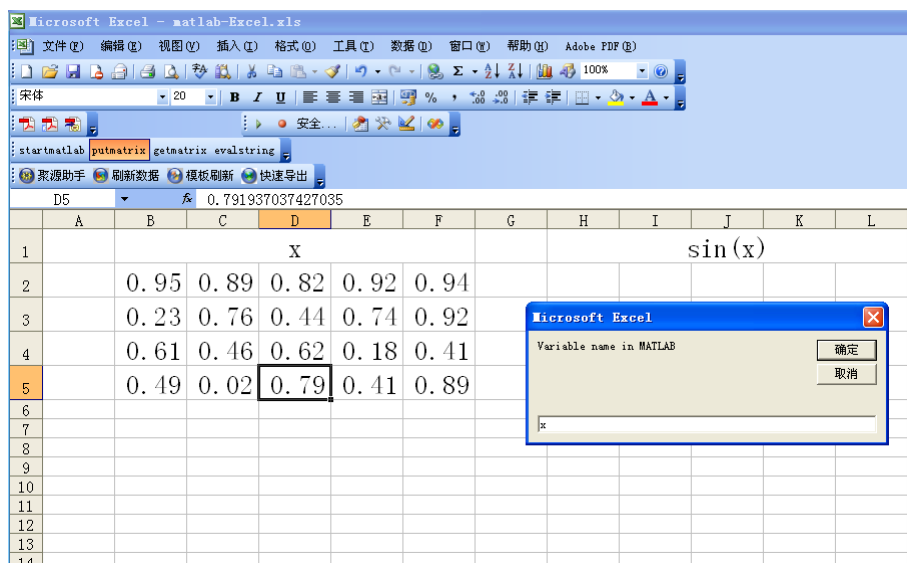


图 7.3-5: exlink 使用方法示意图

在 matlab 中: 可看到传入到 matlab 中的矩阵 x, 就算 $y=\sin(x)$

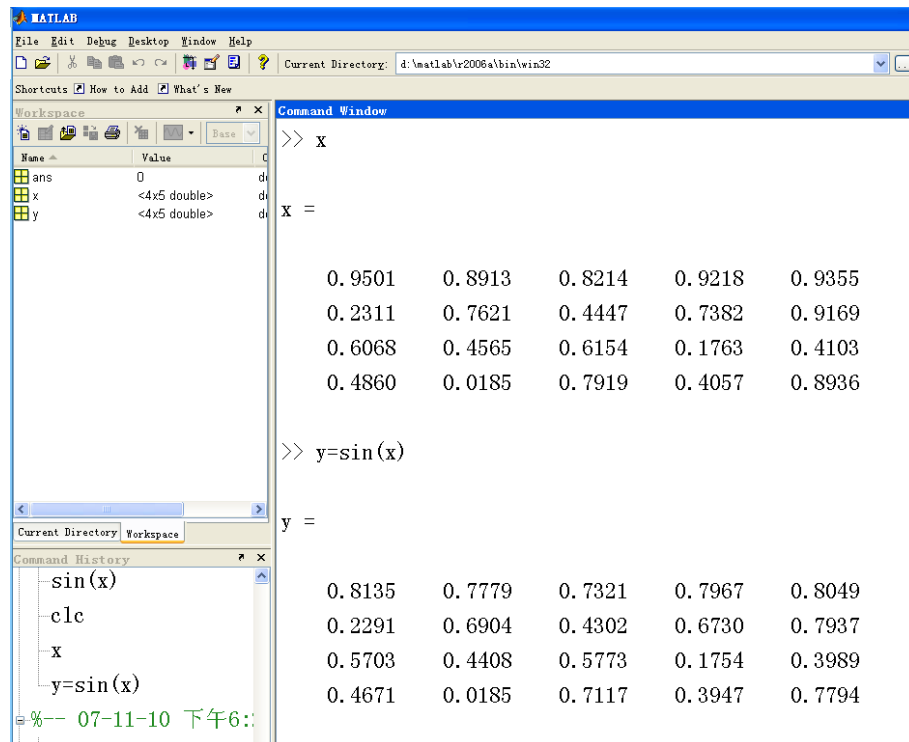


图 7.3-6: exlink 使用方法示意图

Getmatrix: 将 matlab 的数据传输到 Excel 中

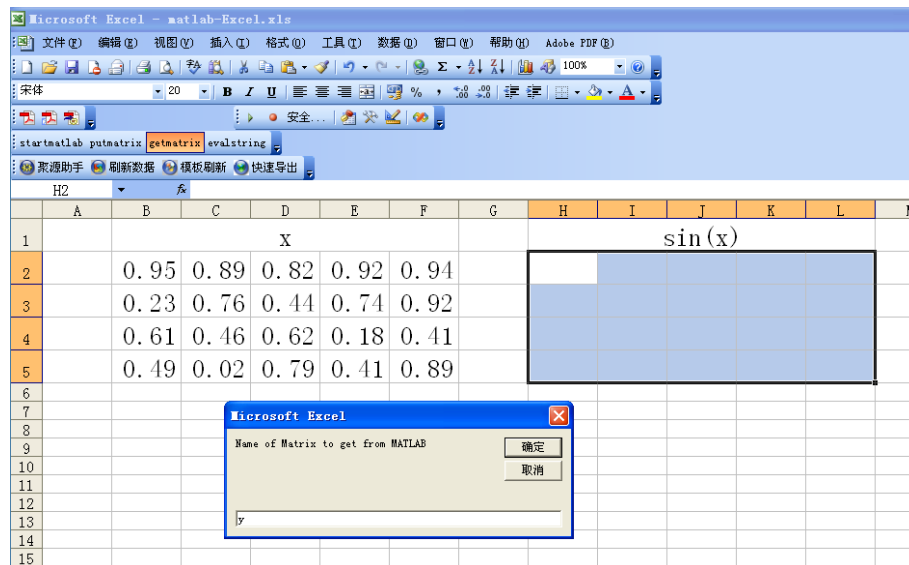


图 7.3-7: exlink 使用方法示意图

点击 getmatrix 出入要传入的矩阵变量名称，确定得

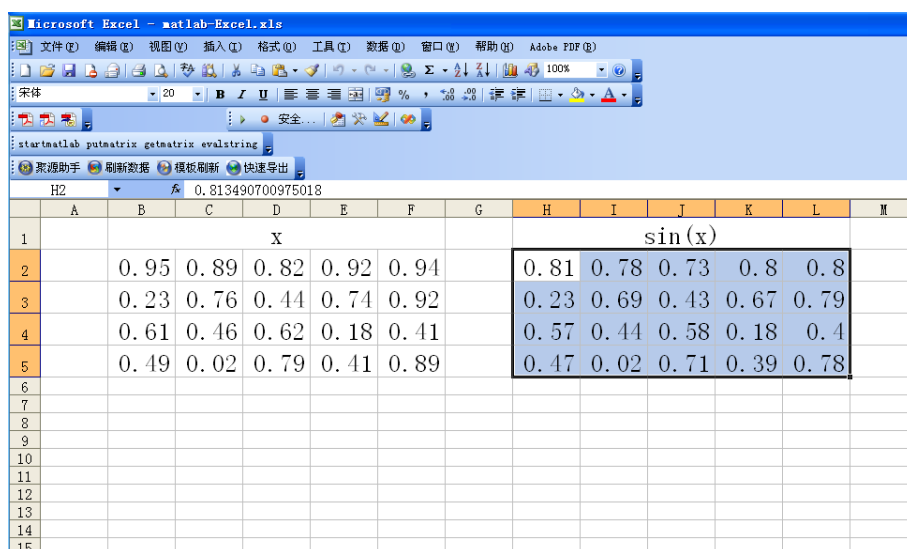


图 7.3-8: exlink 使用方法示意图

Evalstring: 执行 string 的 matlab 命令, 具体可以参看 matlab 的 help

7.3.3 Excel2007 加载与使用宏

加载方法:

点击 Excel 的 office 按钮 → 点击 Excel 选项 → 在加载项中 点击转到见下图

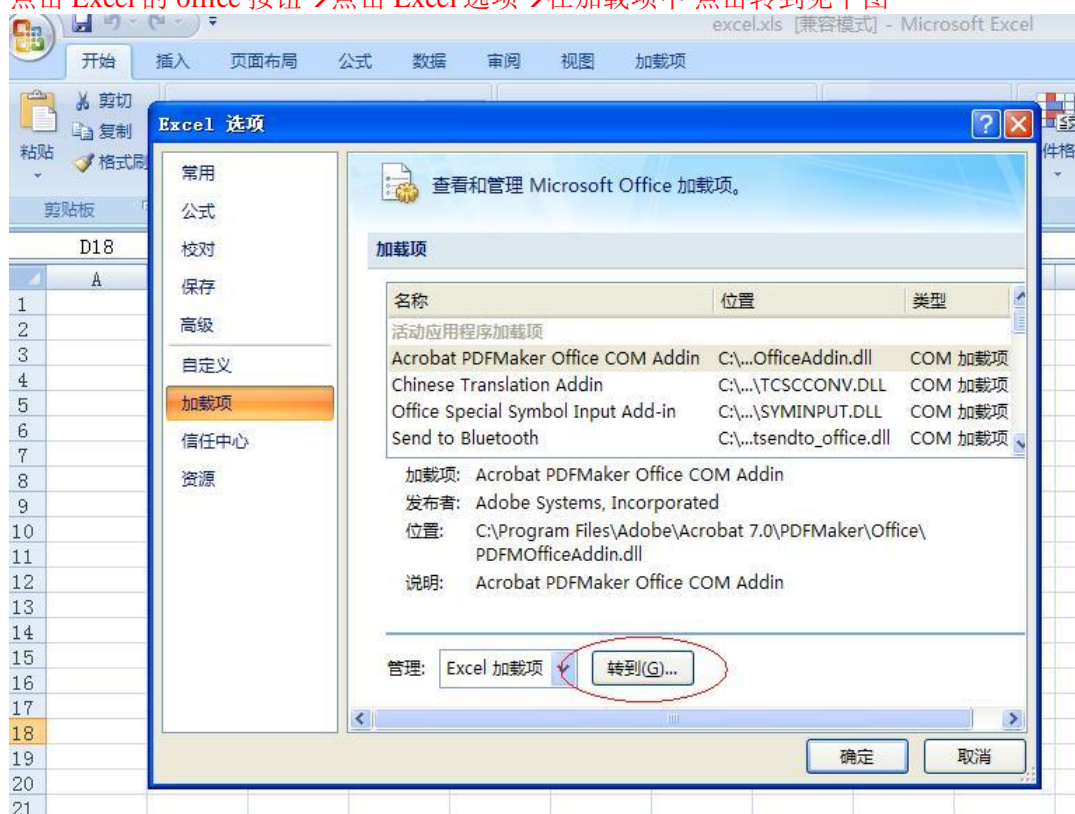


图 7.3-9 Excel2007 加载 exlink

浏览 (matlab 的安装路径) → toolbox 文件夹 → exlink 文件夹 → exclink.xla 文件 (打开)

使用方法:

在 Excel2007 加载项下可以发现 exlink 相关的按钮, 具体使用方法与 exlink 在 Excel2003 中的使用方

法一样。

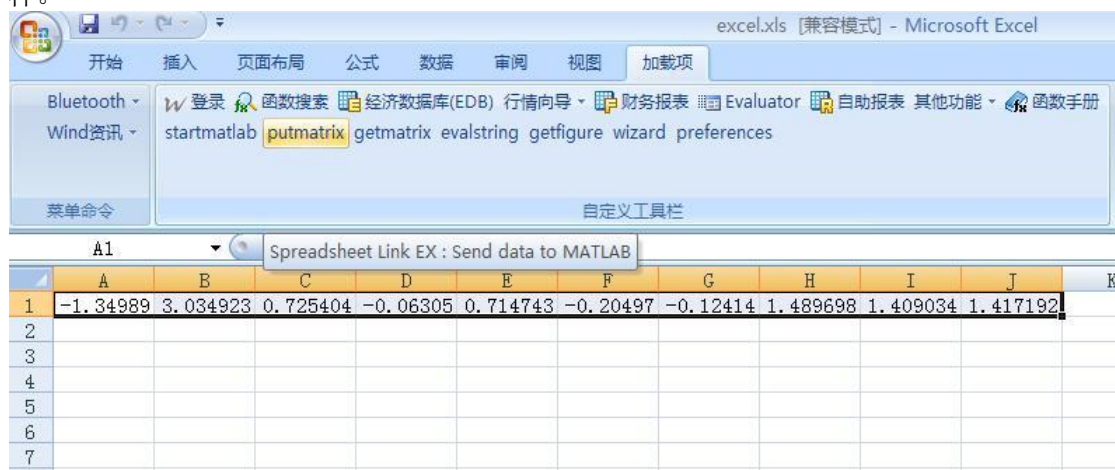


图 7.3-10 Excel2007 使用 exlink

7.4 参考文献

1. 郑志勇. 金融数量分析基于 Matlab 编程. 北京: 北航出版社, 2009
2. Excel 研究组. Excel2007 函数与公式速查手册. 北京: 电子工业出版社, 2008