
MATLAB programming course for beginners, supported by Wagatsuma Lab@Kyutech

Table of Contents

Specifications and requirements	1
Main program	1
Supplementary information to publish	5

/* The MIT License (MIT): Copyright (c) 2022 Hiroaki Wagatsuma and Wagatsuma Lab@Kyutech

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. */

Specifications and requirements

1. @Time : 2022-8-10
2. @Author : Hiroaki Wagatsuma
3. @Site : https://github.com/hirowgit/1A1_matlab_intermediate_course
4. @IDE : MATLAB R2022a
5. @File : lec0b_step4.m

Main program

```
NofD=5;  
maxD=5;  
  
rM=rand(NofD,NofD);
```

```
binaryM=(rM>0.5);

disp('binaryM');
disp(binaryM);
key=find(~binaryM);

doubleM=double(binaryM);
doubleM(key)=3;
disp('doubleM');
disp(doubleM);

intM=floor(rM*maxD)+1;
disp('intM');
disp(intM);

key=find(intM<5);
intM2=intM;
intM2(key)=0;
disp('intM2');
disp(intM2);

s_intM=sort(intM);
disp('s_intM');
disp(s_intM);

% s_intM=[
%     1     2     1     1     2
%     3     2     2     1     4
%     4     2     2     2     4
%     4     2     3     2     4
%     4     2     4     5     5];

% s_intM=[
%     1     2     1     1     1
%     2     3     2     2     1
%     2     3     3     2     1
%     4     5     3     2     3
%     5     5     5     2     5];

s_intM=[
    1     1     1     1     1
    1     2     1     2     3
    1     2     1     3     5
    1     4     3     4     5
    1     5     5     5     5];
s_intM=[
    1     1     1     1     5
    1     2     1     2     5
    1     2     1     3     5
    1     4     3     4     5
    1     5     5     5     5];
disp(s_intM);
```

```
dM=diff(s_intM);

[ki kj]=find(dM>0);

LackNum=setdiff(1:NofD,unique(kj));

sM=sortrows([zeros(length(LackNum),1); ki] [LackNum';kj]],2);
ki=sM(:,1); kj=sM(:,2);

mD=max(max(dM));
ti2=[]; tj2=[];
for i=2:mD
    [ti tj]=find(dM==i);
    if ~isempty(ti)
        sM=sortrows([ repmat(ti,[i-1,1]);ti2] [ repmat(tj,[i-1,1]);tj2]],2);
        ti2=sM(:,1); tj2=sM(:,2);
    end
end

if ~isempty(ti2)
    sM=sortrows([ti2; ki] [tj2; kj]],2);
    ki=sM(:,1); kj=sM(:,2);
end

sect_id=[0 find(diff(kj)>0)' length(kj)];
sect=[sect_id(1:end-1)+1; sect_id(2:end)];
sect_eg=mat2cell(sect',ones(1,NofD),2);

sect_data=cellfun(@(x) sort(ki(x(1):x(2))),sect_eg,'UniformOutput',false);
key=cell2mat(cellfun(@(x) length(x)<=1 &
    sum(x)==0,sect_data,'UniformOutput',false));
% [ 1      1      1      1      1] # {[ 1 1]} # {[      0]} the same position
% [ 5      5      5      5      5] # {[14 14]} # {[      0]} the same position
if sum(key)>0
    sect_data(key)={[]};
end
% disp(sect_data)

NofE_data=cellfun(@(x) diff([0 x' NofD]),sect_data,'UniformOutput',false);
zeroS=cell(size(NofE_data));
zeroE=cell(size(NofE_data));

keyL=find(cell2mat(cellfun(@(x)
    length(x)<maxD,NofE_data,'UniformOutput',false)));
if ~isempty(keyL)
    terM=cell2mat(sect_eg(keyL));

    terM1=terM(:,1);
    trNum=mat2cell(s_intM(1,kj(terM1))',ones(1,length(kj(terM1))));
    zeroS(kj(terM1))=cellfun(@(x) zeros(1,x-1),trNum,'UniformOutput',false);

    trNum=mat2cell(s_intM(NofD,kj(terM1))',ones(1,length(kj(terM1))));
```

```
zeroE(kj(terM1))=cellfun(@(x) zeros(1,NofD-
x),trNum,'UniformOutput',false);
```

```
    NofE_data2=cellfun(@(x,y,z) [x, y,
    z],zeroS,NofE_data,zeroE,'UniformOutput',false);
```

```
else
```

```
    NofE_data2=NofE_data;
```

```
end
```

```
NofE_data_m=cell2mat(NofE_data2);
```

```
binaryM
```

```
    0    0    0    0    1
    0    0    1    1    0
    0    1    0    0    1
    0    1    0    0    1
    0    0    1    1    0
```

```
doubleM
```

```
    3    3    3    3    1
    3    3    1    1    3
    3    1    3    3    1
    3    1    3    3    1
    3    3    1    1    3
```

```
intM
```

```
    2    2    3    1    4
    1    2    3    5    1
    2    5    3    1    5
    3    5    2    2    3
    2    2    5    4    2
```

```
intM2
```

```
    0    0    0    0    0
    0    0    0    5    0
    0    5    0    0    5
    0    5    0    0    0
    0    0    5    0    0
```

```
s_intM
```

```
    1    2    2    1    1
    2    2    3    1    2
    2    2    3    2    3
    2    5    3    4    4
    3    5    5    5    5

    1    1    1    1    5
    1    2    1    2    5
    1    2    1    3    5
    1    4    3    4    5
    1    5    5    5    5
```

Supplementary information to publish

If you want to make a pdf or html file on the code, you can use the code "x_publish_each_codes.m" in the same folder. Please change the file name as " this_file_tag='lec*_step*' " (* will be replaced to the number of the target file).

The code "x_publish_all_codes.m" works for such a publication applying to all codes in the same folder (Note: "x_publish_all_codes_sub.m" should be located in the same folder).

Published with MATLAB® R2022a