MA2507 Computing Mathematics Laboratory: Week 1

1. Generating vectors and matrices

```
>> format compact     % reduce space for display
>> a=[1 2 3]          % generate a vector
a =
     1     2     3
>> a=[1,2,3]
a =
     1     2     3
>> b=4:6
b =
     4     5     6
>> b=4:1:6
b =
     4     5     6
>> b=4:0.5:6
b =
    4.0000    4.5000    5.0000    5.5000    6.0000
>> b=linspace(4,6,5)
b =
    4.0000    4.5000    5.0000    5.5000    6.0000
>> c=[1;1;-2]                % column vector, ";" for next row
c =
     1
     1
    -2
>> A=[1 2 3                   % generate a matrix
      3 4 5
      2 2 1]
A =
     1     2     3
     3     4     5
     2     2     1
>> A=[1 2 3; 3 4 5; 2 2 1]    % ";" for next row
A =
     1     2     3
     3     4     5
     2     2     1
>> A=[1, 2, 3; 3, 4, 5; 2, 2, 1]
A =
     1     2     3
     3     4     5
     2     2     1
>> A(2:3,2:3)                % sub-matrix, not saved, stored in ans
ans =
     4     5
     2     1
>> A(2,3)                   % (2,3) entry of A
ans =
     5
>> diag(A)                  % get the diagonal elements
ans =
     1
     4
     1
>> diag(c)                  % create a matrix with given diagonal
ans =
     1     0     0
     0     1     0
     0     0    -2
>> diag(c,1)                % create a matrix
ans =
     0     1     0     0
     0     0     1     0
     0     0     0    -2
     0     0     0     0
>> diag(c,-1)               % create a matrix
ans =
     0     0     0     0
     1     0     0     0
     0     1     0     0
     0     0    -2     0
>> eye(3)                   % identity matrix
ans =
     1     0     0
     0     1     0
     0     0     1
>> ones(3)                  % matrix with all entries 1
ans =
     1     1     1
     1     1     1
     1     1     1
>> ones(3,3)
ans =
     1     1     1
     1     1     1
     1     1     1
>> ones(3,1)
ans =
     1
     1
     1
>> ones(1,3)
ans =
     1     1     1
>> rand                   % random number, uniform in [0,1]
ans =
    0.8147
>> rand                   % random number, again
ans =
    0.9058
>> rand(3)                % generate 3x3 random matrix
ans =
    0.1270    0.0975    0.9575
    0.9134    0.2785    0.9649
    0.6324    0.5469    0.1576
>> rand(3,1)             % generate 3x1 random vector
ans =
    0.9706
    0.9572
    0.4854
>> rand(3,3)             % generate 3x3 random matrix
ans =
    0.8003    0.9157    0.6557
    0.1419    0.7922    0.0357
    0.4218    0.9595    0.8491
```

2. Basic operations: `+, -, *, \, /, ^, '`, linear systems and least squares

```
>> a=[1 2 3];         % hide results with ";"
>> c=[7 8 9];
>> f = c'             % transpose for real vector
f =
     7
     8
     9
>> a*f                % row multiplies column
ans =
    50
>> f*a                % column multiplies a row
ans =
     7    14    21
     8    16    24
     9    18    27
>> a/10               % divided by 10
ans =
    0.1000    0.2000    0.3000
>> 10\a               % multiplied by (1/10)
```

```
ans =
    0.1000    0.2000    0.3000
>> u=2+3i             % complex number
u =
   2.0000 + 3.0000i
>> v=4+i              % complex number
v =
   4.0000 + 1.0000i
>> i=100              % over-write internal i
i =
   100
>> v=4+i
v =
   104
>> v=4+1i            % complex number without internal i
v =
   4.0000 + 1.0000i
>> u'                 % complex conjugate
ans =
   2.0000 - 3.0000i
>> conj(u)            % complex conjugate
ans =
   2.0000 - 3.0000i
>> A = [1 2 3; 3 4 5; 2 2 1];
>> det(A)                    % determinant
ans =
     2
>> rank(A)
ans =
     3
>> inv(A)                     % inverse matrix
ans =
   -3.0000    2.0000   -1.0000
    3.5000   -2.5000    2.0000
   -1.0000    1.0000   -1.0000
>> A'                         % tranpose for real matrix
ans =
     1     3     2
     2     4     2
     3     5     1
>> b=[1 1 -2]';
>> x=A\b                       % solve Ax=b, IMPORTANT!
x =
    1.0000
   -3.0000
    2.0000
```

```
>> c=[1 1 -2];
>> y=c/A                      % solve yA=c
y =
    2.5000   -2.5000    3.0000
>> B=A(:,1:2)                 % submatrix of A, 1st and 2nd columns
B =
     1     2
     3     4
     2     2
>> z = B\b                    % least squares problem min||Bz-b||
z =
   -3.0000
    2.3333
>> B*z                        % check Bz ?= b
ans =
    1.6667
    0.3333
   -1.3333
>> A(2,3)=5+2i               % reset one entry
A =
   1.0000 + 0.0000i   2.0000 + 0.0000i   3.0000 + 0.0000i
   3.0000 + 0.0000i   4.0000 + 0.0000i   5.0000 + 2.0000i
   2.0000 + 0.0000i   2.0000 + 0.0000i   1.0000 + 0.0000i
>> A’                        % transpose and complex conjugate
ans =
   1.0000 + 0.0000i   3.0000 + 0.0000i   2.0000 + 0.0000i
   2.0000 + 0.0000i   4.0000 + 0.0000i   2.0000 + 0.0000i
   3.0000 + 0.0000i   5.0000 - 2.0000i   1.0000 + 0.0000i
>> transpose(A)              % just transpose
ans =
   1.0000 + 0.0000i   3.0000 + 0.0000i   2.0000 + 0.0000i
   2.0000 + 0.0000i   4.0000 + 0.0000i   2.0000 + 0.0000i
   3.0000 + 0.0000i   5.0000 + 2.0000i   1.0000 + 0.0000i
>> conj(A)                   % complex conjugate
ans =
   1.0000 + 0.0000i   2.0000 + 0.0000i   3.0000 + 0.0000i
   3.0000 + 0.0000i   4.0000 + 0.0000i   5.0000 - 2.0000i
   2.0000 + 0.0000i   2.0000 + 0.0000i   1.0000 + 0.0000i
```

3. Built-in functions `sqrt, exp, log, log10, sin, cos, tan, cot`

```
>> A=[1 2; -1 0]
A =
     1     2
    -1     0
>> exp(A)
ans =
    2.7183    7.3891
    0.3679    1.0000
>> B=(pi/2)*A
B =
    1.5708    3.1416
   -1.5708         0
>> cos(B)
ans =
    0.0000   -1.0000
    0.0000    1.0000
>> sin(B)
ans =
    1.0000    0.0000
   -1.0000         0
>> a = [-2, -2+0.001i, -2-0.001i]
a =
  -2.0000 + 0.0000i  -2.0000 + 0.0010i  -2.0000 - 0.0010i
>> sqrt(a)   % discontinuity along negative real axis
ans =
   0.0000 + 1.4142i   0.0004 + 1.4142i   0.0004 - 1.4142i
```

4. The dot: element-wise operations

```
>> a=[1 2 3]; b=[4 5 6];
>> a.*b    % element-wise multiplication
ans =
     4    10    18
>> a.\b    % element-wise left division
ans =
    4.0000    2.5000    2.0000
>> a./b   % element-wise right division
ans =
    0.2500    0.4000    0.5000
>> 1./a   % vector with elements 1/a(i)
ans =
    1.0000    0.5000    0.3333
>> a.^3    % element-wise power
ans =
     1     8    27
>> b.^a   % element-wise power
ans =
     4    25   216
>> A=[1 2;3 4]  % generate matrix
A =
     1     2
     3     4
>> 2.^A        % matrix with entries 2^A(i,j)
```

```
ans =
     2     4
     8    16
>> B=[2 -1; 1 3]
B =
     2    -1
     1     3
>> A.^B        % matrix with entries A(i,j)^B(i,j)
ans =
    1.0000    0.5000
    3.0000   64.0000
>> A.*B        % matrix with entries A(i,j)*B(i,j)
ans =
     2    -2
     3    12
>> A.\B        % matrix with entries A(i,j)\B(i,j)
ans =
    2.0000   -0.5000
    0.3333    0.7500
>> A./B        % matrix with entries A(i,j)/B(i,j)
ans =
    0.5000   -2.0000
    3.0000    1.3333
>> A(1,2)=2+3i  % reset one element of A
A =
   1.0000 + 0.0000i   2.0000 + 3.0000i
   3.0000 + 0.0000i   4.0000 + 0.0000i
>> A.’          % transpose only
ans =
   1.0000 + 0.0000i   3.0000 + 0.0000i
   2.0000 + 3.0000i   4.0000 + 0.0000i
```