**SCIM212**
AY25/26 SEM 1
nadear

## Math & Trig

$\log$ is natural; $\log_{10}$ use log10. Degrees: use sind, cosd, tand.

```
log_e = log(5); log_10 = log10(5);
log_b = log(5)/log(7); % change-of-base
th = 30; % degree
s = sind(th); c = cosd(th); t = tand(th);
r = sqrt(25); q = nthroot(27, 3); % 3
```

### Rounding

vpa(x,d) uses at least d significant digits

```
money = input('Input the money in US dollars: ')
kg = money / 6.22
fprintf('The amount of money : %.2f\n', money)
fprintf('The weight of apricots is %.2f kg\n', kg)
```

## Control Structures

**if, elseif Conditions**

```
if n < 10
    disp("n smaller 10")
elseif n <= 20
    disp("n between 10 and 20")
else
    disp("n larger than 20")
end
```

**Switch Case**

```
n = input("Enter an integer: ");
switch n
    case -1
        disp("negative one")
    case {0,1,2,3} % check four cases together
        disp("integer between 0 and 3")
    otherwise
        disp("integer value outside interval
             [-1,3]")
end % control structures terminate with end
```

**For-Loop**

```
for i = 1:3
    disp("cool");
end
```

**While-Loop**

```
n = 1;
nFactorial = 1;
while nFactorial < 1e100
    n = n + 1;
    nFactorial = nFactorial * n;
end
```

## Vectors and matrices

```
>> v = [1 2 3 4]
v = 1 2 3 4
>> u = [1; 2; 3; 4]
u = 1
    2
    3
    4
>> x = 1:5;
x = 1 2 3 4 5
>> y = 1:0.5:3;
y = 1.0000 1.5000 2.0000 2.5000 3.0000
>> y ([1 ,3 ,4])
ans = 1.0000 2.0000 2.5000
>> aT = [1 9 5 7];
bT = [1 0 -5 7];
cT = [-3 7 0 -5];
dT = [0 -2 5 9];
M = [aT; bT; cT; dT]
M =
     1 9 5 7
     1 0 -5 7
    -3 7 0 -5
     0 -2 5 9
a = aT';
b = bT';
c = cT';
d = dT';
W = [a, b, c, d]
W =
     1 1 -3 0
     9 0 7 -2
     5 -5 0 5
     7 7 -5 9
A = [-4 3 3; 9 -1 -0.3; -0.1 0 -2];
% (i) Characteristic polynomial
syms q
I = [1 0 0; 0 1 0; 0 0 1];
char = det(A - q*I)
charpoly = det(A - q*eye(3))
% (ii) Factorize the characteristic polynomial
factor(charpoly)
% (iii) Eigenvalues and eigenvectors
[V, D] = eig(A)
>> M(2,1) % element at row 2, col 1
ans = 1
>> M(2,:) % entire 2nd row
ans = 1 0 -5 7
>> M(:,2) % entire 2nd column
ans =
     9
     0
     7
    -2
% 2) Specific rows & columns by vectors or ranges
>> M([1 2],[2 3]) % rows 1,2 and cols 2,3
ans =
     9 5
     0 -5
>> M(2:3,:) % rows 2 to 3, all cols
ans =
     1 0 -5 7
    -3 7 0 -5
>> M(:,2:3) % all rows, cols 2 to 3
ans =
     9 5
     0 -5
     7 0
    -2 5
% 3) Handy patterns
>> M(end,:) % last row
>> M(:,end-1) % next-to-last column
>> M(1:2:end,:) % odd rows, all columns
>> M([1 1 3],[2 4 4]) % repeats allowed (3x3
     result)
% 4) Logical & linear indexing
>> mask = M > 0; M(mask) % all positive entries
     as a column
>> ind = sub2ind(size(M), [1 2 3], [2 3 4]);
>> M(ind)
```

## Plotting functions using MATLAB

**Function handles**

```
>> f = @(x) x^2 +x +1
>> f(2)
ans =
    7
>> f([1 5 3 2])
ans =
    3 31 13 7
>> g = @ (x) x > 5;
>> g(1)
ans =
    logical
    0
>> g(5.5)
ans =
    logical
    1
>> syms x y
r = x^2 - y + 3;
subs(r, x, 4)
ans =
    19 - y
```

$$h(x) = \begin{cases} x^2, & \text{if } x > 5, \\ -x^2, & \text{if } x < 5. \end{cases}$$

```
>> h = @(x) (x>5) .* x .^2 + (x<=5) .* (-x.^2) ;
```

**Plotting the functions**

```
>> f = @(x) sin(x) + log(x) ;
>> fplot(f) ;
>> fplot ( @sin, [-pi,pi])
>> hold on
>> fplot ( @cos, [-pi,pi])
x = linspace(0, 2*pi, 400);
y1 = sin(x); y2 = cos(x);
plot(x, y1, 'LineWidth', 1.2); hold on
plot(x, y2, '--', 'LineWidth', 1.2);
grid on
legend('sin x','cos x','Location','best')
title('Basic Trig'); xlabel('x'); ylabel('y');
```

**Subplots & scatter**

```
subplot(1,2,1); plot(x,y1); title('sin');
subplot(1,2,2); scatter(x,y2,8,x); title('cos');
```

**Contours & surfaces**

```
[xg, yg] = meshgrid(-2:0.05:2, -2:0.05:2);
zg = exp(-(xg.^2 + yg.^2));
figure; contour(xg, yg, zg, 15); axis equal tight
figure; surf(xg, yg, zg); shading interp

[x,y,z]=sphere(30)
mesh(x,y,z)
hold on
surf(x+3,y+2,z-1)
axis equal
xlabel('X');
ylabel('Y');
zlabel('Z');
```

## Symbolic Math

**Create, expand, diff, integrate, solve**

```
syms x y
f = (x+2)^3 * (x-1);
fE = expand(f);
df = diff(f, x);
I1 = int(sin(x)^2, x, 0, pi); % definite integral
S = solve(x^2 - 5*x + 6 == 0, x); % roots 2 and 3
g = (x^3 - 1)/(x - 1);
gS = simplify(g); % simplifies to x^2 + x + 1
%%
y = cos(tan(2^(cot(x)))) * x^(3*x^2 - 2);
dy_dx = diff(y, x)
d2y_dx2 = diff(dy_dx, x)
dy_dx_at_2 = double(subs(dy_dx, x, 2))
d2y_dx2_at_2 = double(subs(d2y_dx2, x, 2))
%%
f_raw = 0.8 * x * exp(-x*(0.1 - y));
K = 1 / int(int(f_raw, y, 0, inf), x, 0, inf);
f = simplify(K * f_raw);
fx = simplify(int(f, y, 0, inf));
fy = simplify(int(f, x, 0, inf));
% (i) P(X > 3)
P_X_gt_3_sym = int(fx, 3, inf);
P_X_gt_3 = double(P_X_gt_3_sym)
% (ii) independence check: f(x,y) ?= fX(x)*fY(y)
is_indep = isAlways(simplify(f - fx*fy) == 0)
% (iii) P(at least one component > 3) = 1 - P(X
     <=3, Y<=3)
P_both_le_3_sym = int(int(f, y, 0, 3), x, 0, 3);
P_at_least_one_gt_3 = double(1 - P_both_le_3_sym)

function y = f(x)
    if (x > -1) && (x <= 1)
        y = x^2;
    elseif x > 1
        y = x - 1;
    elseif (x >= -2) && (x <= -1)
        y = -x + 1;
    else
        y = 0;
    end
end
result = f(f(f(f(f(3)))));
disp(result);

f = 5 * (a*x + (2*b)/x)^3;
fprime = diff(f, x)
eq1 = subs(f, x, 3/2) == 3
eq2 = subs(fprime, x, 3/2) == 30
sol = solve([eq1, eq2], [a, b]);
a_val = simplify(sol.a)
b_val = simplify(sol.b)
```

**Numeric solve & function handle**

```
solnum = vpasolve(exp(-x) == x, x, 0.5); %
     numeric root near 0.567
h = matlabFunction(sin(x)^2 + x, 'Vars', x); %
     fhandle h(x)
```

## Approximation Error

Relative error $E_r = \left|\dfrac{x_{\text{true}} - x_{\text{approx}}}{x_{\text{true}}}\right|$.

```
x_true = pi; x_approx = 3.14;
Er = abs(x_true - x_approx)/abs(x_true);
% stop when target k digits is met: Er<=5*10^(-k)
k = 0; tol = 5*10^(-k);
if Er <= tol
    k = k + 1
end
```