(1)

% Income tax the logical way

inc = [5000 10000 15000 30000 50000];

tax = 0.1 \* inc .\* (inc <= 10000); % (inc <= 10000) creat an logical vector [1 1 0 0 0]

% (inc > 10000 & inc <= 20000) creat an logical vector [0 0 1 0 0]

tax = tax + (inc > 10000 & inc <= 20000).\* (0.2 \* (inc‐10000) + 1000);

tax = tax + (inc > 20000) .\* (0.5 \* (inc‐20000) + 3000);

disp( [inc' tax'] );

%% Exercise 5.5 & 5.7 in textbook p. 125

% 5.5 sum( (salary >32000 ) .\*employees ) ; salary levels are above

% 5.7 units = [200 500 700 1000 1500];cost = cost + 0.02 \* (units <= 500) .\* units;

(2)

%==============================================================

% 6.1.15 Vectorizing nested fors: loan repayment tables

%% forming the table of repayments for a ;oan of $1000 over 15 20 and 25 yrs

% rate% 15 yrs 20 yrs 25 yrs

% 10 10.75 9.65 9.09

% 11 11.37 10.32 9.80

% 12 12.00 11.01 10.53

% 13 12.65 11.72 11.28

%% Exercise to write the expression in p.140 by matlab code

% Method 1:

A = 1000; % amount borrowed

n = 12; % number of payments per year

disp ([' rate% 15 yrs 20 yrs 25 yrs']);

for r = 0.1 : 0.01 : 0.2

fprintf( '%4.0f%', 100 \* r );

for k = 15 : 5 : 25

temp = (1 + r/n) ^ (n\*k);

P = r \* A \* temp / (n \* (temp - 1));

fprintf( '%10.2f', P );

end;

fprintf( '\n' ); % new line

end;

(3)

%==============================================================

% 6.5 MARKOV PROCESSES : A random walk process

% check the textbook p. 150

% After few steps the random walk ended at either home or café

n = 6;

P = zeros(n); % all elements set to zero

for i = 3:6

P(i,i-1) = 2/3;

P(i-2,i-1) = 1/3;

end

P(1,1) = 1;

P(6,6) = 1;

x = [0 1 0 0 0 0]'; % remember x must be a column vector!

for t = 1:50

x = P \* x;

disp( [t x'] )

end

(4)

% To defined function handle

Fhandle=@ (arglist) expression

For example:

Two kinds of functions: anonymous function, and parameterized function.

(1) anonymous function, the parameters value is anonymous (hided in expression)

Fh1= @ (x) 4\*x.^2-50\*x+2;

Fh2= @ (x,y) sqrt(x.^2+y.^2)

Fh3=@(x) (x-1.5).^2;

(2) parameterized function: you can change parameter every time you call it

fh2 = @(x,c) (x-c).^2;% the value of the parameter is not defined

yet

c = 1.5;

fh3 = @ (x) fh2(x,c); % Now the value is defined

d=fh3(3)

% Two build-in function to pass the funtion name as the argument

% feval & fminbnd

% feval(F,x1,...,xn) evaluates the function specified by a function

% handle or function name, F, at the given arguments, x1,...,xn.

fhandle = @sqrt;

feval(fhandle, 9)

feval(fhandle, 25)

f = @(x,y,c) (x-c).^2+y.^2; % The parameterized function.

c = 1.5; % The parameter.

% IN BUILD-IN FUNCTION feval call the function that you define

b = feval(@(x,y) f(x,y,c),0.6,0.5)

% fminbnd Single-variable bounded nonlinear function minimization.

% X = fminbnd(FUN,x1,x2) attempts to find a local minimizer X of the function

% FUN in the interval x1 < X < x2. FUN is a function handle. FUN accepts

% scalar input X and returns a scalar function value F evaluated at X.

Two function forms for the definition of the function handle:

X = fminbnd(@(x) sin(x)+3,2,8)

y=2:0.1:8;

plot(y,sin(y)+3)

%----------------------------------

clear all;clc;

f = @(x,c) (x-c).^2; % The parameterized function.

c = 1.5; % The parameter.

X = fminbnd(@(x) f(x,c),0,2)

% use help to find the fplot

fplot(@(x) f(x,c),[0,2])

% fplot Plot 2-D function

% fplot(FUN) plots the function FUN between the limits of the current

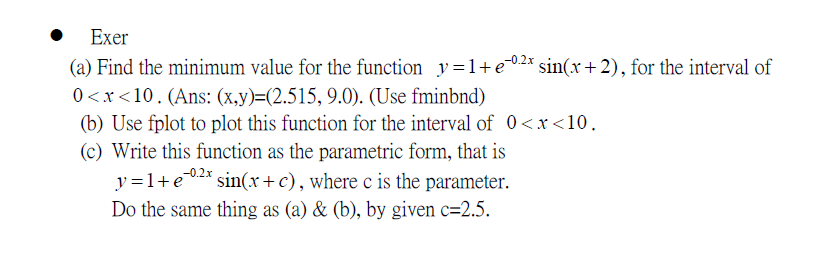
% axes, with a default of [-5 5].

%

% fplot(FUN,LIMS) plots the function FUN between the x-axis limits

% specified by LIMS = [XMIN XMAX].

% Example: fplot(@(x) x.^2.\*sin(1./x),[-1,1])



Sec. 8.1.2 : An exaple in p. 181 :Update processes

K = 0.05;

F = 10;

a = 0; % start time

b = 100; % end time

time = a; % initialize time

T = 25; % initialize temperature

load train % prepare to blow the whistle

dt=5;

opint=10;

% opint = input( 'output interval (minutes): ');

% if opint/dt ~= fix(opint/dt)

% sound(y, Fs) % blow the whistle!

% disp( 'output interval is not a multiple of dt!');

% break

% end

clc

format bank

disp( ' Time Temperature' );

disp( [time T] ) % display initial values

for time = a+dt : dt : b

T = T - K \* dt \* (T - F);

disp( [time T] )

end

end