%%%%% 初探MATLAB

%%% 常用數學函數

**% Example 1**

function y = myFun(x)

y = 0.5\*exp(x/3)-x.\*x.\*sin(x);

%%% 向量矩陣的運算

function [minValue, minIndex] = minxy(matrix)

%MINXY Minimum of a 2D matrix

% Usage: [minValue, minIndex] = minxy(A)

% minValue: the minimum of the matrix A

% minIndex: the 2D index of minValue in A

% Roger Jang, 20010219

[columnMin, columnMinIndex] = min(matrix);

[minValue, tmp] = min(columnMin);

minIndex = [columnMinIndex(tmp) tmp];

%%% 程式流程控制

**% Example 1**

% myTest: my first test M-file.

% Roger Jang, March 3, 1997

fprintf('Start of myTest.m!\n');

for i = 1:3

fprintf('i = %d ---> i^3 = %d\n', i, i^3);

end

fprintf('End of myTest.m!\n');

**% Example 2**

function out = piFun(n)

% piFun: Use sum of sequence to approximate pi

% Roger Jang, 20030726

total=0;

for i=1:n

item = (-1)^(i+1)/(2\*i-1);

total = total+item;

end

out = 4\*total;

**% Example 3**

function out = fibo(n)

% fibo: Fibonacci number

% Roger Jang, 20030726

if n==1

out=0;

return;

elseif n==2

out=1;

return;

else

out=fibo(n-1)+fibo(n-2);

end

**% Example 4**

function out = fibo(n)

% fibo: Fibonacci number using an analytic expression

% Roger Jang, 20030726

r1=(1+sqrt(5))/2;

r2=(1-sqrt(5))/2;

out=(r1^(n-1)-r2^(n-1))/sqrt(5);

**% Example 5**

maxN = 1000;

for n=1:maxN

value = prod(1:n);

if value>realmax

break;

end

end

fprintf('n = %d\n', n);

fprintf('(n-1)! = %d\n', prod(1:n-1));

**% Example 6**

function r=irrFind(cashFlowVec, x0)

r=fzero(@npvCompute, x0);

function npv=npvCompute(x)

n=length(cashFlowVec);

npv=sum(cashFlowVec./((1+x).^(0:n-1))); % Yearly compounding

end

end

**% Example 7**

function r=irrFind(cashFlowVec, x0)

r=fzero(@npvCompute, x0);

function npv=npvCompute(x)

n=length(cashFlowVec);

npv=sum(cashFlowVec./((1+x).^(0:n-1))); % Yearly compounding

end

end

**% Example 8**

function output = fact01(n)

% FACT01 Calculate factorial of a given positive integer (for-loop version)

output = 1;

for i = 1:n,

output = output\*i;

end

**% Example 9**

function output = fact02(n)

% FACT2 Calculate factorial of a given positive integer (recursive version)

if n == 1, % Terminating condition

output = 1;

return;

end

output = n\*fact02(n-1);

**% Example 10**

function out = ranking(x)

% ranking: Find the rank of each element in a vector

% Usage: out = ranking(x)

% x(i) is rank out(i) within the vector x

% Roger Jang, 20030726

if nargin<1, selfdemo; return; end

[sorted, position]=sort(x, 'descend'); % 由大到小排列

n=length(x);

rank=1:n;

[junk, index]=sort(position);

out=rank(index);

% ====== Self demo

function selfdemo

x=[92, 95, 58, 75, 69, 82];

out=ranking(x);

fprintf('x=%s\n', mat2str(x));

fprintf('ranking(x)=%s\n', mat2str(out));