

程式流程控制

MATLAB 在程式撰寫應用中有一些指令會影響程式執行的流程，可以歸納為兩類，一種為迴圈指令，另一種則是條件指令。

MATLAB 迴圈類指令有 for 迴圈 以及 while 迴圈。

for 迴圈

for 迴圈一般用法

for varname = x:y:z

例：

for i=1:1:10 % for i = 起始數值:遞增數值:終止數值

 for j=1:1:10

 a(i,j)=i+j;

 end

end

a

執行結果：

a =

2	3	4	5	6	7	8	9	10	11
3	4	5	6	7	8	9	10	11	12
4	5	6	7	8	9	10	11	12	13
5	6	7	8	9	10	11	12	13	14
6	7	8	9	10	11	12	13	14	15
7	8	9	10	11	12	13	14	15	16
8	9	10	11	12	13	14	15	16	17
9	10	11	12	13	14	15	16	17	18
10	11	12	13	14	15	16	17	18	19
11	12	13	14	15	16	17	18	19	20

上面 sample1.m 的範例中也有 for 迴圈

for 迴圈一般是用來處理已知執行次數的固定重複運算或數值設定工作。

在 for 迴圈的運用中特別注意的是要在景 r 迴圈結尾加上『end』，否則 for 將不會終止。

自訂函數

function [v,w]=f(x,y)

v=x^2+y^2;

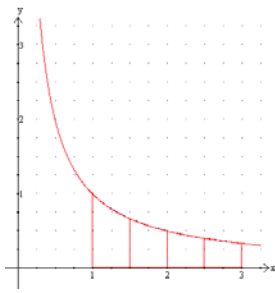
w=x+y;

[v,w]=f(4,-1)

v =17

w =3

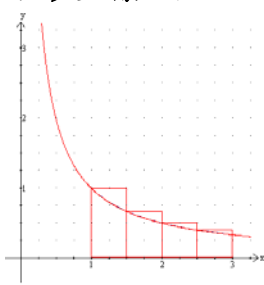
例： 利用矩形左端點法、矩形右端點法、矩形中點法、梯形法、Simpson 法，求 $f(x)=1/x$ 在區間 $[1, 3]$ 分割 $n=4$ 等分時的數值積分。



f.m (自訂函數)

```
function [value]=f(x)
value=1/x;
```

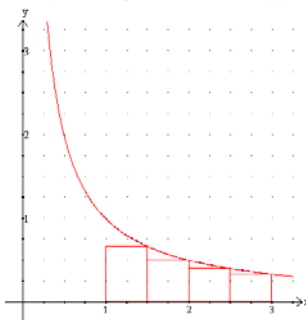
矩形左端點法



```
a=1;
b=3;
n=4;
h=(b-a)/n;
s1=0;
sum=0;
for i=1 :n;
s1=f(a+(i-1)*h) *h;
sum=sum+s1;
end
sum
```

ans =1.2833

矩形右端點法



```
a=1;
b=3;
n=4;
h=(b-a)/n;
```

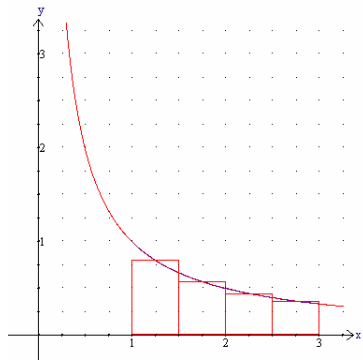
```

s1=0;
sum=0;
for i=1 :n;
s1=f(a+i*h) *h;
sum=sum+s1;
end
sum

```

ans =0.9500

矩形中點法



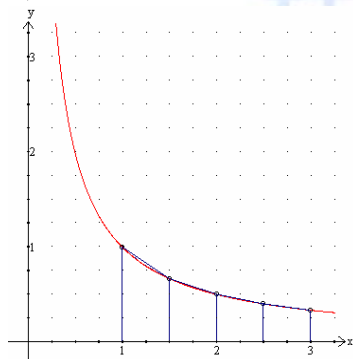
```

a=1;
b=3;
n=4;
h=(b-a)/n;
s1=0;
sum=0;
for i=1 :n;
s1=f(a+(i-1/2)*h) *h ;
sum=sum+s1;
end
sum

```

ans =1.0898

梯形法



```

a=1;
b=3;
n=4;
h=(b-a)/n;

```

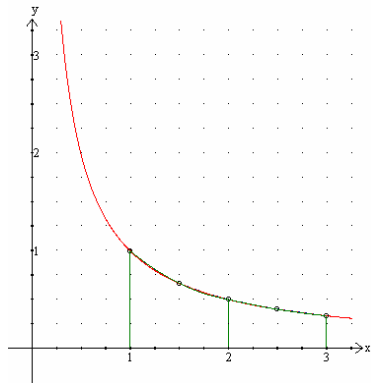
```

s1=0;
sum=0;
for i=1 :n;
s1=(f(a+(i-1)*h)+f(a+i*h)) *h/2;
sum=sum+s1;
end
sum

```

ans =1.1167

Simpson 法



```

a=1;
b=3;
n=4;
h=(b-a)/n;
s1=0;
sum=0;
for i=2 : 2:n;
s1=(f(a+(i-2)*h)+4*f(a+(i-1)*h)+f(a+i*h)) *h/3;
sum=sum+s1;
end
sum

```

ans =1.1000

例： 利用矩形中點法、梯形法、Simpson 法 求 $f(x)=\sqrt{x}\sin(x)$ 在區間 $[0, 4]$ 分割 $n=8$ 等分時的數值積分。

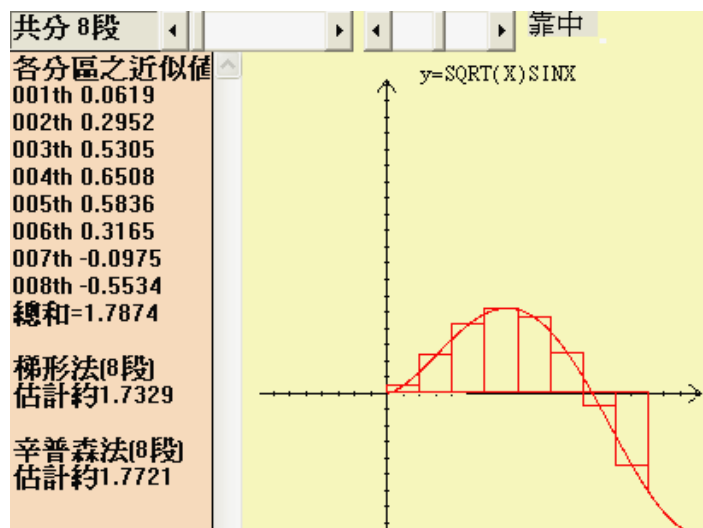
f.m (自訂函數)

```

function [value]=f(x)
value=sqrt(x)*sin(x);

```

矩形中點法



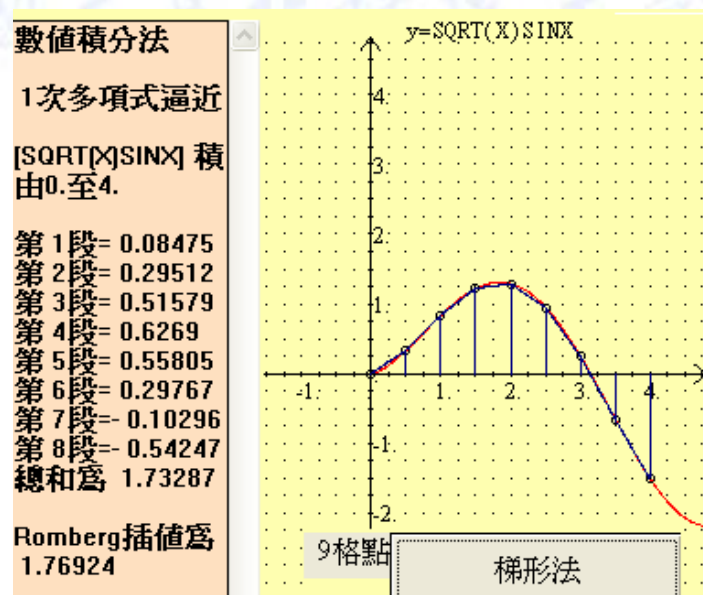
```

a=0;
b=4;
n=8;
h=(b-a)/n;
s1=0;
sum=0;
for i=0 :n-1;
s1=f(a+(i+1/2)*h);
sum=sum+s1;
end
sum*h

```

ans =1.7874

梯形法



```

a=0;
b=4;
n=8;
h=(b-a)/n;
s1=0;

```

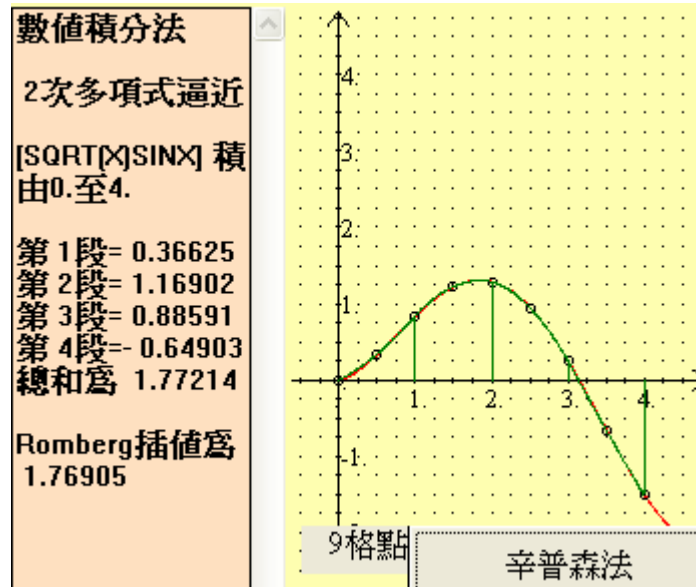
```

sum=0;
for i=0 :n-1;
s1=f(a+i*h)+f(a+(i+1)*h);
sum=sum+s1;
end
sum*h/2

```

ans =1.7329

Simpson 法



```

a=0;
b=4;
n=8;
h=(b-a)/n;
s1=0;
sum=0;
for i=0 : 2:n-2;
s1=f(a+i*h)+4*f(a+(i+1)*h)+f(a+(i+2)*h);
sum=sum+s1;
end
sum*h/3

```

ans =1.7721

```

while 迴圈
while 迴圈一般用法
while expression
    statements
end

```

while 迴圈適用於執行次數未知的運算。

MATLAB 條件類指令可分為 if-else 和 switch case

```

if
if 條件指令的一般用法
if 邏輯條件式
    運算式
end

```

if 條件指令後面的邏輯條件式裡頭的邏輯條件成立（True），則會執行 if 條件指令裡頭的運算式。

if 條件指令可以結合 else 以及 elseif 作更進一步的程式流程控制。

```

if 邏輯條件式
    運算式 A;
else
    運算式 B;
end

```

以及

```

if 邏輯條件式 A
    運算式 A;
elseif 邏輯條件式 B
    運算式 B;
elseif 邏輯條件式 C
    運算式 C;
else
    運算式 D;
end

```

例：

```

a=input('score')
if a>=80
    disp('甲等');
else if a>=70
    disp('乙等');
else if a>=60
    disp('丙等');
else
    disp('丁等');
end
end
end

```

前面我們提過的範例 sample1.m 裡頭也有 if 條件指令。

```

switch case
switch case 的一般用法

```

```

switch 條件式
case 值 A
    運算式 A;
case 值 B
    運算式 B;
.
.
.
end

```

例：

```
a=input('請輸入滿意程度1-5')
```

```

switch a
    case 1
        disp('非常滿意')
    case 2
        disp('滿意')
    case 3
        disp('無意見')
    case 4
        disp('不滿意')
    case 5
        disp('非常不滿意')
    otherwise
        disp('輸入錯誤')
end

```

例：利用二分法解方程式 $e^x - 3x^2 = 0$

```
y=exp(x)-3*x.^2
```

```
xl=3; xr=4 %由圖形觀測而得之初估值
```

```
format long
```

```
while abs(xr-xl)>10^(-6)
```

```
    yold=exp(xl)-3*xl.^2
```

```
    xtmp=(xl+xr)/2; ynew=exp(xtmp)-3*xtmp.^2
```

```
    if yold*ynew<0
```

```
        xr=xtmp
```

```
    else
```

```
        xl=xtmp
```

```
    end
```

```
end
```

```
(xl+xr)/2
```

```
ans =3.73307943344116
```

例：利用牛頓法解方程式 $\cos(x) - x^{1/2} = 0$ 可用下列之語法

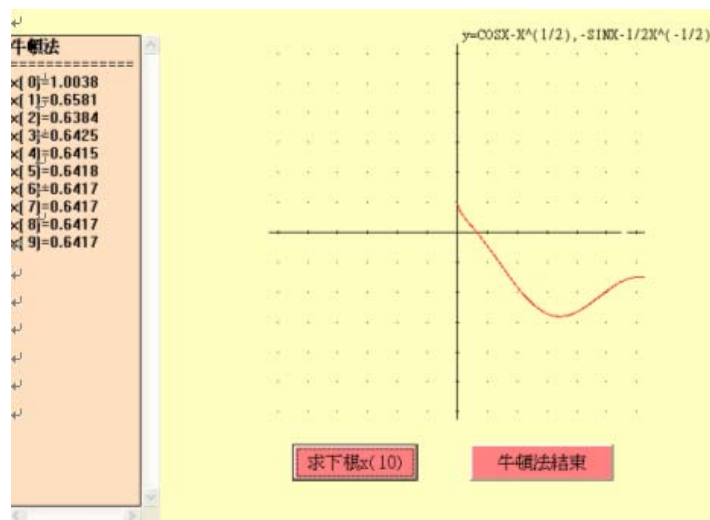
```
a=1
```

```
while abs(cos(a)-a^(1/2))>0.0001
```

```
b=a-(cos(a)-a^(1/2))/(-sin(a)-1/2*a^(-1/2))
```

```
a=b  
end  
a
```

a = 0.6417



國立臺北科技大學

林晶環老師