ΑΡΙΘΜΗΤΙΚΗ ΟΛΟΚΛΗΡΩΣΗ

Χρήση Matlab

Κανόνας τραπεζίου

```
function I = trap(func,a,b,n)
% func = name of function to be integrated
% a, b = integration limits
% n = number of segments (default = 100)
% I = integral estimate
if nargin<3,error('at least 3 input arguments required'),end
if ~(b>a),error('upper bound must be greater than lower'),end
if nargin<4|isempty(n),n=100;end
x = a; h = (b - a)/n;
s=func(a);
for i = 1 : n-1
x = x + h;
s = s + 2*func(x);
end
s = s + func(b);
I = (b - a) * s/(2*n);
```

Χρήση

>> f=@(x) 1/(1+x);

>> trap(f,0,1,2)

ans = 0.70833

>> trap(f,0,1,4)

ans = 0.69702

Επαναληπτικός κανόνας τραπεζίου

```
function T = rectrap(f,a,b,n)
%   f   name of the function
%   a   left  endpoint of [a,b]
%   b   right endpoint of [a,b]
%   n   number of levels for recursion
%   T   recursive trapezoidal rule list
m = 1;
h = b - a;
T = zeros(1,n+1);
T(1) = h*(feval(f,a) + feval(f,b))/2;
for j = 1:n,
  m = 2*m;
  h = h/2;
  s = 0;
  for k=1:m/2,
    x = a + h*(2*k-1);
    s = s + feval(f,x);
  end
  T(j+1) = T(j)/2 + h*s;
end
```

Χρήση

>> f=@(x) 1/(1+x)

>> rectrap(f,0,1,4)

ans =

0.75000 0.70833 0.69702 0.69412
0.69339

Κανόνας του Simpson

```
function s = simprl(f,a,b,m)
%   f   name of the function
%   a   left  endpoint of [a,b]
%   b   right endpoint of [a,b]
%   m    number of subintervals
%   s   Simpson rule quadrature value
h  = (b - a)/(2*m);
s1 = 0;
s2 = 0;
for k=1:m,
  x = a + h*(2*k-1);
  s1 = s1 + feval(f,x);
end
for k=1:(m-1),
  x = a + h*2*k;
  s2 = s2 + feval(f,x);
end
s = h*(feval(f,a)+feval(f,b)+4*s1+2*s2)/3;
```

Χρήση

```
>> f=@(x) 1/(1+x);
>> simprl(f,0,1,2)
ans =  0.69325

>> simprl(f,0,1,4)
ans =  0.69315
```

Κανόνας του Simpson 3/8

```
function r = Simpson38(f,a,b,n)
% f - Matlab inline function
% a,b - integration interval
% n - number of subintervals (panels)
% r - computed value of the integral
h = (b - a) / (n * 3);
r = f(a);
x = a + h;
for i = 1 : n-1
    r = r + 3 * f(x);
    x = x + h;
    r = r + 3 * f(x);
    x = x + h;
    r = r + 2 * f(x);
    x = x + h;
end;
r = r + 3 * f(x);
x = x + h;
r = r + 3 * f(x);
r = r + f(b);
r = r * h*3/8;
```

Χρήση

>> f=@(x) 1/(1+x);

>> Simpson38(f,0,1,1)

ans = 0.69375

>> Simpson38(f,0,1,3)

ans = 0.69316

Romberg

```
function [q,ea,iter]=romberg(func,a,b,es,maxit)
% func = name of function to be integrated
% a, b = integration limits
% es = desired relative error (default = 0.000001%)
% maxit = maximum allowable iterations (default = 30)
% q = integral estimate
% ea = approximate relative error (%)
% iter = number of iterations
if nargin<3,error('at least 3 input arguments required'),end
if nargin<4|isempty(es), es=0.000001;end
if nargin<5|isempty(maxit), maxit=50;end
n = 1;
I(1,1) = trap(func,a,b,n);
iter = 0;
```

Romberg(2)

```
while iter<maxit
iter = iter+1;
n = 2^iter;
I(iter+1,1) = trap(func,a,b,n);
for k = 2:iter+1
j = 2+iter-k;
I(j,k) = (4^(k-1)*I(j+1,k-1)-I(j,k-1))/(4^(k-1)-1);
end
ea = abs((I(1,iter+1)-I(2,iter))/I(1,iter+1))*100;
if ea<=es, break; end
end
q = I(1,iter+1);
```

Χρήση

>> romberg(f,0,1,0.01,10)

ans = 0.69315

Ενσωματωμένη συνάρτηση

```
>> f=@(x) 1/(1+x);
>> quad(f,0,1)
ans =  0.69315
```