
```

% display code
type('rectrap.m');
type('simpsons.m');
type('statError.m');

f = @statError;
a = 0;
b = 3;
n = 200;
err = 1e-5;
[S_t, error, evals] = rectrap(f, a, b, n, err);

fprintf(' Trapezoidal Rule\n');
fprintf('The value of the integral is %f\n', S_t);
fprintf('The number of function evaluations is %d\n', evals);
fprintf('The error is %f\n', error);

delta = 1e-6;
level_max = 50;
[numI, evals, error] = simpsons(f, a, b, delta, 0, level_max, 0);

fprintf(' Simpson''s Rule\n');
fprintf('The value of the integral is %f\n', numI);
fprintf('The number of function evaluations is %d\n', evals);
fprintf('The error is %f\n', error);

function [S_T, error, evals] = rectrap(f, a, b, n, er)
%RECTRAP Recursive trapezoid function
%

h_T = b - a;
S_T = 0.5 * h_T * (f(b) - f(a)) ;

evaluations = 2;

for i = 1:n
    h_i = h_T / 2 ;
    X = a + h_i : h_T : b - h_i;
    X = arrayfun(f, X);
    evaluations = evaluations + length(X);
    S_i = 0.5 * S_T + h_i * sum(X);
    error = (S_i - S_T) / S_i;
    S_T = S_i;
    h_T = h_i;
    if abs(error) < er
        break
    end
end

```

```

end

evals = evaluations;

end

function [numI, eval_num, err] = simpsons(f, a, b, delta, level, level_max,
eval_num)
%SIMPOSONS % adaptive simpsons rule
%

h = b - a;
c = (a + b) / 2;
I_1 = h * (f(a) + 4 * f(c) + f(b)) / 6;
level = level + 1;
d = (a + c) / 2;
e = (c + b) / 2;
I_2 = h * (f(a) + 4 * f(d) + 2 * f(c) + 4 * f(e) + f(b)) / 12;

% count the number of function evaluations
eval_num = eval_num + 3 + 5;

err = abs(I_2 - I_1) / 15;

if level >= level_max
    numI = I_2;
else
    if err <= 15 * delta
        numI = I_2 + (I_2 - I_1) / 15;
    else
        [numI_1, new_eval_num_1, err] = simpsons(f, a, c, delta / 2, level,
level_max, eval_num);
        [numI_2, new_eval_num_2, err] = simpsons(f, c, b, delta / 2, level,
level_max, eval_num);
        numI = numI_1 + numI_2;
        eval_num = new_eval_num_1 + new_eval_num_2;
    end
end

end

% fprintf('level: %d, eval_num: %d\n', level, eval_num);
% fprintf('I_1: %f, I_2: %f, numI: %f\n', I_1, I_2, numI);
end

function [result] = statError(x)
%STATERROR
% this is the fn, for the error funciont
% 2/PI * (integral from 0 to t) of e^(-x^2) dt

```

% we want the non integral part, the integral part is handled seperately

```
persistent constant  
if isempty(constant)  
    constant = 2/sqrt(pi);  
end  
  
result = constant * exp(-x^2);  
  
end
```

```
Trapezoidal Rule  
The value of the integral is 0.999971  
The number of function evaluations is 524289  
The error is 0.000006  
Simpson's Rule  
The value of the integral is 0.999978  
The number of function evaluations is 216  
The error is 0.000003
```

Published with MATLAB® R2023b