

Numerical Integration

Numerical integration is used where an integral is impossible or difficult to solve analytically.

1. Composite Trapezoidal Rule

The approach is to subdivide the interval across which we wish to integrate and approximate the integral of the function with a number of trapezoids. Each subdivision is of equal width h , where

$h = (b-a)/n$ where n is the number of subdivisions.

$$\int_a^b f(x)dx \approx \frac{h}{2} [f(a) + 2f(a+h) + 2f(a+2h) + 2f(a+3h) + \dots + 2f(a+(n-1)h) + f(b)]$$

Here is a program on the class library (**L:\trapex.m**) which integrates $\exp(-x^2)$ over a given interval using the composite trapezoidal rule.

```

                                L:\trapex.m

% trapex.m test program for numerical integration using the composite
% trapezoidal rule to solve the integral of exp(-x^2) between
% a and b

clear; help trapex;
format long; % configures MATLAB to report numbers with more decimal places

a=input('input a (starting value)->');
b=input('input b (end value) ->');
n=input('input number of intervals (n) ->');

h=(b-a)/n;

fa=exp(-a^2); % f(a)
fb=exp(-b^2); %f(b)

ff=0;

for i=2:n
    ff=ff+(2*exp(-(a+(h*(i-1)))^2)); % sum of 2f(a + i(h)) where i = 1 to n-1
end

result=(h/2)*(fa+fb+ff)

% result = f(a) + f(b) + sum of 2f(a + i(h)) where i = 1 to n-1
    
```

2. Composite Simpson's Rule

The approach is similar to the trapezoidal rule, however with Simpson's rule we subdivide the interval across which we wish to integrate and approximate the integral of the function with a number of quadratics

$$\int_a^b f(x)dx \approx \frac{h}{3} \left[f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + 2f(a+4h) + \dots + \dots + 2f(a+(n-2)H) + 4f(a+(n-1)h) + f(b) \right]$$

As with the composite trapezoidal rule the subintervals must have equal width, and n must be an even number.

Note the alternating coefficient of the middle terms – it goes 4,2,4,2 and so on. We can achieve this in MATLAB using a *for* loop. For example, if we use n=10, we want the following

$$h/3 * [f(a) + 4 f(a+h) + 2 f(a+2h) + 4 f(a+3h) + 2 f(a+4h) + 4 f(a+5h) + 2 f(a+6h) + 4 f(a+7h) + 2 f(a+8h) + 4 f(a+9h) + f(b)]$$

One way to achieve this in MATLAB is to split the middle terms into two parts as follows;

Step (1)

```
» n=10;
» for i=2:2:10
h=i-1
end
```

Result of this for loop => h = 1,3,4,5,7,9 which are the 'n' coefficients we require for the 4 f(a+nh) terms.

Step (2)

```
» for i=3:2:n
h=i-1
end
```

Result of this for loop => h = 2,4,6,8 which are the 'n' coefficients we require for the 2 f(a+nh)

Here is a program on the class library (**L:\simpsonex.m**) which integrates $\exp(-x^2)$ over a given interval using the composite simpson's rule.

L:\simpsonex.m

```
% Simpsonex.m - A program for composite Simpsons 1/3 rule
% to numerically integrate a function between a and b with
% n subintervals.
% The example program integrates exp(-x^2)

format long      % sets MATLAB to report more decimal places

clear; help simpsonex;

a=input('input a (starting value)->');
b=input('input b (end value) ->');
n=input('input number of intervals (n) ->');

h=(b-a)/n; % interval with

fa=exp(-(a^2)); % f(a)
fb=exp(-(b^2)); % f(b)

ff=0;

for i=2:2:n; % all 4*f(a+nh) terms to f(b) h=(1,3,5,7,...,n-1)
    x = (a+(i-1)*h);
    fx = exp(-x^2);
    ff = ff + 4*fx;
end

for i=3:2:n; % all 2*f(a+nh) terms to f(b) h=(2,3,4,6,...,n-2)
    x = (a+(i-1)*h);
    fx = exp(-x^2);
    ff = ff + 2*fx;
end

result=(h/3)*(fa+fb+ff) % integral result
                        % with approximation to area under curve
```

END
