Bangladesh University of Engineering and Technology

Numerical Technique Laboratory

EEE 212

Experiment No.: 07

Name of the Experiment: Numerical Integration

Department: EEE

Section: C1

Group: 01

Student No.: 1406131

1406132

Date of Performance: November 12, 2016

Date of Submission: November 18, 2016

- Problem: Find the following triple integration.

$$\int_{-x2}^{x2} \int_{-\left(\frac{x1+x2}{2}\right)}^{\left(\frac{x1+x2}{2}\right)} \int_{-x1}^{x1} e^{-(x^2+2y^2+3z^3)}\, dx\, dy\, dz$$

Here, x1 = 31, x2 = 32

Theory: We will convert this triple integration to double integration and then again convert that double integration to single integration. First of all, we can write a triple integration as

$$\mathbf{I} = \int_{xi}^{xf} \int_{yi}^{yf} \int_{zi}^{zf} f(x, y, z)\, dz\, dy\, dx \text{ ------------------ } (1)$$

Now, let $\mathbf{g}(\mathbf{x}, \mathbf{y}) = \int_{zi}^{zf} \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{z})\, \mathbf{dz}$

$$= \int_{xi}^{xf} \int_{yi}^{yf} g(x, y)\; dy\, dx$$

Again, let $\mathbf{k}(\mathbf{x}) = \int_{yi}^{yf} \mathbf{g}(\mathbf{x}, \mathbf{y})\, \mathbf{dy}$

$$= \int_{xi}^{xf} k(x)\, dx$$

Now we will apply trapezoidal rule.

$g(x,y) = \frac{hz}{2}\, [\, f(x,y,zi) + f(x,y,zf) + 2 \sum_{i=2}^{n} f(x, y, z_i)]$

$k(x) \; = \frac{hy}{2}\, [\, g(x,yi) + g(x,yf) + 2 \sum_{i=2}^{n} g(x, y_i)]$

and finally $\int_{xi}^{xf} k(x)\, dx \; = \frac{hx}{2}\, [\, k(xi) + k(xf) + 2 \sum_{i=2}^{n} k(x_i)]$

After calculating and substituting above function value in equation (1) we will get the total value of triple integration. For simson's 1/3 and 3/8 rule similar approaches will be taken.

Trapezoidal Rule

Code:

```
clc , clear all ;
close all ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms x y z ;
% defining function
f = @(x,y,z) exp(-( x.^2 + 2.*y.^2 + 3.*z.^2 )) ;

% limits of tiple integration
a1 = -31 ; a2 = 31 ; b1 = -31.5 ; b2 = 31.5 ; c1 = -32 ; c2 = 32 ;

% by using integral3 we will get actual value
actual_value = integral3(f,a1,a2,b1,b2,c1,c2) ;
fprintf('Actual value of this Integration is : %.5f\n',actual_value) ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Trapezoidal Rule %%%%%% Trapezoidal Rule %%%%%% Trapezoidal Rule %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

hx = (a2-a1) ;
hy = (b2-b1) ;
hz = (c2-c1) ;

g = (hz/2) * (f(x,y,c1) + f(x,y,c2)) ;  % it will be an expression
% converting g as a function
g = matlabFunction(g) ;
k = (hy/2) * (g(x,b1) + g(x,b2)) ;
% again converting k as a function
k = matlabFunction(k) ;

i0 = (hx/2) * ( k(a1) + k(a2) ) ;

hx = (a2-a1)/2 ;
hy = (b2-b1)/2 ;
hz = (c2-c1)/2 ;
g = 0 ;
k = 0 ;
g = (hz/2) * (f(x,y,c1) + 2*f(x,y,c1+hz) + f(x,y,c2)) ;
g = matlabFunction(g) ;
k = (hy/2) * (g(x,b1) + 2*g(x,b1+hy) + g(x,b2)) ;
k = matlabFunction(k) ;

i1 = (hx/2) * ( k(a1) + 2 * k(a1+hx) + k(a2) ) ;
iteration = 0 ;

e = 10e-2 ;
while abs(i1-i0) > e
    hx = hx / 2 ;
    hy = hy / 2 ;
    hz = hz / 2 ;
    p = c1 : hz : c2 ;
    n = length(p) ;
    g = 0 ;
```

```
    for i = 1 : n - 1
        g = g + .5 * hz * ( f(x,y,p(i))+f(x,y,p(i+1)) ) ;
    end
    g = matlabFunction(g) ;
    p = b1 : hy : b2 ;
    n = length(p) ;
    k = 0 ;
    for i = 1 : n - 1
        k = k + .5 * hy * ( g(x,p(i))+g(x,p(i+1)) )  ;
    end
    k = matlabFunction(k) ;
    p = a1 : hx : a2 ;
    n = length(p) ;
    sum = 0 ;
    for i = 1 : n - 1
        sum = sum + .5 * hx * (k(p(i))+k(p(i+1))) ;
    end
    i0 = i1 ;
    i1 = sum ;
    iteration = iteration + 1 ;
end

fprintf('Approximate value by using trapezoidal rule : %.5f\n',i1) ;
error = (abs(actual_value - i1)/actual_value) * 100 ;
accuracy = 100 - error ;
fprintf('Percentage of Accuracy : %.5f%%\n',accuracy) ;
fprintf('Percentage of Error : %.5f%%\n',error) ;
fprintf('Total Iteration number = %d\n',iteration) ;
```

Output:

Command Window

```
Actual value of this Integration is : 2.27326
Approximate value by using trapezoidal rule : 2.27326
Percentage of Accuracy : 100.00000%
Percentage of Error : 0.00000%
Total Iteration number = 7
fx >>
```

Simson’s 1/3 Rule

Code:

```
clc , clear all ;
close all ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms x y z ;
% defining function
f = @(x,y,z) exp(-( x.^2 + 2.*y.^2 + 3.*z.^2 )) ;
% limits of tiple integration
a1 = -31 ; a2 = 31 ; b1 = -31.5 ; b2 = 31.5 ; c1 = -32 ; c2 = 32 ;

% by using integral3 we will get actual value
actual_value = integral3(f,a1,a2,b1,b2,c1,c2) ;
fprintf('Actual value of this Integration is : %.5f\n',actual_value) ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% simson's 1/3 rule %%%% simson's 1/3 rule %%%% simson's 1/3 rule %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

hx = (a2-a1) ;
hy = (b2-b1) ;
hz = (c2-c1) ;

g = (hz/3) * (f(x,y,c1) + 4*f(x,y,c1+hz)+ f(x,y,c1+2*hz)) ;
g = matlabFunction(g) ;

k = (hy/3) * (g(x,b1) + 4*g(x,b1+hy)+ g(x,b1+2*hy)) ;
k = matlabFunction(k) ;

i0 = (hx/3) * ( k(a1) + 4*k(a1+hx)+ k(a1+2*hz) ) ;

hx = hx/2 ;
hy = hy/2 ;
hz = hz/2 ;

g = (hz/3) * (f(x,y,c1) + 4*f(x,y,c1+hz)+ f(x,y,c1+2*hz)) ;
g = matlabFunction(g) ;

k = (hy/3) * (g(x,b1) + 4*g(x,b1+hy)+ g(x,b1+2*hy)) ;
k = matlabFunction(k) ;

i1 = (hx/3) * ( k(a1) + 4*k(a1+hx)+ k(a1+2*hz) ) ;
iteration = 0 ;
e = 10e-2 ;
while abs(i1-i0) > e
    hx = hx / 2 ;
    hy = hy / 2 ;
    hz = hz / 2 ;

    p = c1 : hz : c2 ;
    n = length(p) ;
    g = 0 ;
    for i = 1 :2: n - 2
        g = g+ (hz/3) * ( f(x,y,p(i))+4*f(x,y,p(i+1))+ f(x,y,p(i+2)) );
```

```
    end
    g = matlabFunction(g) ;
    p = b1 : hy : b2 ;
    n = length(p) ;
    k = 0 ;
    for i = 1 :2: n - 2
        k = k + (hy/3) * ( g(x,p(i))+4*g(x,p(i+1))+g(x,p(i+2)) ) ;
    end
    k = matlabFunction(k) ;
    p = a1 : hx : a2 ;
    n = length(p) ;
    sum = 0 ;
    for i = 1 :2: n - 2
        sum = sum + (hx/3) * ( k(p(i))+4*k(p(i+1))+ k(p(i+2)) ) ;
    end
    i0 = i1 ;
    i1 = sum ;
    iteration = iteration + 1 ;
end

fprintf('Approximate value by using simson''s 1/3 rule is : %f\n',i1) ;
error = (abs(actual_value - i1)/actual_value) * 100 ;
accuracy = 100 - error ;
fprintf('Percentage of Accuracy : %.5f%%\n',accuracy) ;
fprintf('Percentage of Error : %.5f%%\n',error) ;
fprintf('Total Iteration number = %d\n',iteration) ;
```

Output:

```
Command Window
Actual value of this Integration is : 2.27326
Approximate value by using simson's 1/3 rule is : 2.273257
Percentage of Accuracy : 99.99987%
Percentage of Error : 0.00013%
Total Iteration number = 7
fx >>
```

Simson's 3/8 Rule

Code:

```
clc , clear all ;
close all ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms x y z ;
% defining function
f = @(x,y,z) exp(-( x.^2 + 2.*y.^2 + 3.*z.^2 )) ;
% limits of tiple integration
a1 = -31 ; a2 = 31 ; b1 = -31.5 ; b2 = 31.5 ; c1 = -32 ; c2 = 32 ;

% by using integral3 we will get actual value
actual_value = integral3(f,a1,a2,b1,b2,c1,c2) ;
fprintf('Actual value of this Integration is : %.5f\n',actual_value) ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% simson's 3/8 rule %%%% simson's 3/8 rule %%%% simson's 3/8 rule %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

hx = (a2-a1) ;
hy = (b2-b1) ;
hz = (c2-c1) ;

g = ((3*hz)/8) * (f(x,y,c1) + 3*f(x,y,c1+hz)+3*f(x,y,c1+2*hz)+
f(x,y,c1+3*hz)) ;
g = matlabFunction(g) ;

k = ((3*hy)/8) * (g(x,b1) + 3*g(x,b1+hy)+3*g(x,b1+2*hy)+ g(x,b1+3*hy)) ;
k = matlabFunction(k) ;

i0 = ((3*hx)/8) * ( k(a1) + 3*k(a1+hx)+3*k(a1+2*hx)+k(a1+3*hz) ) ;

hx = hx/2 ;
hy = hy/2 ;
hz = hz/2 ;

g = ((3*hz)/8) * (f(x,y,c1) + 3*f(x,y,c1+hz)+3*f(x,y,c1+2*hz)+
f(x,y,c1+3*hz)) ;
g = matlabFunction(g) ;

k = ((3*hy)/8) * (g(x,b1) + 3*g(x,b1+hy)+3*g(x,b1+2*hy)+ g(x,b1+3*hy)) ;
k = matlabFunction(k) ;

i1 = ((3*hx)/8) * ( k(a1) + 3*k(a1+hx)+3*k(a1+2*hx)+k(a1+3*hz) ) ;
iteration = 0 ;
e = 10e-2 ;
while abs(i1-i0) > e
    hx = hx / 2 ;
    hy = hy / 2 ;
    hz = hz / 2 ;

    p = c1 : hz : c2 ;
    n = length(p) ;
    g = 0 ;
```

```
    for i = 1 :3: n - 3
        g = g+ ((3*hz)/8)*(
f(x,y,p(i))+3*f(x,y,p(i+1))+3*f(x,y,p(i+2))+f(x,y,p(i+3)) );
    end
    g = matlabFunction(g) ;
    p = b1 : hy : b2 ;
    n = length(p) ;
    k = 0 ;
    for i = 1 :3: n - 3
        k = k + ((3*hy)/8) * (
g(x,p(i))+3*g(x,p(i+1))+3*g(x,p(i+2))+g(x,p(i+3)) ) ;
    end
    k = matlabFunction(k) ;
    p = a1 : hx : a2 ;
    n = length(p) ;
    sum = 0 ;
    for i = 1 :3: n - 3
        sum = sum + ((3*hx)/8) * ( k(p(i))+3*k(p(i+1))+3*k(p(i+2))+k(p(i+3))
) ;
    end
    i0 = i1 ;
    i1 = sum ;
    iteration = iteration + 1 ;
end

fprintf('Approximate value by using simson''s 3/8 rule is : %.5f\n',i1) ;
error = (abs(actual_value - i1)/actual_value) * 100 ;
accuracy = 100 - error ;
fprintf('Percentage of Accuracy : %.5f%%\n',accuracy) ;
fprintf('Percentage of Error : %.5f%%\n',error) ;
fprintf('Total Iteration number = %d\n',iteration) ;
```

Output:

```
Command Window
Actual value of this Integration is : 2.27326
Approximate value by using simson's 3/8 rule is : 2.27411
Percentage of Accuracy : 99.96249%
Percentage of Error : 0.03751%
Total Iteration number = 7
fx >>
```