[**DOING PHYSICS WITH MATLAB**](http://www.physics.usyd.edu.au/teach_res/mp/mphome.htm)

**MATHEMATICAL ROUTINES**

# COMPUTATION OF TWO-DIMENSIONAL

# INTEGRALS:

# DOUBLE or SURFACE INTEGRALS



Ian Cooper

School of Physics, University of Sydney

ian.cooper@sydney.edu.au

[**DOWNLOAD DIRECTORY FOR MATLAB SCRIPTS**](http://www.physics.usyd.edu.au/teach_res/mp/mscripts)

**math\_integration\_2D.m**

Demonstration mscript evaluating the integral of functions of the form *f*(*x*,*y*) using a two-dimensional form of Simpson’s 1/3 rule. The code can be changed to integrate functions between the specified lower and upper bounds.

**simpson2d.m**

Function to give the integral of a function *f*(*x*,*y*) using a two-dimensional form of Simpson’s 1/3 rule. The format to call the function is

Ixy = simpson2d(f,ax,bx,ay,by)

**NUMERICAL INTEGRATION:**

**COMPUTATION OF TWO-DIMENSIONAL INTEGRALS (DOUBLE OR SURFACE INTEGRALS)**

The function **simpson2d.m** is a very versatile , accurate and easy to implement function that can be used to evaluate a definite integral of a function *f*(*x*,*y*) between lower bounds and an upper bounds .

We want to compute a number expressing the definite integral of the function *f*(*x,y*) between two specific limits (*ax,* *bx)* and (*ay, by*)



The evaluation of such integrals is often called ***quadrature***.

We can estimate the value a double integral by a **two-dimensional version of Simpson’s 1/3 rule**.

# [Simpson’s 1/3 rule](http://www.physics.usyd.edu.au/teach_res/mp/doc/math_integration_1D.pdf)

This rule is based on using a quadratic polynomial approximation to the function *f*(*x*) over a pair of partitions. *N*-1 is the number of partitions where *N* must be **odd** and

Δ*x* ≡ *h* = (*b* – *a*) / (*N*-1). The integral is expressed below and is known as the ***composite Simpson’s 1/3 rule***.



Simpson’s rule can be written vector form as



where .

**c** and **f** are row vectors and **fT** is a column vector.

# Simpson’s [2D] method

The double integral



can be approximated by applying Simpson’s 1/3 rule twice – once for the *x* integration and once for the *y* integration with *N* partitions for both the *x* and *y* values.

*x*-values: 

*y*-values: 

The lower and upper bounds determine the size of the partitions



The *N* *x*-values and *N* *y*-values form a two-dimensional grid of *N* x *N* points. The function *f*(*x*,*y*) and the two-dimensional Simpson’s coefficients are calculated at each grid point. Hence, the function *f*(*x*,*y*) and the two-dimensional Simpson’s coefficients can be represented by *N* x *N* matrices **F** and **S** respectively.

The Simpson matrix **S** for *N* = 5 is

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 x 1 = 1 | 4 x 1 = 4 | 2x1 = 2 | 4x1 = 4 | 1x1 = 1 |
| 1 x 4 = 4 | 4 x 4 = 16 | 2x4 = 4 | 4x4 = 16 | 1x4 = 1 |
| 1 x 2 = 8 | 4 x 2 = 8 | 2x2 = 4 | 4x2= 8 | 1x2 = 1 |
| 1 x 4 = 4 | 4 x 4 = 16 | 2x4 = 4 | 4x4 = 16 | 1x4 = 1 |
| 1 x 1 = 1 | 4 x 1 = 4 | 2x1 = 2 | 4x1 = 4 | 1x1 = 1 |

Therefore, the **two-dimensional Simpson’s rule** which is used to estimate the value of the surface integral can be expressed as



The two-dimensional Simpson’s coefficient matrix **S** for *N* = 9 is

1 4 2 4 2 4 2 4 1

4 16 8 16 8 16 8 16 4

2 8 4 8 4 8 4 8 2

4 16 8 16 8 16 8 16 4

2 8 4 8 4 8 4 8 2

4 16 8 16 8 16 8 16 4

2 8 4 8 4 8 4 8 2

4 16 8 16 8 16 8 16 4

1 4 2 4 2 4 2 4 1

We will consider a number of examples which demonstrates how to apply the two-dimensional Simpson’s rule using the mscript **math\_integration\_2D.m**.

**Example 1**  integrate *x*: 0 🡪 2 and *y*: 1 🡪 5

(1) 

The exact value of the integral can be found analytically and its value is **416**. So we can compare the numerical estimate with the known exact value.

Steps in estimating the integral numerically using **math\_integration\_2D.m** and **simpson2d.m**

* Clear all variables, close any Figure Windows and clear the Command Window:

clear all

close all

clc

* Enter the number of partitions (must be an **odd** number), and the lower and upper bounds for *x* and *y* and calculate the range for the *x* and *y* values

num = 5;

xMin = 0;

xMax = 2;

yMin = 1;

yMax = 5;

x = linspace(xMin,xMax,num);

y = linspace(yMin,yMax,num);

* This is a two-dimensional problem, so we need to specify the values (*x*,*y*) at all grid points which are determined from the upper and lower bounds. We can do this using the Matlab command **meshgrid**. We can then calculate the value of the function *f*(*x*,*y*) at each grid point (*x*,*y*).

[xx yy] = meshgrid(x,y);

f = xx.^2 .\* yy.^3;

To show how the **meshgrid** functions works, see figure (1) and the outputs of the variables *x*, *y*, *xx*, *yy* and *f* that were displayed in the Command Window and the Simpson’s [2D] coefficients calculated with the function **simpson2d.m**.

x = 0 0.5000 1.0000 1.5000 2.0000

xx =

0 0.5000 1.0000 1.5000 2.0000

0 0.5000 1.0000 1.5000 2.0000

0 0.5000 1.0000 1.5000 2.0000

0 0.5000 1.0000 1.5000 2.0000

0 0.5000 1.0000 1.5000 2.0000

y = 1 2 3 4 5

yy =

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

4 4 4 4 4

5 5 5 5 5

f =

0 0.2500 1.0000 2.2500 4.0000

0 2.0000 8.0000 18.0000 32.0000

0 6.7500 27.0000 60.7500 108.0000

0 16.0000 64.0000 144.0000 256.0000

0 31.2500 125.0000 281.2500 500.0000



Fig. 1. The grid points for *N* = 5 and how these points relate to the Matlab matrices.

* Calculate the Simpson [2D] coefficients

% evaluates two dimension Simpson coefficients ---------------------------

sc = 2\*ones(num,1);

sc(2:2:num-1) = 4;

sc(1) = 1;

sc(num) = 1;

scx = meshgrid(sc,sc);

scxy = ones(num,num);

scxy(2:2:num-1,:) = scx(2:2:num-1,:)\*sc(2);

scxy(3:2:num-2,:) = scx(3:2:num-2,:)\*sc(3);

scxy(1,:) = sc';

scxy(num,:) = sc';

* Compute the integral

hx = (bx-ax)/(num-1); hy = (by-ay)/(num-1);

h = hx \* hy / 9;

integral = h \* sum(sum(scxy .\* f));

The complete mscript to compute the integral is

function integral = simpson2d(f,ax,bx,ay,by)

%num must be odd

%1 4 2 4 ...2 4 1

num = length(f);

hx = (bx-ax)/(num-1); hy = (by-ay)/(num-1);

h = hx \* hy / 9;

% evaluates two dimension Simpson coefficients ---------------------------

sc = 2\*ones(num,1);

sc(2:2:num-1) = 4;

sc(1) = 1;

sc(num) = 1;

scx = meshgrid(sc,sc);

scxy = ones(num,num);

scxy(2:2:num-1,:) = scx(2:2:num-1,:)\*sc(2);

scxy(3:2:num-2,:) = scx(3:2:num-2,:)\*sc(3);

scxy(1,:) = sc';

scxy(num,:) = sc';

% evaluates integral -----------------------------------------------------

integral = h \* sum(sum(scxy .\* f));

The exact value of the integral is **416**

With only 5 partitions and 25 (5x5) grid points, the numerical estimate is **416**, the same as the exact value.

**Example 2 Double Integrals and Volumes**

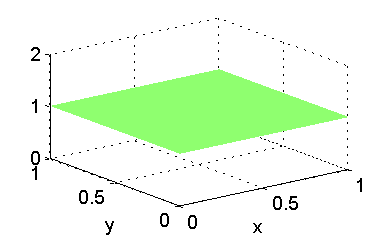


To gain an intuitive feel for double integrals, the volume of the region enclosed by the area *A* is equal to the value of the double integral.

**Volume *V* of a rectangular box**

*f*(*x*,*y*) = *k* height of box *k* > 0

Base of box – the lower bounds (*ax* and *ay*) and upper bounds (*bx* and *by*) determine the area of the rectangular base of the box

Box

*k* = 1 *ax* = 0 *bx* = 1 *ay* = 0 *by* = 1 *N* = 299

Exact volume (analytical) *V* = 1.0000

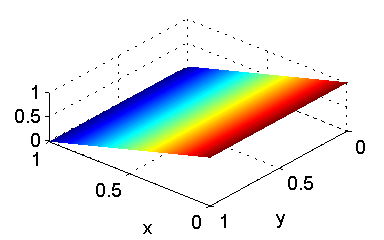
Simpson’s [2D] rule *V* = 1.0000

**Volume *V* of half box**

*f*(*x*,*y*) = 1 - *x*

Base of box – the lower bounds (*ax* and *ay*) and upper bounds (*bx* and *by*) determine the area of the rectangular base of the box





*ax* = 0 *bx* = 1 *ay* = 0 *by* = 1 *N* = 299

Exact volume (analytical) *V* = 0.50000

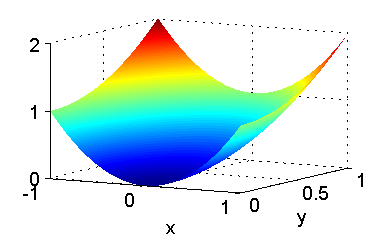
Simpson’s [2D] rule *V* = 0.50000

**Volume *V* of a part-bowl**

*f*(*x*,*y*) = *x*2 + *y*2

Base of box – the lower bounds (*ax* and *ay*) and upper bounds (*bx* and *by*) determine the area of the rectangular base of the surface





*a*x = -1 *bx* = 1 *ay* = 0 *by* = 1 *N* = 299

Exact volume (analytical) *V* = 1.3333

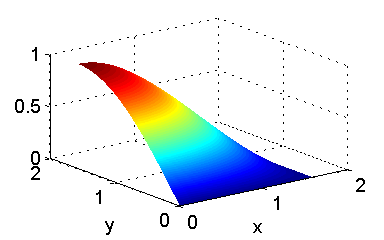
Simpson’s [2D] rule *V* = 1.3333

**Volume *V* over a rectangular base**



Base of box – the lower bounds (*ax* and *ay*) and upper bounds (*bx* and *by*) determine the area of the rectangular base of the surface



*a*x = 0 *bx* = π/2 *ay* = 0 *by* = π/2 *N* = 299

Exact volume (analytical)

*V* = 1.000

Simpson’s [2D] rule

*V* = 1.000000000008578

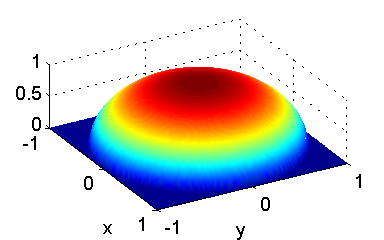
**Volume *of* ahemisphere using Cartesian coordinates**

Volume of a hemisphere of radius *a*  

Function







*a*x = -1 *bx* = 1 *ay* = -1 *by* = 1

Exact volume (analytical)

*V* = 2.094395102393195

Simpson’s [2D] rule

*N* = 99 *V* = 2.094**417986583109**

N = 999 V = 2.094395**646847362**

A logical Matlab function is used to define the function when *y* > 1 – *x*. The code to define the function is

f = real(sqrt(a^2 - xx.^2 - yy.^2));

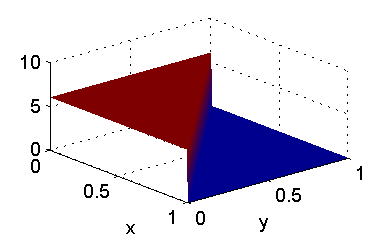
f((xx.^2 + yy.^2) > a^2) = 0;

**Volume *V* over a triangular base**





*a*x = 0 *bx* = 1 *ay* = 0 *by* = 1 height *h* = 6



Exact volume (analytical)

*V* = 3.0000

Simpson’s [2D] rule

*N* = 299 *V* = 3.00**6673873549240**

*N* = 999 *V* = 3.00**1944436635462**

*N* = 999 *V* = 3.00**1944436635462**

*N* = 2999 *V* = 3.000**632027607761**

Even with *N* = 2999 the calculation took less than 1.0 s on a fast Windows computer.

The differences between the exact and computed values is due to the rectangular grid and the condition on the function being zero when *y* > 1 – *x*

(*y* = 1 – *x* is a diagonal line and the grid is rectangular).

A logical Matlab function is used to define the function when *y* > 1 – *x*. The code to define the function is

f = h .\* ones(num,num);

f(yy > 1 - xx) = 0;

**Example 3 Polar coordinates**



where a point (*xP*, *yP*) has polar coordinates (*ρ*, *φ*) where



The grid pattern for the integration is shown in figure (2) for *N* = 9. At each point the function and Simpson [2D] coefficients are calculated.

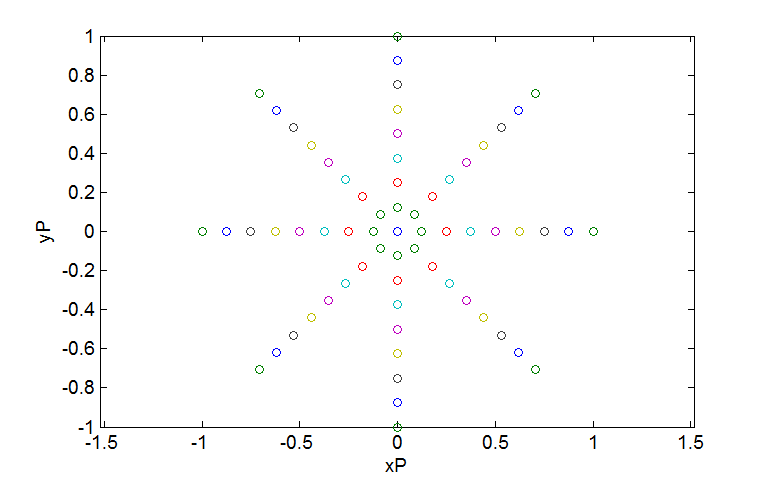


Fig. 2. The grid pattern when using polar coordinates.

Number of partitions *N* = 9 and number of grid points *N* x *N* = 81.

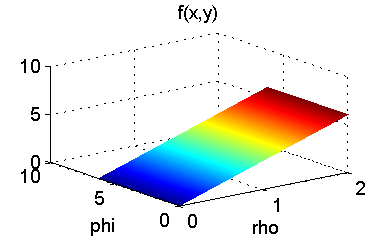
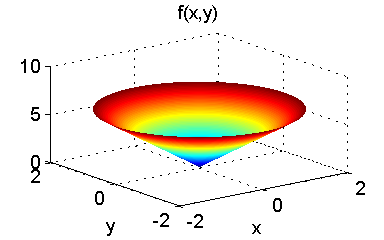
**Volume of a cylinder of radius *a* and height *h***

*a*x = 0 *bx* = 2 *ay* = 0 *by* = 2π *N* = 299

Exact volume (analytical) *V* = 37.69911184307751**7**

Simpson’s [2D] rule  *V* = 37.69911184307751**0**



**Volume of a hemisphere of radius *a***

*a*x = 0 *bx* = 2 *ay* = 0 *by* = 2π *N* = 299

Exact volume (analytical) *V* = 16.7551608191455**62**

Simpson’s [2D] rule  *V* = 16.7551608191455**59**

