**[DOING PHYSICS WITH MATLAB](https://d-arora.github.io/Doing-Physics-With-Matlab/)**

**Ian Cooper**

**matlabvisualphysics@gmail.com**

**A COMPUTATIONAL APPROACH TO**

**ELECTROMAGNETIC THEORY**

**CHAPTER 3**

**INTEGRAL CALCULUS**

**INTEGRATION**

**DOWNLOAD DIRECTORIES FOR MATLAB SCRIPTS**

[**Google drive**](https://drive.google.com/drive/u/3/folders/1j09aAhfrVYpiMavajrgSvUMc89ksF9Jb)

[**GitHub**](https://github.com/D-Arora/Doing-Physics-With-Matlab/tree/master/mpScripts)

**SCRIPTS**

simpson1d.m simpson2d.m

cemCh3.m cemStokes.m

**Reference**

David J Griffiths *Introduction to Electrodynamics* (3rd Edition)

Integration is defined for continuous functions. Thus, if a function is continuous in a region, then the integral of  is defined as another function  such that



The function  satisfies the condition



**SYMBOLIC INTEGRATION**

The symbolic function **int** can be used to evaluate both indefinite and definite integrals as shown in the Script **cemCh3.m (CELL 1)**

%% Symbolic Integration

close all; clc; clear

% Indefintite integral [1D]

syms x

% Input functiuon >>>>>

f = sin(2\*pi\*x)

% Integral

F = int(f,x) 🡪 -cos(2\*pi\*x)/(2\*pi)

% Definite integral [1D]

f = exp(-x)\*sin(x)/x

% Integral

F = int(f,x,0,inf) 🡪 Pi/4

% Indefintite integral [1D]

syms x a

% Input functiuon >>>>>

f = 1/(a^2 + x^2)

% Integral

F = int(f,x) 🡪 F = atan(x/a)/a

% Definite integral

f = x\*exp(-x)

% Integral

f = int(f,x,0, inf) 🡪 1

**NUMERICAL INTEGRATION**

Using Matlab makes it is easy to integrate a function numerically. For [1D] integrals we will only use the [1/3 Simpson rule](https://d-arora.github.io/Doing-Physics-With-Matlab/mpDocs/math_simpson_1D.pdf) with the function **simpson1d.m**

function integral = simpson1d(f,a,b)

% [1D] integration - Simpson's 1/3 rule

% f function a = lower bound b = upper bound

% Must have odd number of data points

% Simpson's coefficients 1 4 2 4 ... 2 4 1

numS = length(f); % number of data points

if mod(numS,2) == 1

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

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

sc(1) = 1; sc(numS) = 1;

h = (b-a)/(numS-1);

integral = (h/3) \* f \* sc;

else

integral = 'Length of function must be an ODD number'

end

**Example 1** **cemCh3.m (CELL 2)**

%% [1] integration using Simpson's rule

clear; close all; clc

% Input: number of grid point N (odd number), lower limit a, upper limit b

N = 999;

a = 0; b = pi/2;

x = linspace(a,b,N);

% Input function >>>>>

f = sin(x);

% Evaluate integral

F = simpson1d(f,a,b);

% Output to Commmand Window

fprintf('integral F = %2.10e \n',F)

🡪 integral F = 1.0000000000e+00

**Example 2** **cemCh3.m (CELL 2)**

% Input: number of grid point N (odd number), lower limit a, upper limit b

N = 999;

a = 0; b = 1;

x = linspace(a,b,N);

% Input function >>>>>

q = 0.3; r = 0.9; s = 6;

f = 1./((x-q).^2+0.01) + 1./((x-r).^2 + 0.04) - s;

% Evaluate integral

F = simpson1d(f,a,b);

% Output to Commmand Window

fprintf('integral F = %2.10e \n',F)

🡪 integral F = 2.9858325396e+01

Exact value F = 2.985832539549867e+011 *excellent agreement*

**[2D] integrals**

We can compute the value of double (area or surface) integrals of the form



You can use the Matlab function **integral2** or the Script **simpson2d.m**

How to use either function is illustrated in the following examples using the Script **cemCh3.m**.

**Example 3** **integral2**   **cemCh3.m (CELL 3)**

%% [2D] integration: Matlab function integral2

clear; close all; clc

% Limits >>>>>

N = 999;

xMin = -3; xMax= 3;

yMin= -4; yMax = 4;

% Grid

% x = linspace(xMin,xMax, N);

% y = linspace(yMin,yMax, N);

% Compute integral

funct = @(x,y) 1 + 2.\*x.^2 + y.^2;

F = integral2(funct,xMin,xMax,yMin,yMax);

% Output to Commmand Window

fprintf('integral F = %2.10e \n',F)

🡪 integral F = 5.9200000000e+02 (exact 592)

**Example 4** **simpson2d.m**   **cemCh3.m (CELL 4)**

%% [2D] integration: simpson2d.m

clear; close all; clc

% Limits >>>>>

N = 999;

xMin = -3; xMax= 3;

yMin= -4; yMax = 4;

% Grid

x = linspace(xMin,xMax, N);

y = linspace(yMin,yMax, N);

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

f = 1 + 2.\*xx.^2 + yy.^2;

F = simpson2d(f,xMin,xMax,yMin,yMax);

% Output to Commmand Window

fprintf('integral F = %2.10e \n',F)

🡪 integral F = 5.9200000000e+02 (exact 592)

**Example 5 area of a semicircle cemCh3.m (CELL 5)**

%% CELL 5: [2D] integration: simpson2d.m circle

clear; close all; clc

% Limits >>>>>

N = 999;

xMin = -1; xMax = 1;

yMin = -1; yMax = 1;

% Grid

x = linspace(xMin,xMax, N);

y = linspace(yMin,yMax, N);

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

f = ones(N,N);

f((xx.^2 + yy.^2) > 1) = 0;

F = simpson2d(f,xMin,xMax,yMin,yMax);

% Output to Commmand Window

fprintf('integral F = %2.10e \n',F)

integral F = 3.1414278122e+00 🡪 

**[3D] integrals**

We can compute the value of triple (volume) integrals of the form



You can use the Matlab function **integral3**.

**Example 6**  **cemCh3.m (CELL 6)**



%% CELL 6: [3D] integration

clear; close all; clc

xmin = 0; xmax = 2;

ymin = 0; ymax = 4;

zmin = 0; zmax = 8;

fun = @(x,y,z) x.^2.\*y.^3.\*z.^4;

F = integral3(fun,xmin,xmax,ymin,ymax,zmin,zmax);

fprintf('integral F = %2.10e \n',F)

🡪 integral F = 1.1184810667e+06

(exact 1.118481066666667e+06)

**Example 7** **cemCh3.m (CELL 7)**



Integral Over the Unit Sphere in Cartesian Coordinates

%% [3D] integration

clear; close all; clc

xmin = -1;

xmax = 1;

ymin = @(x) -sqrt(1 - x.^2);

ymax = @(x) sqrt(1 - x.^2);

zmin = @(x,y) -sqrt(1 - x.^2 - y.^2);

zmax = @(x,y) sqrt(1 - x.^2 - y.^2);

fun = @(x,y,z) x.\*cos(y) + x.^2.\*cos(z);

F = integral3(fun,xmin,xmax,ymin,ymax,zmin,zmax);

fprintf('integral F = %2.10e \n',F)

🡪 integral F = 7.7955545466e-01

**LINE INTEGRALS**

In electromagnetism the line integral is often encountered. A **line integral** is an expression of the form

  
where is a vector function, is an infinitesimal displacement vector and the integral is performed along a prescribed path from *a* to *b*. If the path is a closed loop (*a* = *b*) then we write



At each point on the path, we take the dot product of (evaluated at that point) with the displacement  to the next point.

There are some special vectors, that the value of the integral *F* is **independent** of the integration path.

**Length of a curve**

Consider a continuous curve on the interval [*a*, *b*]. Then the length *L* along of the curve is given by



**Example 8 Line integral of a scalar function cemCh3 (CELL 8)**

The line integral is



The function to be integrated has a singularity at the pint s = 1 + 2 I and thus a path must not pass through it.

Path 1: 0 🡪 2+0i 🡪 2+4i

integral F1 FR = 1.45410e-01 FI =1.85459e+00

integral F2 FR = -4.42859e+00 FI = 8.42859e+00

integral F = F1 + F2 = -4.2831853072e+00 FI = 1.02832e+01

Path 2: 0 🡪 1.5+I 🡪 2+4i

integral F1 FR = -1.74088e+00 FI =1.46830e+00

integral F2 FR = -2.54230e+00 FI = 8.81489e+00

integral F = F1 + F2 = -4.2831853072e+00 FI = 1.02832e+01

Path 3: 0 🡪 1+3i 🡪 2+4i

integral F1 FR = 4.74645e+00 FI =-3.96533e+00

integral F2 FR = 3.53673e+00 FI = 1.68214e+00

integral F = F1 + F2 = 8.2831853072e+00 FI = -2.28319e+00

Path 4: 0 🡪 1+4i 🡪 2+4i

integral F1 FR = 6.13275e+00 FI =-1.57903e+00

integral F2 FR = 2.15044e+00 FI = -7.04152e-01

integral F = F1 + F2 = 8.2831853072e+00 FI = -2.28319e+00

Note: Paths 1 and 2 give the same result as do paths 3 and 4 but the path lengths 3 and 4 are not equal to paths 1 and 2.

%% CELL 8

clear; close all; clc

N = 9999;

% PATH 1

a = 0; b = 1+4i;

s = linspace(a,b,N);

f = ( (s + 1)./(s-1-2\*1i) );

F1 = simpson1d(f,a,b);

a = b; b = 2 + 4i;

s = linspace(a,b,N);

f = ( (s + 1)./(s-1-2\*1i) );

F2 = simpson1d(f,a,b);

F = F1 + F2;

% Output to Commmand Window

disp(' ')

disp('')

fprintf('integral F1 FR = %2.5e FI =%2.5e \n',real(F1), imag(F1))

fprintf('integral F2 FR = %2.5e FI = %2.5e \n',real(F2), imag(F2))

fprintf('integral F = F1 + F2 = %2.10e FI = %2.5e

\n',real(F),imag(F))

**Example 9 Length of a semicircle cemCh3.m (CELL 9)**



The length of the semicircle of radius 1 is LN = 3.14159 ()

**Line integrals of vector fields**

The vector field is given by



and the [3D] smooth curve is given by



The line integral of the vector  along the path C is

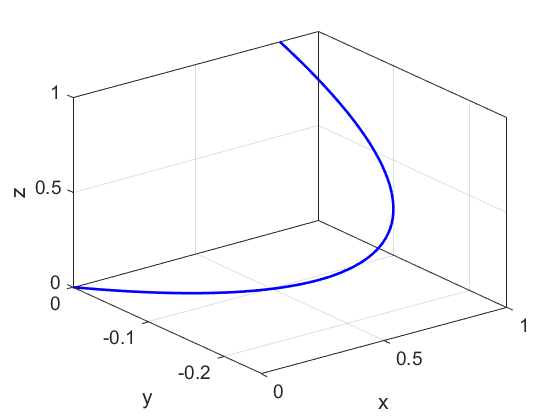


**Example 10 cemCh3.m (CELL 10)**

The vector is 

The curve is 

🡪 line integral L = -2.020078e-01



%% CELL 10

clear; close all; clc

tMin = 0; tMax = 1; N = 999;

t = linspace(tMin,tMax,N);

dt = t(2) - t(1);

x = sin(t); y = t.^2 - t; z = t;

% Vector function

V = [x.^2.\*y.\*z; x.\*y; 2.\*y.\*z];

r = [x; y; z];

% Smooth [3D] curve

rDash = gradient(r,dt);

% Curve gradient

VdotDash = dot(V,rDash);

% Line Integral

L = simpson1d(VdotDash, tMin, tMax);

% Output

fprintf('line integral L = %2.6e \n',L)

% GRAPHICS

figure(1)

plot3(x,y,z,'b','LineWidth',2)

xlabel('x'); ylabel('y'); zlabel('z');

grid on; box on

set(gca,'fontsize',14)

An important line integral is known as the **fundamental theorem of calculus** where the integral of the gradient along a path  from point *a* to point *b* does not depend upon the path from *a* to *b*.



The result is just the difference between *f* values at the ends, regardless of the path of integration.

First suppose that  is a continuous vector field in some domain

D, Then  is a **conservative vector field** if there is a function *f* such that



The function *f* is called a **potential function** for the vector field.

**DIVERGENCE (GAUSS’S or GREEN’S) THEOREM**

**Flux** describes any effect that appears to pass or travel (whether it actually moves or not) through a surface.

The flux  of a vector  is



The symbol implies the integration over a closed surface which encloses a volume. Is a  **unit vector** that is perpendicular to an area element *dA*.

The flux is a scalar. If  then there is a net flow out of the volume across the surface (source) and if  then there is a net flow into the volume across the surface (sink).

The **divergence theorem** (**Gauss’s or Green’s theorem**) is



where  is a volume element.

**Example 11** **cemCh3.m (CELL 11)**

Consider the vector function 

Calculate the surface and volume integrals over a unit cube located at the Origin. (Griffiths Example 1.10)

The surface and volume integrals can be computed using Matlab. There is a considerable amount of coding to find the answers to this problem. However, once the Script is written, it is easy to make changes to compute the integrals for other vector functions and bounded regions.

**Solution** *Inspect the Script to see how the problem was solved*

Results output to Command Window

divV = 2\*x + 2\*y

volume integral (numeric) FN = 2.020202

volume integral (symbolic) FS = 2.000000

surface integrals -0.333 0.333 -0.333 1.333 0.000 1.000

Surface Integral Stot = 2.000000

Note the error in the volume integral computed numerically (FN). The number of grid points used was N = 299. Increasing N improves the accuracy but greatly increases the computation time. You need to manually change the function for the variable divV after running the Script once and looking at the function in the Command Window for .

**Example 12** **cemCh3.m (CELL 12)**

Consider the vector function 

Calculate the surface and volume integrals over a cube of length 2 located at the Origin. (Griffiths Problem 1.32)

It is a bit of an effort to write the Script for Example 11, but it only takes a few minutes to change that Script to compute the answers to this problem.

**Solution**

divV = 3\*x + y + 2\*z

volume integral (numeric) FN = 48.484845

volume integral (symbolic) FS = 48.000000

surface integrals

0.000 8.000 0.000 16.000 0.000 24.000

Surface Integral Stot = 48.000000

**STOKES’ THEOREM**

 simple loop

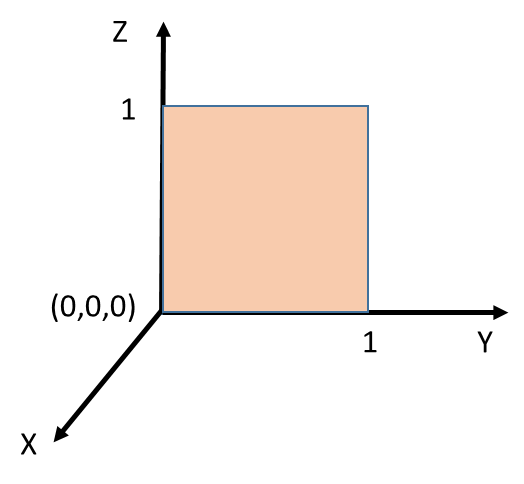
The line integral of the vector  about a closed loop is equal to the integral of the curl of  over the surface bounded by the loop.  the normal to an element of the surface.

**Example 13 cemStokes.m (CELL 1)**

The vector  is given by 

Calculate 

for the square surface shown. Check Stokes’ theorem.



Griffiths (Example 1.11)

**Solution** output to Command Window

Curl grad(V x n)

Dx = 4\*z^2 - 2\*x Dy = 0 Dz = 2\*z

Surface integral SA = 1.333333

Line integral: individual paths

1.000000 1.333333 -1.000000 -0.000000

Line integral SLtot = 1.333333

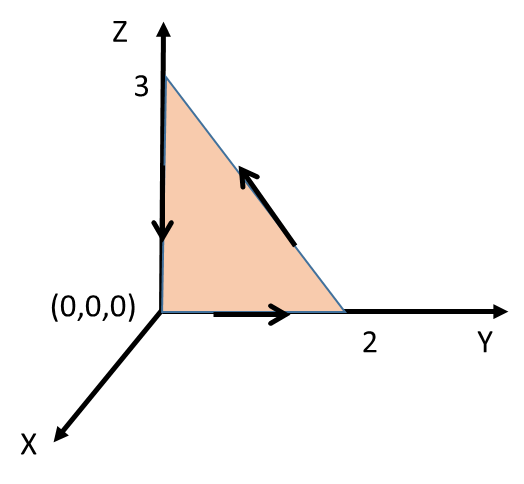
Check the Script to investigate the solution procedure. The parameters for the problem are entered in different sections of the Script. Both symbolic and numeric computations are used.

**Example 14 cemStokes.m (CELL 2)**

The vector  is given by 

Calculate 

for the triangular shown. Check Stokes’ theorem.



Griffiths (Problem 1.34)

**Solution** output to Command Window

Line integral: individual paths 0.000000 0.000000 -4.000000

Line integral SLtot = -4.000000

Curl grad(V x n)

Dx = -2\*y Dy = -3\*z Dz = -x

Surface integral SA = -4.020904

*Comment on Script*

 is computed for a rectangular surface as a NxN matrix curlVdotA. An area NxN matrix **AM** is defined such that each element is equal to 1 if the element corresponds to a grid point within the triangular area, otherwise 0.

nA = [1 0 0]; % surface unit vector

curlVdotA = curlx.\*nA(1) + curly.\*nA(2) + curlz.\*nA(3);

AM = ones(N,N);

for cy = 1:N

for cz = 1:N

if zz(cy,cz) > -(3/2)\*yy(cy,cz)+3; AM(cy,cz) = 0; end

end

end

M = AM.\*curlVdotA;