

Donnish Journal of Educational Research and Reviews. Vol 1(1) pp. 012-017 November, 2014.
http://www.donnishjournals.org/djerr
Copyright © 2014 Donnish Journals

Original Research Paper

Vector Analysis Using MAXIMA

Savaş Tuylu

Computer Science, Nigerian Turkish Nile University, Nigeria.

Accepted 3rd November, 2014.

The methods of calculus lie at the heart of the physical sciences and engineering. Maxima can help you make faster progress, if you are just learning calculus. The examples in this research paper will offer an opportunity to see some Maxima tools in the context of simple examples, but you will likely be thinking about much harder problems you want to solve as you see these tools used here. This research paper includes Vector Analysis.

Keywords: Vector Addition, Vector Subtraction, Dot Product of Vectors, The Cross product, Gradient, The Divergence, Curl, Laplacian Operator, Maxima.

1. INTRODUCTION

Maxima is a system for the manipulation of symbolic and numerical expressions, including differentiation, integration, Taylor series, Laplace transforms, ordinary differential equations, systems of linear equations, polynomials, sets, lists, vectors, matrices, tensors, and more. Maxima yields high precision numeric results by using exact fractions, arbitrary precision integers, and variable precision floating point numbers. Maxima can plot functions and data in two and three dimensions. Maxima source code can be compiled on many computer operating systems, including Windows, Linux, and MacOS X. The source code for all systems and precompiled binaries for Windows and Linux are available at the SourceForge file manager.

Maxima is a descendant of Macsyma, the legendary computer algebra system developed in the late 1960s at the Massachusetts Institute of Technology. It is the only system based on that effort still publicly available and with an active user community, thanks to its open source nature. Macsyma was revolutionary in its day, and many later systems, such as Maple and Mathematica, were inspired by it. The Maxima branch of Macsyma was maintained by William Schelter from 1982 until he passed away in 2001. In 1998 he obtained permission to release the source code under the GNU General Public License (GPL). It was his efforts and skills that made the survival of Maxima possible.

Vector Analysis provides Vector Addition and Subtraction, Dot Product of Vectors, The Cross product, Gradient, The Divergence, Curl, Laplacian Operator, a basic introduction to Vector, some functions and variables of vector operators using Maxima, definitions of Maxima's functions and variables of vector operators, solving functions and variables of vector operators examples using Maxima. MAXIMA has been constantly updated and used by researcher and engineers as well as by students.

2. VECTOR ANALYSIS

Vectors, represented in all three coordinate systems, i.e. cartesian, cylindrical and spherical, must be used in electromagnetics. The vectors are denoted by boldface symbols in order to distinguish vectors from scalars.

2.1. General

A vector field is a function having, at each instant in time, a magnitude at every point in space and a direction. Vector quantities have both magnitude and direction. They are represented graphically by arrows as shown in Figure 2.1. The length and arrowhead show the magnitude and direction of a vector, respectively.

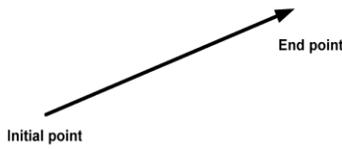


Figure 2.1: A vector

2.2. Vector Analysis Using Maxima

The vector algebra and calculus programs are stored in the package vect of MAXIMA Version 5.19.2 that provides to combine and simplify symbolic expressions of dot products, cross products, gradient, divergence, curl, and laplacian operators of vectors. The command load(vect) is used for vector analysis.

The following operators are defined by the vect package:

Table 2.2

Command	Description
vector1 . vector2	the dot product of vector1 and vector2
vector1 x vector2	the cross product of vector1 and vector2
grad expression n	the gradient of its argument, expression
curl expression	the curl of its argument, expression
laplacian expression	the laplacian of its argument, expression

Examples

```
(%i1) /* Load vector analysis package */
load(vect)$

/* Expression of dot product */
express([A[x],A[y],A[z]].[B[x],B[y],B[z]]);
(%o2) A_z B_z + A_y B_y + A_x B_x

(%i3) /* Expression of cross product */
express([A[x],A[y],A[z]]~[B[x],B[y],B[z]]);
(%o3) [A_y B_z - B_y A_z, B_x A_z - A_x B_z, A_x B_y - B_x A_y]

(%i4) /* Expression of divergence */
express(div([A[x],A[y],A[z]]));
(%o4) d/dz A_z + d/dy A_y + d/dx A_x

(%i5) /* Expression of gradient */
express(grad(f));
(%o5) [d/dx f, d/dy f, d/dz f]

(%i6) /* Expression of curl */
express(curl([A[x],A[y],A[z]]));
(%o6) [d/dy A_z - d/dz A_y, d/dz A_x - d/dx A_z, d/dx A_y - d/dy A_x]

(%i7) /* Expression of laplacian */
express(laplacian(f));
(%o7) d^2/dz^2 f + d^2/dy^2 f + d^2/dx^2 f
```

2.3. Vector Addition and Subtraction

The vector sum of A and B is defined in relation to the graphic sketch of the vectors, as shown in Fig. 2.3. The rule is to put the initial point of one vector to the end point of another vector.

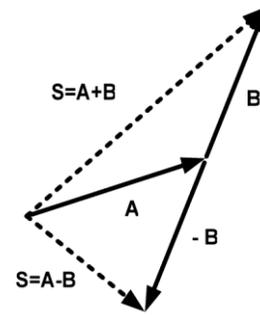


Figure 2.3. Vector addition and Subtraction

Vector addition and subtraction using Maxima

Example 2.3.1:

The vectors A and B are given in cartesian coordinate system as

$$\mathbf{A} = k\mathbf{x} + l\mathbf{y} \quad (2.3.1),$$

$$\mathbf{B} = l\mathbf{y} + k\mathbf{z} \quad (2.3.2).$$

Calculate the vector addition $\mathbf{A} + \mathbf{B}$ and vector subtraction $\mathbf{A} - \mathbf{B}$.

Solution:

```
(%i1) /* Load vector analysis package */
load(vect)$
/* Declare vectors A, B, and S */
declare([A,B,S],nonscalar);
/* Define vector components of A and B */
A:[k,l,0];
B:[0,l,k];
/* Vector addition */
S:A+B;
(%o2) done
(%o3) [k, l, 0]
(%o4) [0, l, k]
(%o5) [k, 2 l, k]

(%i6) /* Vector subtraction */
S:A-B;
(%o6) [k, 0, -k]
```

2.4. Dot Product of Vectors

The dot product of two vectors is the product of one vector which is aligned with another vector therefore it is obtained by the product of their magnitudes multiplied by the cosine of their angle ($0 \leq \theta \leq \pi$) between vectors as

$$\mathbf{A} \cdot \mathbf{B} = |\mathbf{A}| |\mathbf{B}| \cos(\theta_{AB}) \quad (2.4.1).$$

The result is scalar. The dot product of two vectors in cartesian coordinates

$$\mathbf{A} = A_x \mathbf{x} + A_y \mathbf{y} + A_z \mathbf{z} \quad (2.4.2),$$

$$\mathbf{B} = B_x \mathbf{x} + B_y \mathbf{y} + B_z \mathbf{z} \quad (2.4.3),$$

is

$$\mathbf{A} \cdot \mathbf{B} = A_x B_x + A_y B_y + A_z B_z \quad (2.4.4).$$

Same as above, the dot product of two vectors in spherical coordinates

$$\mathbf{A} = A_r \mathbf{r} + A_\theta \boldsymbol{\theta} + A_\phi \boldsymbol{\phi} \quad (2.4.5),$$

$$\mathbf{B} = B_r \mathbf{r} + B_\theta \boldsymbol{\theta} + B_\phi \boldsymbol{\phi} \quad (2.4.6),$$

is

$$\mathbf{A} \cdot \mathbf{B} = A_r B_r + A_\theta B_\theta + A_\phi B_\phi \quad (2.4.7).$$

Dot product using Maxima

Example 2.4.1:

Consider two vectors in cartesian coordinates

$$\mathbf{A} = A_x \mathbf{x} + A_y \mathbf{y} + A_z \mathbf{z}$$

$$\mathbf{B} = B_x \mathbf{x} + B_y \mathbf{y} + B_z \mathbf{z}$$

Calculate the dot product of \mathbf{A} and \mathbf{B} .

Solution:

```
(%i1) /* Load vector analysis package */
load(vect)$

/* Declare vectors A, B, and D */
declare([A,B,D],nonscalar);

(%o2) done
(%i3) /* Dot product of vectors A and B */
D:A.B;

(%o3) A . B
(%i4) express(A.B);
(%o4) A_z B_z + A_y B_y + A_x B_x
```

2.5. The Cross Product

The cross product of two vectors is obtained by the product of their magnitudes multiplied by the sine of their angle ($0 \leq \theta \leq \pi$) between vectors as given in Fig. 2.5.1,

$$\mathbf{A} \times \mathbf{B} = |\mathbf{A}| |\mathbf{B}| \sin(\theta_{AB}) \mathbf{c} \quad (2.5.1),$$

where \mathbf{c} is the unit vector which is normal to the plane determined by these vectors. The direction is obtained by using the right hand rule.

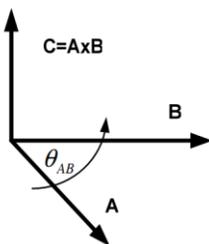


Figure 2.5.1: The Cross product

The cross product of two vectors in cartesian coordinates

$$\mathbf{A} = A_x \mathbf{x} + A_y \mathbf{y} + A_z \mathbf{z} \quad (2.5.2),$$

$$\mathbf{B} = B_x \mathbf{x} + B_y \mathbf{y} + B_z \mathbf{z} \quad (2.5.3),$$

$$\mathbf{A} \times \mathbf{B} = \begin{vmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix} = (A_y B_z - A_z B_y) \mathbf{x} + (A_z B_x - A_x B_z) \mathbf{y} + (A_x B_y - A_y B_x) \mathbf{z} \quad (2.5.4).$$

Cross product using Maxima

Example 2.5.1

Consider two vectors in cartesian coordinates

$$\mathbf{A} = A_x \mathbf{x} + A_y \mathbf{y} + A_z \mathbf{z}, \quad \text{and} \quad \mathbf{B} = B_x \mathbf{x} + B_y \mathbf{y} + B_z \mathbf{z},$$

calculate the cross product of \mathbf{A} and \mathbf{B} .

Solution:

```
(%i1) /* Load vector analysis package */
load(vect)$
/* Declare vectors A, B, and C */
declare([A,B,C],nonscalar);

/* Cross product of vectors A and B */
express(C:A~B);

(%o2) done
(%o3) [A_y B_z - B_y A_z, B_x A_z - A_x B_z, A_x B_y - B_x A_y]
```

2.6. Gradient

The gradient is the rate of change of a scalar function f . It is a vector in the direction of the greatest increase of function f . It is a slope. The ∇f points towards the larger values of given function. If there are three variables, the gradient f in cartesian, cylindrical, and spherical coordinate system, respectively, can be shown as

$$\nabla f = \frac{\partial f}{\partial x} \mathbf{x} + \frac{\partial f}{\partial y} \mathbf{y} + \frac{\partial f}{\partial z} \mathbf{z} \quad (2.6.1),$$

$$\nabla f = \frac{\partial f}{\partial r} \mathbf{r} + \frac{\partial f}{\partial \phi} \boldsymbol{\phi} + \frac{\partial f}{\partial z} \mathbf{z} \quad (2.6.2),$$

$$\nabla f = \frac{\partial f}{\partial r} \mathbf{r} + \frac{1}{r} \frac{\partial f}{\partial \theta} \boldsymbol{\theta} + \frac{1}{r \sin \theta} \frac{\partial f}{\partial \phi} \boldsymbol{\phi} \quad (2.6.3).$$

Gradient using MAXIMA

Example 2.6.1

Consider the scalar function f , calculate the gradient f in cartesian, cylindrical, and spherical coordinate system.

Solution:

```
(%i1) /* Load vector analysis package */
load(vect)$
/* Declare vectors G */
declare(G,nonscalar);

/* Gradient of scalar function f */
G:grad(f);

/* Express the components
in cartesian coordinate system */
express(G);
(%o2) done
(%o3) grad(f)
(%o4) [ $\frac{d}{dx}f, \frac{d}{dy}f, \frac{d}{dz}f$ ]
(%i5) /* r is assumed as positive */
assume(r>0);
/* Use scalefactors for cylindrical
coordinate transformation */
scalefactors([[r*cos(phi),r*sin(phi),z],
r,phi,z]);
(%o5) [r>0]
(%o6) done
(%i7) /* sin(theta) is positive for spherical
coordinate */
assume(sin(theta)>0);
/* Use scalefactors for spherical coordinate
transformation */
scalefactors([[r*sin(theta)*cos(phi),
r*sin(theta)*sin(phi),
r*cos(theta)],r,theta,phi]);
/* Express the components
in spherical coordinate system */
express(G);
(%o7) [sin(theta)>0]
(%o8) done
(%i9)
(%o9) [ $\frac{d}{dr}f, \frac{d}{d\theta}f, \frac{d}{d\phi}f$ ]
```

2.7. The Divergence

The divergence is an operator which shows the magnitude of the vector field's source at the given point. The divergence is scalar and bears a similarity to the derivative of a function. The divergence of a vector \mathbf{A} in cartesian, cylindrical, and spherical coordinate system, respectively, can be shown as

$$\nabla \mathbf{A} = \frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y} + \frac{\partial A_z}{\partial z}$$

(2.7.1),

$$\nabla \mathbf{A} = \frac{1}{r} \frac{\partial(rA_r)}{\partial r} + \frac{1}{r} \frac{\partial A_\phi}{\partial \phi} + \frac{\partial A_z}{\partial z}$$

(2.7.2),

$$\nabla \mathbf{A} = \frac{1}{r^2} \frac{\partial(r^2 A_r)}{\partial r} + \frac{1}{r \sin \theta} \frac{\partial(A_\theta \sin \theta)}{\partial \theta} + \frac{1}{r \sin \theta} \frac{\partial A_\phi}{\partial \phi} \quad (2.7.3).$$

Divergence using Maxima**Example 2.7.1**

Calculate the divergence of vector \mathbf{A} in cartesian, cylindrical, and spherical coordinate systems

Solution:

```
(%i1) /* Load vector analysis package */
load(vect)$
/* Declare vector A */
declare(A,nonscalar);

/* Divergence of A */
d:div(A);

/* Express the components
in cartesian coordinate system */
express(d);
(%o2) done
(%o3) div(A)
(%o4)  $\frac{d}{dz}A_z + \frac{d}{dy}A_y + \frac{d}{dx}A_x$ 
(%i5) /* r is assumed as positive */
assume(r>0);
/* Use scalefactors for cylindrical
coordinate transformation */
scalefactors([[r*cos(phi),r*sin(phi),z],
r,phi,z]);
/* Express the components
in cylindrical coordinate system */
express(d);
(%o5) [r>0]
(%o6) done
(%o7)  $\frac{d}{dz}(rA_z) + \frac{d}{dr}(rA_r) + \frac{d}{d\phi}A_\phi$ 
(%i8) /* sin(theta) is positive for spherical
coordinate */
assume(sin(theta)>0);
/* Use scalefactors for spherical
coordinate transformation */
scalefactors([[r*sin(theta)*cos(phi),
r*sin(theta)*sin(phi),
r*cos(theta)],r,theta,phi]);
/* Express the components
in spherical coordinate system */
express(d);
(%o8) [sin(theta)>0]
(%o9) done
(%o10)  $\frac{d}{d\theta}(rA_\theta \sin(\theta)) + \frac{d}{dr}(r^2 A_r \sin(\theta)) + \frac{d}{d\phi}(A_\phi r)}{r^2 \sin(\theta)}$ 
```

2.8. Curl

The curl is a vector operator which describes the rotation of a vector. The direction of the curl is the axis of the rotation. The curl of a vector \mathbf{A} in cartesian, cylindrical, and spherical coordinate system, respectively, can be shown as:

$$\nabla \times \mathbf{A} = \begin{vmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ A_x & A_y & A_z \end{vmatrix} = \left(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z} \right) \mathbf{x} + \left(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x} \right) \mathbf{y} + \left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y} \right) \mathbf{z} \quad (2.8.1),$$

$$\nabla \times \mathbf{A} = \begin{vmatrix} \frac{1}{r} \mathbf{r} & \phi & \frac{1}{r} \mathbf{z} \\ \frac{\partial}{\partial r} & \frac{\partial}{\partial \phi} & \frac{\partial}{\partial z} \\ A_r & r A_\phi & A_z \end{vmatrix} = \left(\frac{1}{r} \frac{\partial A_z}{\partial \phi} - \frac{\partial A_\phi}{\partial z} \right) \mathbf{r} + \left(\frac{\partial A_r}{\partial z} - \frac{\partial A_z}{\partial r} \right) \phi + \frac{1}{r} \left(\frac{\partial (r A_\phi)}{\partial r} - \frac{\partial A_r}{\partial \phi} \right) \mathbf{z} \quad (2.8.2),$$

$$\nabla \times \mathbf{A} = \frac{1}{r \sin \theta} \left(\frac{\partial (A_\phi \sin \theta)}{\partial \theta} - \frac{\partial A_\theta}{\partial \phi} \right) \mathbf{r} + \frac{1}{r} \left(\frac{1}{\sin \theta} \frac{\partial A_r}{\partial \phi} - \frac{\partial (r A_\phi)}{\partial r} \right) \theta + \frac{1}{r} \left(\frac{\partial (r A_\theta)}{\partial r} - \frac{\partial A_r}{\partial \theta} \right) \mathbf{z} \quad (2.8.3).$$

Curl using Maxima

Example 2.8.1

Calculate the curl of \mathbf{A} in cartesian, cylindrical, and spherical coordinate system.

Solution:

```
(%i1) /* Load vector analysis package */
load(vect)$

/* Declare vector A */
declare(A,nonscalar);

/* Curl of A */
C:=curl(A);

/* Express the components
in cartesian coordinate system */
express(C);
(%o2) done
(%o3) curl(A)
(%o4) [ $\frac{d}{dy} A_z - \frac{d}{dz} A_y$ ,  $\frac{d}{dz} A_x - \frac{d}{dx} A_z$ ,  $\frac{d}{dx} A_y - \frac{d}{dy} A_x$ ]
(%i5) /* r is assumed as positive */
assume(r>0);
/* Use scalefactors for cylindrical coordinate
transformation */
scalefactors([ [r*cos(phi), r*sin(phi), z], r, phi, z]);

/* Express the components
in cylindrical coordinate system */
express(C);
(%o5) [r>0]
(%o6) done
(%o7) [ $\frac{d}{d\phi} A_z - \frac{d}{dz} (A_\phi r)$ ,  $\frac{d}{dz} A_r - \frac{d}{dr} A_z$ ,  $\frac{d}{dr} (A_\phi r) - \frac{d}{d\phi} A_r$ ]
```

```
(%i8) /* sin(theta) is positive for spherical
coordinate */
assume(sin(theta)>0);
```

```
/* Use scalefactors for spherical coordinate
transformation */
scalefactors([ [r*sin(theta)*cos(phi),
r*sin(theta)*sin(phi),
r*cos(theta)], r, theta, phi]);

/* Express the components
in spherical coordinate system */
express(C);
```

```
(%o8) [sin(theta)>0]
```

```
(%o9) done
```

```
(%o10) [ $\frac{d}{d\theta} (A_\phi r \sin(\theta)) - \frac{d}{d\phi} (r A_\theta)$ ,  $\frac{d}{d\phi} A_r - \frac{d}{dr} (A_\phi r \sin(\theta))$ ,  $\frac{d}{dr} (r A_\theta) - \frac{d}{d\theta} A_r$ ]
r^2 sin(theta), r sin(theta), r
```

2.9. Laplacian Operator

The laplacian operator Δ is a second order differential operator, applied Nabla operator ∇ . It is the sum of all second partial derivatives in cartesian coordinates. The laplacian of vector \mathbf{A} in cartesian, cylindrical, and spherical coordinate system, respectively, can be shown as

$$\nabla^2 f = \nabla \nabla f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} \quad (2.9.1),$$

$$\nabla^2 f = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial f}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 f}{\partial \theta^2} + \frac{\partial^2 f}{\partial z^2} \quad (2.9.2),$$

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin^2 \phi} \frac{\partial^2 f}{\partial \theta^2} + \frac{1}{r^2 \sin^2 \phi} \frac{\partial}{\partial \phi} \left(\sin \phi \frac{\partial f}{\partial \phi} \right) \quad (2.9.3).$$

Laplacian using Maxima

Example 2.9.1

Consider the scalar function f , calculate the laplacian f in cartesian, cylindrical, and spherical coordinate system.

Solution:

```
(%i1) /* Load vector analysis package */
load(vect)$
```

```
/* Declare vector L */
declare(L,nonscalar);
```

```
/* Laplacian of scalar function f */
L:=laplacian(f);
```

```
/* Express the components
in cartesian coordinate system */
express(L);
```

```
(%o2) done
```

```
(%o3) laplacian(f)
```

```
(%o4)  $\frac{d^2}{dz^2} f + \frac{d^2}{dy^2} f + \frac{d^2}{dx^2} f$ 
```

```
--> /* Express the components in cylindrical coordinate system */
```

```
(%i1) load(vect);
assume (r>0);
scalefactors([[r*cos(phi), r*sin(phi), z], r, phi, z]);
L:laplacian(f(r, phi, z));
express(L);
expand(%);

(%o1)
C:/PROGRA~2/MAXIMA~1.0-2/share/maxima/5.28.0-2/share/vector/vect.ma
(%i2)
(%o2) [r>0]
(%i3)
(%o3) done
(%i4)
(%o4) laplacian(f(r, phi, z))


$$(\%o5) \frac{\frac{d}{dz} \left( r \left( \frac{d}{dz} f(r, \phi, z) \right) \right) + \frac{d}{dr} \left( r \left( \frac{d}{dr} f(r, \phi, z) \right) \right) + \frac{d}{d\phi} \frac{f(r, \phi, z)}{r}}{r}$$



$$(\%o6) \frac{\frac{d}{dz} \left( r \left( \frac{d}{dz} f(r, \phi, z) \right) \right)}{r} + \frac{\frac{d}{dr} \left( r \left( \frac{d}{dr} f(r, \phi, z) \right) \right)}{r} + \frac{\frac{d}{d\phi} \frac{f(r, \phi, z)}{r}}{r}$$


--> /* sin(theta) is positive for spherical coordinate */
assume(sin(theta)>0);

/* Use scalefactors for spherical coordinate
transformation */
scalefactors([[r*sin(theta)*cos(phi),
r*sin(theta)*sin(phi),
r*cos(theta)], r, theta, phi]);

/* Laplacian of scalar function f(r, theta, phi) */
L:laplacian(f(r, theta, phi));

/* Express the components
in spherical coordinate system */
express(L);

ev(% , diff);

(%i1) load(vect);
assume (sin(theta)>0);
scalefactors([[r*sin(theta)*cos(phi),
r*sin(theta)*sin(phi),
r*cos(theta)], r, theta, phi]);
L:laplacian(f(r, theta, phi));
express(L);
expand(%);

(%o1)
C:/PROGRA~2/MAXIMA~1.0-2/share/maxima/5.28.0-2/share/vector/vect.ma
(%i2)
(%o2) [sin(theta)>0]
(%o3) done
(%i4)
(%o4) laplacian(f(r, theta, phi))


$$(\%o5) \frac{\frac{d}{d\theta} \left( \left( \frac{d}{d\theta} f(r, \theta, \phi) \right) \sin(\theta) \right) + \frac{d}{dr} \left( r^2 \left( \frac{d}{dr} f(r, \theta, \phi) \right) \sin(\theta) \right) + \frac{d}{d\phi} \frac{f(r, \theta, \phi)}{\sin(\theta)}}{r^2 \sin(\theta)}$$



$$(\%o6) \frac{\frac{d}{d\theta} \left( \left( \frac{d}{d\theta} f(r, \theta, \phi) \right) \sin(\theta) \right)}{r^2 \sin(\theta)} + \frac{\frac{d}{dr} \left( r^2 \left( \frac{d}{dr} f(r, \theta, \phi) \right) \sin(\theta) \right)}{r^2 \sin(\theta)} + \frac{\frac{d}{d\phi} \frac{f(r, \theta, \phi)}{\sin(\theta)}}{r^2 \sin(\theta)}$$

```

CONCLUSION

Research paper can apply each and every part of Vectors, help application of the physical sciences and engineering, make faster progress, and help to understand Vectors. The paper particularly helps to understand parts of Calculus and is going to extend to other parts of the Calculus.

ACKNOWLEDGEMENTS

I would like to thank the Maxima developers, Nigerian Turkish Nile University, and Prof. Niyazi ARI for their friendly help.

REFERENCES

- [1] Niyazi ARI, Lecture notes, University of Technology, Zurich, Switzerland.
- [2] Niyazi ARI, Symbolic computation of electromagnetics with Maxima (2013).
- [3] <http://euler.rene-grothmann.de/reference/maximacore.html>.