

July 2018

The Commutator of the Runge Lenz Vector with other operators of the H-atoms physics using Maxima

Carl W. David

University of Connecticut, Carl.David@uconn.edu

Follow this and additional works at: https://opencommons.uconn.edu/chem_educ

 Part of the [Physical Chemistry Commons](#)

Recommended Citation

David, Carl W., "The Commutator of the Runge Lenz Vector with other operators of the H-atoms physics using Maxima" (2018). *Chemistry Education Materials*. 99.
https://opencommons.uconn.edu/chem_educ/99

The Commutator of the Runge Lenz Vector with other operators of the H-atoms physics using Maxima

C. W. David*
Department of Chemistry, Emeritus
University of Connecticut
Storrs, Connecticut 06269-3060

(Dated: July 19, 2018)

Commutators of components of the Runge-Lenz vector and constants of the motion and ladder operators for the H-atom problem are evaluated using brute force calculus and Maxima

I. INTRODUCTION

The arduous task of computing commutators of the Runge-Lenz (see Wikipedia for a good introduction to this topic) vector with other vectors (and scalars) in the quantum mechanics [1, 2] of the H-atom's electron preparatory to creating ladder operators can be obviated by using brute force methods employing symbolic calculus software.

II. PRELIMINARIES

Consider the H-atom's electron of charge e and mass m_e . Assuming the electron is located at \vec{r} at time t (relative to the nucleus), i.e., at

$$\vec{r} = x\hat{i} + y\hat{j} + z\hat{k}$$

(where $x = x(t)$, $y = y(t)$, and $z = z(t)$) then the (classical) velocity of the electron would be

$$\vec{v} = \dot{x}\hat{i} + \dot{y}\hat{j} + \dot{z}\hat{k} = \frac{d\vec{r}}{dt} = v_x\hat{i} + v_y\hat{j} + v_z\hat{k}$$

and a momentum vector

$$\vec{p} = mv_x\hat{i} + mv_y\hat{j} + mv_z\hat{k} = p_x\hat{i} + p_y\hat{j} + p_z\hat{k}$$

A. The classical angular momentum

The angular momentum is defined as the cross product of the radius vector on the momentum vector, i.e.,

$$\vec{L} = \vec{r} \otimes \vec{p} \quad (2.1)$$

aligned perpendicular to the plane of the two component vectors and employing a right hand rule. Alternatively, we have

$$\vec{r} \otimes \vec{p} \equiv \det \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ x & y & z \\ p_x & p_y & p_z \end{vmatrix} \quad (2.2)$$

which expands to

$$\vec{L} = \vec{r} \otimes \vec{p} = \hat{i}(yp_z - zp_y) + \hat{j}(zp_x - xp_z) + \hat{k}(xp_y - yp_x)$$

or

$$\vec{L} = \vec{r} \otimes \vec{p} = \hat{i}(L_x) + \hat{j}(L_y) + \hat{k}(L_z)$$

B. The Classical Runge Lenz Vector

The Classical (Laplace)-Runge-Lenz vector is defined as

$$\vec{A} = \vec{L} \otimes \vec{p} \equiv \det \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ L_x & L_y & L_z \\ p_x & p_y & p_z \end{vmatrix} \quad (2.3)$$

which expands to

$$\vec{A} = \vec{L} \otimes \vec{p} = \hat{i}(L_y p_z - L_z p_y) + \hat{j}(L_z p_x - L_x p_z) + \hat{k}(L_x p_y - L_y p_x)$$

or

$$\vec{A} = \vec{L} \otimes \vec{p} = \hat{i}(A_x) + \hat{j}(A_y) + \hat{k}(A_z)$$

III. QUANTUM MECHANICAL RUNGE LENZ OPERATOR

The Runge Lenz vector operator is

$$\vec{A} = \left\{ \frac{1}{Ze^2\mu} \right\} \left(\vec{L} \otimes \vec{p} - \vec{p} \otimes \vec{L} \right) + \frac{\vec{r}}{r} =$$

also formulated as

$$\vec{A} = \left\{ \frac{1}{Ze^2\mu} \right\} \left(\vec{L} \otimes \vec{p} - i\hbar\vec{p} \right) + \frac{\vec{r}}{r}$$

where, of course, $p_x = -i\hbar\frac{\partial}{\partial x}$, and similarly with y and z .

IV. THE X-COMPONENT OF THE COMMUTATOR OF \vec{A} AND \vec{L}

We need to form the compound operator $A_x L_x$ and its reverse form. The x-component of \vec{A} would be

$$A_x = \left\{ \frac{1}{Ze^2\mu} \right\} [(zp_x p_z - xp_z p_x) - (xp_y p_y - yp_z p_y) - i\hbar p_x] + \frac{x}{r}$$

and L_x , the x-component of \vec{L} would be

$$L_x = yp_z - zp_y$$

There are 24 terms in the "product" of these two, so handling it, even symbolically, is non trivial. Although this can be done using pencil-and-paper methods, there's got to be a better way (other than the symbolic method, which hides the mechanics of the various partial differentiations in a plethora of "sub"-commutators)!

A. Using Maxima

The brute force method is hard to carry out, but ultimately simpler than the above pencil-and-paper method.

As an example, we compute the commutator of the x components of the angular momentum and the Runge-Lenz vector. The Maxima code is shown in Appendix A.

We choose an expanded notation, rather than the most compact. That means that improvements in Maxima coding are self-evident, but ignored here (matters of elegance ...).

V. THE COMMUTATOR OF A_x WITH H , I.E., [A_x, H]

The commutator of the x-component of the Runge-Lenz vector with the Hamiltonian is significantly harder to evaluate symbolically, and almost impossible by the brute force method we emulate using Maxima.

The Maxima input file is shown in Appendix B.

VI. THE COMMUTATOR OF L_z WITH A_+

Our last attempt to use Maxima to generate commutators involves the up-ladder operator A_+ (and of course, the down operator would be similar) commuting with the

z-component of angular momentum (L_z). This is significantly harder to carry out.

We have

$$A_+ = A_x + iA_y$$

while $L_z = xp_y - yp_x$ so

$$[L_z, A_+] = L_z A_+ - A_+ L_z = L_z (A_x + iA_y) - (A_x + iA_y) L_z$$

The code is shown in Appendix C.

Since recognizing the printed output for commx is difficult at best, *test* has been used to get the expected result, i.e., zero comparing the Maxima output to A_+ .

VII. ACKNOWLEDGEMENTS AND COMMENTS

To verify commutator equations in an article in the American Journal of Physics concerning the Ladder Operator Solution to the Hydrogen Atom's electronic energy levels [2] this paper attempts replacing the hand-based computations with machine computations. The was material based on Pauli's article [3] which was a landmark contribution to the early development of Heisenberg mechanics.

At the time, I made what I now regard as a terrible choice, and accepted several equations concerning commutators that Pauli reported. I did not spend the time verifying that they were indeed correct.

Efforts to verify that commutators used in the H-atom ladder operator method for solving the quantum mechanics of the H-atom's electron require bookkeeping of lines and lines of equations which is both tedious and tiring[4, 5].

It is possible to use symbolic calculus programs such as Maxima to "brute force" evaluate commutators eliminating the problems of miscopying and/or mis-transcribing. Since Maxima is "free", it is a quite appealing tool for impoverished students.

Appendix A: Maxima code for [L_x, A_x]

```
load(physical_constants);
```

```

H: (px^2+py^2+pz^2) (2*mu)-Ze_sq/sqrt(x^2+y^2+z^2);
Lx: y*(-%i%h_bar*diff(f(x,y,z),z))-z*(-%i%h_bar*diff(f(x,y,z),y));
Ly: z*(-%i%h_bar*diff(f(x,y,z),x))-x*(-%i%h_bar*diff(f(x,y,z),z))$
Lz: x*(-%i%h_bar*diff(f(x,y,z),y))-y*(-%i%h_bar*diff(f(x,y,z),x))$
Ax: (1/(Z*%e^2*m_e))*(
/*Ly*/(z*(-%i%h_bar*diff((-%i%h_bar*diff(f(x,y,z),z)),x))-
x*(-%i%h_bar*diff((-%i%h_bar*diff(f(x,y,z),z)),z)))
-
/*Lz*/(x*(-%i%h_bar*diff((-%i%h_bar*diff(f(x,y,z),y)),y))-
y*(-%i%h_bar*diff((-%i%h_bar*diff(f(x,y,z),y)),x)))
-%i%h_bar*(diff(f(x,y,z),x))+x*f(x,y,z)/sqrt(x^2+y^2+z^2);

LxAx: substitute(f(x,y,z)=Ax,Lx);
AxLx: substitute(f(x,y,z)=Lx,Ax);
commx: expand(LxAx-AxLx);
commx: commx,diff,expand,factor;

```

Appendix B: Maxima code for $[L_x, H]$

```

/* [Ax,H] */
load(physical_constants);
/*H: (px^2+py^2+pz^2) (2*mu)-Ze_sq/sqrt(x^2+y^2+z^2);*/
px : -%i%h_bar*diff(f(x,y,z),x);
py : -%i%h_bar*diff(f(x,y,z),y);
pz : -%i%h_bar*diff(f(x,y,z),z);
H: ((-%i%h_bar*diff(-%i%h_bar*diff(f(x,y,z),x),x))+
(%i%h_bar*diff(-%i%h_bar*diff(f(x,y,z),y),y))+
(%i%h_bar*diff(-%i%h_bar*diff(f(x,y,z),z),z)))/(2*m_e)
-Z*%e^2*f(x,y,z)/sqrt(x^2+y^2+z^2);

Lx1: y*(pz);
Lx2 : -z*(py);
Ly1: z*(px);
Ly2: -x*(pz);
Lz1: x*(py);
Lz2: -y*(px);
Lx : Lx1+Lx2;
Ly : Ly1+Ly2;
Lz : Lz1+Lz2;
Ax1: substitute(f(x,y,z)=pz,Ly);
Ax2: substitute(f(x,y,z)=py,Lz);
Ax: (1/(Z*%e^2*m_e))*(
(Ax1-Ax2)
-%i%h_bar*(px)
)
+x*f(x,y,z)/sqrt(x^2+y^2+z^2);

AxH: substitute (f(x,y,z)=H,Ax)$
AxH: AxH,diff,expand,factor;

HAX: substitute(f(x,y,z)=Ax,H)$
HAX: HAX,diff,expand,factor;

commx: expand(HAX-AxH);

```

Appendix C: Maxima code for $[L_z, A_+]$

```

/* [Aplus,Lz] */
load(physical_constants);
/*H: (px^2+py^2+pz^2)(2*mu)-Ze_sq/sqrt(x^2+y^2+z^2);*/
px : -%i*%h_bar*diff(f(x,y,z),x);
py : -%i*%h_bar*diff(f(x,y,z),y);
pz : -%i*%h_bar*diff(f(x,y,z),z);
H:((-%i*%h_bar*diff(-%i*%h_bar*diff(f(x,y,z),x),x))+
    (-%i*%h_bar*diff(-%i*%h_bar*diff(f(x,y,z),y),y))+
    (-%i*%h_bar*diff(-%i*%h_bar*diff(f(x,y,z),z),z)))/(2*%m_e)
-Z*%e^2*f(x,y,z)/sqrt(x^2+y^2+z^2);

Lx1:y*(pz);
Lx2 :-z*(py);
Ly1:z*(px);
Ly2:-x*(pz);
Lz1:x*(py);
Lz2:-y*(px);
Lx : Lx1+Lx2;
Ly : Ly1+Ly2;
Lz : Lz1+Lz2;

Ay1:substitute(f(x,y,z)=px,Lz),diff,expand,factor$/*Lz cross px */
Ay2:substitute(f(x,y,z)=pz,Lx),diff,expand,factor$/*Lx cross pz */
Ay: (1/(Z*%e^2*%m_e))*(
    (Ay1-Ay2)
    -%i*%h_bar*(py)
    )
    +y*f(x,y,z)/sqrt(x^2+y^2+z^2),diff,expand,factor;

Ax1:substitute(f(x,y,z)=pz,Ly),diff,expand,factor;
Ax2:substitute(f(x,y,z)=py,Lz),diff,expand,factor;
Ax: (1/(Z*%e^2*%m_e))*(
    (Ax1-Ax2)
    -%i*%h_bar*(px)
    )
    +x*f(x,y,z)/sqrt(x^2+y^2+z^2),diff,expand,factor$

Aplus:(Ax+%i*Ay),diff,expand,factor;

AplusLz: expand(substitute (f(x,y,z)=expand(Lz),expand(Aplus)))$

LzAplus: expand(substitute(f(x,y,z)=expand(Aplus),expand(Lz)))$

commx:(LzAplus-AplusLz),diff,expand,factor;
commx:substitute(sqrt(x^2+y^2+z^2)=r,commx),diff,expand,factor;
commx:substitute(sqrt(x^2=r^2-(y^2+z^2)),commx),diff,expand,factor;
test:commx/%h_bar-Aplus,diff,expand,factor;
test:substitute(sqrt(x^2+y^2+z^2)=r,test),diff,expand,factor;

```

[1] S. M. Blinder, J. Chem. Ed. **78**, 391 (2001).[2] C. W. David, Am. J. Phys. **34**, 984 (1966).

[3] W. Pauli, *Zeit. f. Physik* **36**, 336 (1926).

[4] C. W. David (2006), http://digitalcommons.uconn.edu/chem_educ/14.

[5] C. W. David (2009), http://digitalcommons.uconn.edu/chem_educ/82.