電腦在工程數學上之應用
(Symbolic Computation in Engineering Mathematics)

Chapter 3
Vector and Tensor Analysis

陳俊杉

作者: 陳俊杉、游濟華

∇

Outline

- Introduction
- Basic of vector operator
- Differentiation of Vector
- Vector operator ∇

Unit I

- Vector integrals
- Green's Theorem, Stoke's Theorem and The Divergence Theorem

Unit II

- Orthogonal curvilinear coordinate

Unit III

- Tensor notation
- Tensor analysis in special coordinate

Unit IV

3.0 Introduction

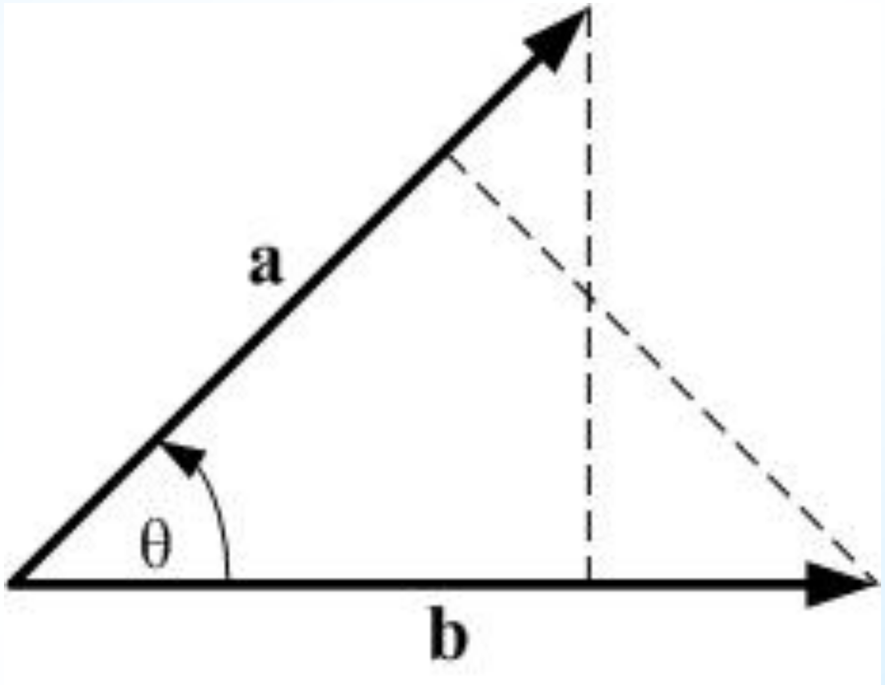
- **In this chapter, the basics of vector and tensor analysis will be introduced in conjunction with the utilization of Maxima in the associated problems.**
  - vector operation, differentiation of vectors, geometry of a space curve, the gradient vector and the vector operator ∇ (3.1-3.3)
  - line integral, surface integral, divergence theorem, Green's theorem, and Stoke's theorem (3.4-3.5)
  - orthogonal curvilinear coordinate (3.6-3.7)
  - tensor notation, basic tensor properties, tensor in orthogonal curvilinear coordinate (3.8-3.9)

3.1 Basics of Vector Operations

- **Inner product (dot product)**

$$\mathbf{a}\square\mathbf{b} = a_x b_x + a_y b_y + a_z b_z$$

$$= |\mathbf{a}||\mathbf{b}|\cos\theta$$

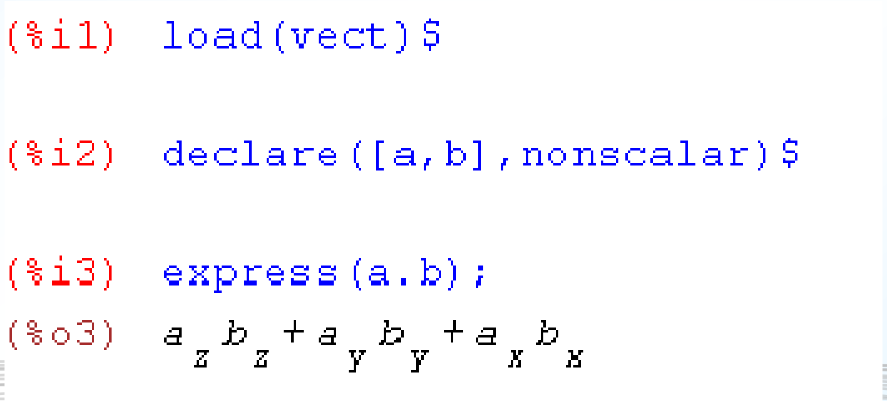


- **Geometric Interpretation**

What is $|\mathbf{a}|\cos\theta$?

3.1 Basics of Vector Operations

- Maxima code
  - (%i1) load(vect)$
  - (%i2) declare([a,b],nonscalar)$
  - (%i3) express(a.b)



```
(%i1) load(vect)$

(%i2) declare([a,b],nonscalar)$

(%i3) express(a.b);
```

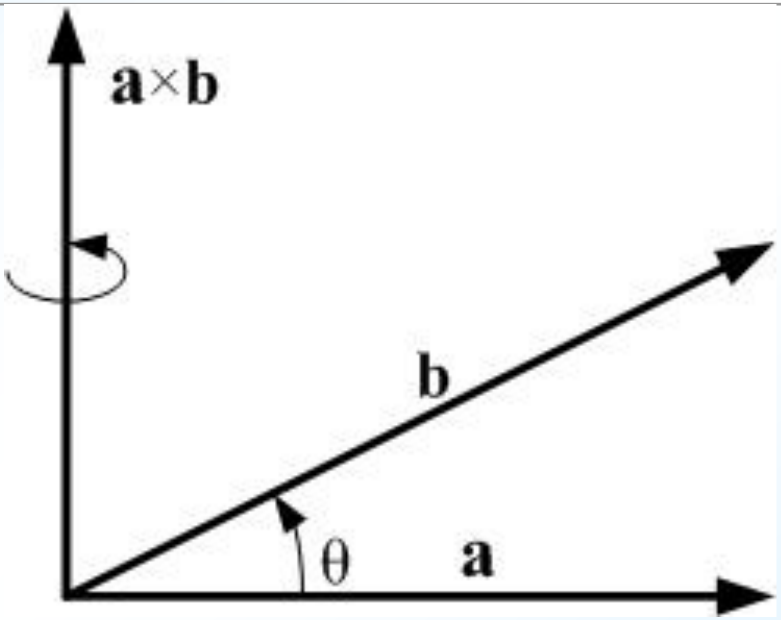
(%o3) $a_z b_z + a_y b_y + a_x b_x$

3.1 Basics of Vector Operations

- **Cross product**

$$\mathbf{a}\times\mathbf{b}=\begin{vmatrix}\mathbf{i} & \mathbf{j} & \mathbf{k}\\ a_x & a_y & a_z\\ b_x & b_y & b_z\end{vmatrix}$$

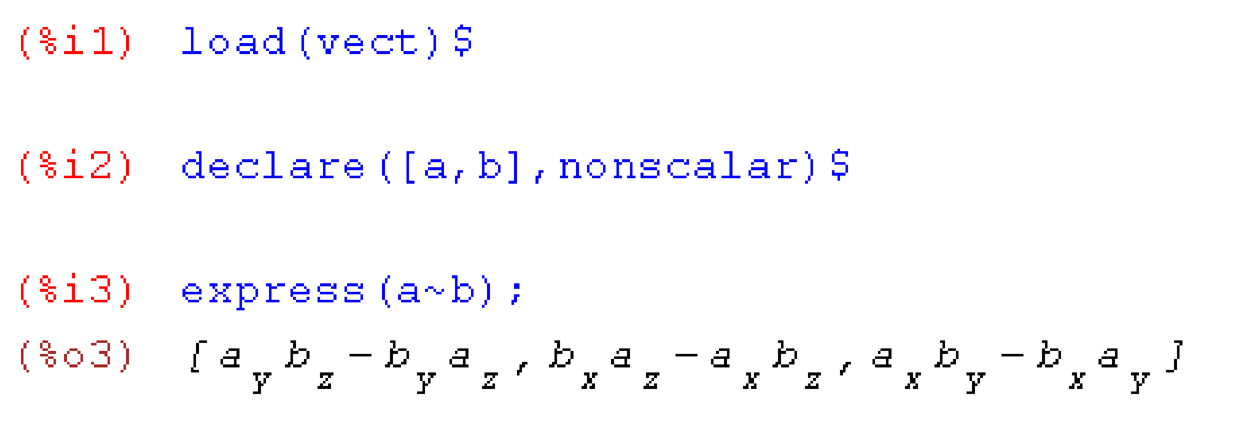
$$=|\mathbf{a}||\mathbf{b}|\sin\theta\hat{\mathbf{n}}$$



- **Geometric interpretation**
  - Magnitude? Direction?

3.1 Basics of Vector Operations

- Maxima code
  - (%i1) load(vect)$
  - (%i2) declare([a,b],nonscalar)$
  - (%i3) express(a~b)



```
(%i1) load(vect)$

(%i2) declare([a,b],nonscalar)$

(%i3) express(a~b);
```

(%o3) $[a_y b_z - b_y a_z, b_x a_z - a_x b_z, a_x b_y - b_x a_y]$

3.1 Basics of Vector Operations

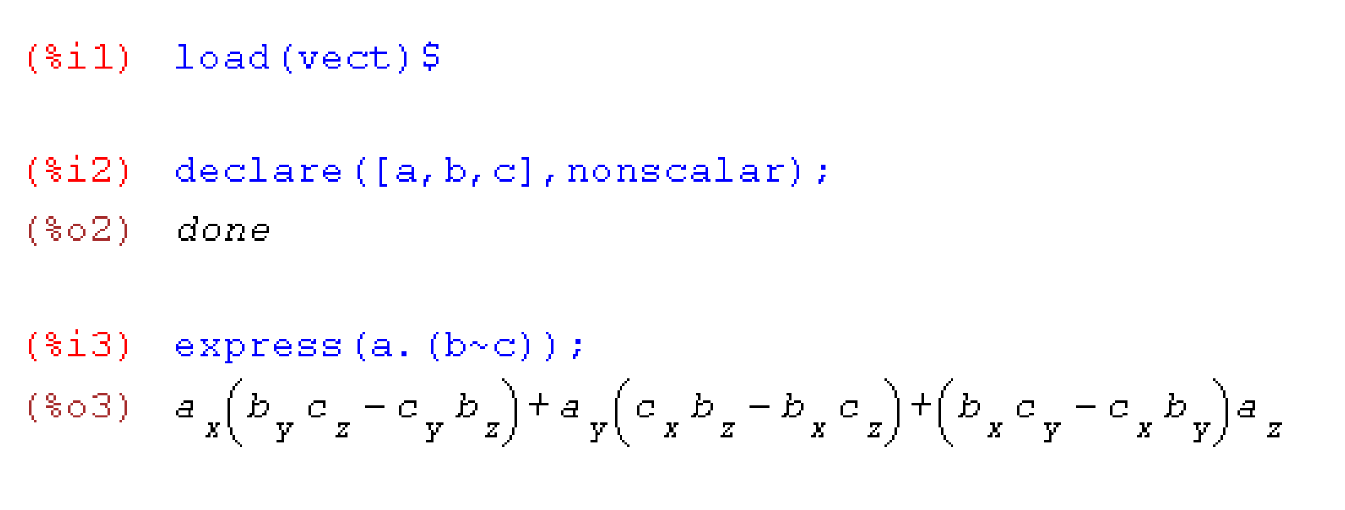
- **Multiple Products**
  - **Scalar triple product**

$$\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c}) = \begin{vmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{vmatrix}$$

  - **Geometric interpretation?**

3.1 Basics of Vector Operations

- Maxima code
  - (%i1) load(vect)$
  - (%i2) declare([a,b,c],nonscalar)
  - (%i3) express(a.(b~c))



```
(%i1) load(vect)$

(%i2) declare([a,b,c],nonscalar);
(%o2) done

(%i3) express(a.(b~c));
```

(%o3) $a_x\left(b_y c_z - c_y b_z\right) + a_y\left(c_x b_z - b_x c_z\right) + \left(b_x c_y - c_x b_y\right) a_z$

3.1 Basics of Vector Operations

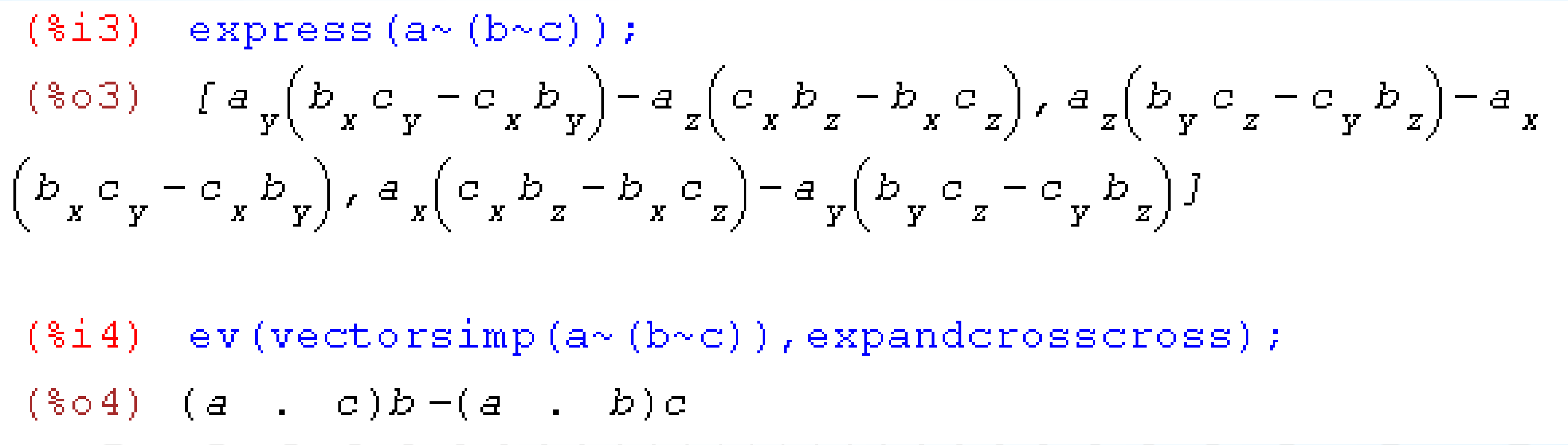
- **Multiple Products**
  - **Vector triple product**

$$\mathbf{a}\times(\mathbf{b}\times\mathbf{c})=\mathbf{b}(\mathbf{a}\cdot\mathbf{c})-\mathbf{c}(\mathbf{a}\cdot\mathbf{b})$$

  - **This is a very useful identity (also known as Lagrange's formula) involving the dot- and cross-products.**
  - **It is easier to remember as “BAC minus CAB”, keeping in mind which vectors are dotted together.**

3.1 Basics of Vector Operations

- Maxima code
  - (%i1) load(vect)$
  - (%i2) declare([a,b,c],nonscalar)$
  - (%i3) express(a~(b~c))
  - (%i4) ev(vectorsimp(a~(b~c),expandcrosscross)



```
(%i3) express(a~(b~c));
```

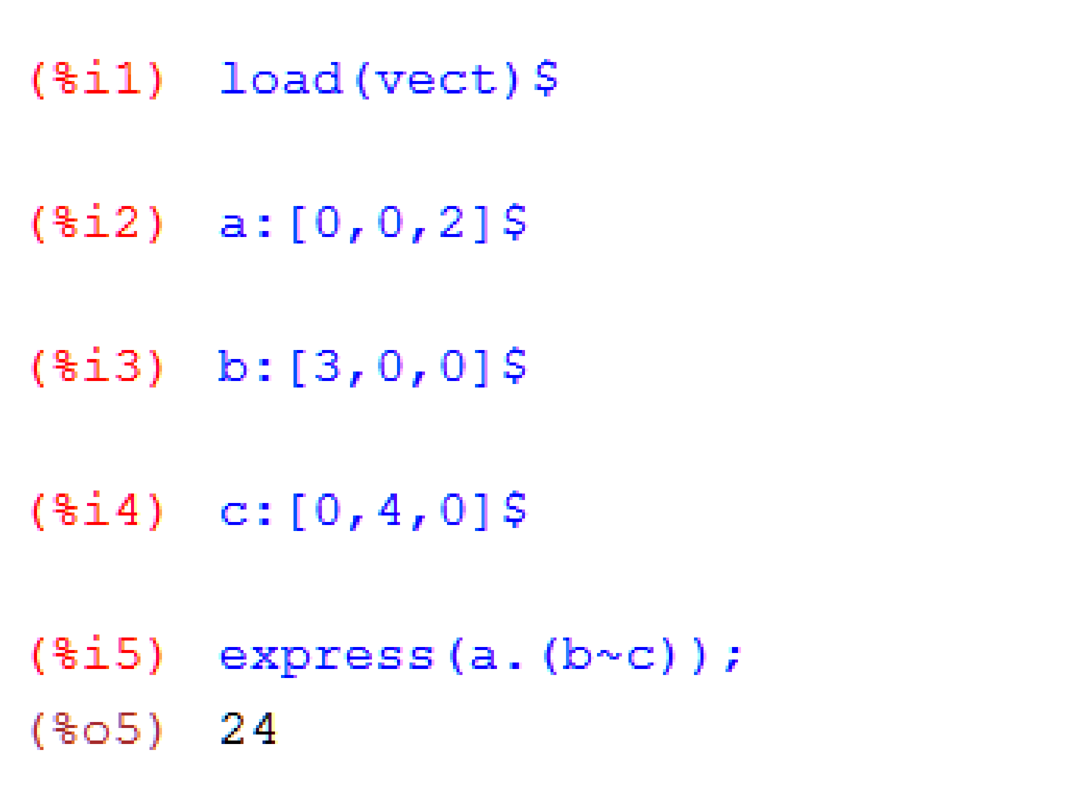
(%o3) $[a_y\left(b_x c_y - c_x b_y\right) - a_z\left(c_x b_z - b_x c_z\right), a_z\left(b_y c_z - c_y b_z\right) - a_x\left(b_x c_y - c_x b_y\right), a_x\left(c_x b_z - b_x c_z\right) - a_y\left(b_y c_z - c_y b_z\right)]$

```
(%i4) ev(vectorsimp(a~(b~c)),expandcrosscross);
```

(%o4) $(a\ .\ c)b-(a\ .\ b)c$

Example 3-1 (a)

- Find the volume of the parallelepiped spanned by the vectors **a** = (-2, 3, 1), **b** = (0, 4, 0), and **c** = (-1, 3, 3).



```
(%i1)  load(vect)$

(%i2)  a:[0,0,2]$

(%i3)  b:[3,0,0]$

(%i4)  c:[0,4,0]$

(%i5)  express(a.(b~c));
(%o5)  24
```

Example 3-1 (b)

- Find the volume of the tetrahedron with vertices at the points (0,0,0), (3,0,0),(0,4,0),and (0,0,2) .
  - (HINT) Volume of tetrahedron is ___ of parallelepiped

3.2 Differentiation of Vectors

- A vector **r** which is expressed as:

$$\mathbf{r} = x(t)\mathbf{i} + y(t)\mathbf{j} + z(t)\mathbf{k}$$

- Differentiation of **r** follows

$$\frac{d\mathbf{r}}{dt} = \frac{dx}{dt}\mathbf{i} + \frac{dy}{dt}\mathbf{j} + \frac{dz}{dt}\mathbf{k}$$

3.3 Vector Operator

- We now move to the consideration of properties of fields.

- Three field operators which reveal interesting collective field properties, viz.
  - the **gradient** of a scalar field,
  - the **divergence** of a vector field, and
  - the **curl** of a vector field.

3.3 Vector Operator ∇

- *vector operator (del)* ∇

$$\nabla = \frac{\partial}{\partial x}\mathbf{i} + \frac{\partial}{\partial y}\mathbf{j} + \frac{\partial}{\partial z}\mathbf{k}$$

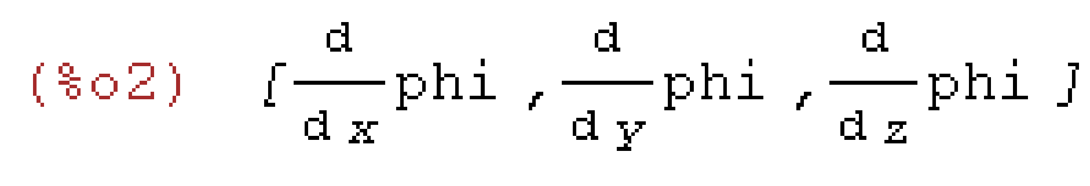
- ϕ is a scalar function of the coordinates *x*, *y* and *z.* The gradient (grad) of ϕ is a vector field pointing in the direction in which the derivative of ϕ is numerically greatest.

$$grad\phi = \nabla\phi = \frac{\partial\phi}{\partial x}\mathbf{i} + \frac{\partial\phi}{\partial y}\mathbf{j} + \frac{\partial\phi}{\partial z}\mathbf{k}$$

  - More Interpretation (see note)

3.3 Vector Operator ∇

- Maxima code
  - (%i1) load(vect)$
  - (%i2) express(grad(phi))



```
        d        d        d
(%o2) [--- phi, --- phi, --- phi]
       d x      d y      d z
```

3.3 Vector Operator ∇

- If **F** is a vector function of *x*, *y* and *z*

$$\mathbf{F} = (F_x, F_y, F_z)$$

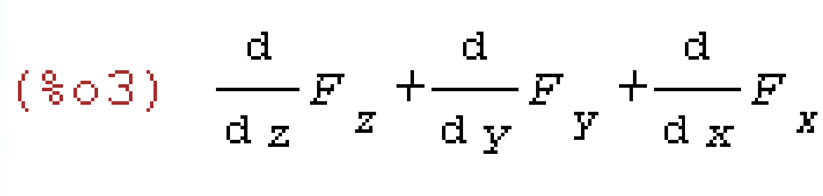
  - ***Divergence***: in vector calculus, the divergence is an operator that measures the magnitude of a vector field’ s source or sink at a given point; the divergence of a vector field is a (signed) scalar.

$$\nabla \cdot \mathbf{F} = \text{div}\ \mathbf{F} = \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} + \frac{\partial \mathbf{F}_z}{\partial z}$$

  - More interpretation (see note)

3.3 Vector Operator ∇

- Maxima code
  - (%i1) load(vect)$
  - (%i2) declare([F],nonscalar)
  - (%i3) express(div(F))



(%o3) $$\frac{d}{d\,z} F_z + \frac{d}{d\,y} F_y + \frac{d}{d\,x} F_x$$

3.3 Vector Operator ∇

- So far we have seen the del operator applied to a scalar field (what is it?); and dotted with a vector field (what is it?). Can we do cross of a vector field?
  - Yes and it is the curl.

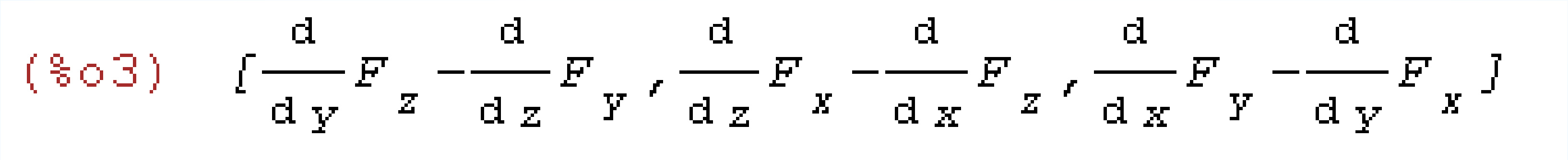
- Curl of **F**

$$\nabla \times \mathbf{F} = \mathrm{curl}\ \mathbf{F} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial z} \\ \mathbf{F}_x & \mathbf{F}_y & \mathbf{F}_z \end{vmatrix}$$

  - **More interpretation (see note)**

3.3 Vector Operator ∇

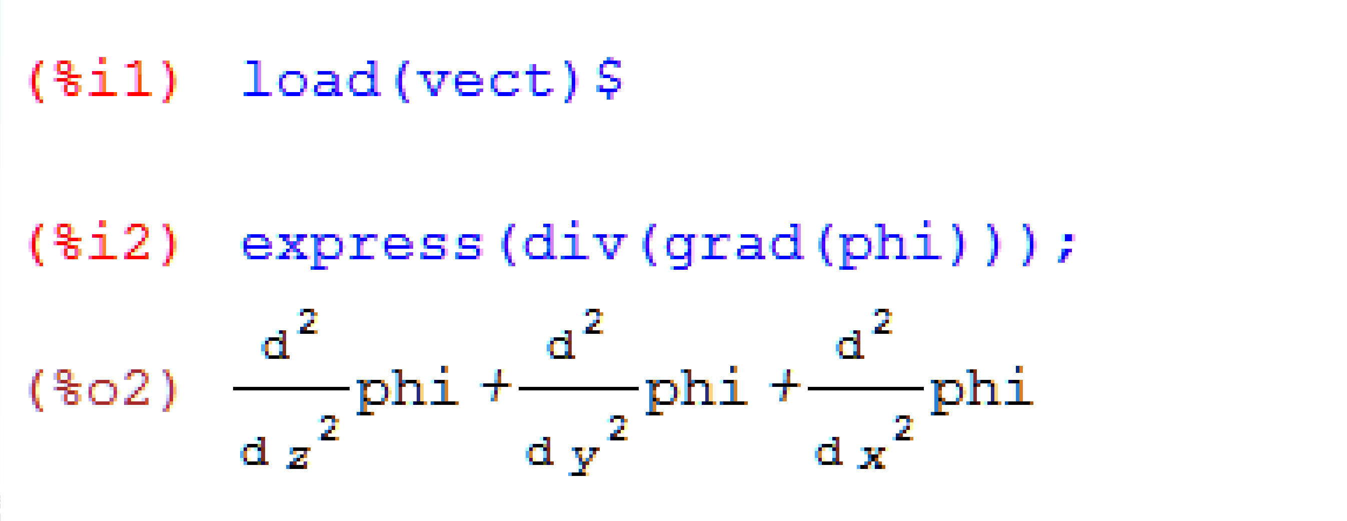
- Maxima code
  - (%i1) load(vect)$
  - (%i2) decalre([F],nonscalar)
  - (%i3) express(curl(F))



(%o3) $$\left[\frac{d}{d\,y} F_z - \frac{d}{d\,z} F_y , \frac{d}{d\,z} F_x - \frac{d}{d\,x} F_z , \frac{d}{d\,x} F_y - \frac{d}{d\,y} F_x\right]$$

Example 3-3 (a)

- Recall that grad of *any* scalar field is a vector field. Recall also that we can compute the divergence of any vector field. So we can certainly compute $\nabla\Box(\nabla\phi)$, even if we don't know what it means yet.



```
(%i1) load(vect)$

(%i2) express(div(grad(phi)));
         2          2          2
        d          d          d
(%o2)  ---- phi + ---- phi + ---- phi
          2          2          2
        dz         dy         dx
```

What is it?

Example 3-3 (b)

- Evaluate $\nabla \cdot (\mathbf{p} \times \mathbf{q})$ with Maxima

<Sol>

  - (%i1) load(vect);
  - (%i2) declare([p,q],nonscalar)$;
  - (%i3) express(div(p~q));