Nabla operator

Mauro Mongiardo[1]

[1] Department of Engineering, University of Perugia, Perugia, Italy.

Table of contents

Gradient

Gradient definition

Let us consider a scalar function $\phi(\mathbf{r})$, which is single valued and continuous in a volume V. Physically, the function ϕ may represent an electric potential, a temperature, etc.

At a point P the function will take the value ϕ_P.

Now suppose to draw a sphere of radius Δs centered in P. We can check the values of $\phi(\mathbf{r})$ on this sphere and, in general, there will be a point Q on which the variation $\Delta\phi = \phi_P - \phi_Q$ is maximum.

This defines also the direction from P to Q. This direction is taken as the direction of a new vector which is called the *gradient*.

Gradient definition

The magnitude of the gradient is is defined as the value $\Delta\phi/\Delta s$ in this preferred direction.

Thus the gradient of a scalar may be defined as

$$grad\phi = \nabla\phi = \mathbf{a}_{max} \lim_{\Delta s \to 0} \left(\frac{\Delta\phi}{\Delta s} \right)_{max} \tag{1}$$

where $\mathbf{a}_{max}$ is a unit vector pointing in the direction of maximum $\frac{\Delta\phi}{\Delta s}$.

It is noted that the definition (1) is not dependent on a a particular coordinate system.

Gradient in coordinates

In rectangular coordinates, (1) reduces to

$$\nabla\phi = \mathbf{a}_x \frac{\partial\phi}{\partial x} + \mathbf{a}_y \frac{\partial\phi}{\partial y} + \mathbf{a}_z \frac{\partial\phi}{\partial z}\,. \tag{2}$$

In circular–cylinder coordinates , Δs in the angular direction ψ is not equal to $\Delta\psi$ but $\delta s = \rho\Delta\psi$. Thus we have

$$\nabla\phi = \mathbf{a}_\rho \frac{\partial\phi}{\partial \rho} + \frac{\mathbf{a}_\psi}{\rho} \frac{\partial\phi}{\partial \psi} + \mathbf{a}_z \frac{\partial\phi}{\partial z}\,. \tag{3}$$

Similarly, in spherical coordinates,

$$\nabla\phi = \mathbf{a}_r \frac{\partial\phi}{\partial r} + \frac{\mathbf{a}_\psi}{r\sin\theta} \frac{\partial\phi}{\partial \psi} + \frac{\mathbf{a}_\theta}{r} \frac{\partial\phi}{\partial \theta}\,. \tag{4}$$

Divergence

Divergence

Divergence is a scalar function. Let us consider a field, at each point of which a vector **F** is specified. We can associate with each point P a scalar quantity, the divergence of **F**, defined as

$$div\, \mathbf{F} = \nabla \cdot \mathbf{F} = \lim_{\Delta\tau \to 0} \frac{1}{\Delta\tau} \oiint \mathbf{F} \cdot d\mathbf{s}\,. \tag{5}$$

The point P can be considered enclosed in a surface **s** of any shape, the volume within the surface being $\Delta\tau$. Take the total outward flux of the vector **F** through the bounding surface. The limit of the total outward flux, per unit volume, as the surface shrinks about P, is defined as div **F** at the point P.

In general, div **F** is different for each point in the field.

In rectangular coordinates the divergence take the following form

$$\nabla \cdot \mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}\,. \tag{6}$$

Equation (20) is often used as the definition of divergence. However, since this equation is only valid in rectangular coordinates, it is better to use (5) as the definition for the divergence.

In circular cylindrical coordinates the divergence has the following expression

$$\nabla \cdot \mathbf{F} = \frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho F_\rho\right) + \frac{1}{\rho}\frac{\partial F_\phi}{\partial \phi} + \frac{\partial F_z}{\partial z}\,. \tag{7}$$

For spherical coordinates,

$$\nabla \cdot \mathbf{F} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 F_r\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial \theta}\left(F_\theta \sin\theta\right) + \frac{1}{r\sin\theta}\frac{\partial F_\phi}{\partial \phi}\,. \tag{8}$$

Divergence evaluation

In many circumstances it is possible to evaluate the divergence without calculation.

Whenever physical intuition indicates that the flux entering $\Delta\tau$ is the same as the flux leaving it, the integral in (5) must be zero and consequently also the divergence is zero.

In electrostatics, for example, one visualizes electric flux lines between charges. At any point P in a region without charges, the integral in (5) must be zero, and we can say immediately that $\nabla \cdot \mathbf{D} = 0$ at such a point. Only if there is a charge distribution of density ρ (coulomb m^{-3}) at P we will have $\nabla \cdot \mathbf{D} = \rho$, i.e. a measure of the strength of the source at P.

Similarly, one may say immediately that for the magnetic field $\nabla \cdot \mathbf{B} = 0$ always since there are no magnetic charges and the flux lines invariably form closed loops.

Divergence theorem or Gauss theorem

Divergence theorem or Gauss theorem

Let $\mathbf{A}(\mathbf{r})$ be any vector function of position, continuous together with its first derivative throughout a volume V bounded by a surface S. The divergence theorem states that

$$\oint_S \mathbf{A}(\mathbf{r}) \cdot \mathbf{n} \, dS = \int_V \nabla \cdot \mathbf{A}(\mathbf{r}) \, dV \tag{9}$$

As a matter of fact the Gauss theorem is therefore used to define the divergence.

Curl

Curl

Consider an incremental element of area Δs and denote the unit normal to it by $\mathbf{u}_n$. It is possible to define the vector curl $\mathbf{F}$, represented symbolically as $\nabla \times \mathbf{F}$, as

$$curl\,\mathbf{F} = \nabla \times \mathbf{F} = \lim_{\Delta s \to 0} \frac{1}{\Delta s} \left[\mathbf{u}_n \oint \mathbf{F} \cdot d\mathbf{l} \right]_{max} . \qquad (10)$$

Note that in general we should repeat this procedure for three orthogonal surfaces and sum their contributions.

However, assuming assuming to know the direction of the curl (i.e. considering the surface such that the result is max), we can use the above expression.

Curl Expressions

In rectangular coordinates the curl takes the following form:

$$\nabla \times \mathbf{F} = \mathbf{u}_x \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} \right) + \mathbf{u}_y \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x} \right) + \mathbf{u}_z \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right) . \tag{11}$$

The above expressions can be easily memorized by forming a determinant with the versors on the first line, the partial derivatives on the second line and the vector itself on the third line.

For cylindrical coordinates, the following expression holds:

$$\begin{aligned} \nabla \times \mathbf{F} \quad = \quad & \mathbf{u}_\rho \left(\frac{1}{\rho} \frac{\partial F_z}{\partial \phi} - \frac{\partial F_\phi}{\partial z} \right) \\ & + \mathbf{u}_\phi \left(\frac{\partial F_\rho}{\partial z} - \frac{\partial F_z}{\partial \rho} \right) \\ & + \mathbf{u}_z \left[\frac{1}{\rho} \frac{\partial}{\partial \rho} \left(\rho F_\phi \right) - \frac{1}{\rho} \frac{\partial F_\rho}{\partial \phi} \right] \end{aligned} \tag{12}$$

For spherical coordinates:

$$\begin{aligned} \nabla \times \mathbf{F} \quad = \quad & \mathbf{u}_r \frac{1}{r \sin \theta} \left[\frac{\partial}{\partial \theta} \left(F_\phi \sin \theta \right) - \frac{\partial F_\theta}{\partial \phi} \right] \\ & + \mathbf{u}_\theta \frac{1}{r} \left[\frac{1}{\sin \theta} \frac{\partial F_r}{\partial \phi} - \frac{\partial}{\partial r} \left(r F_\theta \right) \right] \\ & + \mathbf{u}_\phi \frac{1}{r} \left[\frac{\partial}{\partial r} \left(r F_\theta \right) - \frac{\partial F_r}{\partial \theta} \right] \end{aligned} \tag{13}$$

Curl theorem or Stokes theorem

Curl theorem or Stokes theorem

Let $\mathbf{A}(\mathbf{r})$ be any vector function of position, continuous together with its first derivative throughout an arbitrary surface S bounded by a contour C, assumed to be resolvable into a finite number of regular arcs.

The Stokes theorem states that

$$\oint_C \mathbf{A}(\mathbf{r}) \cdot d\ell = \int_S [\nabla \times \mathbf{A}(\mathbf{r})] \cdot \mathbf{n} \, dS \tag{14}$$

where $d\ell$ is an element of length along C and $\mathbf{n}$ is a unit vector normal to the positive side of the element area dS.

This relationship is an equation defining the curl.

Scalar Laplacian

Scalar Laplacian

The scalar Laplacian is a second order operator, corresponding to the notation $\nabla \cdot \nabla w$ or div grad w, that is we have to take the divergence of the gradient of w

$$div\,(grad\;w) = \nabla^2 w = \nabla \cdot \nabla w \tag{15}$$

In the following are reported the Laplacian expressions in cartesian, circular-cylindrical and spherical coordinate systems, respectively.

$$\nabla^2 w = \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \tag{16}$$

$$\nabla^2 w = \frac{1}{\rho}\frac{\partial}{\partial\,\rho}\left(\rho\frac{\partial w}{\partial \rho}\right) + \frac{1}{\rho^2}\frac{\partial^2 w}{\partial \phi^2} + \frac{\partial^2 w}{\partial z^2} \tag{17}$$

$$\nabla^2 w = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial w}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial w}{\partial\theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 w}{\partial\phi^2} \tag{18}$$

Summary

Rectangular coordinates

$$\nabla w = \mathbf{u}_x \frac{\partial w}{\partial x} + \mathbf{u}_y \frac{\partial w}{\partial y} + \mathbf{u}_z \frac{\partial w}{\partial z} \tag{19}$$

$$\nabla \cdot \mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \tag{20}$$

$$\nabla \times \mathbf{F} = \mathbf{u}_x \left(\frac{\partial F_z}{\partial\, y} - \frac{\partial F_y}{\partial\, z} \right) + \mathbf{u}_y \left(\frac{\partial F_x}{\partial\, z} - \frac{\partial F_z}{\partial\, x} \right) + \mathbf{u}_z \left(\frac{\partial F_y}{\partial\, x} - \frac{\partial F_x}{\partial\, y} \right) \tag{21}$$

$$\nabla^2 w = \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \tag{22}$$

Cylindrical coordinates

$$\nabla w = \mathbf{u}_\rho \frac{\partial w}{\partial \rho} + \mathbf{u}_\phi \frac{1}{\rho} \frac{\partial w}{\partial \phi} + \mathbf{u}_z \frac{\partial w}{\partial z} \tag{23}$$

$$\nabla \cdot \mathbf{F} = \frac{1}{\rho} \frac{\partial}{\partial \rho} \left(\rho F_\rho \right) + \frac{1}{\rho} \frac{\partial F_\phi}{\partial \phi} + \frac{\partial F_z}{\partial z} \tag{24}$$

$$\begin{aligned} \nabla \times \mathbf{F} \quad = \quad & \mathbf{u}_\rho \left(\frac{1}{\rho} \frac{\partial F_z}{\partial \phi} - \frac{\partial F_\phi}{\partial z} \right) \\ & + \mathbf{u}_\phi \left(\frac{\partial F_\rho}{\partial z} - \frac{\partial F_z}{\partial \rho} \right) \\ & + \mathbf{u}_z \left[\frac{1}{\rho} \frac{\partial}{\partial \rho} \left(\rho F_\phi \right) - \frac{1}{\rho} \frac{\partial F_\rho}{\partial \phi} \right] \end{aligned} \tag{25}$$

$$\nabla^2 w = \frac{1}{\rho} \frac{\partial}{\partial \rho} \left(\rho \frac{\partial w}{\partial \rho} \right) + \frac{1}{\rho^2} \frac{\partial^2 w}{\partial \phi^2} + \frac{\partial^2 w}{\partial z^2} \tag{26}$$

Spherical coordinates

$$\nabla w = \mathbf{u}_r \frac{\partial w}{\partial r} + \mathbf{u}_\theta \frac{1}{r} \frac{\partial w}{\partial \theta} + \mathbf{u}_\phi \frac{1}{r \sin\theta} \frac{\partial w}{\partial \phi} \tag{27}$$

$$\nabla \cdot \mathbf{F} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 F_r \right) + \frac{1}{r \sin\theta} \frac{\partial}{\partial \theta} \left(F_\theta \sin\theta \right) + \frac{1}{r \sin\theta} \frac{\partial F_\phi}{\partial \phi} \tag{28}$$

$$\begin{aligned} \nabla \times \mathbf{F} \quad = \quad & \mathbf{u}_r \frac{1}{r \sin\theta} \left[\frac{\partial}{\partial \theta} \left(F_\phi \sin\theta \right) - \frac{\partial F_\theta}{\partial \phi} \right] \\ & + \mathbf{u}_\theta \frac{1}{r} \left[\frac{1}{\sin\theta} \frac{\partial F_r}{\partial \phi} - \frac{\partial}{\partial r} \left(r F_\phi \right) \right] \\ & + \mathbf{u}_\phi \frac{1}{r} \left[\frac{\partial}{\partial r} \left(r F_\theta \right) - \frac{\partial F_r}{\partial \theta} \right] \end{aligned} \tag{29}$$

$$\nabla^2 w = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial w}{\partial r} \right) + \frac{1}{r^2 \sin\theta} \frac{\partial}{\partial \theta} \left(\sin\theta \frac{\partial w}{\partial \theta} \right) + \frac{1}{r^2 \sin^2\theta} \frac{\partial^2 w}{\partial \phi^2} \tag{30}$$

Vector operation with CAS

Vector operation with CAS: Cartesian coordinates

A code for performing vector algebra with wxMaxima in cartesian coordinates is reported next. With small modifications the code can be used to compute divergence gradient curl etc. of specific functions.

```
/* [wxMaxima batch file version 1] [ DO NOT EDIT BY HAND! ]*/
/* [ Created with wxMaxima version 11.08.0 ] */

/* [wxMaxima: input   start ] */
kill(all);

/* vect_op_rectangular */

print("the package vect allows to perform vector operations")$
load(vect);

print("select the coordinate system by using the appropriate scalefactors")$

scalefactors(cartesian3d);
/*
scalefactors(polar);
scalefactors(spherical);
*/

print("Version for symbolic general expressions")$

/* give either the w expression or its dependence */

/* w : 1/sqrt(x^2 +z^2 +y^2);
example of function in rectangular coordinates */

depends(w,[x,y,z]); /*  */

/*  specific function
Fx : x/(x^2+y^2+z^2)^(3/2);
Fy : x*y*z + y/(x^2+y^2+z^2)^(3/2);
Fz : z/(x^2+y^2+z^2)^(3/2);
*/
 /* */
depends (Fx,[x,y,z]);
depends (Fy,[x,y,z]);
depends (Fz,[x,y,z]);

Fvect: matrix([Fx, Fy, Fz]);

print(" ==================================================")$
print(" RECTANGULAR COORDINATES")$
print(" ==================================================")$
print("Gradient of w")$
g : grad (w);
express (g);
ev (%, diff)$


print("Divergence")$
d : div ([Fx, Fy, Fz]);
express (d);
ev (%, diff)$

print("Curl")$
curl ([Fx, Fy, Fz]);
express (%);
ev (%, diff)$

print(" ==================================================")$
print("second order operators")$
print(" ==================================================")$
print("Expression of the Laplacian in rectangular coordinates")$
aa:laplacian w;
/* scalefactors(cartesian3d); */
express(aa);
ev (%, diff)$

print("laplacian is div(grad(w))")$

g : grad (w);
express (g);
gradw : ev (%, diff)$

d : div (gradw);
express (d);
ev (%, diff)$

print("Now evaluate curl curl of a vector")$
print("Curl")$
curl ([Fx, Fy, Fz]);
express (%);
curlF : ev (%, diff)$

curl(curlF);
express (%);
curlcurlF : ev (%, diff)$
/* [wxMaxima: input   end   ] */

/* Maxima can't load/batch files which end with a comment! */
"Created with wxMaxima"$

/* [wxMaxima batch file version 1] [ DO NOT EDIT BY HAND! ]*/
/* [ Created with wxMaxima version 11.08.0 ] */

/* [wxMaxima: input   start ] */
kill(all);
assume(r>0);     /* Arghh */
assume(%rho>0); /* Arghh */

print("the package vect allows to perform vector operations")$
load(vect);

print("select the coordinate system by using the appropriate scalefactors")$

print("Better to use phi since %phi is reserved")$

scalefactors([[%rho*cos(phi),%rho*sin(phi),z],%rho,phi,z]);
depends(w,[%rho,phi,z]);

/*  this allow to select the desired scalefactors
scalefactors(cartesian3d);  rectangular
scalefactors(polar);     cylindrical 2D
scalefactors([[%rho*cos(%phi),%rho*sin(%phi),z],%rho,%phi,z]);
cylindrical 3D
scalefactors(spherical);
*/

/* give either the w expression or its dependence */

w : log(%rho);  /* example of scalar function */
 /*  depends(t,[%rho,phi,z]);  */

/*  specific function
A[%rho] : 0 $
A[%phi] : 0$
A[z] : log(%rho)$

*/

/* set the first secon and third component */
F[x] : A[%rho];
F[y] : A[%phi];
F[z] : A[z];



/* depends(f,[r,theta,phi]) */

print("  Cylindrical COORDINATES")$
print("Gradient of w")$
g : grad (w);
express (g);
gradw : ev (%, diff);


print("Divergence")$
d : div ([F[x], F[y], F[z]]);
express (d);
ev (%, diff);

print("Curl")$
curl ([F[x], F[y], F[z]]);
express (%);
curl_F : ev (%, diff);

print("second order operators")$
print("Expression of the Laplacian in cylindrical coordinates")$
aa:laplacian w;
/* scalefactors(cartesian3d); */
express(aa);
ev (%, diff);

print("laplacian is div(grad(w))")$

g : grad (w);
express (g);
gradw : ev (%, diff);

d : div (gradw);
express (d);
ev (%, diff);

print("Now evaluate curl curl of a vector")$
print("Curl")$
curl ([F[x], F[y], F[z]]);
express (%);
curlF : ev (%, diff);

curl(curlF);
express (%);
curlcurlF : ev (%, diff);
print("----------------------------------------")$

print("curl of gradient is always zero")$
curl (gradw);
express (%);
curlgradw : ev (%, diff);

print("----------------------------------------")$
print("divergence of a curl is always zero");
d_curl_F : div (curl_F);
express (d);
ev (%, diff);

print("Bye")$
/* [wxMaxima: input   end   ] */

/* Maxima can't load/batch files which end with a comment! */
"Created with wxMaxima"$
```

Vector operation with CAS: Cylindrical coordinates

```
/* [wxMaxima batch file version 1] [ DO NOT EDIT BY HAND! ]*/
/* [ Created with wxMaxima version 11.08.0 ] */

/* [wxMaxima: input   start ] */
kill(all);
assume(r>0);     /* Arghh */
assume(%rho>0); /* Arghh */

print("the package vect allows to perform vector operations")$
load(vect);

print("select the coordinate system by using the appropriate scalefactors")$

print("Better to use phi since %phi is reserved")$

scalefactors([[%rho*cos(phi),%rho*sin(phi),z],%rho,phi,z]);
depends(w,[%rho,phi,z]);

/*  this allow to select the desired scalefactors
scalefactors(cartesian3d);  rectangular
scalefactors(polar);     cylindrical 2D
scalefactors([[%rho*cos(%phi),%rho*sin(%phi),z],%rho,%phi,z]);
cylindrical 3D
scalefactors(spherical);
*/

/* give either the w expression or its dependence */

w  : log(%rho);  /* example of scalar function */
 /*  depends(t,[%rho,phi,z]);  */

/*  specific function
A[%rho] : 0 $
A[%phi] : 0$
A[z] : log(%rho)$

*/

/* set the first secon and third component */
F[x] : A[%rho];
F[y] : A[%phi];
F[z] : A[z];



/* depends(f,[r,theta,phi]) */

print("  Cylindrical COORDINATES")$
print("Gradient of w")$
g : grad (w);
express (g);
gradw : ev (%, diff);


print("Divergence")$
d : div ([F[x], F[y], F[z]]);
express (d);
ev (%, diff);

print("Curl")$
curl ([F[x], F[y], F[z]]);
express (%);
curl_F : ev (%, diff);

print("second order operators")$
print("Expression of the Laplacian in cylindrical coordinates")$
aa:laplacian w;
/* scalefactors(cartesian3d); */
express(aa);
ev (%, diff);

print("laplacian is div(grad(w))")$

g : grad (w);
express (g);
gradw : ev (%, diff);

d : div (gradw);
express (d);
ev (%, diff);

print("Now evaluate curl curl of a vector")$
print("Curl")$
curl ([F[x], F[y], F[z]]);
express (%);
curlF : ev (%, diff);

curl(curlF);
express (%);
curlcurlF : ev (%, diff);
print("--------------------------------------------")$

print("curl of gradient is always zero")$
curl (gradw);
express (%);
curlgradw : ev (%, diff);

print("--------------------------------------------")$
print("divergence of a curl is always zero");
d_curl_F : div (curl_F);
express (d);
ev (%, diff);

print("Bye")$
/* [wxMaxima: input   end   ] */

/* Maxima can't load/batch files which end with a comment! */
"Created with wxMaxima"$
```

Vector operation with CAS: Spherical coordinates

```
/* [wxMaxima batch file version 1] [ DO NOT EDIT BY HAND! ]*/
/* [ Created with wxMaxima version 11.08.0 ] */

/* [wxMaxima: input   start ] */
kill(all);
assume(r>0);     /* Arghh */
assume(%rho>0); /* Arghh */

print("the package vect allows to perform vector operations")$
load(vect);

print("select the coordinate system by using the appropriate scalefactors")$

scalefactors(spherical);

/*  this allow to select the desired scalefactors
scalefactors(cartesian3d);
scalefactors(polar);
scalefactors(spherical);
*/

/* give either the w expression or its dependence */

  w : 1/r; /*   example of scalar function */
 /*  depends(w,[x,y,z]);  */

/*  specific function * %e^{-%i * k * r} */
A[r] : 1/r * cos(%theta)  $
A[%theta] : - 1/r * sin(%theta)  $
A[%phi] : 0$

A[%theta] : - 1/r * sin(%theta) *sin(phi)  $
A[%phi] : cos(phi) + sin(phi)$



/* set the first secon and third component */
F[x] : A[r];
F[y] : A[%theta];
F[z] : A[%phi];

Avect: matrix([A[r], F[y], F[z]]);

/* depends(f,[r,theta,phi]) */

print(" SPHERICAL COORDINATES")$
print("Gradient of w")$
g : grad (w);
express (g);
gradw : ev (%, diff);


print("Divergence")$
d : div ([F[x], F[y], F[z]]);
express (d);
ev (%, diff);

print("Curl")$
curl ([F[x], F[y], F[z]]);
express (%);
curl_F : ev (%, diff);

print("second order operators")$
print("Expression of the Laplacian in rectangular coordinates")$
aa:laplacian w;
/* scalefactors(cartesian3d); */
express(aa);
ev (%, diff);

print("laplacian is div(grad(w))")$

g : grad (w);
express (g);
gradw : ev (%, diff);

d : div (gradw);
express (d);
ev (%, diff);

print("Now evaluate curl curl of a vector")$
print("Curl")$
curl ([F[x], F[y], F[z]]);
express (%);
curlF : ev (%, diff);

curl(curlF);
express (%);
curlcurlF : ev (%, diff);
print("————————————————————————————————")$

print("curl of gradient is always zero")$
curl (gradw);
express (%);
curlgradw : ev (%, diff);

print("————————————————————————————————")$
print("divergence of a curl is always zero");
d_curl_F : div (curl_F);
express (d);
ev (%, diff);
/* [wxMaxima: input   end   ] */

/* Maxima can't load/batch files which end with a comment! */
"Created with wxMaxima"$
```