

Oct. 15 2007

Lecture 13: Differential Operations on Vectors

Reading:

Kreyszig Sections: 9.8, 9.9 (pages 410–413, 414–416)

Generalizing the Derivative

The number of different ideas, whether from physical science or other disciplines, that can be understood with reference to the “meaning” of a derivative from the calculus of scalar functions, is very very large. Our ideas about many topics, such as price elasticity, strain, stability, and optimization, are connected to our understanding of a derivative.

In vector calculus, there are generalizations to the derivative from basic calculus that act on a scalar and give another scalar back:

gradient (∇): A derivative on a scalar that gives a vector.

curl ($\nabla \times$): A derivative on a vector that gives another vector.

divergence ($\nabla \cdot$): A derivative on a vector that gives scalar.

Each of these have “meanings” that can be applied to a broad class of problems.

The gradient operation on $f(\vec{x}) = f(x, y, z) = f(x_1, x_2, x_3)$,

$$\text{grad} f = \nabla f \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) f \quad (13-1)$$

has been discussed previously. The curl and divergence will be discussed below.

Lecture 13 MATHEMATICA® Example 1

Scalar Potentials and their Gradient Fields

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](http://pruffle.mit.edu/3.016-2007) from <http://pruffle.mit.edu/3.016-2007>.

An example of a scalar potential, due three point charges in the plane, is visualized. Methods for computing a gradient are presented.

Simple 2 D 1/r potential

```
potential[x_, y_, xo_, yo_] :=  
-1/Sqrt[(x-xo)^2 + (y-yo)^2]
```

1

A field source located a distance 1 south of the origin

```
HoleSouth[x_, y_] :=  
potential[x, y, Cos[3 Pi / 2], Sin[3 Pi / 2]]
```

2

```
HoleNorthWest[x_, y_] :=  
potential[x, y, Cos[Pi / 6], Sin[Pi / 6]]
```

3

```
HoleNorthEast[x_, y_] :=  
potential[x, y, Cos[5 Pi / 6], Sin[5 Pi / 6]]
```

4

Function that returns the two dimensional (x,y) gradient field of any function declared a function of two arguments:

```
gradfield[scalarfunction_] :=  
{D[scalarfunction[x, y], x] // Simplify,  
D[scalarfunction[x, y], y] // Simplify}
```

5

Generalizing the function to any arguments:

```
gradfield[scalarfunction_, x_, y_] :=  
{D[scalarfunction[x, y], x] // Simplify,  
D[scalarfunction[x, y], y] // Simplify}
```

6

The sum of three potentials:

```
ThreeHolePotential[x_, y_] :=  
HoleSouth[x, y] +  
HoleNorthWest[x, y] + HoleNorthEast[x, y]
```

7

f(x,y) visualization of the scalar potential:

```
Plot3D[ThreeHolePotential[x, y],  
{x, -2, 2}, {y, -2, 2}]
```

8

Contour visualization of the three-hole potential

```
ContourPlot[ThreeHolePotential[x, y],  
{x, -2, 2}, {y, -2, 2}, PlotPoints -> 40,  
ColorFunction -> {Hue[1 - #* 0.66] &}]
```

9

1: This is the 2D 1/r-potential; here *potential* takes four arguments: two for the location of the charge and two for the position where the “test” charge “feels” the potential.

2-4: These are three fixed charge potentials, arranged at the vertices of an equilateral triangle.

5: *gradfield* is an example of a function that takes a scalar function of x and y and returns a vector with component derivatives: the gradient vector of the scalar function of x and y .

6: However, the previous example only works for functions of x and y explicitly. This expands *gradfield* to other Cartesian coordinates other than x and y .

7: *ThreeHolePotential* is the superposition of the three potentials defined in 2-4.

8: *Plot3D* is used to visualize the superposition of the potentials due to the three charges.

9: *ContourPlot* is an alternative method to visualize this scalar field. The option *ColorFunction* points to an example of a *Pure Function*—a method of making functions that do not operate with the usual “square brackets.” Pure functions are indicated with the *&* at the end; the *#* is a place-holder for the pure function’s argument.

Divergence and Its Interpretation

The divergence operates on a vector field that is a function of position, $\vec{v}(x, y, z) = \vec{v}(\vec{x}) = (v_1(\vec{x}), v_2(\vec{x}), v_3(\vec{x}))$, and returns a scalar that is a function of position. The scalar field is often called the divergence field of \vec{v} , or simply the divergence of \vec{v} .

$$\operatorname{div} \vec{v}(\vec{x}) = \nabla \cdot \vec{v} = \frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y} + \frac{\partial v_3}{\partial z} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) \cdot (v_1, v_2, v_3) = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) \cdot \vec{v} \quad (13-2)$$

Think about what the divergence means.

Lecture 13 MATHEMATICA® Example 2
 Visualizing the Gradient Field and its Divergence: The Laplacian

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](#) from <http://pruffle.mit.edu/3.016-2007>.

A visualization gradient field of the potential defined in the previous example is presented. The divergence of the gradient $\nabla \cdot \nabla \phi = \nabla^2 \phi$ (i.e., the result of the Laplacian operator ∇^2) is computed and visualized.

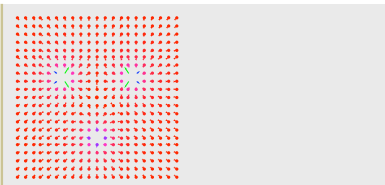
Gradient field of three-hole potential

```
gradthreehole = gradfield[ThreeHolePotential]
```

1

```
Needs["VectorFieldPlots"];
VectorFieldPlots`VectorFieldPlot[
gradthreehole, {x, -2, 2}, {y, -2, 2},
ScaleFactor -> 0.2, ColorFunction ->
(Hue[1 - #1 0.66] &), PlotPoints -> 21]
```

2



Function that takes a two-dimensional vector function of (x,y) as an argument and returns its divergence

```
divergence[{xcomp_, ycomp_}] :=
Simplify[D[xcomp, x] + D[ycomp, y]]
```

3

```
divgradthreehole = divergence[
gradfield[ThreeHolePotential]] // Simplify
```

4

Plotting the divergence of the gradient

($\nabla \cdot (\nabla f)$ is the "Laplacian" $\nabla^2 f$, sometimes indicated with symbol Δf)

```
Plot3D[divgradthreehole,
{x, -2, 2}, {y, -2, 2}, PlotPoints -> 60]
```

5

- 1: We use our previously defined function *gradfield* to compute the gradient of *ThreeHolePotential* everywhere in the plane.
- 2: `PlotVectorField` is in the `VectorFieldPlots` package. Because a gradient produces a vector field from a scalar potential, arrows are used at discrete points to visualize it.
- 3: The divergence operates on a vector and produces a scalar. Here, we define a function, *divergence*, that operates on a 2D-vector field of *x* and *y* and returns the sum of the component derivatives. Therefore, taking the divergence of the gradient of a scalar field returns a scalar field that is naturally associated with the original—its physical interpretation is (minus) the rate at which gradient vectors “diverge” from a point.
- 4–5: We compute the divergence of the gradient of the scalar potential. This is used to visualize the Laplacian field of *ThreeHolePotential*.

Coordinate Systems

The above definitions are for a Cartesian (x, y, z) system. Sometimes it is more convenient to work in other (spherical, cylindrical, etc) coordinate systems. In other coordinate systems, the derivative operations ∇ , $\nabla \cdot$, and $\nabla \times$ have different forms. These other forms can be derived, or looked up in a mathematical handbook, or specified by using the MATHEMATICA® package “VectorAnalysis.”

Lecture 13 MATHEMATICA® Example 3

Coordinate Transformations

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](#) from <http://pruffle.mit.edu/3.016-2007>.

Examples of *Coordinate Transformations* obtained from the `VectorAnalysis` package are presented.

It is no surprise that many of these differential operations already exist in Mathematica packages.

```
<< "VectorAnalysis"
```

1

Converting between coordinate systems

The spherical coordinates expressed in terms of the cartesian x,y,z

```
CoordinatesFromCartesian[
{x, y, z}, Spherical[r, theta, phi]]
```

2

$$\left\{ \sqrt{x^2 + y^2 + z^2}, \right. \\ \left. \text{ArcCos}\left[\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right], \text{ArcTan}[x, y] \right\}$$

The cartesian coordinates expressed in terms of the spherical r θ φ

```
CoordinatesToCartesian[
{r, theta, phi}, Spherical[r, theta, phi]]
```

3

```
{r Cos[phi] Sin[theta],
 r Sin[phi] Sin[theta], r Cos[theta]}
```

The equation of a line through the origin in spherical coordinates

```
Simplify[
CoordinatesFromCartesian[{a t, b t, c t},
Spherical[r, theta, phi]], t > 0]
```

4

1-2: `CoordinatesFromCartesian` from the `VectorAnalysis` package transforms three Cartesian coordinates, named in the first argument-list, into one of many coordinate systems named by the second argument.

3: `CoordinatesToCartesian` transforms one of many different coordinate systems, named in the second argument, into the three Cartesian coordinates, named in the first argument (which is a list).

4: For example, this would be the equation of a line radiating from the origin in spherical coordinates.

Lecture 13 MATHEMATICA® Example 4

Frivolous Example Using Geodesy, VectorAnalysis, and CityData.

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](http://pruffle.mit.edu/3.016-2007) from <http://pruffle.mit.edu/3.016-2007>.

We compute distances from Boston to Paris along different routes.

```
(The following will not work unless you have an active internet
connection)
CityData["Boston", "Latitude"]
CityData["Marseille", "Latitude"]
CityData["Paris", "Longitude"]
SphericalCoordinatesofCity[
  cityname_String] := {
  6378.1, CityData[cityname, "Latitude"]
  Degree,
  CityData[cityname, "Longitude"] Degree}
SphericalCoordinatesofCity["Boston"]
LatLong[city_String] :=
  {CityData[city, "Latitude"],
  CityData[city, "Longitude"]}
CartesianCoordinatesofCity[
  cityname_String] := CoordinatesToCartesian[
  SphericalCoordinatesofCity[cityname],
  Spherical[r, theta, phi]]
CartesianCoordinatesofCity["Paris"]
MinimumTunnel[city1_String, city2_String] :=
  Norm[CartesianCoordinatesofCity[city1] -
  CartesianCoordinatesofCity[city2]]
MinimumTunnel["Boston", "Paris"]
Needs["Geodesy"]
SphericalDistance[
  LatLong["Paris"], LatLong["Boston"]]
SpheroidalDistance[
  LatLong["Paris"], LatLong["Boston"]]
```

- 1–3:** `CityData` provides downloadable data. The data includes—among many other things—the latitude and longitude of many cities in the database. This show that Marseilles is north of Boston (which I found to be surprising).
- 4–5:** `SphericalCoordinatesofCity` takes the string-argument of a city name and uses `CityData` to compute its spherical coordinates (i.e., $(r_{\text{earth}}, \theta, \phi)$ are same as (average earth radius = 6378.1 km, latitude, longitude)). We use `Degree` which is numerically $\pi/180$.
- 6:** `LatLong` takes the string-argument of a city name and uses `CityData` to return a list-structure for its latitude and longitude. We will use this function below.
- 7–8:** `CartesianCoordinatesofCity` uses a coordinate transform and `SphericalCoordinatesofCity`
- 9–10:** If we imagine traveling *through* the earth instead of around it, we would use the `Norm` of the difference of the Cartesian coordinates of two cities.
- 11–12:** Comparing the great circle route using `SphericalDistance` (from the `Geodesy` package) to the Euclidean distance, is a result that surprises me. It would save only about 55 kilometers to dig a tunnel to Paris—sigh.
- 13:** `SpheroidalDistance` accounts for the earth's extra waistline for computing great-circle distances.

Lecture 13 MATHEMATICA® Example 5

Gradient and Divergence Operations in Other Coordinate Systems

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](http://pruffle.mit.edu/3.016-2007) from <http://pruffle.mit.edu/3.016-2007>.

A $1/r^n$ -potential is used to demonstrate how to obtain gradients and divergences in other coordinate systems.

```

SimplePot[x_, y_, z_, n_] :=
  1
  (x^2 + y^2 + z^2)^(n/2)
1
gradsp = Grad[
  SimplePot[x, y, z, 1], Cartesian[x, y, z]]
2
{
  -x
  (x^2 + y^2 + z^2)^(3/2)
  -y
  (x^2 + y^2 + z^2)^(3/2)
  -z
  (x^2 + y^2 + z^2)^(3/2)
}
The above is equal to ∇ / (|| r̄ ||)^3
3
SimplePot[r_, n_] := 1/r^n
4
gradsphere = Grad[SimplePot[r, 1], Spherical[r, θ, φ]]
5
Grad[SimplePot[r, 1], Cylindrical[r, θ, z]]
6
Grad[SimplePot[r, 1], ProlateSpheroidal[r, θ, φ]]
7
GradSimplePot[x_, y_, z_, n_] :=
  Evaluate[Grad[SimplePot[x, y, z, n],
  Cartesian[x, y, z]]]
8
Div[GradSimplePot[x, y, z, n],
  Cartesian[x, y, z]] // Simplify
9
Div[GradSimplePot[x, y, z, 1],
  Cartesian[x, y, z]] // Simplify
0

```

- 1: *SimplePot* is the simple $1/r^n$ potential in Cartesian coordinates.
- 2: `Grad` is defined in the `VectorAnalysis`: in this form it takes a scalar function and returns its gradient in the coordinate system defined by the second argument.
- 3: An alternate form of *SimplePot* is defined in terms of a single coordinate; if r is the spherical coordinate $r^2 = x^2 + y^2 + z^2$ (referring back to a Cartesian (x, y, z)), then this is equivalent the function in 1.
- 4: Here, the gradient of $1/r$ is obtained in spherical coordinates; it is equivalent to the gradient in 2, but in spherical coordinates.
- 5: Here, the gradient of $1/r$ is obtained in cylindrical coordinates, but it is not equivalent to 2 nor 4, because in cylindrical coordinates, (r, θ, z) , $r^2 = x^2 + y^2$, even though the form appears to be the same.
- 6: Here, the gradient of $1/r$ is obtained in prolate spheroidal coordinates.
- 7: We define a function for the x - y - z gradient of the $1/r^n$ scalar potential. `Evaluate` is used in the function definition, so that `Grad` is not called each time the function is used.
- 8: The Laplacian ($\nabla^2(1/r^n)$) has a particularly simple form, $n(n-1)/r^{2+n}$
- 9: By inspection of $\nabla^2(1/r^n)$ or by direct calculation, it follows that $\nabla^2(1/r)$ vanishes identically.

Curl and Its Interpretation

The curl is the vector-valued derivative of a vector function. As illustrated below, its operation can be geometrically interpreted as the rotation of a field about a point.

For a vector-valued function of (x, y, z) :

$$\vec{v}(x, y, z) = \vec{v}(\vec{x}) = (v_1(\vec{x}), v_2(\vec{x}), v_3(\vec{x})) = v_1(x, y, z)\hat{i} + v_2(x, y, z)\hat{j} + v_3(x, y, z)\hat{k} \quad (13-3)$$

the curl derivative operation is another vector defined by:

$$\text{curl } \vec{v} = \nabla \times \vec{v} = \left(\left(\frac{\partial v_3}{\partial y} - \frac{\partial v_2}{\partial z} \right), \left(\frac{\partial v_1}{\partial z} - \frac{\partial v_3}{\partial x} \right), \left(\frac{\partial v_2}{\partial x} - \frac{\partial v_1}{\partial y} \right) \right) \quad (13-4)$$

or with the memory-device:

$$\operatorname{curl} \vec{v} = \nabla \times \vec{v} = \det \begin{pmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ v_1 & v_2 & v_3 \end{pmatrix} \quad (13-5)$$

For an example, consider the vector function that is often used in Brakke's Surface Evolver program:

$$\vec{w} = \frac{z^n}{(x^2 + y^2)(x^2 + y^2 + z^2)^{\frac{n}{2}}} (y\hat{i} - x\hat{j}) \quad (13-6)$$

This will be shown below, in a MATHEMATICA® example, to have the property:

$$\nabla \times \vec{w} = \frac{nz^{n-1}}{(x^2 + y^2 + z^2)^{1+\frac{n}{2}}} (x\hat{i} + y\hat{j} + z\hat{k}) \quad (13-7)$$

which is spherically symmetric for $n = 1$ and convenient for turning surface integrals over a portion of a sphere, into a path-integral, over a curve, on a sphere.

Lecture 13 MATHEMATICA® Example 6

Computing and Visualizing Curl Fields

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](http://pruffle.mit.edu/3.016-2007) from <http://pruffle.mit.edu/3.016-2007>.

Examples of curls are computing for a particular family of vector fields. Visualization is produced with the `VectorFieldPlot3D` function from the `VectorFieldPlots` package.

```

LeavingKansas[x_, y_, z_, n_] :=
  (y, -x, 0)
  -----
  z^n
  (x^2 + y^2) (x^2 + y^2 + z^2)^(n/2)
Needs["VectorFieldPlots`"];
VectorFieldPlot3D[LeavingKansas[x, y, z, 3],
  {x, -1, 1}, {y, -1, 1},
  {z, -0.5, 0.5}, VectorHeads -> True,
  ColorFunction -> (Hue[#1 0.66] &),
  PlotPoints -> 21, ScaleFactor -> 0.5`]
VectorFieldPlot3D[
  LeavingKansas[x, y, z, 3], {x, 0, 1},
  {y, 0, 1}, {z, 0, 0, 0.5}, VectorHeads -> True,
  ColorFunction -> (Hue[#1 0.66] &),
  PlotPoints -> 15, ScaleFactor -> 0.5]
Curl[LeavingKansas[x, y, z, 3],
  Cartesian[x, y, z]] // Simplify
Glenda[x_, y_, z_, n_] :=
  Simplify[Curl[LeavingKansas[x, y, z, n],
  Cartesian[x, y, z]]]
VectorFieldPlot3D[
  Evaluate[Glenda[x, y, z, 1]],
  {x, -0.5, 0.5}, {y, -0.5, 0.5},
  {z, -0.25, 0.25}, VectorHeads -> True,
  ColorFunction -> (Hue[#1 0.66] &),
  PlotPoints -> 21]
Demonstrate that the divergence of the curl vanishes for the above
function independent of n
DivCurl =
Div[Glenda[x, y, z, n], Cartesian[x, y, z]]
Simplify[DivCurl]

```

- 1: *LeavingKansas* is the family of vector fields indicated by 13-6.
- 2-3: The function will be singular for $n > 1$ along the z - axis. This singularity will be reported during the numerical evaluations for visualization. There are two visualizations—the second one is over a sub-region but is equivalent because of the cylindrical symmetry of *LeavingKansas*. The singularity in the second case could be removed easily by excluding points near $z = 0$, but MATHEMATICA® seems to handle this fine without doing so.
- 4-6: This demonstrates the assertion, that for Eq. 13-7, the curl has cylindrical symmetry for arbitrary n , and spherical symmetry for $n = 1$.
- 7-8: This demonstrates that the divergence of the curl of \vec{w} vanishes for any n ; this is true for any differentiable vector field.

One important result that has physical implications is that the curl of a gradient is always zero:
 $f(\vec{x}) = f(x, y, z):$

$$\nabla \times (\nabla f) = 0 \quad (13-8)$$

Therefore if some vector function $\vec{F}(x, y, z) = (F_x, F_y, F_z)$ can be derived from a scalar potential, $\nabla f = \vec{F}$, then the curl of \vec{F} must be zero. This is the property of an exact differential $df = (\nabla f) \cdot$

$(dx, dy, dz) = \vec{F} \cdot (dx, dy, dz)$. Maxwell's relations follow from equation 13-8:

$$\begin{aligned} 0 &= \frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} = \frac{\partial \frac{\partial f}{\partial z}}{\partial y} - \frac{\partial \frac{\partial f}{\partial y}}{\partial z} = \frac{\partial^2 f}{\partial z \partial y} - \frac{\partial^2 f}{\partial y \partial z} \\ 0 &= \frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x} = \frac{\partial \frac{\partial f}{\partial x}}{\partial z} - \frac{\partial \frac{\partial f}{\partial z}}{\partial x} = \frac{\partial^2 f}{\partial x \partial z} - \frac{\partial^2 f}{\partial z \partial x} \\ 0 &= \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} = \frac{\partial \frac{\partial f}{\partial y}}{\partial x} - \frac{\partial \frac{\partial f}{\partial x}}{\partial y} = \frac{\partial^2 f}{\partial y \partial x} - \frac{\partial^2 f}{\partial x \partial y} \end{aligned} \quad (13-9)$$

Another interpretation is that gradient fields are *curl-free*, *irrotational*, or *conservative*.

The notion of “conservative” means that, if a vector function can be derived as the gradient of a scalar potential, then integrals of the vector function over any path is zero for a closed curve—meaning that there is no change in “state;” energy is a common state function.

Here is a picture that helps visualize why the curl invokes names associated with spinning, rotation, etc.

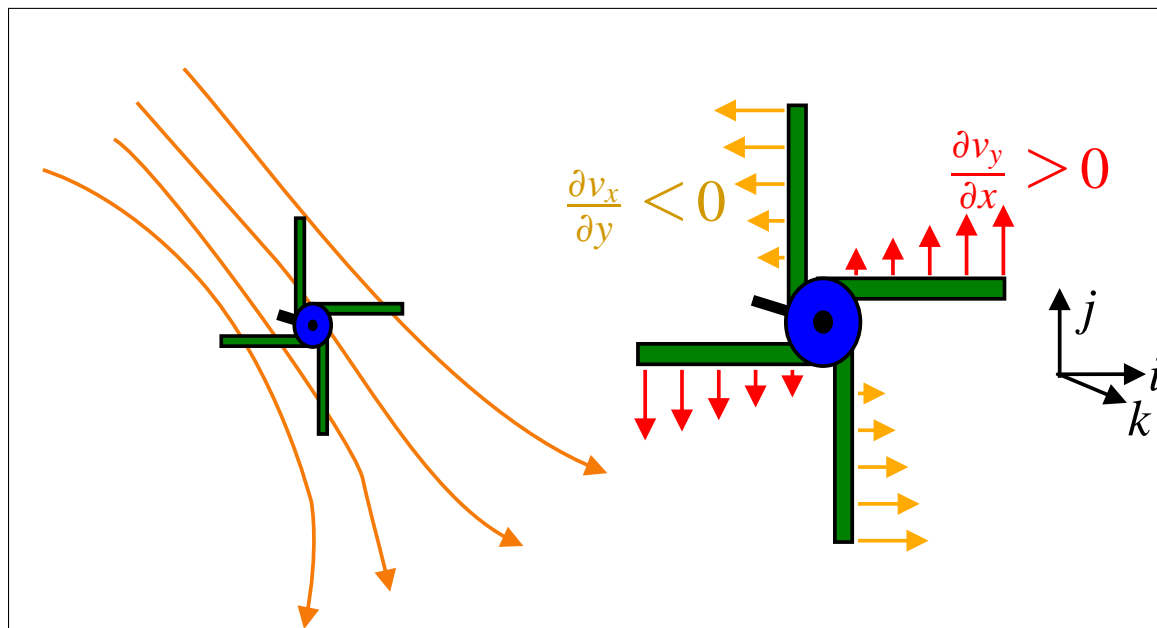


Figure 13-10: Consider a small paddle wheel placed in a set of stream lines defined by a vector field of position. If the v_y component is an increasing function of x , this tends to make the paddle wheel want to spin (positive, counter-clockwise) about the \hat{k} -axis. If the v_x component is a decreasing function of y , this tends to make the paddle wheel want to spin (positive, counter-clockwise) about the \hat{k} -axis. The net impulse to spin around the \hat{k} -axis is the sum of the two.

Note that this is independent of the reference frame because a constant velocity $\vec{v} = \text{const.}$ and the local acceleration $\vec{v} = \nabla f$ can be subtracted because of Eq. 13-10.

Another important result is that divergence of any curl is also zero, for $\vec{v}(\vec{x}) = \vec{v}(x, y, z)$:

$$\nabla \cdot (\nabla \times \vec{v}) = 0 \quad (13-10)$$