Lecture 13: Differential Operations on Vectors

Reading:
Kreyszig Sections: 9.8, 9.9 (pages410–413, 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 acts on a scalar and gives 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 $f(\vec{x}) = f(x, y, z) = f(x_1, x_2, x_3)$,

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

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

Lecture 13 MATHEMATICA® Example 1

Gradients and Laplacians on Scalar Potentials

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

An example of a scalar potential due three point charges in the plane is visualized. Methods for computing a gradient and the divergence of a gradient (Laplacian) are presented.

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.

4: This is the third of 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…

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.

8: `Plot3D` is used to visualize the superposition of the three charge potentials defined as *ThreeHolePotential* .

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.

12: `PlotVectorField` is in the `Graphics‘PlotField‘` package. Because a gradient produces a vector field from a scalar potential, arrows are used at discrete points to visualize it.

14: The divergence operates on a vector and produces a scalar. 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.

```
1  potential[x_ , y_, xo_ , yo_] :=
     -1/Sqrt[(x - xo)^2 + (y - yo)^2]
2  HoleSouth[x_, y_] := potential[x, y, Cos[3 Pi/2], Sin[3 Pi/2]]
3  HoleNorthWest[x_ , y_] :=
     potential[x, y, Cos[Pi/6], Sin[ Pi/6]]
4  HoleNorthEast[x_ , y_] :=
     potential[x, y, Cos[5 Pi/6], Sin[5 Pi/6]]
5  gradfield[scalarfunction_] :=
     {D[scalarfunction[x, y], x] // Simplify,
       D[scalarfunction[x, y], y] // Simplify}
6  gradfield[scalarfunction_, x_ , y_] :=
     {D[scalarfunction[x, y], x] // Simplify,
       D[scalarfunction[x, y], y] // Simplify}
7  ThreeHolePotential[x_, y_] := HoleSouth[x, y] +
       HoleNorthWest[x, y] + HoleNorthEast[x, y]
8  Plot3D[ThreeHolePotential[x, y], {x, -2, 2}, {y, -2, 2}]
9  ContourPlot[ThreeHolePotential[x, y], {x, -2, 2}, {y, -2, 2},
     PlotPoints → 40, ColorFunction → (Hue[1 - # * 0.66] &)]
10 gradthreehole = gradfield[ThreeHolePotential]
11 << Graphics`PlotField`
12 PlotVectorField[gradthreehole,
     {x, -2, 2}, {y, -2, 2}, ScaleFactor → 0.2,
     ColorFunction → (Hue[1 - # * 0.66] &), PlotPoints → 21]
13 divergence[{xcomp_ , ycomp_}] :=
     Simplify[D[xcomp, x] + D[ycomp, y]]
14 divgradthreehole =
     divergence[gradfield[ThreeHolePotential]] // Simplify
15 Plot3D[divgradthreehole,
     {x, -2, 2}, {y, -2, 2}, PlotPoints -> 60]
```

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 $\vec{v}$.

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

Think about what the divergence means,

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 2

Coordinate Transformations

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

Examples of *Coordinate Transformations* obtained from the `Calculus‘VectorAnalysis‘` package. An frivolous example of computing distances from Boston to Paris along different routes using data from the `Miscellaneous‘CityData‘` package.

```
1  << Calculus`VectorAnalysis`
   Converting between coordinate systems
2  CoordinatesFromCartesian[{x, y, z}, Spherical[r, theta, phi]]
3  CoordinatesToCartesian[{r, theta, phi}, Spherical[r, theta, phi]]
4  Simplify[CoordinatesFromCartesian[
      {a t, b t, c t}, Spherical[r, theta, phi]], t > 0]
   An example of calculating the positions of cities in cartesian
   and spherical coordinates.
5  << Miscellaneous`CityData`
6  boston = CityData["Boston", CityPosition]
7  paris = CityData["Paris", CityPosition]
8  SphericalCoordinatesofCity[cityname_String] :=
     {6378.1,
      (2 Pi)/360 ToDegrees[CityData[cityname, CityPosition][[1]]],
      (2 Pi)/360 ToDegrees[ CityData[cityname, CityPosition][[2]]]
     }
9  SphericalCoordinatesofCity["Boston"]
10 CartesianCoordinatesofCity[cityname_String] :=
     CoordinatesToCartesian[SphericalCoordinatesofCity[
        cityname], Spherical[r, theta, phi]]
11 CartesianCoordinatesofCity["Paris"]
12 MinimumTunnel[city1_String, city2_String] :=
     Norm[CartesianCoordinatesofCity[city1] −
        CartesianCoordinatesofCity[city2]]
13 MinimumTunnel["Boston", "Paris"]
14 SphericalDistance[boston, paris] // N
15 SpheroidalDistance[boston, paris] // N
```

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

3: `CoordinatesFromCartesian` transforms one of many different coordinate systems, named in the second argument into three cartesian coordinates, named in the first argument-list.

7: `CityData` in the `Calculus‘VectorAnalysis‘` package can give the latitude and longitude of cities in the database—in this case Boston and Paris.

8: *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)). `ToDegrees` is from the `Miscellaneous‘Geodesy‘` package and converts a (degree, minutes, seconds)-structure to degrees.

10: *CartesianCoordinatesofCity* uses a coordinate transform and *SphericalCoordinatesofCity* to compute cartesian coordinates.

12: Imagining that a tunnel could be constructed between two cities, this function would calculate the minimum distance between cities.

14: Comparing the great circle route using `SphericalDistance` to the euclidian distance is a result that suprises me. It would save only about 55 kilometers to dig a tunnel to Paris—sigh.

15: `SpheroidalDistance` accounts for the earth's extra waistline for computing minimum distances.

Lecture 13 MATHEMATICA® Example 3

Gradient and Divergence Operations in Other Coordinate Systems

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

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

```
1  SimplePot[x_ , y_ , z_, n_] := 1/(x^2 + y^2 + z^2)^(n/2)
2  gradsp = Grad[SimplePot[x, y, z, 1], Cartesian[x, y, z]]
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
```

1: *SimplePot* is an example function—a $1/r^n$ potential in cartesian coordinates.

2: `Grad` is defined in the `Calculus‘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 here in spherical coordinates.

4: Here, the gradient of $1/r$ is obtained in spherical coordinates.

5: Here, the gradient of $1/r$ is obtained in cylindrical coordinates.

6: Here, the gradient of $1/r$ is obtained in prolate spheriodal coordinates.

8: The laplacian ($\nabla^2(1/r^n)$) has a particularly simple form...

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):

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

the curl derivative operation is another vector defined by:

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

or with the memory-device:

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

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

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

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

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

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 4

Computing and Visualizing Curl Fields

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

Examples of curls are computing for a particular family of vector fields. Visualization is produced with the `PlotVectorField3D` function from the `Graphics‘PlotField3D‘`.

1: *LeavingKansas* is the family of vector fields indicated by 13-6.

4: The function will be singular for $n > 1$ along the $z-axis$, this singularity will be reported during the numerical evaluations for visualization.

5: Here, the singularity is removed by testing the value of the argument and returning a fixed value along the singular axis.

7: Alternatively, the singular axis can be avoided by explicitly removing it from the domain of plotting.

9: This demonstrates the assertion (13-7) about the cylindrical symmetry of this curl for $n = 1$.

10: Visualizing the curl for $n = 3$: note that the field is points up with large magnitude near the vortex at the origin.

11: Demonstrate that the divergence of the curl of $\vec{w}$ vanishes for any n—this is true for any differentiable vector field.

```
1   LeavingKansas[x_, y_, z_ , n_] :=
        z^n / ((x^2 + y^2) (x^2 + y^2 + z^2)^(n/2)) {y, -x, 0}
2   LeavingKansas[x, y, z, 3]
3   << Graphics`PlotField3D`
4   PlotVectorField3D[LeavingKansas[x, y, z, 3],
      {x, -1, 1}, {y, -1, 1}, {z, -.5, .5}, VectorHeads → True,
      ColorFunction → ((Hue[# * .66]) &),
      PlotPoints → 15, ScaleFactor → 0.5]
5   LeavingKansasNicely[x_, y_, z_ , n_] :=
      Module[{CindRadsq = x^2 + y^2},
        CindRadsq =
          If[CindRadsq ≤ 10^-4, 10^-4, CindRadsq, CindRadsq];
        z^n / (CindRadsq (CindRadsq + z^2)^(n/2)) {y, -x, 0}]
6   PlotVectorField3D[LeavingKansasNicely[x, y, z, 3],
      {x, -1, 1}, {y, -1, 1}, {z, -.5, .5}, VectorHeads → True,
      ColorFunction → ((Hue[# * .66]) &),
      PlotPoints → 15, ScaleFactor → 0.5]
7   PlotVectorField3D[LeavingKansas[x, y, z, 3],
      {x, .01, 1}, {y, .01, 1}, {z, .01, .5}, VectorHeads → True,
      ColorFunction → ((Hue[# * .66]) &),
      PlotPoints → 15, ScaleFactor → 0.5]
8   Curl[LeavingKansas[x, y, z, 3], Cartesian[x, y, z]] // Simplify
9   Glenda[x_, y_, z_, n_] :=
      Simplify[Curl[LeavingKansas[x, y, z, n], Cartesian[x, y, z]]]
10  Glenda[x, y, z, 1]
11  PlotVectorField3D[Evaluate[Glenda[x, y, z, 3]],
      {x, 0, .5}, {y, 0, .5}, {z, 0.1, .5}, VectorHeads → True,
      ColorFunction → ((Hue[# * .66]) &), PlotPoints → 7]
12  DivCurl = Div[Glenda[x, y, z, n], Cartesian[x, y, z]]
13  Simplify[DivCurl]
```

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

$$\nabla \times (\nabla f) = 0 \tag{13-8}$$

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:

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

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.

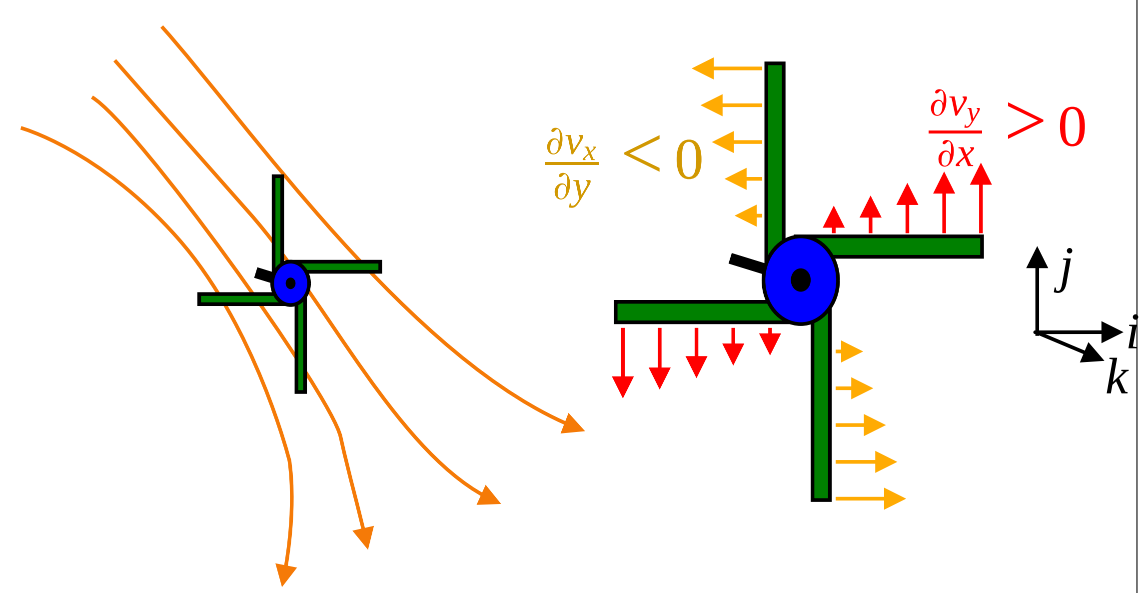

Figure 13-10: Consider a small paddle wheel placed in a set of stream lines defined by a vector field of position. If the v_y component is an increasing function of x, this tends to make the paddle wheel want to spin (positive, counter-clockwise) about the $\hat{k}$-axis. If the v_x component is a decreasing 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 $\vec{v} =$ const. and the local acceleration $\vec{v} = \nabla f$ can subtracted because of Eq. 13-10.

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

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