

Oct. 17 2012

Lecture 14: Integrals along a Path

Reading:

Kreyszig Sections: 10.1, 10.2, 10.3

Integrals along a Curve

Consider the type of integral that everyone learns initially:

$$E(b) - E(a) = \int_a^b f(x) dx \quad (14-1)$$

The equation implies that f is integrable and

$$dE = f dx = \frac{dE}{dx} dx \quad (14-2)$$

so that the integral can be written in the following way:

$$E(b) - E(a) = \int_a^b dE \quad (14-3)$$

where a and b represent “points” on some *line* where E is to be evaluated.

Of course, there is no reason to restrict integration to a straight line—the generalization is the integration along a curve (or a path) $\vec{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$.

$$E(b) - E(a) = \int_{\vec{x}(a)}^{\vec{x}(b)} \vec{f}(\vec{x}) \cdot d\vec{x} = \int_a^b g(x(\vec{t})) dt = \int_a^b \nabla E \cdot \frac{d\vec{x}}{dt} dt = \int_a^b dE \quad (14-4)$$

This last set of equations assumes that the gradient exists—i.e., there is some function E that has the gradient $\nabla E = \vec{f}$.

Path-Independence and Path-Integration

If the function being integrated along a simply-connected path (Eq. 14-4) is a gradient of some scalar potential, then the path between two integration points does not need to be specified: the integral is independent of path. It also follows that for closed paths, the integral of the gradient of a scalar potential is zero.⁶ A simply-connected path is one that does not self-intersect or can be shrunk to a point without leaving its domain.

There are familiar examples from classical thermodynamics of simple one-component fluids that satisfy this property:

$$\oint dU = \oint \nabla_{\vec{S}} U \cdot d\vec{S} = 0 \quad \oint dS = \oint \nabla_{\vec{S}} S \cdot d\vec{S} = 0 \quad \oint dG = \oint \nabla_{\vec{S}} G \cdot d\vec{S} = 0 \quad (14-5)$$

$$\oint dP = \oint \nabla_{\vec{S}} P \cdot d\vec{S} = 0 \quad \oint dT = \oint \nabla_{\vec{S}} T \cdot d\vec{S} = 0 \quad \oint dV = \oint \nabla_{\vec{S}} V \cdot d\vec{S} = 0 \quad (14-6)$$

Where \vec{S} is any other set of variables that sufficiently describe the equilibrium state of the system (i.e., $U(S, V)$, $U(S, P)$, $U(T, V)$, $U(T, P)$ for U describing a simple one-component fluid).

The relation $\text{curl } f = \nabla \times \nabla f = 0$ provides method for testing whether some *general* $\vec{F}(\vec{x})$ is independent of path. If

$$\vec{0} = \nabla \times \vec{F} \quad (14-7)$$

or equivalently,

$$0 = \frac{\partial F_j}{\partial x_i} - \frac{\partial F_i}{\partial x_j} \quad (14-8)$$

for all variable pairs x_i, x_j , then $\vec{F}(\vec{x})$ is independent of path. These are the Maxwell relations of classical thermodynamics.

⁶In fact, there are some extra requirements on the domain (i.e., the space of all paths that are supposed to be path-independent) where such paths are defined: the scalar potential must have continuous second partial derivatives everywhere in the domain.

Lecture 14 MATHEMATICA® Example 1

Path Dependence of Integration of Vector Function: Non-Vanishing Curl

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](http://pruffle.mit.edu/3.016-2012) from <http://pruffle.mit.edu/3.016-2012>.

The path dependence of a vector field with a non-vanishing curl ($\vec{v}(\vec{x}) = xyz(\hat{i} + \hat{k} + \hat{z})$) is demonstrated with a family of closed curves.

```
Needs["VectorAnalysis"];
VectorFunction = {x y z, x y z, y x z}
CurlVectorFunction = Simplify[
  Curl[VectorFunction, Cartesian[x, y, z]]]
ConditionsOfZeroCurl =
Table[0 == CurlVectorFunction[[i]], {i, 3}]
FindInstance[ConditionsOfZeroCurl, {x, y, z}]

Any curve that wraps around the cylinder can be parameterized as (x(t),
y(t), z(t)) = (R cos(t), R sin(t), A Pz,z(t)) where Pz,z(t) = Pz,z(t + 2π).
Therefore ds = (-R sin(t), R cos(t), P'z,z(t)) dt = (-y(t), x(t), A P'z,z(t)) dt
The integrand for an integral of "VectorFunction" around such a curve is
(written in terms of an arbitrary P(t)):
vf = VectorFunction.(-y, x, Amp D[P[t], t]) /.
{x -> Radius Cos[t], y -> Radius Sin[t],
z -> Amp P[t]} // Simplify
The integral depends on the choice of P(t)
PathDepInt = Integrate[vf, {t, 0, 2 Pi}]
PathDepInt /. P -> Sin
PathDepInt /. P -> Cos
PathDepInt /.
{P[t] -> t (t - 2 Pi), P'[t] -> D[t (t - 2 Pi), t]}
pdigen = PathDepInt /.
{P[t] -> Cos[n t], P'[t] -> D[Cos[n t], t]}
Simplify[pdigen, n ∈ Integers]
Limit[pdigen, n -> 1]
```



Any curve that wraps around the cylinder can be parameterized as $(x(t), y(t), z(t)) = (R \cos(t), R \sin(t), A P_{z,z}(t))$ where $P_{z,z}(t) = P_{z,z}(t + 2\pi)$.

Therefore $ds = (-R \sin(t), R \cos(t), P'_{z,z}(t)) dt = (-y(t), x(t), A P'_{z,z}(t)) dt$
The integrand for an integral of "VectorFunction" around such a curve is (written in terms of an arbitrary $P(t)$):

```
vf = VectorFunction.(-y, x, Amp D[P[t], t]) /.
{x -> Radius Cos[t], y -> Radius Sin[t],
z -> Amp P[t]} // Simplify
```

The integral depends on the choice of $P(t)$

```
PathDepInt = Integrate[vf, {t, 0, 2 Pi}]
PathDepInt /. P -> Sin
PathDepInt /. P -> Cos
PathDepInt /.
{P[t] -> t (t - 2 Pi), P'[t] -> D[t (t - 2 Pi), t]}
pdigen = PathDepInt /.
{P[t] -> Cos[n t], P'[t] -> D[Cos[n t], t]}
Simplify[pdigen, n ∈ Integers]
Limit[pdigen, n -> 1]
```

- 1: *VectorFunction* (xyz, xyz, xyz) is an example vector field that has a *non-vanishing curl*. The curl is computed with `Curl` which is in the `VectorAnalysis` package. Here, the particular coordinate system is specified with `Cartesian` argument to `Curl`.
- 2–3: The curl vanishes only at the origin—this is shown with `FindInstance` called with a list of equations corresponding to the vanishing curl.
- 4: This is the integrand $\vec{v} \cdot d\vec{s}$ computed as indicated in the figure, $d\vec{s} = (-y(t), x(t), P'(t))dt$. $P(\theta)$ represents any periodic function, but $(x, y) = R(\cos \theta, \sin \theta)$ representing paths that wrap around cylinders.
- 5: *PathDepInt* is an integral for \vec{v} represented by *VectorFunction* an arbitrary path wrapping around the cylinder.
- 6–9: These are examples of a computation by using a replacement for a periodic $P(\theta)$ (i.e., each of the $P(\theta)$ begin and end at the same point, but the path between differs). The examples use $P(t) = \sin(t)$, $\cos(t)$, and $t(t - 2\pi)$. That the results differ shows that \vec{v} is path-dependent—this is a general result for non-vanishing curl vector functions.
- 9–10: These results show that, for *some* closed paths, the result will be path-independent (here, for $P(t) = \cos(nt)$ the path-integral vanishes for integer n . This doesn't imply path-independence for all paths.
- 11: Our last result seems to contradict the result in 7 for which the integral was zero. However, computing the limit resolves the contradiction.

Lecture 14 MATHEMATICA® Example 2

Examples of Path-Independence of Curl-Free Vector Fields

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](#) from <http://pruffle.mit.edu/3.016-2012>.

A curl-free vector field can be generated from any scalar potential, in this case $\vec{w} = \nabla e^{xyz} = \vec{w}(\vec{x}) = e^{xyz}(yz\hat{i} + zx\hat{k} + xy\hat{z})$ will be shown to be curl-free.

```

Try the path dependence with a
conservative (curl free, or exact) Vector
Function:

Start with a scalar potential
temp = Grad[Exp[x y z], Cartesian[x, y, z]] 1
Create another vector function that should have a zero curl
AnotherVFunction = {e^{y^2} y z, e^{y^2} x z, e^{y^2} x y}
Simplify[
Curl[AnotherVFunction, Cartesian[x, y, z]] 2
anothervf =
AnotherVFunction.{-y, x, D[P[t], t]} /.
{x -> Radius Cos[t], y -> Radius Sin[t],
z -> P[t]} // Simplify 3
The integral depends doesn't on the choice of P(t)
PathDepInt = Integrate[anothervf, t] 4
e^{Radius^2 Cos[t] P[t]} Sin[t]
(PathDepInt /. t -> 2 Pi) - (PathDepInt /. t -> 0) 5
0

```

- 1: To ensure that we will have a zero-curl, a vector field is generated from a gradient of a scalar potential. The curl vanishes because $\nabla \times \nabla f = 0$.
- 2: This is a demonstration that the curl does indeed vanish.
- 3: Here is the integrand for $\oint \vec{v} \cdot d\vec{s}$ for the family of paths that wrap around a cylinder for the particular case of this conservative fields.
- 4: This is the general result for the family of curves indicated by $P(\theta)$.
- 5: This demonstrates that the path integral closes for any periodic $P(\theta)$ —which is the same as the condition that the curve is closed.

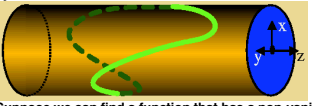
Lecture 14 MATHEMATICA® Example 3

Examples of Path-Independence of Curl-Free Vector Fields on a Restricted Subspace

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](#) from <http://pruffle.mit.edu/3.016-2012>.

If a path-integral is path-dependent for an arbitrary three path, it is possible that path-independence can occur over closed paths restricted to some surface where the curl vanishes. To find a function that is curl-free on a restricted subspace (for example, the vector function $\vec{v}(\vec{x}) = (x^2 + y^2 - R^2)\hat{z}$ vanishes on the surface of a cylinder) one needs to find a \vec{m} such that $\nabla \times \vec{m} = \vec{v}$ (for this case

□ For a last example, suppose the curl vanishes on the cylindrical surface defined above:



Suppose we can find a function that has a non-vanishing curl on this surface

We want to find a function which is generally non-curl free, but for which the curl vanishes on a surface. Let's pick the cylinder as our surface.

`VanishOnCylinder = x^2 + y^2 - Radius^2` 1

If a function can be found, that has the following curl, then we will have constructed such a function.

`CurlOfOneStooge = {0, 0, VanishOnCylinder}` 2

It is easy to see that this is the curl of Stooge, where we construct Stooge by integrating.

`Stooge =`
`{-1/2 Integrate[VanishOnCylinder, y],`
`1/2 Integrate[VanishOnCylinder, x], 0}` 3

In fact, we could add to Stooge, any vector function that has vanishing curl—there are an infinite number of these

`Simplify[Curl[Stooge, Cartesian[x, y, z]]]` 4

Its integral doesn't care which path around the cylinder it takes, the integrand doesn't depend on P(t)

`WhyIOughta = Stooge.{-y, x, D[P[t], t]} /.`
`{x -> Radius Cos[t],`
`y -> Radius Sin[t], z -> P[t]} // Expand` 5

This is the value for "any" path on the cylinder that is closed.

`Integrate[WhyIOughta, {t, 0, 2 Pi}]` 6

$$\frac{\pi \text{Radius}^4}{2}$$

- 1–3: This demonstrates a method to find a vector field for which the curl that vanishes on a on a surface. This is an example for the cylinder surface. The zero constraint, *VanishOnCylinder*, is used to produce a vector field that will represent the curl, *CurlOfOneStooge*. The formula for the curl is integrated to find the vector function, *Stooge*, that has the specified curl.
- 4: This demonstrates that the curl is what we designed it to be.
- 5–6: This demonstrates that the integral of *Stooge* is path-independent on the cylinder and its value is $-\pi R^4/2$.

Multidimensional Integrals

Perhaps the most straightforward of the higher-dimensional integrations (e.g., vector function along a curve, vector function on a surface) is a scalar function over a domain such as, a rectangular block in two dimensions, or a block in three dimensions. In each case, the integration over a dimension is uncoupled from the others and the problem reduces to pedestrian integration along a coordinate axis.

Sometimes difficulty arises when the domain of integration is not so easily described; in these cases, the limits of integration become functions of another integration variable. While specifying the limits of integration requires a bit of attention, the only thing that makes these cases difficult is that the

integrals become tedious and lengthy. MATHEMATICA® removes some of this burden.

A short review of various ways in which a function's variable can appear in an integral follows:

	The Integral	Its Derivative
Function of limits	$p(x) = \int_{\alpha(x)}^{\beta(x)} f(\xi) d\xi$	$\frac{dp}{dx} = f(\beta(x)) \frac{d\beta}{dx} - f(\alpha(x)) \frac{d\alpha}{dx}$
Function of integrand	$q(x) = \int_a^b g(\xi, x) d\xi$	$\frac{dq}{dx} = \int_a^b \frac{\partial g(\xi, x)}{\partial x} d\xi$
Function of both	$r(x) = \int_{\alpha(x)}^{\beta(x)} g(\xi, x) d\xi$	$\begin{aligned} \frac{dr}{dx} &= f(\beta(x)) \frac{d\beta}{dx} - f(\alpha(x)) \frac{d\alpha}{dx} \\ &\quad + \int_{\alpha(x)}^{\beta(x)} \frac{\partial g(\xi, x)}{\partial x} d\xi \end{aligned}$

Using Jacobians to Change Variables in Thermodynamic Calculations

Changing of variables is a topic in multivariable calculus that often causes difficulty in classical thermodynamics.

This is an extract of my notes on thermodynamics: <http://pruffle.mit.edu/3.00/>

Alternative forms of differential relations can be derived by changing variables.

To change variables, a useful scheme using Jacobians can be employed:

$$\begin{aligned} \frac{\partial(u, v)}{\partial(x, y)} &\equiv \det \begin{vmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{vmatrix} \\ &= \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \\ &= \left(\frac{\partial u}{\partial x} \right)_y \left(\frac{\partial v}{\partial y} \right)_x - \left(\frac{\partial u}{\partial y} \right)_x \left(\frac{\partial v}{\partial x} \right)_y \\ &= \frac{\partial u(x, y)}{\partial x} \frac{\partial v(x, y)}{\partial y} - \frac{\partial u(x, y)}{\partial y} \frac{\partial v(x, y)}{\partial x} \end{aligned} \tag{14-9}$$

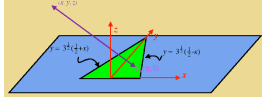
Lecture 14 MATHEMATICA® Example 4

Integrals over Variable Domains

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](http://pruffle.mit.edu/3.016-2012) from <http://pruffle.mit.edu/3.016-2012>.

This will demonstrate how MATHEMATICA® handles multiple integrals; in particular, when the domains depend on the integration variables. The goal is to find a function that will give the potential in the vicinity of a triangular patch with uniform charge density.

We will attempt to model the energy of ion just above one half of a triangular capacitor. Suppose there is a uniformly charged surface ($\rho = \text{charge/area}=1$) occupying an equilateral triangle in the $z=0$ plane:



what is the energy (voltage) of a unit positive charge located at (x,y,z)

The electrical potential goes like $\frac{1}{r}$, therefore the potential of a unit charge located at (x,y,z) from a small surface patch at $(\xi,\eta,0)$ is

$$\frac{\rho d\xi d\eta}{r} = \frac{d\xi d\eta}{\sqrt{(x-\xi)^2 + (y-\eta)^2 + z^2}}$$

Therefore it remains to integrate this function over the domain $\eta \in (0, \frac{\sqrt{3}}{2})$

and $\xi \in (\frac{\eta}{\sqrt{3}} - \frac{1}{2}, (\frac{1}{2} - \frac{\eta}{\sqrt{3}}))$

$$\int_0^{\frac{\sqrt{3}}{2}} \int_{\frac{\eta}{\sqrt{3}} - \frac{1}{2}}^{\frac{1}{2} - \frac{\eta}{\sqrt{3}}} \frac{d\xi d\eta}{\sqrt{(x-\xi)^2 + (y-\eta)^2 + z^2}}$$

```
Integrate[f[x, y], {y, 0, 1}, {x, xi, xf}]
Integrate[f[x, y], {y, yi, yf}, {x, xi, xf}]
Integrate[f[x, y], {y, yi, yf}, {x, xi[y], xf[y]}]
```

1

For example, consider the difference in the following two cases:
First, we integrate over x and y using the two iterators in `Integrate` with the order $\{y, 0, 1\}, \{x, 0, 1\}$. Then explicitly using two separate steps

```
Integrate[Exp[3 x], {y, 0, 1}, {x, 0, 1}]
interx = Integrate[Exp[3 x], {x, 0, 1}]
Integrate[interx, {y, 0, 1}]
```

2

Compared to
integrate over x and y using the two iterators in `Integrate` with the order $\{x, 0, 1\}, \{y, 0, 1\}$. Then explicitly using two separate steps

```
Integrate[Exp[3 x], {x, 0, 1}, {y, 0, 1}]
intery = (Integrate[Exp[3 x], {y, 0, 1}])
Integrate[intery, {x, 0, 1}]
```

3

1: These examples demonstrate that MATHEMATICA® integrates over the last iterator which appears in the argument-list of `Integrate` first: LIFI-FILI (last iterator, first integrated; first iterator, last integrated).

2-3: Here we demonstrate the order of integration explicitly, by first integrating with two iterators, and then integrating in two step-sequence. The methods are equivalent.

Lecture 14 MATHEMATICA® Example 5

Potential near a Charged and Shaped Surface Patch: Brute Force

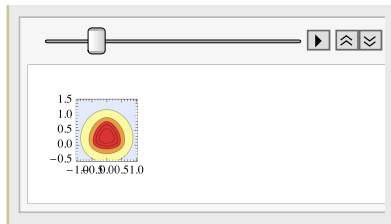
Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](http://pruffle.mit.edu/3.016-2012) from <http://pruffle.mit.edu/3.016-2012>.

An example of a multiple integral and its numerical evaluation for the triangular charged patch.

```

TrianglePotentialNumeric[x_, y_, z_] :=
NIntegrate[1/Sqrt[(x-ξ)²+(y-η)²+z²], {η, 0,
Sqrt[3]/2}, {ξ, η/Sqrt[3]-1/2, 1/2-η/Sqrt[3]]]
TrianglePotentialNumeric[1, 3, .01]
Plot[TrianglePotentialNumeric[x, x, 1/40],
{x, -1, 1}]
cplot[h_] := cplot[h] = ContourPlot[
TrianglePotentialNumeric[x, y, h],
{x, -1, 1}, {y, -0.5, 1.5}, Contours ->
Table[v, {v, .25, 2, .25}], ColorFunction ->
ColorData["TemperatureMap"],
ColorFunctionScaling -> False,
PlotPoints -> 11]
Timing[cplot[1/10]]
Row[{TextCell[
"Computing ContourPlots a different
h: Progress: ", "Text"],
ProgressIndicator[Dynamic[h], {0, .5}]]]
cplots = Table[cplot[h], {h, .025, .5, .025}];
ListAnimate[cplots]

```



- 1: MATHEMATICA® can't seem to find a closed-form solution to this integral over the triangular domain, However, the energy can be integrated numerically. Here is a function that calls `NIntegrate` for a location given by its arguments. We will call this function at different heights z . Multidimensional integration is generally computationally expensive.
- 2–3: Here are examples calling the numerical function `TrianglePotentialNumeric`. First, the function is evaluated at a single point; next, it is evaluated and plotted along a 45° -line parallel in the $z = 1/40$ plane.
- 4: The function `cplot` calls `TrianglePotentialNumeric` repeatedly at variable x and y to generate a `ContourPlot` at height specified by the argument to `cplot`. These plots will eventually appear in an animation, so `ColorFunctionScaling` is set to false so that the colors will be consistent between frames. The `Contours` are set explicitly so that they are also consistent across frames. `Timing` indicates that each plot consumes a large number of cpu cycles.
- 5: Because each frame is expensive to compute, it is not a good idea to compute them *within* an animation. Here, we use `Table` to generate individual frames (n.b., the `cplots` stores its previous calculations in memory). Because this is time consuming, we add a *progress monitor* that will dynamically update as each `cplot[h]` is computed. We use `ProgressIndicator` on the argument `Dynamic[h]`. `Dynamic` informs MATHEMATICA® that a particular variable will be changing; therefore the object that calls it will need to be updated.
- 6: We use `ListAnimate` on the pre-computed frames.