


$$\min_{P \in \Omega} \int \log\left(\frac{dP}{dQ}\right) dP$$



 You are here

Carlos C. Rodríguez

Checking Stokes' Theorem for a general triangle in 3D

Given the positions of three points $\vec{p}_0, \vec{p}_1, \vec{p}_2$ where $\vec{p}_j = x_j\hat{x} + y_j\hat{y} + z_j\hat{z}$

for $j = 0, 1, 2$, they define an oriented plane segment S . This plane segment is the triangle with the points as its vertices and oriented by the order $[\vec{p}_0, \vec{p}_1, \vec{p}_2]$ which means that we go around the boundary visiting the vertices in that order. This order corresponds to the choice of normal vector field $\vec{N} = (\vec{p}_1 - \vec{p}_0) \times (\vec{p}_2 - \vec{p}_0)$

The Fundamental Theorem of (all) Calculus says that:

$$\int_{\partial S} \omega = \int_S d\omega$$

where

$$\omega = f_1(x, y, z)dx + f_2(x, y, z)dy + f_3(x, y, z)dz$$

is any 1-form with the scalar functions f_1, f_2, f_3 smooth functions defined on a region that contains S and where ∂S is the oriented boundary of S and the differential is simply,

$$d\omega = df_1 \wedge dx + df_2 \wedge dy + df_3 \wedge dz$$

A 1-form, like ω is just a linear function of (tangent) vectors or if we define the concatenation of two curves as their sum we can think of ω as a linear function on curves. Give a one dimensional oriented smooth thingee (a parameterized curve) to ω and it will give you back a number independent of the way in which you choose to parameterize the curve. This number is the line integral of the 1-form along the curve and the mapping from curves to numbers is linear. From this point of view the purely algebraic fact that $d^2\omega = 0$ (i.e. taking the differential twice) is just the purely geometrical fact that $\partial^2 S = \emptyset$ (i.e. the boundary of a boundary is empty; In math lingo taking the boundary is the adjoint to taking the differential).

The algebra of differential forms unifies all the scattered around fundamental theory like theorems including: The standard fundamental theorem of calculus (I), i.e.,

$$\int_{[a,b]} df(x) = -f(a) + f(b)$$

The fundamental theorem for line integrals, i.e.,

$$\int_C dV(x,y,z) = -V(\vec{A}) + V(\vec{B})$$

where C is any smooth curve from \vec{A} to \vec{B} and $V(x,y,z)$ is a scalar field (the scalar potential). Green's theorem on the plane, i.e.

$$\int_{\partial D} P(x,y)dx + Q(x,y)dy = \int_D \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dA$$

Stokes' theorem,

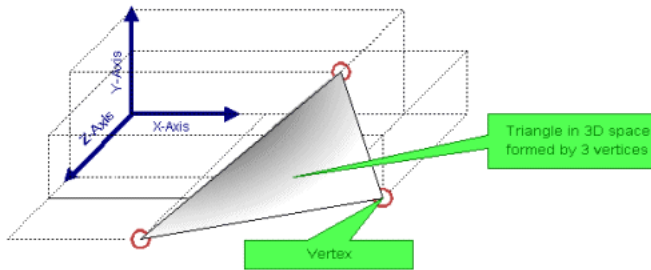
$$\int_{\partial S} \vec{F} \cdot d\vec{r} = \int_S (\nabla \times \vec{F}) \cdot d\vec{S}$$

and Gauss' divergence theorem,

$$\int_{\partial V} \vec{F} \cdot d\vec{S} = \int_V \nabla \cdot \vec{F} dV$$

In this notebook we show that Stokes' theorem is in fact true for a general triangle in 3D. Now, if you think about it a little, by noticing that any smooth surface can be well approximated by splitting it into little triangles (a.k.a. triangulation duh!) we are (almost) proving that Stokes Theorem is true for any (at least triangulatable (is that a word?)) surface.

So here is the surface S .



We are going to need line and surface integration so we "attach" the functions SurfInt.sage defined on our web site.

```

_____ Sage code _____
attach(DATA+"SurfInt.sage")

```

Start by defining the points (p_0, p_1, p_2) by their general coordinates and the boundary segments C_{01}, C_{02}, C_{12} connecting the points (e.g. C_{12} is the straight line segment from p_1 to p_2).

```

_____ Sage code _____
var('x y z x0 y0 z0 x1 y1 z1 x2 y2 z2 t')
p0 = vector(x0,y0,z0)
p1 = vector(x1,y1,z1)
p2 = vector(x2,y2,z2)
C01(t) = tuple((1-t)*p0+t*p1)
C02(t) = tuple((1-t)*p0+t*p2)
C12(t) = tuple((1-t)*p1+t*p2)

```

Since we have already defined all these goodies, let's not miss the opportunity to check that the area of the triangle computed with the aid of the geometric definition of the cross product (i.e., A_0), coincides with the value given by integration over the surface S (what we call "Area").

```

_____ Sage code _____
f(x,y,z)=1
S(u,v) = tuple((1-u-v)*p0+u*p1+v*p2)
Area = surf_intds(f,S,(u,0,1-v),(v,0,1))
A0 = (p1-p0).cross_product(p2-p0).norm()/2
(Area-A0).full_simplify()

```

0
nice!

Sage code

```
F(x,y,z)=(0,0,2*x+y)
cF(x,y,z)=curl(F)
Flux1 = surf_int(cF,S,(u,0,1-v),(v,0,1))
Flux1
```

$$\frac{1}{2}(z_0 - z_2)(y_0 - y_1) + (z_0 - z_2)(x_0 - x_1) - \frac{1}{2}(z_0 - z_1)(y_0 - y_2) - (z_0 - z_1)(x_0 - x_2)$$

Sage code

```
L1 = line_int(F,C01,(t,0,1))
L2 = line_int(F,C12,(t,0,1))
L3 = line_int(F,C02,(t,1,0))
L1;L2;L3
```

$$\begin{aligned} &-\frac{1}{2}(z_0 - z_1)(2x_0 + 2x_1 + y_0 + y_1) \\ &-\frac{1}{2}(z_1 - z_2)(2x_1 + 2x_2 + y_1 + y_2) \\ &\frac{1}{2}(z_0 - z_2)(2x_0 + 2x_2 + y_0 + y_2) \end{aligned}$$

Sage code

```
(Flux1 - (L1+L2+L3)).full_simplify()
```

0

In the next cell we try something a bit more general for the field. Notice that every vector field in a region of R^3 can be written as, $F = f\hat{x} + g\hat{y} + h\hat{z}$ where f, g, h are smooth functions of (x, y, z) in the region where F is defined. We have that

$$\nabla \times F = \nabla \times (f\hat{x}) + \nabla \times (g\hat{y}) + \nabla \times (h\hat{z})$$

Therefore, if we can show that Stokes' theorem is true for any smooth vector field parallel to one of the axis then by linearity it will be true for every vector field. Moreover, every smooth enough scalar function can be well approximated by a (multivariate) power series. With these considerations, the fact that Stokes' theorem holds for the vector fields of the form

$$F = x^i y^j z^k \hat{z},$$

as we check in the cell below, makes it very likely that it will hold for all vector fields.

Sage code

```
n = floor(6*random()) for i in range(3)
F(x,y,z) = (0,0,x^n0*y^n1*z^n2)
cF(x,y,z) = curl(F)
Flux2 = surf_int(cF,S,(u,0,1-v),(v,0,1),evaluate=True)
L1 = line_int(F,C01,(t,0,1))
L2 = line_int(F,C12,(t,0,1))
L3 = line_int(F,C02,(t,1,0))
(Flux2 - (L1+L2+L3)).full_simplify(), n
```

(0,[3,3,3])

Now Let's Interact!

Sage interact code

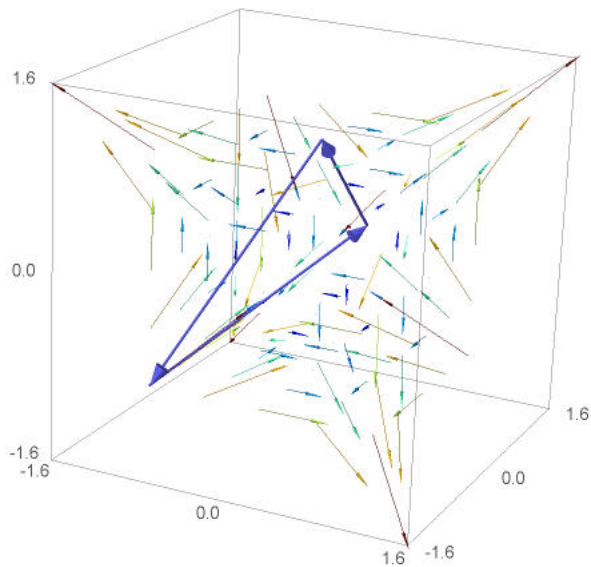
```
%hide
x,y,z = var('x,y,z')
interact
def _(f = input_box(default=y*z), g=input_box(default=-x*y+x^3-z),
    h=input_box(default=0),
    p0=input_box(default=(-1,-1,-1)),
    p1=input_box(default=(0,1,0)),
    p2=input_box(default=(0,0,1))):
    vector_field = plot_vector_field3d((f,g,h),
                                       (x,-1,1),(y,-1,1),(z,-1,1))

    p01 = arrow3d(p0,p1)
    p12 = arrow3d(p1,p2)
    p20 = arrow3d(p2,p0)
    tri = p01+p12+p20
    result = vector_field+tri
    F(x,y,z)=(f,g,h)
    v0 = vector(p0)
    v1 = vector(p1)
    v2 = vector(p2)
    C01(t) = tuple((1-t)*v0+t*v1)
    C02(t) = tuple((1-t)*v0+t*v2)
    C12(t) = tuple((1-t)*v1+t*v2)
    S(u,v) = tuple((1-u-v)*v0+u*v1+v*v2)
    cF(x,y,z) = curl(F)
```

```

Flux = surf_int(cF,S,(u,0,1-v),(v,0,1),evaluate=True)
L1 = line_int(F,C01,(t,0,1))
L2 = line_int(F,C12,(t,0,1))
L3 = line_int(F,C02,(t,1,0))
LI = (L1+L2+L3)
html(r"Make sure the points are in the  $(-1,1)^{3}$  cube.")
html(r" ") # new line
html(r" $\int_S (\nabla \times F) \cdot dS = %s$  and
 $\int_{\partial S} F \cdot dr = %s + %s + %s = %s$ "
%(latex(Flux),latex(L1),latex(L2),latex(L3),
latex(LI)))
result.show(xmin=-1,xmax=1,ymin=-1,ymax=1,zmin=-1,
zmax=1)

```



Carlos Rodriguez <carlos@math.albany.edu>