Stokes' Theorem

Based on Mathemation Video Stokes' Theorem - Examples I

Written by Daniel Volinski at danielvolinski@yahoo.es

(%i2)
```
info:build_info()$info@version;
```

(%o2)

5.38.1

(%i2)
```
reset()$kill(all)$
```

(%i1)
```
derivabbrev:true$
```

(%i2)
```
ratprint:false$
```

(%i3)
```
fpprintprec:5$
```

(%i4)
```
load(linearalgebra)$
```

(%i5)
```
if get('draw,'version)=false then load(draw)$
```

(%i6)
```
wxplot_size:[1024,768]$
```

(%i7)
```
if get('drawdf,'version)=false then load(drawdf)$
```

(%i8)
```
set_draw_defaults(xtics=1,ytics=1,ztics=1,xyplane=0,nticks=100,
xaxis=true,xaxis_type=dots,xaxis_width=3,
yaxis=true,yaxis_type=dots,yaxis_width=3,
zaxis=true,zaxis_type=dots,zaxis_width=3,
background_color=light_gray)$
```

(%i9)
```
if get('vect,'version)=false then load(vect)$
```

(%i10)
```
norm(u):=block(ratsimp(radcan(√(u.u))))$
```

(%i11)
```
normalize(v):=block(v/norm(v))$
```

(%i12)
```
angle(u,v):=block([junk:radcan(√((u.u)*(v.v)))],acos(u.v/junk))$
```

(%i13)
```
mycross(va,vb):=[va[2]*vb[3]-va[3]*vb[2],va[3]*vb[1]-va[1]*vb[3],va[1]*vb[2]-va[2]*vb[1]]$
```

(%i14)
```
if get('diff_form,'version)=false then load(diff_form)$
```

(%i15)
```
inv_i1(_pform):=block([a_],a_:makelist(coeff(_pform,basis[i]),i,1,dim),
list_matrix_entries(a_ . sqrt(diag(norm_table))))$
```

(%i16)
```
declare(trigsimp,evfun)$
```

Stokes' Theorem

$$\oint_C \vec{F}\cdot \mathrm{d}\vec{r} = \iint_S (\nabla \times \vec{F})\cdot \mathrm{d}\vec{s}$$

Let $\vec{F}(x,y,z) = \langle e^z, x\,y\,z, x^3\rangle$ and let C be the path of straight line segments shown down below. Evaluate $\int_C \vec{F}\cdot \mathrm{d}\vec{r}$.

Define the space $\mathbb{R}^3$

(%i17) `ζ:[x,y,z]$`

(%i18) `dim:length(ζ)$`

(%i19) `scalefactors(ζ)$`

Vector field $\vec{F} \in \mathbb{R}^3$

(%i20) `ldisplay(F:[0,x*z,-x*y])$`

$$F = [0, xz, -xy] \qquad (\%t20)$$

(%i21) `ldisplay(F:[exp(z),x*y*z,x^3])$`

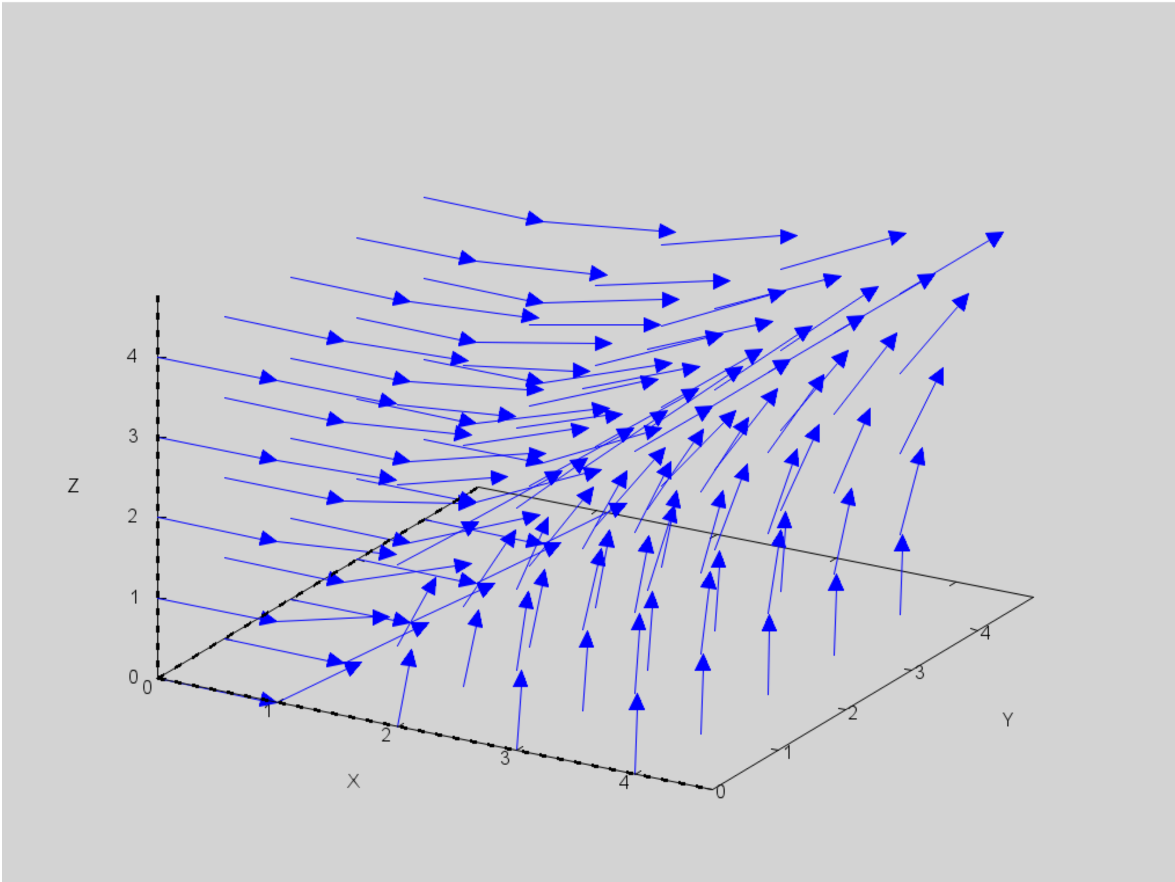
$$F = [e^z, xyz, x^3] \qquad (\%t21)$$

3D Direction field

(%i23)
```
/* vector origins are (x,y,z)| x,y=1,...,5 */
coord:setify(makelist(k,k,0,4))$
points3d:listify(cartesian_product(coord,coord,coord))$
```

(%i25)
```
/* compute vectors at the given points */
define(vf3d(x,y,z),vector(ζ,F))$
vect3:makelist(vf3d(k[1],k[2],k[3]),k,points3d)$
```

(%i26) `wxdraw3d([head_length=0.1,color=blue,head_angle=25,unit_vectors=true],vect3)$`



(%t26)

Calculate $\nabla \cdot \vec{F} \in \mathbb{R}$

(%i27)
```
ldisplay(divF:ev(express(div(F)),diff))$
```

$$divF = xz \tag{\%t27}$$

Calculate $\nabla \times \vec{F} \in \mathbb{R}^3$

(%i28)
```
ldisplay(curlF:ev(express(curl(F)),diff))$
```

$$curlF = [-xy, e^z - 3x^2, yz] \tag{\%t28}$$

Calculate $*\mathrm{d} * \vec{F}^\flat$

(%i29)
```
fstar_with_clf(ζ,ζ,nest2([h_st,d,h_st,vtof1],F));
```

$$xz \tag{\%o29}$$

Verify $\nabla \cdot \vec{F} = *\mathrm{d} * \vec{F}^\flat$

(%i30)
```
is(%=divF);
```

$$\text{true} \tag{\%o30}$$

Calculate $(\nabla \times \vec{F})^\flat \in \mathcal{A}^1(\mathbb{R}^3)$

(%i31)
```
format(fstar_with_clf(ζ,ζ,vtof1(curlF)),%poly(Dx,Dy,Dz));
```

$$Dy\ \left(e^z - 3x^2\right) + Dzyz - Dxxy \tag{\%o31}$$

Calculate $*\mathrm{d}\vec{F}^\flat \in \mathcal{A}^1(\mathbb{R}^3)$

(%i32)
```
format(fstar_with_clf(ζ,ζ,nest2([h_st,d,vtof1],F)),%poly(Dx,Dy,Dz));
```

$$Dy\ \left(e^z - 3x^2\right) + Dzyz - Dxxy \tag{\%o32}$$

Verify $(\nabla \times \vec{F})^\flat = *\mathrm{d}\vec{F}^\flat$

(%i33)
```
is(%=%th(2));
```

$$\text{true} \tag{\%o33}$$

Calculate $(*\mathrm{d}\vec{F}^\flat)^\sharp$

(%i34)
```
fstar_with_clf(ζ,ζ,inv_i1(%th(2)));
```

$$[-xy, e^z - 3x^2, yz] \tag{\%o34}$$

Verify $\nabla \times \vec{F} = (*\mathrm{d}\vec{F}^\flat)^\sharp$

(%i35)
```
is(%=curlF);
```

$$\text{true} \tag{\%o35}$$

Work form $\vec{F}^\flat = \alpha \in \mathcal{A}^1(\mathbb{R}^3)$

```
(%i37) fstar_with_clf(ζ,ζ,vtof1(F))$
       ldisplay(α:format(%,%poly(Dx,Dy,Dz)))$
```

$$\alpha = Dx\, e^z + Dy xyz + Dz\, x^3 \tag{\%t37}$$

Calculate d$\alpha \in \mathcal{A}^2(\mathbb{R}^3)$

```
(%i39) fstar_with_clf(ζ,ζ,nest2([d,vtof1],F))$
       ldisplay(dα:format(%,%poly(Dx,Dy,Dz)))$
```

$$d\alpha = Dx\, Dz\, \left(3x^2 - e^z\right) + Dx\, Dy yz - Dy\, Dz xy \tag{\%t39}$$

Flux form $\beta \in \mathcal{A}^2(\mathbb{R}^3)$

```
(%i41) fstar_with_clf(ζ,ζ,vtof2(F))$
       ldisplay(β:format(%,%poly(Dx,Dy,Dz)))$
```

$$\beta = Dy\, Dz\, e^z - Dx\, Dz xyz + Dx\, Dy\, x^3 \tag{\%t41}$$

```
(%i43) fstar_with_clf(ζ,ζ,β@Dx+β@Dy+β@Dz)$
       ldisplay(ω:format(%,%poly(Dx,Dy,Dz)))$
```

$$\omega = Dx\, Dy\, Dz\, \left(e^z + xyz + x^3\right) \tag{\%t43}$$

Calculate d$\beta \in \mathcal{A}^3(\mathbb{R}^3)$

```
(%i45) fstar_with_clf(ζ,ζ,nest2([d,vtof2],F))$
       ldisplay(dβ:format(%,%poly(Dx,Dy,Dz)))$
```

$$d\beta = Dx\, Dy\, Dz xz \tag{\%t45}$$

```
(%i46) fstar_with_clf(ζ,ζ,diff(ζ,z)|(diff(ζ,y)|(diff(ζ,x)|dβ)));
```

$$xz \tag{\%o46}$$

End Points

```
(%i50) A:[1,0,0]$B:[1,2,0]$P:[0,2,1]$Q:[0,0,1]$
```

Trajectories and their derivatives

```
(%i58) C_1:A*(1-t)+B*t$C\'_1:diff(C_1,t)$
       C_2:B*(1-t)+P*t$C\'_2:diff(C_2,t)$
       C_3:P*(1-t)+Q*t$C\'_3:diff(C_3,t)$
       C_4:Q*(1-t)+A*t$C\'_4:diff(C_4,t)$
```

Line integrals according to Vector Calculus

```
(%i62) I_1:'integrate(ev(F,map("=",ζ,C_1)).C\'_1,t,0,1)$
       I_2:'integrate(ev(F,map("=",ζ,C_2)).C\'_2,t,0,1)$
       I_3:'integrate(ev(F,map("=",ζ,C_3)).C\'_3,t,0,1)$
       I_4:'integrate(ev(F,map("=",ζ,C_4)).C\'_4,t,0,1)$
```

Total line integral according to Vector Calculus

```
(%i63) ldisplay(I_1+I_2+I_3+I_4=box(ev(I_1+I_2+I_3+I_4,integrate)))$
```

$$\int_0^1 -e^t - t^3 + 3t^2 - 3t + 1dt + \int_0^1 e^{1-t} - t^3 dt = (0) \qquad (\%t63)$$

Line integrals according to Differential Forms

```
(%i67) I_1:'integrate(fstar_with_clf(ζ,ζ,C\'_1|ev(α,map("=",ζ,C_1))),t,0,1)$
       I_2:'integrate(fstar_with_clf(ζ,ζ,C\'_2|ev(α,map("=",ζ,C_2))),t,0,1)$
       I_3:'integrate(fstar_with_clf(ζ,ζ,C\'_3|ev(α,map("=",ζ,C_3))),t,0,1)$
       I_4:'integrate(fstar_with_clf(ζ,ζ,C\'_4|ev(α,map("=",ζ,C_4))),t,0,1)$
```

Total line integral according to Differential Forms

```
(%i68) ldisplay(I_1+I_2+I_3+I_4=box(ev(I_1+I_2+I_3+I_4,integrate)))$
```

$$\int_0^1 -e^t - t^3 + 3t^2 - 3t + 1dt + \int_0^1 e^{1-t} - t^3 dt = (0) \qquad (\%t68)$$

Surface $\vec{S} \in \mathbb{R}^3$

(%i69) `S:[x,y,1-x]$`

Normal $\vec{N} \in \mathbb{R}^3$

(%i70) `ldisplay(N:mycross(diff(S,x),diff(S,y)))$`

$$N = [1, 0, 1] \tag{\%t70}$$

Calculate $(\nabla \times \vec{F}) \circ \vec{S}$

(%i71) `ldisplay(curlFoS:subst(map("=",ζ,S),curlF))$`

$$curlFoS = [-xy, e^{1-x} - 3x^2, (1-x)\, y] \tag{\%t71}$$

Integrand according to Vector Calculus

(%i72) `ldisplay(integrand:expand(curlFoS.N))$`

$$integrand = y - 2xy \tag{\%t72}$$

Integrand according to Differential Forms

(%i73) `ldisplay(integrand:fstar_with_clf(ζ,ζ,diff(S,y)|(diff(S,x)|ev(dα,map("=",ζ,S)))))$`

$$integrand = y - 2xy \tag{\%t73}$$

Surface integral

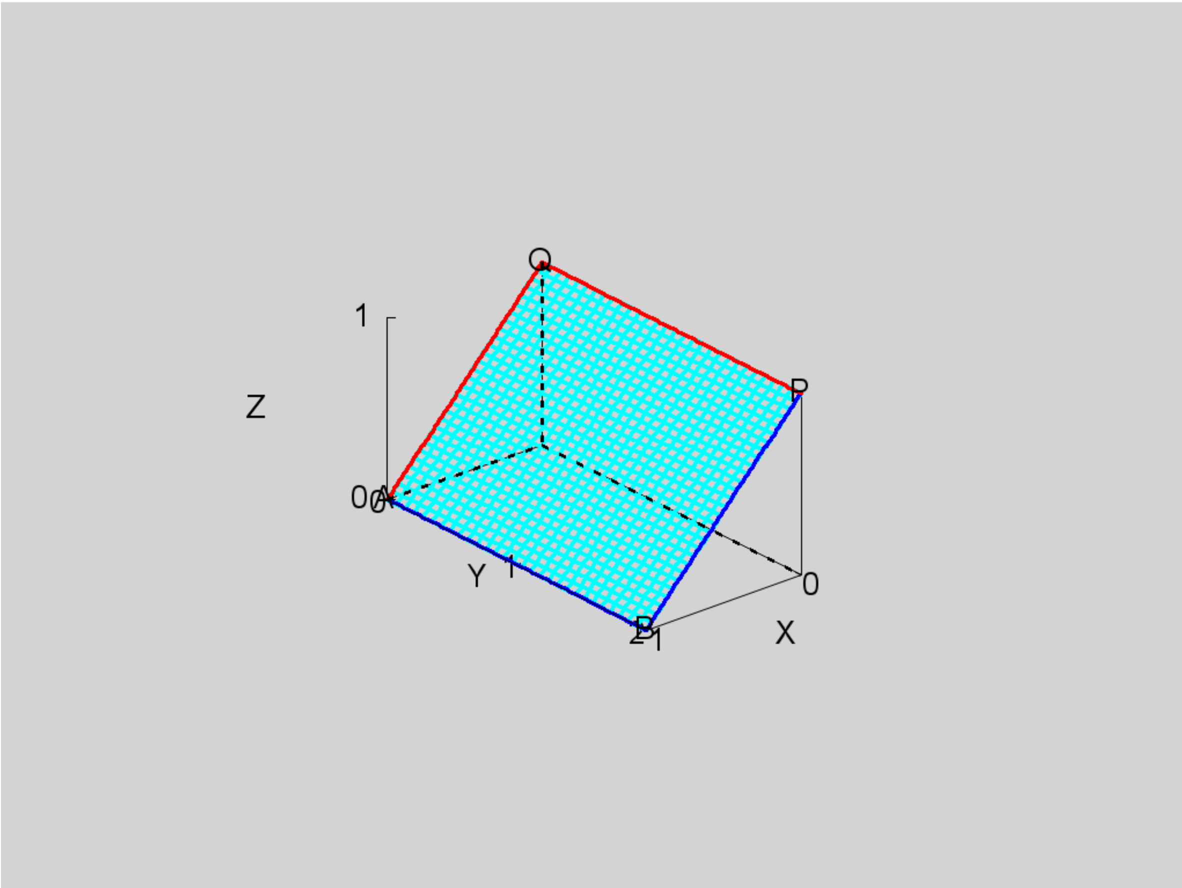
(%i74) `I:'integrate('integrate(integrand,y,0,2),x,0,1)$`

(%i75) `ldisplay(I=box(ev(I,integrate)))$`

$$\int_0^1 \int_0^2 y - 2xydy\, dx = \boxed{0} \tag{\%t75}$$

Graphics

(%i76)
```
wxdraw3d(nticks=100,proportional_axes=xyz,line_width=3,
font="Courier-Bold",font_size=20,view=[65,140],
color=cyan,
apply(parametric_surface,append(S,[x,0,1,y,0,2])),
color=blue,
apply(parametric,append(C_1,[t,0,1])),
apply(parametric,append(C_2,[t,0,1])),
color=red,
apply(parametric,append(C_3,[t,0,1])),
apply(parametric,append(C_4,[t,0,1])),
color=black,
apply(label,[append(["A"],A)]),
apply(label,[append(["B"],B)]),
apply(label,[append(["P"],P)]),
apply(label,[append(["Q"],Q)]))$
```



(%t76)