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('cartan,'version)=false then load(cartan)$
```

(%i15)
```
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 $\mathbb{R}^3$

```
(%i16) ζ:[x,y,z]$
(%i17) dim:length(ζ)$
(%i18) scalefactors(ζ)$
(%i19) init_cartan(ζ)$
```

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

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

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

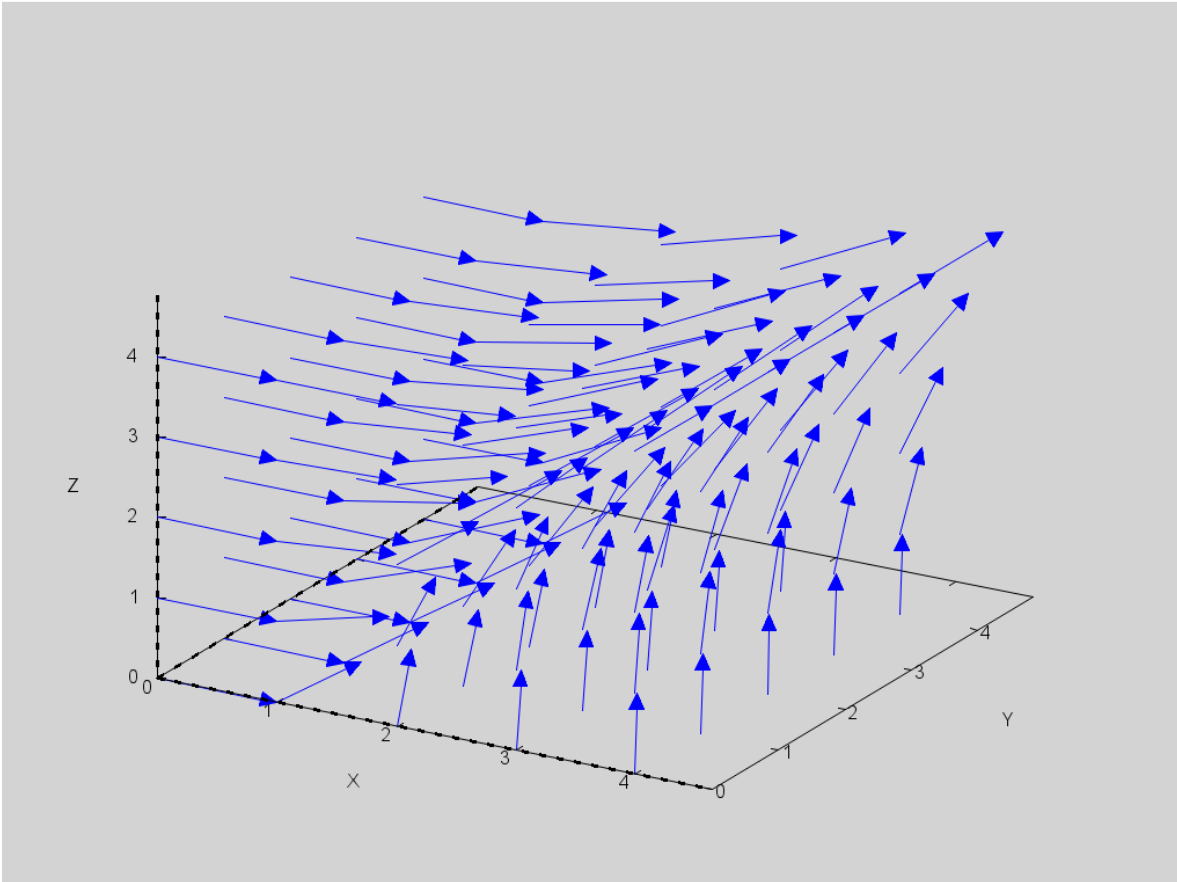
```
(%i21) ldisplay(F:[exp(z),x*y*z,x^3])$
```

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

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

(%i27) `ldisplay(curlF:ev(express(curl(F)),diff))$`

$$curlF = [-xy, e^z - 3x^2, yz] \tag{\%t27}$$

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

(%i28) `ldisplay(divF:ev(express(div(F)),diff))$`

$$divF = xz \tag{\%t28}$$

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

(%i29) `ldisplay(α:edit(F.cartan_basis))$`

$$\alpha = x^3\, dz + xyz\, dy + e^z\, dx \tag{\%t29}$$

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

(%i30) `ldisplay(dα:edit(ext_diff(α)))$`

$$d\alpha = -xy\, dy\, dz + \left(3x^2 - e^z\right)\, dx\, dz + yz\, dx\, dy \tag{\%t30}$$

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

(%i31)
```
ldisplay(β:F[1]*cartan_basis[2]~cartan_basis[3]+
F[2]*cartan_basis[3]~cartan_basis[1]+
F[3]*cartan_basis[1]~cartan_basis[2])$
```

$$\beta = e^z\, dy\, dz - xyz\, dx\, dz + x^3\, dx\, dy \tag{\%t31}$$

(%i32) `ldisplay(ω:factor(edit(β~dx+β~dy+β~dz)))$`

$$\omega = \left(e^z + xyz + x^3\right)\, dx\, dy\, dz \tag{\%t32}$$

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

(%i33) `ldisplay(dβ:edit(ext_diff(β)))$`

$$d\beta = xz\, dx\, dy\, dz \tag{\%t33}$$

(%i34) `diff(ζ,z)|(diff(ζ,y)|(diff(ζ,x)|dβ));`

$$xz \tag{\%o34}$$

End Points

```
(%i38) A:[1,0,0]$B:[1,2,0]$P:[0,2,1]$Q:[0,0,1]$
```

Trajectories and their derivatives

```
(%i46) 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

```
(%i50) 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

```
(%i51) ldisplay(I_1+I_2+I_3+I_4=box(ev(I_1+I_2+I_3+I_4,integrate)))$
```

$$\int_0^1 (1-t)^3 - e^t dt + \int_0^1 e^{1-t} - t^3 dt = \boxed{0} \qquad (\%t51)$$

Line integrals according to Differential Forms

```
(%i55) I_1:'integrate(C\'_1|ev(α,map("=",ζ,C_1)),t,0,1)$
       I_2:'integrate(C\'_2|ev(α,map("=",ζ,C_2)),t,0,1)$
       I_3:'integrate(C\'_3|ev(α,map("=",ζ,C_3)),t,0,1)$
       I_4:'integrate(C\'_4|ev(α,map("=",ζ,C_4)),t,0,1)$
```

Total line integral according to Differential Forms

```
(%i56) ldisplay(I_1+I_2+I_3+I_4=box(ev(I_1+I_2+I_3+I_4,integrate)))$
```

$$\int_0^1 -e^t - t^3 + 3t^2 - 3t + 1 dt + \int_0^1 e^{1-t} - t^3 dt = \boxed{0} \qquad (\%t56)$$

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

(%i57) `S:[x,y,1-x]$`

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

(%i58) `ldisplay(N:mycross(diff(S,x),diff(S,y)))$`

$$N = [1, 0, 1] \qquad (\%t58)$$

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

(%i59) `ldisplay(curlFoS:subst(map("=",ζ,S),curlF))$`

$$curlFoS = [-xy, e^{1-x} - 3x^2, (1-x)\, y] \qquad (\%t59)$$

Integrand according to Vector Calculus

(%i60) `ldisplay(integrand:expand(curlFoS.N))$`

$$integrand = y - 2xy \qquad (\%t60)$$

Integrand according to Differential Forms

(%i61) `ldisplay(integrand:diff(S,y)|(diff(S,x)|ev(dα,map("=",ζ,S))))$`

$$integrand = y - 2xy \qquad (\%t61)$$

Surface integral

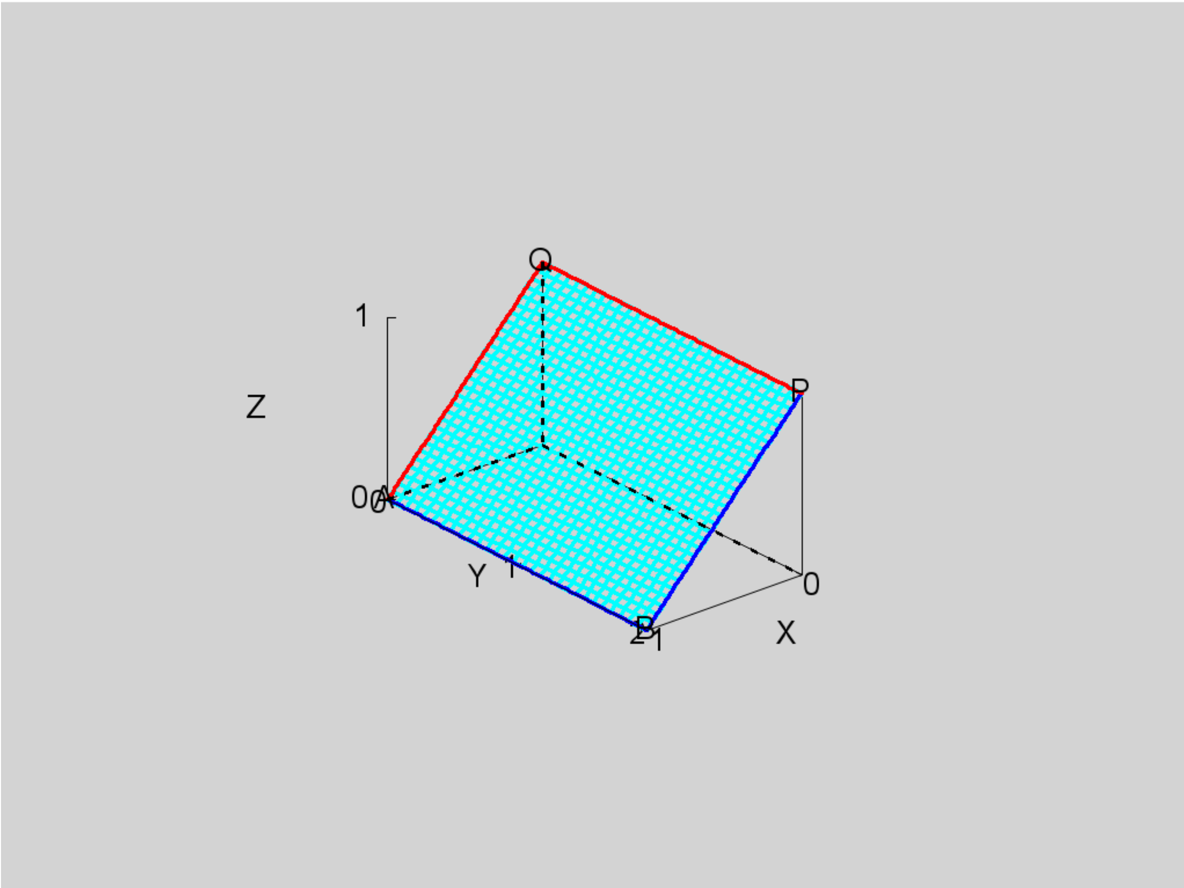
(%i62) `I:'integrate('integrate(integrand,y,0,2),x,0,1)$`

(%i63) `ldisplay(I=box(ev(I,integrate)))$`

$$\int_0^1 \int_0^2 y - 2xydy\, dx = (0) \qquad (\%t63)$$

Graphics

(%i64)
```
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)]))$
```



(%t64)