2D and 3D Measurements

©1999 Bill Davis, Horacio Porta and Jerry Uhl
Produced by Bruce Carpenter Published by Math Everywhere, Inc.
www.matheverywhere.com

VC.11 3D Flow Along
Basics

B.1) Measuring flow along a 3D curve

□B.1.a)

How do you measure the flow of a 3-dimensional vector field
Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}
along a curve C given in parametric form
P[t] = {x[t], y[t], z[t]}
with a ≤ t ≤ b?

□ **Answer:**

Just about the same way you do it in 2D.

You measure the net flow of Field[x, y, z] along C by calculating the 3D path integral

$$\begin{aligned}&\int_C \text{Field}\,.\,\text{unittan}\,ds\\ &= \int_C m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz\\ &= \int_C (m[x[t], y[t], z[t]]\,x'[t]\\ &\qquad + n[x[t], y[t], z[t]]\,y'[t]\\ &\qquad + p[x[t], y[t], z[t]]\,z'[t])\,dt\end{aligned}$$

If this path integral calculates out as positive, then the net flow of Field[x, y, z] along C is in the direction specified by the parameterization.

If this path integral calculates out as negative, then the net flow of Field[x, y, z] along C is against the direction specified by the parameterization.

□B.1.b)

Here is a 3D vector field:

```
Clear[Field, m, n, p, x, y, z]
m[x_, y_, z_] = y^2 z^3;
n[x_, y_, z_] = 1/3 x y z^3;
p[x_, y_, z_] = 1/2 x y^2 z^2;
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]}
```

$\{y^2 z^3,\ \frac{1}{3} x y z^3,\ \frac{1}{2} x y^2 z^2\}$

Here's a curve C in three dimensions:

```
Clear[t]
{x[t_], y[t_], z[t_]} = {t, t + Sin[1.6 t], t - 1}
```

```
{t, t + Sin[1.6 t], -1 + t}
```

This curve is given to start at:

```
a = 0.6;
start = {x[a], y[a], z[a]}
```

```
{0.6, 1.41919, -0.4}
```

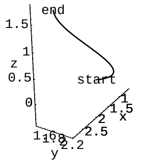
And this curve is given to end at:

```
b = 2.8;
end = {x[b], y[b], z[b]}
```

```
{2.8, 1.82688, 1.8}
```

Here's a look at C:

```
Clear[P, t, Cplot, labels, setup]
P[t_] = {x[t], y[t], z[t]};
Cplot = ParametricPlot3D[
  Evaluate[P[t]], {t, a, b}, DisplayFunction → Identity];
labels =
 {Graphics3D[Text["start", start]], Graphics3D[Text["end", end]]};

setup = Show[Cplot, labels, ViewPoint → CMView,
  Axes → Automatic, AxesLabel → {"x", "y", "z"}, Boxed → False,
  DisplayFunction → $DisplayFunction];
```



Measure the net flow of Field[x, y, z] along C, and interpret the result.

□ **Answer:**

The measurement

$$\begin{aligned}&\int_C \text{Field}\,.\,\text{unittan}\,ds\\ &= \int_C m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz\\ &= \int_C (m[x[t], y[t], z[t]]\,x'[t]\\ &\qquad + n[x[t], y[t], z[t]]\,y'[t]\\ &\qquad + p[x[t], y[t], z[t]]\,z'[t])\,dt\end{aligned}$$

measures the flow of the vector field

Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}

along C:

```
NIntegrate[m[x[t], y[t], z[t]] x′[t] +
  n[x[t], y[t], z[t]] y′[t] + p[x[t], y[t], z[t]] z′[t],
 {t, a, b}]
```

```
16.2563
```

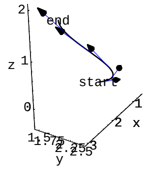
Positive.

This means that the net flow of Field[x, y, z] along C is fairly strong in the direction of the parameterization of C.

To see what this means, just plot C and some of its unit tangent vectors.

```
Clear[unittan]
unittan[t_] = TrigExpand[P′[t] / Sqrt[P′[t] . P′[t]]];
```

```
jump = (b - a)/5;
Show[
 setup, Table[Arrow[unittan[t], Tail → P[t]], {t, a, b - jump, jump}]];
```



There you go with the flow.

B.2) The curl of a 3D vector field

Start with a cleared 3D vector field:

```
Clear[Field, m, n, p, x, y, z]
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]}
```

```
{m[x, y, z], n[x, y, z], p[x, y, z]}
```

The curl, curlField[x, y, z], of Field[x, y, z] is given by:

You can detect some cyclic patterns here.

```
Clear[curlField]
curlField[x_, y_, z_] = {D[p[x, y, z], y] - D[n[x, y, z], z],
  D[m[x, y, z], z] - D[p[x, y, z], x], D[n[x, y, z], x] - D[m[x, y, z], y]}
```

$\{-n^{(0,0,1)}[x, y, z] + p^{(0,1,0)}[x, y, z],\ m^{(0,0,1)}[x, y, z] - p^{(1,0,0)}[x, y, z],\ -m^{(0,1,0)}[x, y, z] + n^{(1,0,0)}[x, y, z]\}$

That's quite a pill to swallow, but the computer will swallow it for you.
You'll want to use it to help finger a 3D vector field.

□B.2.a) Using curlField[x, y, z] to help finger a 3D vector field.

Stick the tail of a unit vector V at a point {a, b, c}.
Push your finger onto the point of V so that V spikes through the center of the bones of your finger, and the tip of your finger is at {a, b, c}.
How do you use
curlField[a, b, c]
to determine whether the tip of your finger feels a net clockwise or counterclockwise swirl resulting from the flow of the given 3D vector field
Field[x, y, z]?

□**Answer:**

Here's how it works:

→ If curlField[a, b, c] . V > 0,

then you feel Field[x, y, z] swirling in the counterclockwise way at the tip of your finger.

→ If curlField[a, b, c] . V < 0,

then you feel Field[x, y, z] swirling in the clockwise way at the tip of your finger.

→ If curlField[a, b, c] . V = 0,

then you feel no net swirl at all.

Try it out for a given vector field, a given unit vector V, and a given point {a, b, c}:

```
Clear[Field, curlField, unitvector, s, t, x, y, z, m, n, p];
unitvector[s_, t_] := N[{Sin[s] Cos[t], Sin[s] Sin[t], Cos[s]}]
m[x_, y_, z_] = x - 2 z;
n[x_, y_, z_] = y - 2 x^2;
p[x_, y_, z_] = z - 2 y;
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]};
curlField[x_, y_, z_] =
 {D[p[x, y, z], y] - D[n[x, y, z], z], D[m[x, y, z], z] - D[p[x, y, z], x],
  D[n[x, y, z], x] - D[m[x, y, z], y]};
curlField[0, 2, 1] . unitvector[π/3, π/4]
```

```
-2.44949
```

Negative.

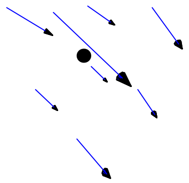
This tells you that when you push your finger onto

V = unitvector[$\frac{\pi}{3}$, $\frac{\pi}{4}$]

so that V spikes through the center of your finger and the tip of your finger is at {0, 2, 1}, then your finger feels a net CLOCKWISE swirl.

Confirm this by plotting what's happening near {0, 2, 1} using the plotting option ViewPoint → V to look from the tip of V to the tail of V at the point {0, 2, 1}:

```
point = {0, 2, 1};
V = unitvector[π/3, π/4];
h = 0.3;
scalefactor = 0.08;
fieldplot =
 Table[Arrow[Field@@(point + {x, y, z}), Tail → N[point + {x, y, z}],
   VectorColor → Blue, ScaleFactor → scalefactor],
  {x, -h, h, 2 h}, {y, -h, h, 2 h}, {z, -h, h, 2 h}];
pointplot = Graphics3D[{PointSize[0.07], Point[point]}];

swirlplot = Show[pointplot, fieldplot, PlotRange → All,
  Boxed → False, ViewPoint → V];
```



Put the tip of your finger at the plotted point on the screen and feel that net clockwise flow!

Just as the calculation predicted!

Try it for a different point and a different unit vector:

```
curlField[0, 0, 0] . unitvector[3π/2, π/6]
```

```
2.73205
```

Positive.

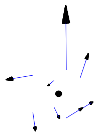
This tells you that when you push your finger onto

V = unitvector[$\frac{3\pi}{2}$, $\frac{\pi}{6}$]

so that V spikes though the center of your finger and the tip of your finger is at {0, 0, 0}, then your finger feels a net COUNTERCLOCKWISE swirl.

Confirm with a plot:

```
point = {0, 0, 0};
V = unitvector[3π/2, π/6];
h = 0.2;
scalefactor = 0.4;
fieldplot =
 Table[Arrow[Field@@(point + {x, y, z}), Tail → N[point + {x, y, z}],
   VectorColor → Blue, ScaleFactor → scalefactor],
  {x, -h, h, 2 h}, {y, -h, h, 2 h}, {z, -h, h, 2 h}];
pointplot = Graphics3D[{PointSize[0.07], Point[point]}];

swirlplot = Show[pointplot, fieldplot, PlotRange → All,
  Boxed → False, ViewPoint → V];
```



Put the tip of your finger at the plotted point on the screen and feel that net clockwise flow!

Math happens again.

B.3) Stokes's formula for using the curl to measure the swirl of a vector field in 3D

In 2D, everyone likes the form of the Gauss-Green formula that tells you that if

Field[x, y] = {m[x, y], n[x, y]}

and R is a region in two dimensions with boundary curve C, then

$$\int\int_R \text{rotField}\, dA$$
$$= \oint_C \text{Field}\,.\,\text{unittan}\, ds$$
$$= \oint_C m[x, y]\, dx + n[x, y]\, dy$$
= net flow of Field[x, y] along C.

Stokes's formula is the 3D analogue of this formula. Stokes's formula says that if you go with a 3D vector field

Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}

and R is a surface in three dimensions with boundary curve C, then

$$\int\int_R \text{curlField}\,.\,\text{topunitnormal}\, dA$$
$$= \oint_C \text{Field}\,.\,\text{unittan}\, ds$$
$$= \oint_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz$$
= net flow of Field[x, y, z] along C.

Before you can make complete sense of this formula, you have to deal with a few objections.

□B.3.a.i) Two objections.

→ Objection 1:

The integral

$$\oint_C \text{Field}\,.\,\text{unittan}\, ds$$

insists on a counterclockwise parameterization for the boundary curve C. But counterclockwise parameterization makes no sense in three dimensions, so Stokes's formula

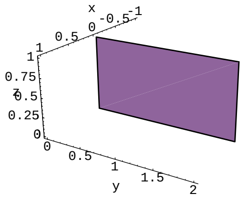
$$\int\int_R \text{curlField}\,.\,\text{topunitnormal}\, dA = \oint_C \text{Field}\,.\,\text{unittan}\, ds$$

makes no sense.

→ Objection 2:

Look at this piece of the yz-plane:

```
yzplanepiece =
  Graphics3D[Polygon[{{0, 0, 0}, {0, 2, 0}, {0, 2, 1}, {0, 0, 1}}]];
Show[yzplanepiece, Axes → True, AxesLabel → {"x", "y", "z"},
  ViewPoint → CMView, Boxed → False, PlotRange → All,
  DisplayFunction → $DisplayFunction];
```



Because this surface sits in the yz-plane, you can't say whether the front side or the back side is the top side.

Calculating

$$\int\int_R \text{curlField}\,.\,\text{topunitnormal}\, dA$$

depends on knowing which side of the surface is the top side. So again Stokes's formula

$$\int\int_R \text{curlField}\,.\,\text{topunitnormal}\, dA = \oint_C \text{Field}\,.\,\text{unittan}\, ds$$

makes no sense.

How do you get around these legitimate objections?

□ **Answer:**

Taken separately, each of these two objections ruins any possibility of making sense out of Stokes's formula, but if you deal with both of them simultaneously, then you can get a full understanding of what Stokes's formula says.

First, you pick one side of the surface at your own pleasure and call it the top side.

```
The side you pick for the top side may not be the same
  as the side designated the top side by the person
       sitting at the computer next to you.
```

Once you pick the top side of R, then you agree that you are going in the counterclockwise direction on C if, as you walk around C, R is on your left. With these two agreements carved in stone, Stokes's formula

$$\int\int_R \text{curlField}\,.\,\text{topunitnormal}\, dA = \oint_C \text{Field}\,.\,\text{unittan}\, ds$$
$$(= \int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz)$$

for a given 3D vector field

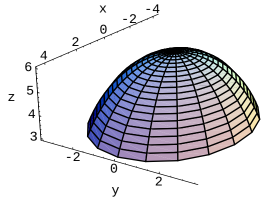
Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}

and a given surface R with boundary curve C makes perfect sense provided you agree to parameterize C in the counterclockwise direction dictated by your choice of the top side of the surface according to the agreement above.

□B.3.a.ii) The designated top side.

Here is a surface R which is a cap of an ellipsoid:

```
Clear[x, y, z, s, t]
{x[s_, t_], y[s_, t_], z[s_, t_]} =
 {5 Sin[s] Cos[t], 4 Sin[s] Sin[t], 6 Cos[s]};
{slow, shigh} = {0, π/3};
surfaceplot = ParametricPlot3D[{x[s, t], y[s, t], z[s, t]},
   {s, slow, shigh}, {t, 0, 2 π}, Boxed → False, PlotRange → All,
   ViewPoint → CMView, AxesLabel → {"x", "y", "z"}];
```



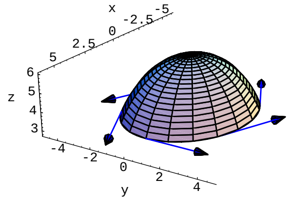
A parameterization of the boundary curve C is:

```
Clear[P]
P[t_] = {x[shigh, t], y[shigh, t], z[shigh, t]}
```

$$\left\{\frac{5}{2}\sqrt{3}\ \text{Cos}[t],\ 2\sqrt{3}\ \text{Sin}[t],\ 3\right\}$$

Here are the surface R, its boundary curve C, and a few tangent vectors reflecting the direction of this parameterization of C:

```
Cplot = ParametricPlot3D[
   Evaluate[P[t]], {t, 0, 2 π}, DisplayFunction → Identity];
tangents = Table[Arrow[P′[t], Tail → P[t]], {t, 0, 2 π, π/3}];
Show[surfaceplot, Cplot, tangents, Boxed → False,
 PlotRange → All, AxesLabel → {"x", "y", "z"}, ViewPoint → CMView,
 DisplayFunction → $DisplayFunction];
```



If you want to call this parameterization of boundary curve C counterclockwise in accordance with the agreement above, then which side of the cap must you designate as the top side?

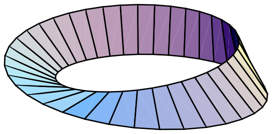
□ **Answer:**

Walk around the boundary in the direction indicated by the tangents. As you walk, your left foot hits on the visible side (the high side). This is your designated top side.

□B.3.a.iii) A third objection:

Take a look at this sorry excuse for a surface:

```
Clear[P, x, y, z, t, unittan, mainunitnormal, binormal, moebius]
P[t_] = 2 {-Cos[t], Sin[t], 0};
unittan[t_] = TrigExpand[P′[t] / √(P′[t] . P′[t])];
mainunitnormal[t_] =
 unittan′[t] / (√TrigExpand[unittan′[t] . unittan′[t]]);
binormal[t_] = TrigExpand[unittan[t] × mainunitnormal[t]];
moebius[s_, t_] =
 P[t] + s (Sin[t/2] mainunitnormal[t] - Cos[t/2] binormal[t]);
ParametricPlot3D[Evaluate[moebius[s, t]], {s, -0.5, 0.5}, {t, -π, π},
 PlotPoints → {2, 40}, ViewPoint → CMView, PlotRange → All,
 Boxed → False, Axes → False];
```



This surface, which folks like to call a "Moebius strip," is unfortunate enough to have only one side. How the hell are you going to designate a top side for a surface that has only one side?

□**Answer:**

You can't.

It's impossible to choose a top side.

The upshot:

It's impossible to use Stokes's formula for a surface that has only one side.

□B.3.a.iv)

What else do you have to worry about?

□**Answer:**

Stokes's formula works only for surfaces that are not the complete boundary of a solid region.

For example, Stokes's formula is OK if R is a cap of a sphere, but Stokes's formula does not work if R is the whole sphere. This is the case simply because a whole sphere does not have a boundary curve, but any cap of a sphere does have a boundary curve.

□B.3.b.i) Stokes's formula helps to explain why the finger test works.

Take a 3D vector field Field[x, y, z].
Stick the tail of a unit vector V at the point $\{x_0, y_0, z_0\}$.
Push your finger onto V so that V spikes through the center of your finger and the tip of your finger is at $\{x_0, y_0, z_0\}$.
Why does the tip of your finger feel a counterclockwise swirl if
 $\text{curlField}[x_0, y_0, z_0] . V > 0$?
Why does the tip of your finger feel a clockwise swirl if
 $\text{curlField}[x_0, y_0, z_0] . V < 0$?

□**Answer:**

Put a plane with normal vector V through the point

$\{x_0, y_0, z_0\}$.

Put a small circle C with center at $\{x_0, y_0, z_0\}$ in this plane. Call R the part of this plane that is within the small circle. Designate the top side by taking the direction V points. This gives you

V = topunitnormal

for this little surface R.

Stokes's formula tells you:

net flow of Field[x, y, z] along C

$$= \int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz$$

$$= \int\int_R \text{curlField} . \text{topunitnormal}\, dA$$

$$= \int\int_R \text{curlField} . V\, dA.$$

Here's the kicker:

If

$\text{curlField}[x_0, y_0, z_0] . V > 0$,

then

curlField[x, y, z] . V > 0

for all {x, y, z}'s close to $\{x_0, y_0, z_0\}$. So if C is small enough so that

curlField[x, y, z] . V > 0

at all {x, y, z}'s on R and inside C, then you see that

net flow of Field[x, y, z] along C

$$= \int\int_R \text{curlField} . V\, dA > 0.$$

So the net flow of Field[x, y, z] along such a small circle C is counterclockwise.

This is why the tip of your finger feels a counterclockwise swirl when you stick the tail of V at the point $\{x_0, y_0, z_0\}$, and push your finger onto V so that V spikes through the center of your finger and the tip of your finger is at $\{x_0, y_0, z_0\}$.

Similarly, if $\text{curlField}[x_0, y_0, z_0] . V < 0$ and you stick the tail V at the point $\{x_0, y_0, z_0\}$, and push your finger onto V so that V spikes through the center of your finger and the tip of your finger is at $\{x_0, y_0, z_0\}$, then your fingertip feels a clockwise swirl.

□B.3.b.ii)

Given a 3D vector field Field[x, y, z], when you stick the tail of a unit vector V at the point {x, y, z} and push your finger onto V so that V spikes through the center of your finger and the tip of your finger is at {x, y, z}, then what unit vector V do you want to use so that your fingertip feels the greatest possible counterclockwise swirl?
What does the direction of curlField[x, y, z] tell you?
What does the length of curlField[x, y, z] tell you?

□**Answer:**

The swirl your fingertip feels is measured by

curlField[x, y, z] . V

= ||curlField[x, y, z]|| ||V|| Cos[angle between]

= ||curlField[x, y, z]|| Cos[angle between]

because V is a unit vector.

This is biggest when

Cos[angle between] = 1 = Cos[0].

So you feel the biggest counterclockwise swirl when you take the unit vector V to be in the same direction as curlField[x, y, z].

In other words, to feel the most counterclockwise swirl, you take

$$V = \frac{\text{curlField}[x,y,z]}{\sqrt{\text{curlField}[x,y,z].\text{curlField}[x,y,z]}}$$

The upshots:

→ The direction of curlField[x, y, z] gives the direction of the axis of the actual swirl of Field[x, y, z] at {x, y, z}.

→ The length of curlField[x, y, z] measures how vigorous the swirl at {x, y, z} is.

Check this out:

```
Clear[x, y, z, m, n, p, Field, curlField]
m[x_, y_, z_] = x + 2 z^2;
n[x_, y_, z_] = y - 2 x^2;
p[x_, y_, z_] = z + 4 y^2;
```

```
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]};
curlField[x_, y_, z_] = {D[p[x, y, z], y] - D[n[x, y, z], z],
  D[m[x, y, z], z] - D[p[x, y, z], x], D[n[x, y, z], x] - D[m[x, y, z], y]}
```

```
{8 y, 4 z, -4 x}
```

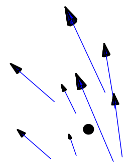
To feel the greatest possible counterclockwise swirl of this vector field at {1, 1, 1}, you should use:

```
point = {1, 1, 1};
                    curlField[1, 1, 1]
V = ──────────────────────────────────────────
     √curlField[1, 1, 1] . curlField[1, 1, 1]
```

$$\left\{\sqrt{\frac{2}{3}}, \frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}\right\}$$

Take a look down the vector V from its tip to its tail at {1, 1, 1}:

```
h = 0.3;
scalefactor = 0.1;
fieldplot =
 Table[Arrow[N[Field @@ (point + {x, y, z})], Tail → N[point + {x, y, z}],
   VectorColor → Blue, ScaleFactor → scalefactor],
  {x, -h, h, 2 h}, {y, -h, h, 2 h}, {z, -h, h, 2 h}];
pointplot = Graphics3D[{PointSize[0.07], Point[point]}];

fingerplot = Show[pointplot, fieldplot, Boxed → False, ViewPoint → V];
```



Powerfully counterclockwise just as you expected.

Math really happens.

□B.3.b.iii)

Where did the name "curl" come from?

□**Answer:**

Take a look at the end of the answer to part ii) above.

□B.3.c.i)

Take a surface R with boundary curve C. Designate a top side of R.
The path integral

$$\int_C \text{Field} . \text{unittan}\, ds$$
$$= \int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz$$

measures the net flow of a given 3D vector field

Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}

along C.
If you know that

curlField[x, y, z] . topunitnormal[x, y, z] > 0

at all points {x, y, z} on the surface R, then how do you also know that the net flow of Field[x, y, z] along the boundary curve C is counterclockwise?

□**Answer:**

You calculate the flow of Field[x, y, z] along C by calculating

$$\int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz.$$

Stokes's formula tells you that this is the same as calculating

$$\int\int_R \text{curlField} . \text{topunitnormal}\, dA$$

Here's the kicker:

If

curlField[x, y, z] . topunitnormal[x, y, z] > 0

at all points of R, then the

flow of Field[x, y, z] along C

$$= \int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz$$
$$= \int\int_R \text{curlField} . \text{topunitnormal}\, dA > 0,$$

because curlField[x, y, z] . topunitnormal[x, y, z] > 0 for all {x, y, z}'s on R.

So, if you know that

curlField[x, y, z] . topunitnormal[x, y, z] > 0

for all {x, y, z}'s on R, then you know that the flow – along – C measurement is positive.

This means that the net flow of Field[x, y, z] along the boundary curve C of R is counterclockwise.

□B.3.c.ii)

Take a surface R with boundary curve C. If you know that

curlField[x, y, z] = 0

at all points {x, y, z} on the surface R, then how do you also know that the net flow of Field[x, y, z] along C is 0?

□**Answer:**

You calculate the net flow of Field[x, y, z] along C by calculating

$$\int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz.$$

Stokes's formula tells you that this is the same as calculating

$$\int\int_R \text{curlField} . \text{topunitnormal}\, dA.$$

Here's the kicker:

If

curlField[x, y, z] = 0

at all points of R, then the

net flow of Field[x, y, z] along C

$$= \int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz$$
$$= \int\int_R \text{curlField} . \text{topunitnormal}\, dA$$
$$= \int\int_R 0\, dA$$
$$= 0,$$

because curlField[x, y, z] = 0 for all {x, y, z}'s on R.

So

curlField[x, y, z] = 0

for all {x, y, z}'s on R guarantees that the flow – along – C measurement is 0. This means that the net flow of Field[x, y, z] along the boundary curve C of R is 0.

□B.3.c.iii) Irrotational vector fields.

Why do the fancy folks call a vector field Field[x, y, z] with

curlField[x, y, z] = 0

at all points {x, y, z} irrotational?

□**Answer:**

As you saw above if

curlField[x, y, z] = 0

for all points {x, y, z}, then the flow of Field[x, y, z] along any closed curve is 0.

This rules out any vortex or other swirl.

B.4) An attempt to explain Stokes's formula as an outgrowth of the 2D Gauss-Green formula

```
      You are perfectly free to call this explanation
               esoteric, elite, abstruse,
         or any other adjective you want to go with.
        A detailed understanding is not required here.
What is hoped for is that you begin to see that someone can
    explain the nitty-gritty behind Stokes's formula
 as an outgrowth of the basic 2D Gauss-Green formula which,
     in turn, is an outgrowth of the fundamantal formula.
```

□B.4.a)

Explain the reasoning behind Stokes's formula.

□**Answer:**

Ultimately Stoke's formula is a consequence of the Gauss-Green formula in two dimensions which, in turn, is a consequence of the fundamental formula of calculus in one variable.

The fundamental formula of calculus strikes again!

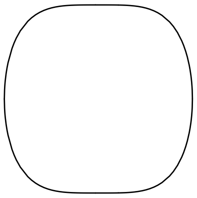
You can explain Stokes's formula as a rather easy consequence of the Gauss-Green formula, but there is a lot of bookkeeping (bean counting).

The first step is to reduce the explanation of the overall formula to an explanation for a very simple surface.

□Step One: Reduction to parallelograms.

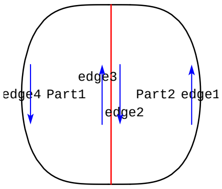
Look at a picture and think of it as a view of a surface R in three dimensions.

```
            Don't read the code; just look at the picture.
surface = Plot[{√(1 - x^4), -√(1 - x^4)}, {x, -1, 1}, AspectRatio → Automatic,
  Axes → None];
```



Split the surface into two pieces and label:

```
            Don't read the code; just look at the picture.
splitter = Graphics[{RGBColor[1, 0, 0], Line[{{0, 1}, {0, -1}}]}];
labels = Graphics[{Text["Part1", {-1/2, 0}], Text["Part2", {1/2, 0}],
    Text["edge1", {1, 0}], Text["edge2", {.15, -.2}],
    Text["edge3", {-.15, .2}], Text["edge4", {-1, 0}]}];
arrows = {Vector[{.9, -1/3}, {.9, 1/3}], Vector[{-.9, 1/3}, {-.9, -1/3}],
  Vector[{.1, 1/3}, {.1, -1/3}], Vector[{-.1, -1/3}, {-.1, 1/3}]};
Show[{surface, splitter, labels, arrows},
 PlotRange → {{-1.2, 1.2}, {-1.1, 1.1}}];
```



Now instead of one surface, you've got two surfaces with their own boundaries. To integrate a function around the boundary of Part2, you integrate out around edge1, and then down along edge2. To integrate around the boundary of Part1, you integrate up along edge3 and out around edge4.

Look at what happens:

$$\int_{\text{edge1}} \text{F . unittan } ds + \int_{\text{edge2}} \text{F . unittan } ds$$
$$+ \int_{\text{edge3}} \text{F . unittan } ds + \int_{\text{edge4}} \text{F . unittan } ds$$
$$= \int_{\text{edge1}} \text{F . unittan } ds + \int_{\text{edge4}} \text{F . unittan } ds,$$

because edge2 and edge3 are the same curves going in opposite directions.

What does it say?

It says that adding the integrals around the boundaries of the surfaces Part1 and Part2 gives the same result as integrating around the boundary of the whole region R before you split it.

And what does that say to you?

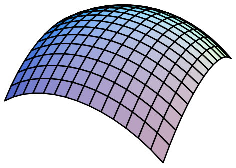
It tells you that to explain Stokes's formula

$$\int\int_R \text{curlField . topunitnormal } dA = \oint_C \text{Field . unittan } ds,$$

you can cut the original surface R up into small pieces, and if you can explain the formula for the small pieces and their boundary curves, then you can add the formulas for the small pieces to arrive at a full explanation of the formula for the whole original surface R.

Here is another view of R but this time R is cut up into little pieces:

```
            Don't read the code; just look at the picture.
Plot3D[√(1 - x^2 - y^2), {x, -1/2, 1/2}, {y, -1/2, 1/2}, Boxed → False,
 Axes → None];
```



So explaining the formula for the whole surface is the same as justifying it for each of the little parallelogram-like parts you see above and adding the results together.

Now things are a lot simpler because each of the smaller parts is very close to being a parallelogram. So if you can explain why Stokes's formula holds for a parallelogram, you'll have explained Stokes's formula for R.

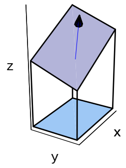
□Step Two: Explaining Stokes's formula for a parallelogram.

You're not out of the woods yet. This is the part that is complicated ... not hard, just complicated. It's really a job for those bean counting dweebs who like to do a lot of bookkeeping. Recall what needs to be explained:

$$\int\int_R \text{curlField . topunitnormal } dA = \oint_C \text{Field . unittan } ds,$$

where the surface R is a parallelogram, and its boundary C is the edge of the parallelogram. Here's a picture with normal vector included:

```
            Don't read the code; just look at the picture.
a = 1/3 {1, 1, 1};
aa = 1/3 {1, 1, 0};
b = {-1/2, 0, 1/4};
bb = {-1/2, 0, 0};
c = {0, -1/3, 1/8};
cc = {0, -1/3, 0};
para = Graphics3D[
  {RGBColor[1, 0.5, 0], Polygon[{a, a + b, a + b + c, a + c, a}]}];
base = Graphics3D[{RGBColor[0.8, 0.8, 0.8],
   Polygon[{aa, aa + bb, aa + bb + cc, aa + cc, aa}]}];
normal = Vector[a + (b + c)/3, a + (b + c)/3 + 2 b × c];
lines = Graphics3D[{Line[{a, aa}], Line[{a + b, aa + bb}],
   Line[{a + c, aa + cc}], Line[{a + b + c, aa + bb + cc}]}];

Show[lines, normal, para, base, ViewPoint → CMView, Ticks → None,
 Axes → Automatic, AxesLabel → {"x", "y", "z"}, Boxed → False,
 PlotRange → All];
```



The upper parallelogram is the region R you are working with, and the lower parallelogram is the shadow of the upper parallelogram on the xy-plane. When you parameterize to integrate around the boundary C on the top, you will be directly over a counterclockwise parameterization of the boundary of the lower parallelogram in the xy-plane. Ultimately you are going to use the two-dimensional Gauss-Green formula applied to the lower parallelogram which will be called R_{uv} when you parameterize with x = u and y = v. The boundary of R_{uv} will be called C_{uv}.

Now set up some calculations:

```
Clear[x, y, z, m, n, p, Field]
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]}
{m[x, y, z], n[x, y, z], p[x, y, z]}
```

The upper parallelogram with the normal vector pointing up out of it lies in a plane. Write the equation for this plane as

$$z = c + a\,x + b\,y,$$

or parametrically by:

```
Clear[u, v, a, b, c]
{x[u_, v_], y[u_, v_], z[u_, v_]} = {u, v, c + a x[u, v] + b y[u, v]}
{u, v, c + a u + b v}
```

As {u, v} varies through R_{uv}, {x[u, v], y[u, v], z[u, v]} varies through R.

Look at the normal vector:

```
Clear[normal]
normal[u_, v_] =
  D[{x[u, v], y[u, v], z[u, v]}, u] × D[{x[u, v], y[u, v], z[u, v]}, v]
{-a, -b, 1}
```

Because this normal has a positive third slot, it points up, just as in the picture above.

```
Clear[topunitnormal]
topunitnormal[u_, v_] = normal[u, v] / √(normal[u, v] . normal[u, v])
```

$$\left\{-\frac{a}{\sqrt{1+a^2+b^2}},\ -\frac{b}{\sqrt{1+a^2+b^2}},\ \frac{1}{\sqrt{1+a^2+b^2}}\right\}$$

This means that when you calculate the path integral you will have to parameterize the boundary curve of the upper parallelogram so that as you walk in the direction of the parameterization, your left foot touches the high side of the surface R.

Now see what

$$\int\int_R \text{curlField} \,.\, \text{topunitnormal}\, dA$$

$$= \int\int_{R_{uv}} \text{curlField}[x[u, v], y[u, v], z[u, v]] \,.\, \text{topunitnormal}[u, v]\ SA_{xyz}[u,$$

looks like:

```
Clear[SAxyz]
SAxyz[u_, v_] = √(normal[u, v] . normal[u, v])
√(1 + a^2 + b^2)
```

```
Clear[curlField]
curlField[x_, y_, z_] = {D[p[x, y, z], y] - D[n[x, y, z], z],
   D[m[x, y, z], z] - D[p[x, y, z], x], D[n[x, y, z], x] - D[m[x, y, z], y]}
{-n^(0,0,1)[x, y, z] + p^(0,1,0)[x, y, z], m^(0,0,1)[x, y, z] - p^(1,0,0)[x, y, z],
 -m^(0,1,0)[x, y, z] + n^(1,0,0)[x, y, z]}
```

The integrand is:

```
integrand =
 Simplify[curlField[x[u, v], y[u, v], z[u, v]] . topunitnormal[u, v]
   SAxyz[u, v]]
-b m^(0,0,1)[u, v, c + a u + b v] + a n^(0,0,1)[u, v, c + a u + b v] -
 m^(0,1,0)[u, v, c + a u + b v] - a p^(0,1,0)[u, v, c + a u + b v] +
 n^(1,0,0)[u, v, c + a u + b v] + b p^(1,0,0)[u, v, c + a u + b v]
```

Here's the bookkeeping trick:

Once you have done this, you can reduce the whole explanation to the two dimensional Gauss-Green formula. Start by making a couple of off-the-wall but delicious choices for new 2-dimensional functions.

```
Clear[mm, nn, FF, rotFF]
mm[u_, v_] = m[x[u, u], y[u, v], z[u, v]] + a p[x[u, v], y[u, v], z[u, v]];
nn[x_, y_] = n[x[u, v], y[u, v], z[u, v]] + b p[x[u, v], y[u, v], z[u, v]];
FF[u_, v_] = {mm[u, v], nn[u, v]}
{m[u, v, c + a u + b v] + a p[u, v, c + a u + b v],
 n[u, v, c + a u + b v] + b p[u, v, c + a u + b v]}
```

Look at rotFF[u, v]:

```
rotFF[u_, v_] = Expand[D[nn[u, v], u] - D[mm[u, v], v]]
-b m^(0,0,1)[u, v, c + a u + b v] + a n^(0,0,1)[u, v, c + a u + b v] -
 m^(0,1,0)[u, v, c + a u + b v] - a p^(0,1,0)[u, v, c + a u + b v] +
 n^(1,0,0)[u, v, c + a u + b v] + b p^(1,0,0)[u, v, c + a u + b v]
```

rotFF[u, v] looks like the quantity you got for the integrand up above. Is it?

```
integrand == rotFF[u, v]
True
```

You bet it is!

This tells you that

$$\int\int_R \text{curlField} \,.\, \text{topunitnormal}\, dA$$
$$= \int\int_{R_{uv}} \text{rotFF}[u, v]\, du\, dv$$

and the two dimensional Gauss-Green formula says that

$$\int\int_{R_{uv}} \text{rotFF}[u, v]\, du\, dv$$
$$= \oint_{C_{uv}} \text{mm}[u, v]\, du + \text{nn}[u, v]\, dv$$

where C_{uv} is the boundary of R_{uv} .

Now

$$\oint_{C_{uv}} \text{mm}[u, v]\, du + \text{nn}[u, v]\, dv$$

$$= \oint_{C_{uv}} (m[x[u, v], y[u, v], z[u, v]] + a\,p[x[u, v], y[u, v], z[u, v]])\, du + (n[x[u, v], y[u, v], z[u, v]] + b\,p[x[u, v], y[u, v], z[u, v]])\, dv$$

$$= \oint_{C_{uv}} m[x[u, v], y[u, v], z[u, v]]\, du + n[x[u, v], y[u, v], z[u, v]]\, dv + p[x[u, v], y[u, v], z[u, v]]\,(a\, du + b\, dv)\,.$$

Now look at:

```
z[u, v]
c + a u + b v
```

Read off

$$dz = a\, du + b\, dv,$$

and plug into the above, remembering that x = u and y = v, to realize that when you parameterize and calculate

$$\oint_C \text{Field} \,.\, \text{unittan}\, ds$$
$$\oint_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz,$$

then you get

$$\oint_{C_{uv}} \text{mm}[u, v]\, du + \text{nn}[u, v]\, dv$$
$$= \int\int_{R_{uv}} \text{rotFF}[u, v]\, du\, dv$$
$$= \int\int_R \text{curlField} \,.\, \text{topunitnormal}\, dA.$$

And now the formula

$$\oint_C \text{Field} \,.\, \text{unittan}\, ds = \int\int_R \text{curlField} \,.\, \text{topunitnormal}\, dA$$

has been explained.

Review the explanation to see why Stokes's formula is just a dressed-up version of the basic two dimensional Gauss-Green formula.

VC.11 3D Flow Along Tutorials

T.1) Stokes's formula in theory and practice

Stokes's formula says that if

Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]},

and if R is a surface in three dimensions with boundary curve C, and you have made the right specifications of the top side and of what counterclockwise means, then you are guaranteed that

$$\int\int_R \text{curlField . topunitnormal}\, dA$$
$$= \oint_C \text{Field . unittan}\, ds$$
$$= \oint_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz$$
$$= \text{net flow of Field[x, y, z] along C.}$$

□T.1.a)

How good a theoretical tool is Stokes's formula?

□Answer:

It's a superb theoretical tool.

Check out B.3) to see how Stokes's formula confirms that the counterclockwise swirl of a 3D vector field Field[x, y, z] in the direction of a unit vector V at a point $\{x_0, y_0, z_0\}$ is measured by

$\text{curlField}[x_0, y_0, z_0]\,.\,V$.

This, in turn, tells you that if you stick the tail of

$\text{curlField}[x_0, y_0, z_0]$

at the point $\{x_0, y_0, z_0\}$, then you get the axis of the direction of the greatest counterclockwise swirl of Field[x, y, z] at the point $\{x_0, y_0, z_0\}$.

This is the direction of the curl of the flow represented by Field[x, y, z].

Stokes's formula is a great theoretical tool that sets up concrete calculations.

□T.1.b.i)

How good a calculational tool is Stoke's formula?

□Answer:

Look at Stokes's formula again.

$$\int\int_R \text{curlField . topunitnormal}\, dA$$
$$= \oint_C \text{Field . unittan}\, ds$$
$$= \oint_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz$$
$$= \text{net flow of Field[x, y, z] along C.}$$

If curlField[x, y, z] calculates out to {0, 0, 0}, then you don't have to do any extra work to learn that the flow of Field[x, y, z] along any closed curve is 0.

In fact, if you have a closed curve C, think of it as wire and throw a big bed sheet over the wire. The part of the sheet inside C defines a surface R with boundary curve C.

If curlField[x, y, z] is always {0, 0, 0}, you get

$$0 = \int\int_R 0\, dA$$
$$= \int\int_R \{0, 0, 0\}\text{ . topunitnormal}\, dA$$
$$= \int\int_R \text{curlField . topunitnormal}\, dA$$
$$= \oint_C \text{Field . unittan}\, ds$$
$$= \oint_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz$$
$$= \text{net flow of Field[x, y, z] along C.}$$

This means that if curlField[x, y, z] is always {0, 0, 0}, then the net flow of Field[x, y, z] along any closed curve is 0.

And Stokes's formula gives you this juicy calculational fact with no work on your part.

You've got to agree that if curlField[x, y, z] is always 0, then Stokes's formula is an excellent calculational tool.

□T.1.b.ii)

What happens when curlField[x, y, z] does not calculate out to {0, 0, 0}?

□Answer:

Take another look at Stokes's formula:

$$\int\int_R \text{curlField . topunitnormal}\, dA$$
$$= \oint_C \text{Field . unittan}\, ds$$
$$= \oint_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz$$
$$= \text{net flow of Field[x, y, z] along C.}$$

If curlField[x, y, z] is not always 0, then you are probably better off ignoring Stokes's formula and calculating the path integral

$$\oint_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz$$

directly.

Reason:

Most surface integrals are harder to calculate than most path integrals.

□T.1.c)

Here are two vector fields in three dimensions:

```
Clear[x, y, z, m1, n1, p1, m2, n2, p2, Field1, Field2]
Field1[x_, y_, z_] = E^-y {2 x Sin[x^2 + z^2], Cos[x^2 + z^2], 2 z Sin[x^2 + z^2]};
{m1[x_, y_, z_], n1[x_, y_, z_], p1[x_, y_, z_]} = Field1[x, y, z]
```

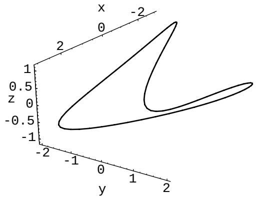
```
{2 E^-y x Sin[x^2 + z^2], E^-y Cos[x^2 + z^2], 2 E^-y z Sin[x^2 + z^2]}
```

```
Field2[x_, y_, z_] = {(x - 1)^2, x y, (z - y)^2};
{m2[x_, y_, z_], n2[x_, y_, z_], p2[x_, y_, z_]} = Field2[x, y, z]
```

```
{(-1 + x)^2, x y, (-y + z)^2}
```

Here is a closed curve C in three dimensions:

```
Clear[t, x, y, z]
{x[t_], y[t_], z[t_]} = {3 Sin[2 t], 2 Cos[t], Sin[t] + 0.2 Cos[3 t]};
{tlow, thigh} = {0, 2 π};
Clear[P]
P[t_] = {x[t], y[t], z[t]};
Cplot = ParametricPlot3D[Evaluate[P[t]], {t, tlow, thigh},
  ViewPoint → CMView, Boxed → False, Axes → Automatic,
  AxesLabel → {"x", "y", "z"}];
```



Measure the flow of Field1[x, y, z] along this curve, and then measure the flow of Field2[x, y, z] along this curve.

□Answer:

Take a look at the curls:

```
Clear[curlField1, x, y, z]
curlField1[x_, y_, z_] = {D[p1[x, y, z], y] - D[n1[x, y, z], z],
  D[m1[x, y, z], z] - D[p1[x, y, z], x],
  D[n1[x, y, z], x] - D[m1[x, y, z], y]}
```

```
{0, 0, 0}
```

Ah-ha!

With no further work, you know that the flow of Field1[x, y, z] along this curve is 0. In fact, thanks to Stokes's formula, you also know that the flow of Field1[x, y, z] along any other closed curve is also 0.

Now look at the second vector field and its curl:

```
Clear[curlField2]
curlField2[x_, y_, z_] = {D[p2[x, y, z], y] - D[n2[x, y, z], z],
  D[m2[x, y, z], z] - D[p2[x, y, z], x],
  D[n2[x, y, z], x] - D[m2[x, y, z], y]}
```

```
{-2 (-y + z), 0, y}
```

No such luck this time; curlField2[x, y, z] does not calculate out to {0, 0, 0}. To measure the flow of Field2[x, y, z] along the given curve, calculate the path integral

$$\oint_C \text{Field2}\,.\,\text{unittan}\,ds$$
$$= \oint_C \text{m2}[x, y, z]\,dx + \text{n2}[x, y, z]\,dy + \text{p2}[x, y, z]\,dz$$
$$= \int_{\text{tlow}}^{\text{thigh}} (\text{m2}[x[t], y[t], z[t]]\,x'[t]$$
$$+ \text{n2}[x[t], y[t], z[t]]\,y'[t]$$
$$+ \text{p2}[x[t], y[t], z[t]]\,z'[t])\,dt$$

```
{x[t_], y[t_], z[t_]} = P[t];
NIntegrate[m2[x[t], y[t], z[t]] x′[t] +
  n2[x[t], y[t], z[t]] y′[t] + p2[x[t], y[t], z[t]] z′[t],
 {t, tlow, thigh}, AccuracyGoal → 2]
```

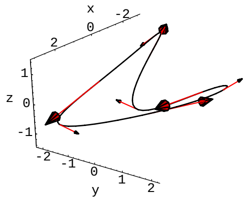
```
-18.8496
```

Negative.

This means the net flow of Field2[x, y, z] along C is fairly strong in the direction opposite the direction of the parameterization of C.

To see what this means, just plot C and some of its tangent vectors.

```
scalefactor = 0.5;
Show[Cplot, Table[Arrow[P′[t], Tail → P[t], VectorColor → Red,
    ScaleFactor → scalefactor], {t, tlow, thigh, (thigh - tlow)/8}],
 ViewPoint → CMView, Boxed → False, Axes → Automatic,
 AxesLabel → {"x", "y", "z"}];
```



The net flow of Field2[x, y, z] along C is in the direction opposite the direction indicated by the tangent vectors above.

T.2) Path independence

□T.2.a)

You've got a vector field

Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}

and you are fortunate enough to find that curlField[x, y, z] is always 0.

You also have a curve C_1 with parameterization

$P_1[t] = \{x_1[t], y_1[t], z_1[t]\}$ with $a_1 \le t \le b_1$,

and you have a curve C_2 with parameterization

$P_2[t] = \{x_2[t], y_2[t], z_2[t]\}$ with $a_2 \le t \le b_2$.

In addition, you are given that

$P_1[a_1] = P_2[a_2]$ and $P_1[b_1] = P_2[b_2]$;

so that both curves start at the same point and end at the same point.

Say how you know in advance with no calculation that

$$\int_{C_1} m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz$$
$$= \int_{a_1}^{b_1} (m[x_1[t], y_1[t], z_1[t]]\,x_1'[t]$$
$$+ n[x_1[t], y_1[t], z_1[t]]\,y_1'[t]$$
$$+ p[x_1[t], y_1[t], z_1[t]]\,z_1'[t])\,dt$$

is the same as

$$\int_{C_2} m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz$$
$$= \int_{a_2}^{b_2} (m[x_2[t], y_2[t], z_2[t]]\,x_2'[t]$$
$$+ n[x_2[t], y_2[t], z_2[t]]\,y_2'[t]$$
$$+ p[x_2[t], y_2[t], z_2[t]]\,z_2'[t])\,dt$$

□**Answer:**

$$\int_{C_1} m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz$$
$$- \int_{C_2} m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz$$
$$= \oint_C m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz,$$

where C is the closed curve made by joining C_1 and C_2.

Think of C as rigid wire, and hang a sheet over the wire being careful that the sheet doesn't pass through any singularities. Call R the part of the sheet bounded by C. Remembering that curlField[x, y, z] = {0, 0, 0} at all points on R, use Stokes's formula to see that

$$0 = \int\int_R \{0, 0, 0\}\,.\,\text{topunitnormal}\,dA$$
$$= \int\int_R \text{curlField}\,.\,\text{topunitnormal}\,dA$$
$$= \oint_C \text{Field}\,.\,\text{unittan}\,ds$$
$$= \oint_C m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz$$
$$= \int_{C_1} m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz$$
$$- \int_{C_2} m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz$$

This tells you that

$$\int_{C_1} m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz$$
$$= \int_{C_2} m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz$$

T.3) Work

□T.3.a)

What is work?

□Answer:

What's work for some folks is fun for other folks. Trig identities come to mind; they always seem to be work for the math student but fun for the math teacher. The physicists have their own technical notion of work.

Just as in two dimensions, the physicists envision a 3D vector field

Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}

to represent the force (push) on an object positioned at {x, y, z}. In this interpretation, the vector field Field[x, y, z] is called a force field.

Next, the physicists say that if an object goes along a 3D curve C specified by

P[t] = {x[t], y[t], z[t]} with a ≤ t ≤ b,

then the work done by a force field Field[x, y, z] on the object during the duration of the object's trip is measured by the path integral

$$\begin{aligned}&\int_0^{\text{length}} \text{Field}\,.\,\text{unittan}\, ds\\ &= \int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz\\ &= \int_a^b (m[x[t], y[t], z[t]]\, x'[t]\\ &\quad + n[x[t], y[t], z[t]]\, y'[t]\\ &\quad + p[x[t], y[t], z[t]]\, z'[t])\, dt\end{aligned}$$

This might not be your own notion of work, but notice that it is not you who does this work.

The physicists have a pretty good reason for using this word for this measurement. Think of it this way:

If at a point on the trip,

→ Field . unittan > 0, then at this point the force field Field[x, y, z] is working this much to push the object and the object does no work at all.

But if

→ Field . unittan < 0, then at this point the force field Field[x, y, z] is against the object; the object is working this much, and the force field Field[x, y, z] does not work at all.

With this in mind, you can think of

$$\int_0^{\text{length}} \text{Field}\,.\,\text{unittan}\, ds$$

as a measurement of

the force field' s work – the object' s work.

If

$$\int_0^{\text{length}} \text{Field}\,.\,\text{unittan}\, ds > 0,$$

then the force field Field[x, y, z] did most of the work during the object's trip.

But if

$$\int_0^{\text{length}} \text{Field}\,.\,\text{unittan}\, ds > 0,$$

then the object did most of the work during the object's trip.

If you agree that the object's work represents negative work for the force field, then you'll probably agree that

$$\begin{aligned}&\int_0^{\text{length}} \text{Field}\,.\,\text{unittan}\, ds\\ &= \int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz\\ &= \int_a^b (m[x[t], y[t], z[t]]\, x'[t]\\ &\quad + n[x[t], y[t], z[t]]\, y'[t]\\ &\quad + p[x[t], y[t], z[t]]\, z'[t])\, dt\end{aligned}$$

is a reasonable measurement of the work done by the force field during the duration of the trip.

□T.3.b)

Is there a difference between net flow along the curve and work?

□Answer:

Mathematically there is no difference, because they are both measured by the same integral.

The difference is the interpretation.

When you are talking about flow – along – the – curve, then you envision

Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}

as the velocity vector at {x, y, z} of a fluid flow.

The fluid is flowing and the curve is just sitting there.

When you are talking about work, then you envision

Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}

as the force on an object at {x, y, z} moving on a curve.

This time the force field is just sitting there and the object is moving on the curve.

□T.3.c)

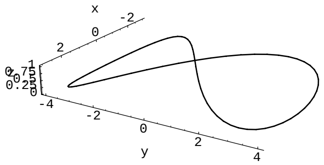
Here's a force field:

```
Clear[x, y, z, m, n, p, Field]
Field[x_, y_, z_] = E^-y {1 - x, y, z - 2};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
```

```
{E^-y (1 - x), E^-y y, E^-y (-2 + z)}
```

Here's a space curve:

```
Clear[t, P]
{x[t_], y[t_], z[t_]} = {3 Sin[2 t], 4 Sin[t], Cos[t]^2};
P[t_] = {x[t], y[t], z[t]};
{tlow, thigh} = {0, 2 π};
Cplot = ParametricPlot3D[Evaluate[P[t]], {t, tlow, thigh},
  ViewPoint → CMView, Boxed → False, Axes → Automatic,
  AxesLabel → {"x", "y", "z"}];
```



An object starts at:

```
P[tlow]
```

```
{0, 0, 1}
```

And makes one trip around the curve, ending the trip at:

```
P[thigh]
```

```
{0, 0, 1}
```

Which way should you send the object around C to make the force field do most of the work?

□Answer:

Take a look at the curl:

```
Clear[curlField]
curlField[x_, y_, z_] = {D[p[x, y, z], y] - D[n[x, y, z], z],
  D[m[x, y, z], z] - D[p[x, y, z], x], D[n[x, y, z], x] - D[m[x, y, z], y]}
```

```
{-E^-y (-2 + z), 0, E^-y (1 - x)}
```

Too bad!

If curlField[x, y, z] had calculated out to be {0, 0, 0}, the work measurement would have been 0; it wouldn't have mattered which way you sent the object.

To measure the work done by the force field in moving the object along C in the given parameterization of C, calculate the path integral

$$\begin{aligned}&\int_C \text{Field}\,.\,\text{unittan}\, ds\\ &= \int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz\\ &= \int_{\text{tlow}}^{\text{thigh}} (m[x[t], y[t], z[t]]\, x'[t]\\ &\quad + n[x[t], y[t], z[t]]\, y'[t]\\ &\quad + p[x[t], y[t], z[t]]\, z'[t])\, dt\end{aligned}$$

```
NIntegrate[m[x[t], y[t], z[t]] x′[t] +
  n[x[t], y[t], z[t]] y′[t] + p[x[t], y[t], z[t]] z′[t],
 {t, tlow, thigh}, AccuracyGoal → 2]
```

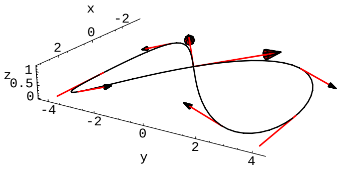
−242.111

Very negative.

This means that to make the force field do the lion's share of the work, the object should move in the direction OPPOSITE to the parameterization of C.

To see which direction that is, just plot C and a few of its NEGATIVE tangent vectors.

```
scalefactor = 0.5;
Show[Cplot, Table[Arrow[(-P′[t]), Tail → P[t], VectorColor → Red,
                                                    thigh - tlow
    ScaleFactor → scalefactor], {t, tlow, thigh, ------------}],
                                                         8
 ViewPoint → CMView, Boxed → False, Axes → Automatic,
 AxesLabel → {"x", "y", "z"}];
```



This is the way you should send the object around C to make the force field do most of the work.

If you send the object the other way, then the force field will work against the object's motion.

How do you try to come up with a function f[x, y, z] with

Field[x, y, z] = gradf[x, y, z]?

□ **Answer:**

You just take the procedure that worked so well in two dimensions, and make appropriate adjustments:

The goal is to come up with a function f[x, y, z] with

gradf[x, y, z] = Field[x, y, z].

To do this, fix any point {a, b, c} you like, and parameterize a line C (or other curve) running from {a, b, c} to the variable point {x, y, z}:

{0, 0, 0} is usually a good choice for {a, b, c}.

```
Clear[X, Y, Z, t]
{a, b, c} = {0, 0, 0};
fixedpoint = {a, b, c};
variablepoint = {X, Y, Z};
tlow = 0;
thigh = 1;
{x[t_], y[t_], z[t_]} = fixedpoint + t (variablepoint - fixedpoint)
```

{t X, t Y, t Z}

To get a function f[x, y, z] with

gradf[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]},

all you gotta do is set

$$f[x, y, z] = \int_C m[x, y, z]\, dx + n[x, y, z]\, dy + p[x, y, z]\, dz,$$

where C is the line (or other curve) running from the fixed point {a, b, c} to the variable point {x, y, z}:

```
Clear[f]
                  thigh
f[X_, Y_, Z_] = ∫      (m[x[t], y[t], z[t]] x′[t] +
                  tlow
      n[x[t], y[t], z[t]] y′[t] + p[x[t], y[t], z[t]] z′[t])
    ⅆt
```

$-2 + E^{4 Z} + X\,(-2 + Y Z) + \text{Cos}[5 Y Z]$

Try it out:

T.4) The gradient test in 3D

□T.4.a)

The gradient test for a 2D vector field

Field[x, y] = {m[x, y], n[x, y]}

is run by checking whether

rotField[x, y] = D[n[x, y], x] − D[m[x, y], y] = 0.

If Field[x, y] passes this test, then you know how to go about trying to find a function f[x, y] with

gradf[x, y] = Field[x, y].

What is the three-dimensional version of the gradient test for a 3D vector field?

□ **Answer:**

No surprise here; you just check whether

curlField[x, y, z] = {0, 0, 0}.

To see why, check on what the curl of any gradient field is:

```
Clear[f, gradf, Field, x, y, z, m, n, p, curlField]
gradf[x_, y_, z_] =
  {D[f[x, y, z], x], D[f[x, y, z], y], D[f[x, y, z], z]};
Field[x_, y_, z_] = gradf[x, y, z];
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z];
curlField[x_, y_, z_] = {D[p[x, y, z], y] - D[n[x, y, z], z],
   D[m[x, y, z], z] - D[p[x, y, z], x], D[n[x, y, z], x] - D[m[x, y, z], y]}
```

{0, 0, 0}

And that's that.

□T.4.b)

Here's a 3D vector field which passes the gradient test:

```
Clear[Field, x, y, z, m, n, p, curlField]
Field[x_, y_, z_] =
  {-2 + y z, x z - 5 z Sin[5 y z], 4 E^(4 z) + x y - 5 y Sin[5 y z]};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z];
curlField[x_, y_, z_] = {D[p[x, y, z], y] - D[n[x, y, z], z],
   D[m[x, y, z], z] - D[p[x, y, z], x], D[n[x, y, z], x] - D[m[x, y, z], y]}
```

{0, 0, 0}

```
Clear[gradf]
gradf[x_, y_, z_] = {D[f[x, y, z], x], D[f[x, y, z], y], D[f[x, y, z], z]}
```

$\{-2 + y z,\ x z - 5 z\,\text{Sin}[5 y z],\ 4 E^{4 z} + x y - 5 y\,\text{Sin}[5 y z]\}$

Compare:

```
Field[x, y, z]
```

$\{-2 + y z,\ x z - 5 z\,\text{Sin}[5 y z],\ 4 E^{4 z} + x y - 5 y\,\text{Sin}[5 y z]\}$

```
gradf[x, y, z] == Field[x, y, z]
```

True

Great.

This tells you that

gradf[x, y, z] = Field[x, y, z],

just as you wanted.

See what happens when you go with a different fixed point:

```
Clear[X, Y, Z, t, f]
{a, b, c} = {3, 0, 1};
fixedpoint = {a, b, c};
variablepoint = {X, Y, Z};
tlow = 0;
thigh = 1;
{x[t_], y[t_], z[t_]} = fixedpoint + t (variablepoint - fixedpoint);
                  thigh
f[X_, Y_, Z_] = ∫      (m[x[t], y[t], z[t]] x′[t] +
                  tlow
      n[x[t], y[t], z[t]] y′[t] + p[x[t], y[t], z[t]] z′[t])
    ⅆt
```

$5 - E^{4} + E^{4 Z} + X\,(-2 + Y Z) + \text{Cos}[5 Y Z]$

Try it out:

```
Clear[gradf]
gradf[x_, y_, z_] = {D[f[x, y, z], x], D[f[x, y, z], y], D[f[x, y, z], z]}
```

$\{-2 + y z,\ x z - 5 z\,\text{Sin}[5 y z],\ 4 E^{4 z} + x y - 5 y\,\text{Sin}[5 y z]\}$

Compare:

```
Field[x, y, z]
```

$\{-2 + y z,\ x z - 5 z\,\text{Sin}[5 y z],\ 4 E^{4 z} + x y - 5 y\,\text{Sin}[5 y z]\}$

```
ExpandAll[gradf[x, y, z] - Field[x, y, z], Trig → True]
```

{0, 0, 0}

Yes!

gradf[x, y, z] = Field[x, y, z]

It feels great!

Each time you change the fixed point {a, b, c}, you make a different function f[x, y, z] whose gradient is Field[x, y, z].

Of course, this cannot be expected to work when you start with a 3D vector field that does not pass the 3D gradient test.

VC.11 3D Flow Along
Give it a Try!

G.1) Fingering a 3D vector field

□G.1.a.i)

Here's a 3D vector field:

```
Clear[x, y, z, m, n, p, Field]
Field[x_, y_, z_] = {(y - z)^2, (z - x)^2, (x - y)^2};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
{(y - z)^2, (-x + z)^2, (x - y)^2}
```

Stick the tail of the following unit vectors V at the indicated points {a, b, c}. Push your finger onto V so that V spikes though the center of your finger and the tip of your finger is at {a, b, c}.
Use curlField[a, b, c] to help you report on whether the tip of your finger reveals a counterclockwise net swirl, a net clockwise swirl, or no swirl at all.
Illustrate your results with a plot.

□G.1.a.i)

```
{a, b, c} = {1, 0, 3};
V = {-3, 2, 1}/Sqrt[3^2 + 2^2 + 1^2];
```

□G.1.a.ii)

```
{a, b, c} = {1.13, -2.67, -1.58};
V = {2, 8, 1}/Sqrt[2^2 + 8^2 + 1^2];
```

□G.1.a.iii)

```
{a, b, c} = {2.13, 1.38, 4.34};
V = {-6, 4.5, 1.5}/Sqrt[6^2 + 4.5^2 + 1.5^2];
```

□G.1.a.iv)

For the vector field Field[x, y, z] as set above, there are many points at which your finger will feel no net clockwise or counterclockwise swirl no matter how you align your finger. Describe these points.

□Tip:

They all lie on a certain line.
Your job is to say what this line is.

□G.1.b)

Here is another three dimensional vector field:

```
Clear[x, y, z, m, n, p, Field]
Field[x_, y_, z_] = {1.3 (y - z), 0.2 (z - x), 0.04 (x^2 - y)};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
{1.3 (y - z), 0.2 (-x + z), 0.04 (x^2 - y)}
```

Your job is to find the points at which this field rotates the very least.

□Big tip:

At a point {x, y, z}, the axis of the greatest counterclockwise swirl is in the direction of the unit vector

$$\text{Vbig} = \frac{\text{curlF[x,y,z]}}{\sqrt{\text{curlF[x,y,z].curlF[x,y,z]}}}.$$

The strength of the counterclockwise swirl in this direction is

$$\text{curlF[x, y, z]} \,.\, \text{Vbig}$$
$$= \frac{\text{curlF[x,y,z].curlF[x,y,z]}}{\sqrt{\text{curlF[x,y,z].curlF[x,y,z]}}}$$
$$= \sqrt{\text{curlF[x, y, z]} \,.\, \text{curlF[x, y, z]}}$$

So the points at which the field rotates the very least are the points at which

curlF[x, y, z] . curlF[x, y, z]

is as small as it can be.

□G.1.c)

Here is a new 3D vector field:

```
Clear[x, y, z, m, n, p, Field]
Field[x_, y_, z_] = {z x^2, x (y - 1)^2, 3 y z^2};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
{x^2 z, x (-1 + y)^2, 3 y z^2}
```

Take any three dimensional unit vector

V = {a, b, 0} with a > 0 and b > 0,

and put the tail of V at any unspecified point {x, y, z} with

x ≥ 0 and z ≥ 0.

Push your finger onto V so that V spikes though the center of your finger, and so the tip of your finger is at {x, y, z}. How do you know in advance that the tip of your finger will feel a net counterclockwise swirl?

G.2) Flow-along measurements and estimates

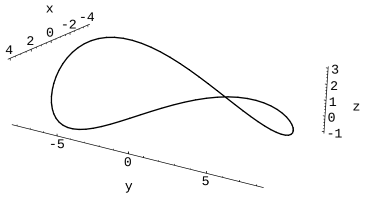
□G.2.a)

Here are two 3D vector fields:

```
Clear[x, y, z, m1, n1, p1, m2, n2, p2, Field1, Field2]
Field1[x_, y_, z_] = E^(-y^2) {x, y, z};
{m1[x_, y_, z_], n1[x_, y_, z_], p1[x_, y_, z_]} = Field1[x, y, z]
{E^(-y^2) x, E^(-y^2) y, E^(-y^2) z}
Field2[x_, y_, z_] = {y z, x z, x y};
{m2[x_, y_, z_], n2[x_, y_, z_], p2[x_, y_, z_]} = Field2[x, y, z]
{y z, x z, x y}
```

Here's a closed curve C in three dimensions:

```
Clear[t, P]
{x[t_], y[t_], z[t_]} = {4 Cos[t], 8 Sin[t], 2 Sin[2 t] + 1};
P[t_] = {x[t], y[t], z[t]};
Cplot =
 ParametricPlot3D[Evaluate[P[t]], {t, 0, 2 π}, ViewPoint → CMView,
  Boxed → False, Axes → Automatic, AxesLabel → {"x", "y", "z"}];
```



Measure the flow of Field1[x, y, z] along C, and then measure the flow of Field2[x, y, z] along C.
Describe the direction of the net flow of each vector field along C.

□G.2.b)

Take a 3D surface R with boundary curve C. If you are given a vector field Field[x, y, z] with the extra property that

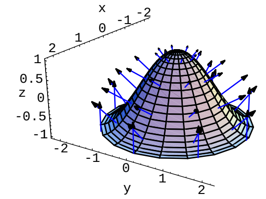
curlField[x, y, z]

with its tail at {x, y, z} is tangent to the surface R at all points {x, y, z} on the surface R, then how does Stokes's formula tell you that the net flow of Field[x, y, z] along C is 0?

□G.2.c.i)

Here's a surface shown with a selection of normal vectors:

```
Clear[x, y, z, s, t]
x[s_, t_] = 2 s Cos[t];
y[s_, t_] = 2 s Sin[t];
z[s_, t_] = Cos[π s];
{{slow, shigh}, {tlow, thigh}} = {{0, 1}, {0, 2 π}};
surface = ParametricPlot3D[Evaluate[{x[s, t], y[s, t], z[s, t]}],
  {s, slow, shigh}, {t, tlow, thigh}, DisplayFunction → Identity];
Clear[normal]
normal[s_, t_] =
 N[D[{x[s, t], y[s, t], z[s, t]}, s] × D[{x[s, t], y[s, t], z[s, t]}, t]];
sjump = 0.2;
tjump = π/4;
scalefactor = 0.2;
normals = Table[Arrow[normal[s, t],
   Tail → {x[s, t], y[s, t], z[s, t]}, ScaleFactor → scalefactor],
  {s, slow + sjump, shigh, sjump}, {t, tlow + tjump, thigh, tjump}];
Show[surface, normals, ViewPoint → CMView, Boxed → False,
 AxesLabel → {"x", "y", "z"}, DisplayFunction → $DisplayFunction];
```

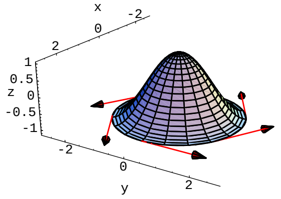


You come across a 3D vector field Field[x, y, z] and happen to realize that

curlField[x[s, t], y[s, t], z[s, t]] . normal[s, t] > 0

no matter what s and t you go with. This means that as you look down from the top of each normal, you see Field[x, y, z] swirling around the tail of each normal vector in the counterclockwise way. Give a common-sense explanation of why this guarantees that the net flow of Field[x, y, z] along the boundary curve of this surface is in the direction indicated by the following plot:

```
Clear[P]
P[t_] = {x[shigh, t], y[shigh, t], z[shigh, t]};
Cplot = ParametricPlot3D[
  Evaluate[P[t]], {t, tlow, thigh}, DisplayFunction → Identity];
scalefactor = 1;
tangents = Table[Arrow[scalefactor P′[t],
    Tail → P[t], VectorColor → Red], {t, 0, 2 π, π/3}];
Show[surface, Cplot, tangents, Boxed → False,
 PlotRange → All, ViewPoint → CMView, AxesLabel → {"x", "y", "z"},
 DisplayFunction → $DisplayFunction];
```

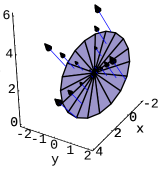


Back up your common-sense explanation with an explanation based on Stokes's formula.

□G.2.c.ii)

Here's a surface shown with a selection of normal vectors:

```
Clear[x, y, z, s, t, normal]
x[s_, t_] = 2 s Cos[t];
y[s_, t_] = 2 s Sin[t];
z[s_, t_] = 2 - x[s, t];
{{slow, shigh}, {tlow, thigh}} = {{0, 1}, {0, 2 π}};
surface = ParametricPlot3D[Evaluate[{x[s, t], y[s, t], z[s, t]}],
  {s, slow, shigh}, {t, tlow, thigh},
  PlotPoints → {2, Automatic}, DisplayFunction → Identity];
normal[s_, t_] =
 N[D[{x[s, t], y[s, t], z[s, t]}, s] × D[{x[s, t], y[s, t], z[s, t]}, t]];
sjump = 0.3;
tjump = π/2;
scalefactor = 0.5;
normals = Table[Arrow[normal[s, t],
    Tail → {x[s, t], y[s, t], z[s, t]}, ScaleFactor → scalefactor],
  {s, slow + 0.1, shigh, sjump}, {t, tlow + tjump, thigh, tjump}];

Show[surface, normals, ViewPoint → CMView, Boxed → False,
 AxesLabel → {"x", "y", "z"}, DisplayFunction → $DisplayFunction];
```



Here's a 3D vector field:

```
Clear[Field, m, n, p, x, y, z, curlField]
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = {z, x^3 + 2 x + y, z^2 + x};
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]}
```

```
{z, 2 x + x^3 + y, x + z^2}
```

```
curlField[x_, y_, z_] = {D[p[x, y, z], y] - D[n[x, y, z], z],
  D[m[x, y, z], z] - D[p[x, y, z], x], D[n[x, y, z], x] - D[m[x, y, z], y]}
```

```
{0, 0, 2 + 3 x^2}
```

Now look at this:

```
curlField[x[s, t], y[s, t], z[s, t]] . normal[s, t]
```

```
(4. s Cos[t]^2 + 4. s Sin[t]^2) (2 + 3 x[s, t]^2)
```

Remembering that 0 = slow ≤ s ≤ 1 = shigh, use the information immediately above to say which direction is the direction of the net flow of the given vector field on the boundary curve of the surface.

G.3) Work

□G.3.a)

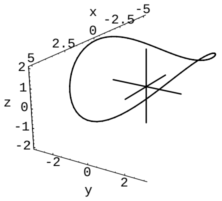
Here's a 3D force field:

```
Clear[x, y, z, m, n, p, Field]
Field[x_, y_, z_] = {z (x - 1)^2, x (y - 1)^2, 3 y z^2};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
```

```
{(-1 + x)^2 z, x (-1 + y)^2, 3 y z^2}
```

Here's a space curve:

```
Clear[t, P]
{x[t_], y[t_], z[t_]} = {5 Sin[t], 3 Cos[t], 2 Cos[t]^2};
P[t_] = {x[t], y[t], z[t]};
Cplot = ParametricPlot3D[
  Evaluate[P[t]], {t, 0, 2 π}, DisplayFunction → Identity];
h = 2;
coordinateaxes = {Graphics3D[Line[{{-h, 0, 0}, {h, 0, 0}}]],
  Graphics3D[Line[{{0, -h, 0}, {0, h, 0}}]],
  Graphics3D[Line[{{0, 0, -h}, {0, 0, h}}]]};

Show[Cplot, coordinateaxes, ViewPoint → CMView,
 Boxed → False, Axes → Automatic, AxesLabel → {"x", "y", "z"},
 DisplayFunction → $DisplayFunction];
```



An object starts at

```
P[0]
```

```
{0, 3, 2}
```

and makes one trip around the curve, ending the trip at:

```
P[2 π]
```

```
{0, 3, 2}
```

Which way should you send the object around C to make the force field do most of the work? Or does it matter?

□G.3.b)

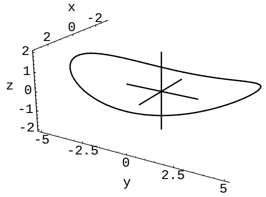
Here's another 3D force field:

```
Clear[x, y, z, m, n, p, Field]
Field[x_, y_, z_] = Sin[(x - 2)^2 + y^2 + (z - 1)^2] {x - 2, y, z - 1};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
```

```
{(-2 + x) Sin[(-2 + x)^2 + y^2 + (-1 + z)^2], y Sin[(-2 + x)^2 + y^2 + (-1 + z)^2],
 (-1 + z) Sin[(-2 + x)^2 + y^2 + (-1 + z)^2]}
```

Here's another space curve:

```
Clear[t, P]
{x[t_], y[t_], z[t_]} = {3 Cos[t], 5 Sin[t], Sin[t]^2};
P[t_] = {x[t], y[t], z[t]};
Cplot = ParametricPlot3D[
  Evaluate[P[t]], {t, 0, 2 π}, DisplayFunction → Identity];
h = 2;
coordinateaxes = {Graphics3D[Line[{{-h, 0, 0}, {h, 0, 0}}]],
  Graphics3D[Line[{{0, -h, 0}, {0, h, 0}}]],
  Graphics3D[Line[{{0, 0, -h}, {0, 0, h}}]]};

Show[Cplot, coordinateaxes, ViewPoint → CMView,
 Boxed → False, Axes → Automatic, AxesLabel → {"x", "y", "z"},
 DisplayFunction → $DisplayFunction];
```



An object starts at:

```
P[0]
```

```
{3, 0, 0}
```

And makes one trip around the curve, ending the trip at:

```
P[2 π]
```

```
{3, 0, 0}
```

Which way should you send the object around C to make the force field do most of the work? Or does it matter?

□G.3.c)

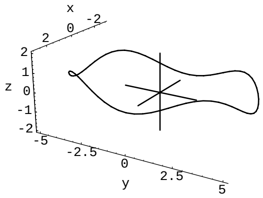
Here is a cleared 3D gradient force field:

```
Clear[x, y, z, m, n, p, gradfF, f]
gradf[x_, y_, z_] =
  {D[f[x, y, z], x], D[f[x, y, z], y], D[f[x, y, z], z]};
Field[x_, y_, z_] = gradf[x, y, z];
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
```

```
{f^(1,0,0)[x, y, z], f^(0,1,0)[x, y, z], f^(0,0,1)[x, y, z]}
```

Here is a space curve:

```
Clear[t, P]
{x[t_], y[t_], z[t_]} = {3 Cos[t], 5 Sin[t], Sin[2 t]^2};
P[t_] = {x[t], y[t], z[t]};
Cplot = ParametricPlot3D[
   Evaluate[P[t]], {t, 0, 2 π}, DisplayFunction → Identity];
h = 2;
coordinateaxes = {Graphics3D[Line[{{-h, 0, 0}, {h, 0, 0}}]],
   Graphics3D[Line[{{0, -h, 0}, {0, h, 0}}]],
   Graphics3D[Line[{{0, 0, -h}, {0, 0, h}}]]};

Show[Cplot, coordinateaxes, ViewPoint → CMView,
  Boxed → False, Axes → Automatic, AxesLabel → {"x", "y", "z"},
  DisplayFunction → $DisplayFunction];
```



An object starts at:

```
P[0]
```

```
{3, 0, 0}
```

And makes one trip around the curve, ending the trip at:

```
P[2 π]
```

```
{3, 0, 0}
```

Does it matter which way you send the object around C to make the force field do most of the work? Why?

G.4) The gradient test in 3D

Here are two 3D vector fields. One of these vector fields passes the gradient test and the other does not.

```
Clear[x, y, z, m1, n1, p1, m2, n2, p2, Field1, Field2]
Field1[x_, y_, z_] = {y + z, x - 2 z + 3, x - 2 y - 4 z + 3};
{m1[x_, y_, z_], n1[x_, y_, z_], p1[x_, y_, z_]} = Field1[x, y, z]
```

```
{y + z, 3 + x - 2 z, 3 + x - 2 y - 4 z}
```

```
Field2[x_, y_, z_] = {y + 2 z, x - 2 z + 3, x - 2 y + 3 z + 3};
{m2[x_, y_, z_], n2[x_, y_, z_], p2[x_, y_, z_]} = Field2[x, y, z]
```

```
{y + 2 z, 3 + x - 2 z, 3 + x - 2 y + 3 z}
```

□G.4.a.i)

Which of these two vector fields passes the three-dimensional gradient test?

□G.4.a.ii)

Come up with function f[x, y, z] whose gradient field is the same as the 3D vector field in part i) that passed the gradient test.

□G.4.b) Voltage.

Here is the 3D electric field resulting from a single point charge of strength q placed at the point {a, b, c}

```
Clear[Field, m, n, p, x, y, z, q, a, b, c]
point = {a, b, c};
Field[x_, y_, z_] = q ({x, y, z} - point)/((x - a)^2 + (y - b)^2 + (z - c)^2)^(3/2);
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
```

$$\left\{\frac{q\,(-a+x)}{\left((-a+x)^2+(-b+y)^2+(-c+z)^2\right)^{3/2}},\ \frac{q\,(-b+y)}{\left((-a+x)^2+(-b+y)^2+(-c+z)^2\right)^{3/2}},\ \frac{q\,(-c+z)}{\left((-a+x)^2+(-b+y)^2+(-c+z)^2\right)^{3/2}}\right\}$$

Confirm that Field[x, y, z] passes the gradient test, and try to come up a function f[x, y, z] with

gradf[x, y, z] = Field[x, y, z].

Some folks call f[x, y, z] a potential function for Field[x, y, z].

The function f[x, y, z] produces a measurement that is proportional to the voltage at {x, y, z} resulting from placing a point charge of strength q at the point {a, b, c} in the case that {a, b, c} ≠ {0, 0, 0}.

□Tip:

Make your fixed point = {0, 0, 0}.

G.5) Path independence and path dependence

□G.5.a)

The curve C_1 is parameterized to run on a straight line starting at {0, 0, 0} and ending at {1, 2, 1} as the parameter t runs from 0 to 1:

```
Clear[x1, y1, z1, t]
{x1[t_], y1[t_], z1[t_]} = {0, 0, 0} + t {1, 2, 1}
```

```
{t, 2 t, t}
```

```
startC1 = {x1[0], y1[0], z1[0]}
```

```
{0, 0, 0}
```

```
endC1 = {x1[1], y1[1], z1[1]}
```

```
{1, 2, 1}
```

The curve C_2 is also parameterized to run on a parabolic arc starting at {0, 0, 0} and ending at {1, 2, 1} as the parameter t runs from 0 to 1:

```
Clear[x2, y2, z2, t]
{x2[t_], y2[t_], z2[t_]} = {6 t^2 - 5 t, 2 t^2, t^2}
```

```
{-5 t + 6 t^2, 2 t^2, t^2}
```

```
startC2 = {x2[0], y2[0], z2[0]}
```

```
{0, 0, 0}
```

```
endC2 = {x2[1], y2[1], z2[1]}
```

```
{1, 2, 1}
```

The curve C_3 is also parameterized to run on another curve starting at {0, 0, 0} and ending at {1, 2, 1} as the parameter t runs from 0 to 1:

```
Clear[x3, y3, z3, t]
{x3[t_], y3[t_], z3[t_]} = {t, 2 Sin[π t/2]^2, t^2}
```

```
{t, 2 Sin[π t/2]^2, t^2}
```

```
startC3 = {x3[0], y3[0], z3[0]}
```

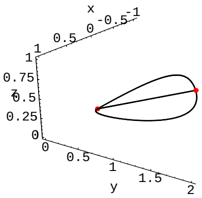
```
{0, 0, 0}
```

```
endC3 = {x3[1], y3[1], z3[1]}
```

```
{1, 2, 1}
```

Here is how they look:

```
Clear[P1, P2, P3]
P1[t_] = {x1[t], y1[t], z1[t]};
C1plot = ParametricPlot3D[
   Evaluate[P1[t]], {t, 0, 1}, DisplayFunction → Identity];
P2[t_] = {x2[t], y2[t], z2[t]};
C2plot = ParametricPlot3D[
   Evaluate[P2[t]], {t, 0, 1}, DisplayFunction → Identity];
P3[t_] = {x3[t], y3[t], z3[t]};
C3plot = ParametricPlot3D[
   Evaluate[P3[t]], {t, 0, 1}, DisplayFunction → Identity];
points = {Graphics3D[{Red, PointSize[0.02], Point[P1[0]]}],
   Graphics3D[{Red, PointSize[0.02], Point[P1[1]]}]};

Show[C1plot, C2plot, C3plot, points, ViewPoint → CMView, Boxed → False,
  AxesLabel → {"x", "y", "z"}, DisplayFunction → $DisplayFunction];
```



All three run from {0, 0, 0} to {1, 2, 1}.

Here are calculations of the three path integrals

$$\int_{C_1} (6\,x - 4\,y + 3\,z)\,dx - (4\,x + 4\,z)\,dy + (3\,x - 4\,y)\,dz$$

$$\int_{C_2} (6\,x - 4\,y + 3\,z)\,dx - (4\,x + 4\,z)\,dy + (3\,x - 4\,y)\,dz$$

$$\int_{C_3} (6\,x - 4\,y + 3\,z)\,dx - (4\,x + 4\,z)\,dy + (3\,x - 4\,y)\,dz$$

in order:

```
Clear[m, n, p, x, y, z]
m[x_, y_, z_] = 6 x - 4 y + 3 z;
n[x_, y_, z_] = -(4 x + 4 z);
p[x_, y_, z_] = 3 x - 4 y;
Chop[NIntegrate[m[x1[t], y1[t], z1[t]] x1′[t] +
    n[x1[t], y1[t], z1[t]] y1′[t] + p[x1[t], y1[t], z1[t]] z1′[t],
   {t, 0, 1}]]
-10.
```

```
Chop[NIntegrate[m[x2[t], y2[t], z2[t]] x2′[t] +
    n[x2[t], y2[t], z2[t]] y2′[t] + p[x2[t], y2[t], z2[t]] z2′[t],
   {t, 0, 1}]]
-10.
```

```
Chop[NIntegrate[m[x3[t], y3[t], z3[t]] x3′[t] +
    n[x3[t], y3[t], z3[t]] y3′[t] + p[x3[t], y3[t], z3[t]] z3′[t],
   {t, 0, 1}]]
-10.
```

Keep the same curves, but tweak the integrals slightly by putting in a little minus sign on the middle term.
Here are the resulting calculations:

$$\int_{C_1} (6x - 4y + 3z)\,dx + (4x + 4z)\,dy + (3x - 4y)\,dz$$
$$\int_{C_2} (6x - 4y + 3z)\,dx + (4x + 4z)\,dy + (3x - 4y)\,dz$$
$$\int_{C_3} (6x - 4y + 3z)\,dx + (4x + 4z)\,dy + (3x - 4y)\,dz$$

in order:

```
Clear[mm, nn, pp, x, y, z]
mm[x_, y_, z_] = 6 x - 4 y + 3 z;
nn[x_, y_, z_] = +(4 x + 4 z);
pp[x_, y_, z_] = 3 x - 4 y;
NIntegrate[mm[x1[t], y1[t], z1[t]] x1′[t] +
  nn[x1[t], y1[t], z1[t]] y1′[t] + pp[x1[t], y1[t], z1[t]] z1′[t],
 {t, 0, 1}]
6.
```

```
NIntegrate[mm[x2[t], y2[t], z2[t]] x2′[t] +
  nn[x2[t], y2[t], z2[t]] y2′[t] + pp[x2[t], y2[t], z2[t]] z2′[t],
 {t, 0, 1}]
-7.33333
```

```
NIntegrate[mm[x3[t], y3[t], z3[t]] x3′[t] +
  nn[x3[t], y3[t], z3[t]] y3′[t] + pp[x3[t], y3[t], z3[t]] z3′[t],
 {t, 0, 1}]
2.75772
```

Explain how you could have predicted in advance that the first group of three integrals would have calculated out to the same value.
Explain why it would have been quite a surprise if the second group of three path integrals had all calculated out to be equal.

□G.5.b)

C_1 is a line segment parameterized to start at {1, −3, 5} and to end at {−8, 7, 4}.
C is any other curve parameterized to start at {1, −3, 5} and to end at {−8, 7, 4}.
Explain how you know that

$$\int_C (-18x^2 + 24y + 2xz)\,dx + (24x - 4z)\,dy + (x^2 - 4y)\,dz$$

$$= \int_{C_1} (-18x^2 + 24y + 2xz)\,dx + (24x - 4z)\,dy + (x^2 - 4y)\,dz.$$

Then use this fact to help calculate

$$\int_C (-18x^2 + 24y + 2xz)\,dx + (24x - 4z)\,dy + (x^2 - 4y)\,dz$$

for ANY curve C parameterized to start at {1, −3, 5} and end at {−8, 7, 4}.

□G.5.c.i)

You are given a 3D vector field
Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}
and a closed curve C (like a deformed circle). You are also given the additional information that:
→ Field[x, y, z] has only a couple of singularities and none of them lie on C, and
→ curlField[x, y, z] = {0, 0, 0} at all points {x, y, z} that are not singularities.
Parameterize C, and call the direction of your parameterization counterclockwise. Use Stokes's formula to explain why you are guaranteed that the path integral

$$\oint_C m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz = 0.$$

□Tip:

Think of a surface R whose boundary curve is C with the extra property that none of the singularities is on the surface R.

□G.5.c.ii)

Agree or disagree with each of the following statements:
→ In 2D, if you go with a vector field
Field[x, y] = {m[x, y], n[x, y]}
with the property that rotField[x, y] = 0 at all {x, y}'s that are not singularities, then it is possible that

$$\oint_C m[x, y]\,dx + n[x, y]\,dy \neq 0$$

for a closed curve C that does not pass through any of the singularities.
→ In 3D, if you go with a vector field
Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]}
with the property that curlField[x, y, z] = {0, 0, 0} at all {x, y, z} that are not singularities, then it is possible that

$$\oint_C m[x, y, z]\,dx + n[x, y, z]\,dy + p[x, y, z]\,dz \neq 0$$

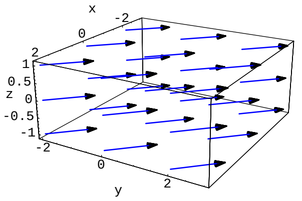
for a closed curve C that does not pass through any of the singularities.

G.6) Parallel flow and irrotational flow

□G.6.a.i)

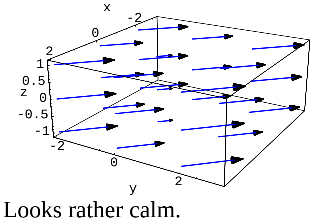
Here is a 3D vector field consisting of equal parallel vectors:

```
Clear[Field1, m1, n1, p1, x, y, z]
Field1[x_, y_, z_] = {-1, 1, 0};
{m1[x_, y_, z_], n1[x_, y_, z_], p1[x_, y_, z_]} = Field1[x, y, z];
Field1plot =
 Table[Arrow[Field1[x, y, z], Tail → {x, y, z}, VectorColor → Blue],
  {x, -2, 2, 2}, {y, -2, 2, 2}, {z, -1, 1, 1}];

Show[Field1plot, ViewPoint → CMView, Axes → Automatic,
 AxesLabel → {"x", "y", "z"}, PlotRange → All];
```



A calming, steady flow.
Here's another 3D vector field flowing in the same direction.

```
Clear[Field2, m2, n2, p2]
Field2[x_, y_, z_] = 0.3 (√(x^2 + y^2) + 1) {-1, 1, 0};
{m2[x_, y_, z_], n2[x_, y_, z_], p2[x_, y_, z_]} = Field2[x, y, z];
Field2plot =
 Table[Arrow[Field2[x, y, z], Tail → {x, y, z}, VectorColor → Blue],
  {x, -2, 2, 2}, {y, -2, 2, 2}, {z, -1, 1, 1}];

Show[Field2plot, ViewPoint → CMView, Axes → Automatic,
 AxesLabel → {"x", "y", "z"}, PlotRange → All];
```



Looks rather calm.
The farther {x, y, 0} is from {0, 0, 0}, the faster the flow of Field2[x, y, z] at {x, y, z}.
Both fields represent parallel flows.
Field1[x, y, z] represents flow at a steady speed, but the flow represented by Field2[x, y, z] is of variable speed.
The question here is, "Are both flows irrotational?"

□Tip:

A flow described by a vector field Field[x, y, z] is irrotational if curlField[x, y, z] = {0, 0, 0}.

□G.6.a.ii)

Continue with the same vector fields Field1[x, y, z] and Field2[x, y, z] as in part i).
Is it possible to make a paddle wheel rotate by sticking the paddle wheel into the flow described Field1[x, y, z]? If so, then how would you align the paddle wheel to get the most action?
Is it possible to make a paddle wheel rotate by sticking the paddle wheel into the flow described Field2[x, y, z]? If so, then how would you align the paddle wheel to get the most action?

□G.6.a.iii)

Continue with the same vector fields Field1[x, y, z] and Field2[x, y, z] as in part i).
Does either field have any sources or sinks?
Why is it sort of natural that Field2[x, y, z] should have lots of sources and lots of sinks?

□G.6.b.i)

Here is a general 3D vector field flowing in the direction of {1, 1, 0} with speed of the flow at {x, y, z} determined by the x and y entries in {x, y, z}:

```
Clear[Field, m, n, p, x, y, z, s]
Field[x_, y_, z_] = s[x, y] {1, 1, 0};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
```

```
{s[x, y], s[x, y], 0}
```

How must D[s[x, y], x] and D[s[x, y], y] be related to ensure that the flow represented by this vector field is irrotational?

□G.6.b.ii)

Here again is a general 3D vector field flowing in the direction of {1, 1, 0} with speed of the flow at {x, y, z} determined by the x and y entries in {x, y, z}:

```
Clear[Field, m, n, p, x, y, z, s]
Field[x_, y_, z_] = s[x, y] {1, 1, 0};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
```

```
{s[x, y], s[x, y], 0}
```

How must D[s[x, y], x] and D[s[x, y], y] be related to ensure that the flow represented by this vector field has no sources or sinks?

□G.6.b.iii)

Continue with the same set-up as in parts b.i) and b.ii).

```
Clear[Field, m, n, p, x, y, z, s]
Field[x_, y_, z_] = s[x, y] {1, 1, 0};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
```

```
{s[x, y], s[x, y], 0}
```

Look at:

```
Clear[curlField]
curlField[x_, y_, z_] = {D[p[x, y, z], y] - D[n[x, y, z], z],
   D[m[x, y, z], z] - D[p[x, y, z], x], D[n[x, y, z], x] - D[m[x, y, z], y]}
```

$\{0, 0, -s^{(0,1)}[x, y] + s^{(1,0)}[x, y]\}$

And look at:

```
Clear[divField]
divField[x_, y_, z_] =
  D[m[x, y, z], x] + D[n[x, y, z], y] + D[p[x, y, z], z]
```

$s^{(0,1)}[x, y] + s^{(1,0)}[x, y]$

Use what you see to say why it is impossible to build a non-constant function s[x, y] so that the parallel flow represented by

Field[x, y, z] = s[x, y] {1, 1, 0}

is both irrotational and is without sources or sinks.

□Tip:

s[x, y] is constant if

D[s[x, y], x] = $s^{(1,0)}$[x, y] = 0

and

D[s[x, y], y] = $s^{(0,1)}$[x, y] = 0

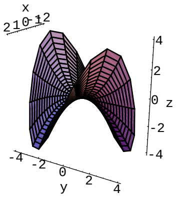
no matter what x and y are.

G.7) Clockwise or counterclockwise?

□G.7.a)

Here's a surface R:

```
Clear[x, y, z, s, t]
x[s_, t_] = 2 s Sin[t];
y[s_, t_] = 4 s Cos[t];
z[s_, t_] = y[s, t] x[s, t];
{{slow, shigh}, {tlow, thigh}} = {{0, 1}, {0, 2 π}};
surface = ParametricPlot3D[
  Evaluate[{x[s, t], y[s, t], z[s, t]}], {s, slow, shigh},
  {t, tlow, thigh}, PlotPoints → {Automatic, 24}, Boxed → False,
  PlotRange → All, ViewPoint → CMView, AxesLabel → {"x", "y", "z"}];
```



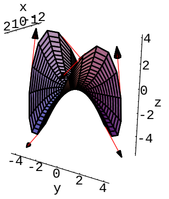
A parameterization of the boundary curve C is:

```
Clear[P]
P[t_] = {x[shigh, t], y[shigh, t], z[shigh, t]}
```

```
{2 Sin[t], 4 Cos[t], 8 Cos[t] Sin[t]}
```

Here are the surface R, its boundary curve C, and a few tangent vectors reflecting the direction of this parameterization of C:

```
Cplot = ParametricPlot3D[Evaluate[P[t]],
   {t, tlow, thigh}, PlotPoints → 24, DisplayFunction → Identity];
scalefactor = 0.5;
tangents = Table[Arrow[P′[t], Tail → P[t],
    VectorColor → Red, ScaleFactor → scalefactor], {t, 0, 2 π, π/3}];

Show[surface, Cplot, tangents, Boxed → False, PlotRange → All,
 ViewPoint → CMView, AxesLabel → {"x", "y", "z"}];
```

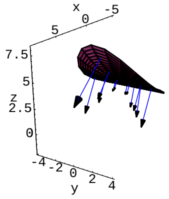


If you want to call this parameterization counterclockwise in accordance with the agreement made with regard to Stokes's formula, then which side of the surface must you designate as the top side?

□G.7.b)

Here is a surface together with some of its normal vectors:

```
Clear[x, y, z, s, t, normal]
x[s_, t_] = 5 s Sin[t];
y[s_, t_] = 4 s Cos[t];
z[s_, t_] = 3 E^(s Sin[t]);
{{slow, shigh}, {tlow, thigh}} = {{0, 1}, {0, 2 π}};
surface = ParametricPlot3D[Evaluate[{x[s, t], y[s, t], z[s, t]}],
   {s, slow, shigh}, {t, tlow, thigh}, DisplayFunction → Identity];
normal[s_, t_] = TrigExpand[
   D[{x[s, t], y[s, t], z[s, t]}, s] × D[{x[s, t], y[s, t], z[s, t]}, t]];
scalefactor = 0.2;
normals = Table[Arrow[normal[s, t],
    Tail → {x[s, t], y[s, t], z[s, t]}, ScaleFactor → scalefactor],
   {s, slow + 0.2, shigh - 0.2, 0.3}, {t, tlow, thigh, π/2}];
setup = Show[surface, normals, ViewPoint → CMView, Boxed → False,
   AxesLabel → {"x", "y", "z"}, DisplayFunction → $DisplayFunction];
```



Here is a parameterization of the boundary curve of this surface:

```
Clear[P]
P[t_] = {x[shigh, t], y[shigh, t], z[shigh, t]}
```

$\{5\,\mathrm{Sin}[t], 4\,\mathrm{Cos}[t], 3\,E^{\mathrm{Sin}[t]}\}$

If you decree that the normals plotted above are top normals, then are you forced to say that the parameterization P[t] above is counterclockwise, or are you forced to say that it is clockwise?

G.8) The symbols △(nabla) and ∇ (del)

□G.8.a) ∇ ×Field[x, y, z] = curlField[x, y, z].

Most reasonable folks agree that calculating the curl of a 3D vector field by hand is a loathsome chore, but calculating it by machine is not much trouble:

```
Clear[Field, m, n, p, x, y, z, curlField]
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]};
curlField[x_, y_, z_] = {D[p[x, y, z], y] - D[n[x, y, z], z],
  D[m[x, y, z], z] - D[p[x, y, z], x], D[n[x, y, z], x] - D[m[x, y, z], y]}
```

```
{-n^(0,0,1)[x, y, z] + p^(0,1,0)[x, y, z], m^(0,0,1)[x, y, z] - p^(1,0,0)[x, y, z],
 -m^(0,1,0)[x, y, z] + n^(1,0,0)[x, y, z]}
```

Nevertheless if your back is pressed to the wall, you can do this calculation by hand. Here's how to remember how to do it:
Letting Dx mean differentiation with repect to x,
letting Dy mean differentiation with repect to y, and
letting Dz mean differentiation with repect to z, form the matrix:

```
Clear[Dx, Dy, Dz]
MatrixForm[{{{1, 0, 0}, {0, 1, 0}, {0, 0, 1}}, {Dx, Dy, Dz},
  {m[x, y, z], n[x, y, z], p[x, y, z]}}]
```

```
( {1, 0, 0}   {0, 1, 0}   {0, 0, 1} )
(    Dx          Dy          Dz     )
( m[x, y, z]  n[x, y, z]  p[x, y, z] )
```

For

Field[x, y, z] = {m[x, y, z], n[x, y, z], p[x, y, z]},

you can calculate curlField[x, y, z] by hand by taking the determinant of this matrix by expanding across the top row.
This means that curlField[x, y, z] is given by

{1, 0, 0} (Dy (p[x, y, z]) − Dz (n[x, y, z])) −
 {0, 1, 0} ((Dx (p[x, y, z]) − Dz (m[x, y, z])) +
  {0, 0, 1} (Dx (n[x, y, z]) − Dy (m[x, y, z]))

Check:

```
Clear[handcurlField]
handcurlField[x_, y_, z_] =
 {1, 0, 0} (D[p[x, y, z], y] - D[n[x, y, z], z]) -
  {0, 1, 0} (D[p[x, y, z], x] - D[m[x, y, z], z]) +
  {0, 0, 1} (D[n[x, y, z], x] - D[m[x, y, z], y]);
handcurlField[x, y, z] == curlField[x, y, z]
```

```
True
```

Yes ma'am.
Another way to do the hand calculation is to put

∇ = {Dx, Dy, Dz}:

(Most folks call the upside down triangle ∇ by the name of "del.")

```
del = {Dx, Dy, Dz};
```

And calculating curlField[x, y, z] by taking the cross product

∇ ×Field[x, y, z]:

```
del×Field[x, y, z]
```

```
{-Dz n[x, y, z] + Dy p[x, y, z], Dz m[x, y, z] - Dx p[x, y, z],
 -Dy m[x, y, z] + Dx n[x, y, z]}
```

Still interpreting Dx, Dy, and Dz as above, this agrees with:

```
curlField[x, y, z]
```

```
{-n^(0,0,1)[x, y, z] + p^(0,1,0)[x, y, z], m^(0,0,1)[x, y, z] - p^(1,0,0)[x, y, z],
 -m^(0,1,0)[x, y, z] + n^(1,0,0)[x, y, z]}
```

Give a hand calculation of

curlField[x, y, z] = ∇ ×Field[x, y, z]

for

Field[x, y, z] = {x − y, y − 2 z, z − 3 x}.

Use the machine to check yourself.

□G.8.a.ii) ∇ . Field[x, y, z] = divField[x, y, z]

Look at:

```
Clear[Field, m, n, p, x, y, z]
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]}
```

```
{m[x, y, z], n[x, y, z], p[x, y, z]}
```

```
del = {Dx, Dy, Dz}
```

```
{Dx, Dy, Dz}
```

Here is ∇ . Field[x, y, z]:

```
del.Field[x, y, z]
```

```
Dx m[x, y, z] + Dy n[x, y, z] + Dz p[x, y, z]
```

Why do a lot of folks say

divField[x, y, z] = ∇ . Field[x, y, z]?

□G.8.a.iii)

$$\triangle f[x,y,z] = \nabla\,.\,\nabla f[x,y,z] = \frac{\partial^2 f[x,y,z]}{\partial x^2} + \frac{\partial^2 f[x,y,z]}{\partial y^2} + \frac{\partial^2 f[x,y,z]}{\partial z^2}$$

Now you know the neat formulas

divField[x, y, z] = ∇ . Field[x, y, z]

and

curlField[x, y, z] = ∇ ×Field[x, y, z]

for hand calculation of the divergence and the curl.
Here's another formula you sometimes see:
For a function f[x, y, z], put

$$\triangle f[x,y,z] = \frac{\partial^2 f[x,y,z]}{\partial x^2} + \frac{\partial^2 f[x,y,z]}{\partial y^2} + \frac{\partial^2 f[x,y,z]}{\partial z^2}.$$

(Most folks call the triangle △ by the name of "nabla" and they call △f[x, y, z] the "Laplacian" of f[x, y, z].)
What do these same folks mean when they write

△f[x, y, z] = ∇ . ∇f[x, y, z]?

And what do these same folks mean when they write

gradf[x, y, z] = ∇f[x, y, z]?

□ **Tip:**

Look at:

```
Clear[x, y, z, f, Dx, Dy, Dz]
del = {Dx, Dy, Dz}
del.(del f[x, y, z])
```

```
{Dx, Dy, Dz}
```

```
Dx^2 f[x, y, z] + Dy^2 f[x, y, z] + Dz^2 f[x, y, z]
```

```
del f[x, y, z]
```

```
{Dx f[x, y, z], Dy f[x, y, z], Dz f[x, y, z]}
```

□G.8.b.i)

When you start with a given 3D vector field Field[x, y, z], then

curlField[x, y, z] = ∇ ×Field[x, y, z]

is a new 3D vector field. Try it out:

```
Clear[Field, m, n, p, x, y, z]
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} =
  {x^2 + 2 y^2, 2 y^2 - z^2, 3 z^2 - x^2};
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]}
```

```
{x^2 + 2 y^2, 2 y^2 - z^2, -x^2 + 3 z^2}
```

Here comes the new 3D vector field:

```
Clear[curlField, newm, newn, newp]
curlField[x_, y_, z_] =
  {D[p[x, y, z], y] - D[n[x, y, z], z], D[m[x, y, z], z] - D[p[x, y, z], x],
   D[n[x, y, z], x] - D[m[x, y, z], y]};
{newm[x_, y_, z_], newn[x_, y_, z_], newp[x_, y_, z_]} =
  curlField[x, y, z]
```

```
{2 z, 2 x, -4 y}
```

Does this new vector field have any sources or sinks?

□G.8.b.ii)

Try it again with a new vector field:

```
Clear[Field, m, n, p, x, y, z]
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} =
  {Sin[x] Cos[3 y], Cos[y] Sin[3 z], Sin[y] Cos[3 z]};
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]}
```

```
{Cos[3 y] Sin[x], Cos[y] Sin[3 z], Cos[3 z] Sin[y]}
```

Here comes the new 3D vector field:

```
Clear[curlField, newm, newn, newp]
curlField[x_, y_, z_] =
  {D[p[x, y, z], y] - D[n[x, y, z], z], D[m[x, y, z], z] - D[p[x, y, z], x],
   D[n[x, y, z], x] - D[m[x, y, z], y]};
{newm[x_, y_, z_], newn[x_, y_, z_], newp[x_, y_, z_]} =
  curlField[x, y, z]
```

```
{-2 Cos[y] Cos[3 z], 0, 3 Sin[x] Sin[3 y]}
```

Does this new vector field have any sources or sinks?

□G.8.b.iii)

Try it again starting with a cleared vector field:

```
Clear[Field, m, n, p, x, y, z]
Field[x_, y_, z_] = {m[x, y, z], n[x, y, z], p[x, y, z]}
```

```
{m[x, y, z], n[x, y, z], p[x, y, z]}
```

Here comes the new 3D vector field:

```
Clear[curlField, newm, newn, newp]
curlField[x_, y_, z_] =
  {D[p[x, y, z], y] - D[n[x, y, z], z], D[m[x, y, z], z] - D[p[x, y, z], x],
   D[n[x, y, z], x] - D[m[x, y, z], y]};
{newm[x_, y_, z_], newn[x_, y_, z_], newp[x_, y_, z_]} =
  curlField[x, y, z]
```

$\{-n^{(0,0,1)}[x, y, z] + p^{(0,1,0)}[x, y, z], m^{(0,0,1)}[x, y, z] - p^{(1,0,0)}[x, y, z], -m^{(0,1,0)}[x, y, z] + n^{(1,0,0)}[x, y, z]\}$

Does this new vector field have any sources or sinks other than possibly at singularities?

□G.8.b.iv)

Lots of folks like to write
divcurlField[x, y, z] = ∇ . (∇ ×Field[x, y, z]) = 0
for all {x, y, z}'s other than singularities.
Why do they like to do this?

□Tip:

If you think of ∇ as a vector, then you can take a lot of poetic license and say that ∇ . (∇ ×Field[x, y, z]) = 0 because you can say ∇ is perpendicular to ∇ ×Field[x, y, z].

□G.8.b.v)

Does anything worth writing up happen when you start with a 3D vector field Field[x, y, z] and calculate the curl of curlField[x, y, z] which is given by
curlcurlField[x, y, z] = ∇ ×(∇ ×Field[x, y, z])?

G.9) Windpower

□G.9.a)

An electrical generator is driven by the wind. When you put the unit into the field, and the wind swirls in the clockwise way along the front of the unit, the generator charges a battery. If the wind swirls in the clockwise way, the generator discharges the battery.
You are looking at a wind flow given by the vector field:

```
Clear[Field, x, y, z, m, n, p]
Field[x_, y_, z_] = {2 x^2 - 3 y^2, 4 y - 6 z, z - 2 y};
{m[x_, y_, z_], n[x_, y_, z_], p[x_, y_, z_]} = Field[x, y, z]
```

$\{2 x^2 - 3 y^2, 4 y - 6 z, -2 y + z\}$

If you stick the generator into this flow at the point {1, 1, 2}, then how should you align the unit (front to back) to make the generator charge the battery as much as possible?

G.10) How 2D rotField[x, y] is related to 3D curlField[x, y, z]

□G.10.a)

Look at this cleared 2D vector field:

```
Clear[m, n, x, y, Field, rotField]
Field[x_, y_] = {m[x, y], n[x, y]}
```

```
{m[x, y], n[x, y]}
```

Here is rotField[x, y]:

```
rotField[x_, y_] = D[n[x, y], x] - D[m[x, y], y]
```

$-m^{(0,1)}[x, y] + n^{(1,0)}[x, y]$

Now make this 2D vector field into a 3D field as follows:

```
Clear[Field3D, z]
Field3D = Append[Field[x, y], 0]
```

```
{m[x, y], n[x, y], 0}
```

Calculate curlField3D[x, y, z], and say how it is related to rotField[x, y].