2D and 3D Measurements

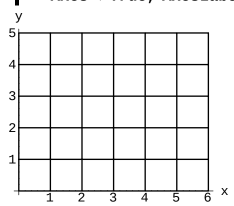
Produced by Bruce Carpenter   Published by Math Everywhere, Inc.
www.matheverywhere.com

VC.07 Transforming 2D Integrals
Basics

B.1) uv-paper and xy-paper

Here's a part of the usual xy-paper grid:

```
Clear[x, y]
{xlow, xhigh} = {0, 6};
{ylow, yhigh} = {0, 5};
xygrid = Show[
  Table[Graphics[Line[{{xlow, y}, {xhigh, y}}]], {y, ylow, yhigh}],
  Table[Graphics[Line[{{x, ylow}, {x, yhigh}}]], {x, xlow, xhigh}],
  Axes → True, AxesLabel → {"x", "y"}];
```

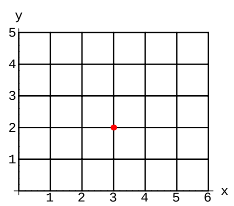


As you well know, y is constant on the horizontal lines, and x is constant on the vertical lines.
If you want to locate a point like {3, 2}, you can do it by going to the point at which the grid lines

x = 3 and y = 2

cross each other:

```
point = {3, 2};
Show[xygrid, Graphics[{Red, PointSize[0.03], Point[point]}]];
```



This isn't the only way to locate the point {3, 2}.
You can go with:

```
Clear[u, v, x, y]
{u[x_, y_], v[x_, y_]} = {x^2 - y^2, x y}
```

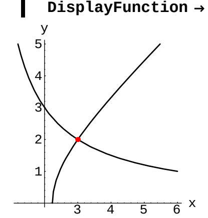
$\{x^2 - y^2, x\,y\}$

And you can say that the point {3, 2} is the point at which the level curves

u[x, y] = u[3, 2] and v[x, y] = v[3, 2]

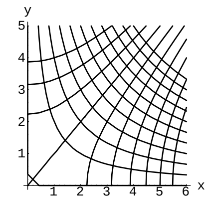
cross each other:

```
ulevelcurve = ContourPlot[Evaluate[u[x, y]],
  {x, xlow, xhigh}, {y, ylow, yhigh}, Contours → {u[3, 2]},
  ContourShading → False, DisplayFunction → Identity];
vlevelcurve = ContourPlot[Evaluate[v[x, y]],
  {x, xlow, xhigh}, {y, ylow, yhigh}, Contours → {v[3, 2]},
  ContourShading → False, DisplayFunction → Identity];
uvlevelcurves = Show[ulevelcurve, vlevelcurve, Frame → False];

Show[uvlevelcurves, Graphics[{Red, PointSize[0.03], Point[{3, 2}]}],
 Axes → True, AxesLabel → {"x", "y"}, PlotRange → All,
 DisplayFunction → $DisplayFunction];
```



You can plot a whole grid of level curves of u[x, y] and v[x, y]:

```
ulevelcurves = ContourPlot[Evaluate[u[x, y]], {x, xlow, xhigh},
  {y, ylow, yhigh}, Contours → {-15, -10, -5, 0, 5, 10, 15, 20, 25, 30},
  ContourShading → False, DisplayFunction → Identity];
vlevelcurves = ContourPlot[Evaluate[v[x, y]], {x, xlow, xhigh},
  {y, ylow, yhigh}, Contours → {0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20},
  ContourShading → False, DisplayFunction → Identity];
uvGridonxyPaper = Show[ulevelcurves, vlevelcurves, Frame → False];

Show[uvGridonxyPaper, Axes → True, AxesLabel → {"x", "y"},
 DisplayFunction → $DisplayFunction];
```

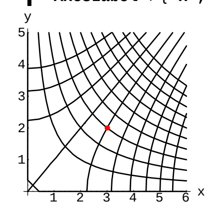


Totally cool.
As above, you can locate the point {3, 2} on xy-paper as the point where the level curves

u[x, y] = u[3, 2], and
v[x, y] = v[3, 2]

cross each other:

```
Show[uvGridonxyPaper,
 Graphics[{Red, PointSize[0.03], Point[{3, 2}]}], Axes → True,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```

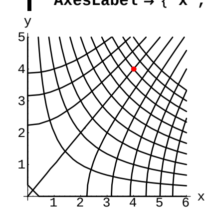


Or you can locate the point {4, 4} on xy-paper as the point where the level curves

u[x, y] = u[4, 4], and
v[x, y] = v[4, 4]

cross each other:

```
Show[uvGridonxyPaper,
 Graphics[{Red, PointSize[0.03], Point[{4, 4}]}], Axes → True,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```



Play.

□B.1.a.i)

Continue to go with the u[x, y] and v[x, y] above.
What do folks mean when they talk about the uv-coordinates of a point with xy-coordinates {x, y}?

□Answer:

That's easy.

The uv-coordinates of a point with xy-coordinates {x, y} are:

```
{u[x, y], v[x, y]}
```

$\{x^2 - y^2, x\,y\}$

For instance, the uv-coordinates of the point with xy-coordinates {3, 1} are:

```
{u[3, 1], v[3, 1]}
```

{8, 3}

In other words, the level curves

u[x, y] = 8 and v[x, y] = 3

cross at the point {x, y} = {3, 1}.

Not a whole lot to it.

□B.1.a.ii)

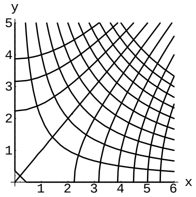
How do you make uv-paper?

□Answer:

With a little imagination and a good graphics system like Mathematica.

To see how little is involved, look at the uv-grid on xy-paper and imagine the plot to be made on a rubber sheet:

```
Show[uvGridonxyPaper, Axes → True, AxesLabel → {"x", "y"},
  DisplayFunction → $DisplayFunction];
```



What you see on the rubber sheet are the level curves

$u[x, y] = k$

for $k = -15, -10, -5, 0, 5, 10, 15, 20, 25$, and 30, and

$v[x, y] = c$

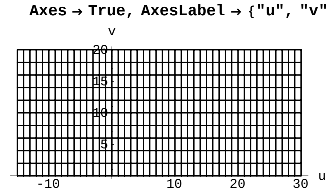
for $c = 0, 2, 4, 6, 8, 10, 12, 14, 16, 18$, and 20.

To make uv-paper, stretch and compress the rubber sheet so that all of the curves $u[x, y] = k$ and $v[x, y] = c$ become straight lines crossing perpendicularly.

Then apply new axes corresponding to the curves $v[x, y] = 0$ and $u[x, y] = 0$.

Here's what you get:

```
{ulow, uhigh, ujump} = {-15, 30, 5};
{vlow, vhigh, vjump} = {0, 20, 2};
uvgrid = Show[Table[
   Graphics[Line[{{ulow, v}, {uhigh, v}}]], {v, vlow, vhigh, vjump}],
  Table[Graphics[Line[{{u, vlow}, {u, vhigh}}]], {u, ulow, uhigh}],
  Axes → True, AxesLabel → {"u", "v"}];
```



This shows off the advantage of plotting on uv-paper; the hard-to-handle curved uv-grid on xy-paper becomes an easily dealt with grid of straight lines on uv-paper.

A little imagination and Mathematica will take you a long way.

□B.1.a.iii)

Continue to use

$u[x, y] = x^2 - y^2$, and

$v[x, y] = x\,y$.

How does the hyperbola $y = \frac{2}{x}$ plot out on uv-paper?

How does the hyperbola $x^2 - y^2 = 4$ plot out on uv-paper?

How does the circle $(x - 1)^2 + y^2 = 1$ plot out on uv-paper?

□Answer:

```
Clear[u, v, x, y]
u[x_, y_] = x^2 - y^2
```

$x^2 - y^2$

```
v[x_, y_] = x y
```

$x\,y$

→ uv-paper plot of $y = \frac{2}{x}$:

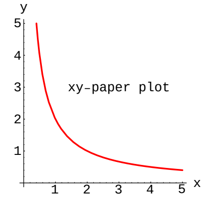
A parametric equation of the hyperbola $y = \frac{2}{x}$ is:

```
Clear[x, y, t]
{x[t_], y[t_]} = {t, 2/t}
```

$\{t, \frac{2}{t}\}$

Here's part of this hyperbola on xy-paper:

```
ParametricPlot[{x[t], y[t]}, {t, 2/5, 5},
  PlotStyle → {{Thickness[0.01], Red}}, AxesLabel → {"x", "y"},
  Epilog → Text["xy-paper plot", {3, 3}]];
```



The xy-paper point

$\{x[t], y[t]\}$

plots out on uv-paper at the uv-paper point

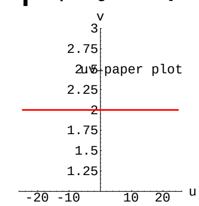
$\{u[x[t], y[t]], v[x[t], y[t]]\}$.

Here's the uv-paper plot of the same part of the hyperbola

$y = \frac{2}{x}$

plotted above on xy-paper:

```
ParametricPlot[{u[x[t], y[t]], v[x[t], y[t]]},
  {t, 2/5, 5}, PlotStyle → {{Thickness[0.01], Red}},
  PlotRange → {1, 3}, AxesLabel → {"u", "v"}, AspectRatio → 1,
  Epilog → Text["uv-paper plot", {10, 2.5}]];
```



Gee whiz.

The hyperbola $y = \frac{2}{x}$ plots out as the line $v = 2$.

This is no accident because the hyperbola

$y = \frac{2}{x}$

is the level curve

$x\,y = 2$

and this is the same as the level curve

$v[x, y] = x\,y = 2$.

When you stretched the xy-paper into uv-paper, this level curve became the line $v = 2$.

→ uv-paper plot of $x^2 - y^2 = 4$:

You don't need the machine to say how this looks on uv-paper.

When you remember that

$u[x, y] = x^2 - y^2$,

you see that the hyperbola

$x^2 - y^2 = 4$

plots out on uv-paper on the line

$u = 4$.

→ uv-paper plot of $(x - 1)^2 + y^2 = 1$:

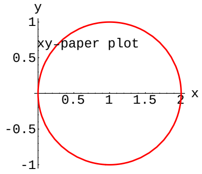
A parametric equation of the circle $(x - 1)^2 + y^2 = 1$ is:

```
Clear[x, y, t]
{x[t_], y[t_]} = {1, 0} + {Cos[t], Sin[t]}
```

$\{1 + \text{Cos}[t], \text{Sin}[t]\}$

Here's this circle on xy-paper in true scale:

```
ParametricPlot[{x[t], y[t]}, {t, 0, 2 π},
  PlotStyle → {{Thickness[0.01], Red}}, AxesLabel → {"x", "y"},
  AspectRatio → Automatic, Epilog → Text["xy-paper plot", {0.7, 0.7}]];
```



The xy-paper point

{x[t], y[t]}

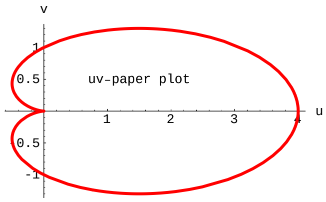
plots out on uv-paper at the uv-paper point

{u[x[t], y[t]], v[x[t], y[t]]}.

Here's the true scale uv-paper plot of the circle

$(x - 1)^2 + y^2 = 1$:

```
ParametricPlot[{u[x[t], y[t]], v[x[t], y[t]]}, {t, 0, 2 π},
 PlotStyle → {{Thickness[0.01], Red}}, AxesLabel → {"u", "v"},
 AspectRatio → Automatic, Epilog → Text["uv-paper plot", {1.5, 0.5}]];
```



Buns.

When you stretched and compressed the original rubber xy-paper to make the uv-paper, the circle got squashed.

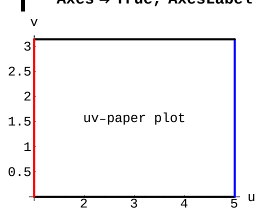
The cusp (corner) at {0, 0} is hard to ignore; you'll find why it happened in one of the GiveItaTry problems.

□B.1.b.i)

Often you'll want to start with uv-paper and go to xy-paper. Here's an example involving polar parameterization:

Go with the uv-paper rectangle, 1 ≤ u ≤ 5 and 0 ≤ v ≤ π:

```
uvpaperplot =
 Show[Graphics[{Thickness[0.01], Line[{{1, 0}, {5, 0}}]}],
  Graphics[{Blue, Thickness[0.01], Line[{{5, 0}, {5, π}}]}],
  Graphics[{Thickness[0.01], Line[{{5, π}, {1, π}}]}],
  Graphics[{Red, Thickness[0.01], Line[{{1, π}, {1, 0}}]}],
  Graphics[Text["uv-paper plot", {3, π/2}]], AspectRatio → Automatic,
  Axes → True, AxesLabel → {"u", "v"}];
```



Go to xy-paper with the polar functions

x[u, v] = u Cos[v] and y[u, v] = u Sin[v]

and plot this rectangle on xy-paper.

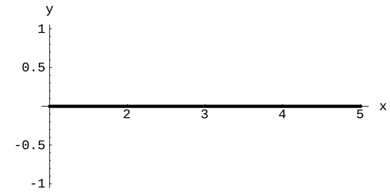
□ **Answer:**

```
Clear[x, y, u, v]
{x[u_, v_], y[u_, v_]} = {u Cos[v], u Sin[v]}
{u Cos[v], u Sin[v]}
```

The bottom side of the rectangle runs with

1 ≤ u ≤ 5 and v = 0:

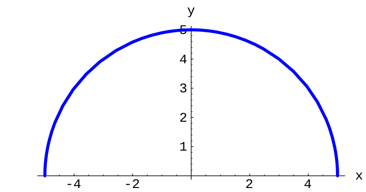
```
xybottom = ParametricPlot[{x[u, 0], y[u, 0]}, {u, 1, 5},
  PlotStyle → {{Thickness[0.01]}}, AxesLabel → {"x", "y"}];
```



The right side of the rectangle runs with

u = 5 and 0 ≤ v ≤ π:

```
xyright = ParametricPlot[{x[5, v], y[5, v]}, {v, 0, π},
  PlotStyle → {{Blue, Thickness[0.01]}}, AxesLabel → {"x", "y"}];
```

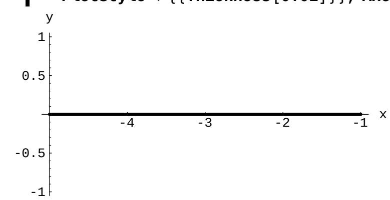


The top side of the rectangle runs with

1 ≤ u ≤ 5 and v = π:

There is no need for a counterclockwise parameterization here.

```
xytop = ParametricPlot[{x[u, π], y[u, π]}, {u, 1, 5},
  PlotStyle → {{Thickness[0.01]}}, AxesLabel → {"x", "y"}];
```

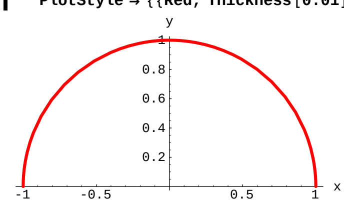


The left side of the rectangle runs with

u = 1 and 0 ≤ v ≤ π:

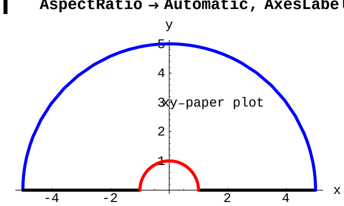
There is no need for a counterclockwise parameterization here.

```
xyleft = ParametricPlot[{x[1, v], y[1, v]}, {v, 0, π},
  PlotStyle → {{Red, Thickness[0.01]}}, AxesLabel → {"x", "y"}];
```



Here they are all assembled:

```
xypaperplot = Show[xybottom, xyright,
  xytop, xyleft, Graphics[Text["xy-paper plot", {1.5, 3}]],
  AspectRatio → Automatic, AxesLabel → {"x", "y"}];
```



Half a donut.

□B.1.b.ii)

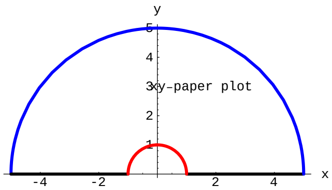
How does this bear on the topic of double integrals?

□ **Answer:**

It's sometimes handier than a can opener. You'll see why later, but just to whet your appetite, ask yourself this:

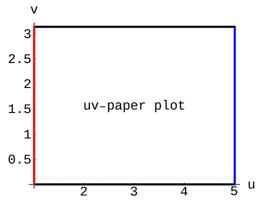
If you are setting up a double integral over a region R, would you prefer R to look like this:

```
Show[xypaperplot];
```



Or this:

```
Show[uvpaperplot];
```



This lesson is all about the art of replacing ornery regions by rectangles.

Go on and enjoy.

B.2) Linearizing the grids

□B.2.a.i)

Here is a microscopic plot of part of the xy-grid coming from

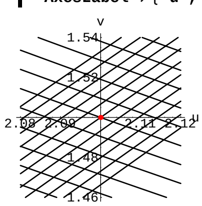
$x[u, v] = u^2 - v^2$ and $y[u, v] = u\,v$

in the vicinity of the uv-paper point

{a, b} = {2.1, 1.5}

plotted on uv-paper:

```
{a, b} = {2.1, 1.5};
jump = 0.02;
Clear[u, v, h, x, y]
{x[u_, v_], y[u_, v_]} = {u^2 - v^2, u v};
xlevelcurves = ContourPlot[Evaluate[x[u, v]],
  {u, a - jump, a + jump}, {v, b - 2 jump, b + 2 jump},
  Contours → Table[x[a, b] + h, {h, -5 jump, 5 jump, jump}],
  ContourShading → False, DisplayFunction → Identity];
ylevelcurves = ContourPlot[Evaluate[y[u, v]],
  {u, a - jump, a + jump}, {v, b - 2 jump, b + 2 jump},
  Contours → Table[y[a, b] + h, {h, -5 jump, 5 jump, jump}],
  ContourShading → False, DisplayFunction → Identity];
xyGridonuvPaper = Show[xlevelcurves, ylevelcurves, Frame → False];

Show[xyGridonuvPaper,
 Graphics[{Red, PointSize[0.03], Point[{a, b}]}], Axes → True,
 AxesLabel → {"u", "v"}, DisplayFunction → $DisplayFunction];
```



Here's the same thing for the linearizations of x[u, v] and y[u, v] at the same point {a, b}.

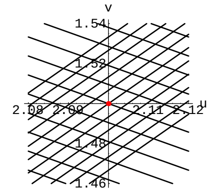
```
Clear[linearx, lineary, gradx, grady]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
linearx[u_, v_] = Expand[x[a, b] + gradx[a, b] . {u - a, v - b}]
```

```
-2.16 + 4.2 u - 3. v
```

```
lineary[u_, v_] = Expand[y[a, b] + grady[a, b] . {u - a, v - b}]
```

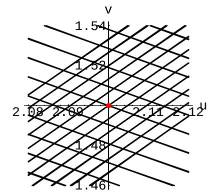
```
-3.15 + 1.5 u + 2.1 v
```

```
linearxlevelcurves = ContourPlot[Evaluate[linearx[u, v]],
  {u, a - jump, a + jump}, {v, b - 2 jump, b + 2 jump},
  Contours → Table[linearx[a, b] + h, {h, -5 jump, 5 jump, jump}],
  ContourShading → False, DisplayFunction → Identity];
linearylevelcurves = ContourPlot[Evaluate[lineary[u, v]],
  {u, a - jump, a + jump}, {v, b - 2 jump, b + 2 jump},
  Contours → Table[lineary[a, b] + h, {h, -5 jump, 5 jump, jump}],
  ContourShading → False, DisplayFunction → Identity];
linearxyGridonuvPaper =
 Show[linearxlevelcurves, linearylevelcurves, Frame → False];

Show[linearxyGridonuvPaper,
 Graphics[{Red, PointSize[0.03], Point[{a, b}]}], Axes → True,
 AxesLabel → {"u", "v"}, DisplayFunction → $DisplayFunction];
```



Compare:

```
Show[xyGridonuvPaper, linearxyGridonuvPaper,
 Graphics[{Red, PointSize[0.03], Point[{a, b}]}], Axes → True,
 AxesLabel → {"u", "v"}, DisplayFunction → $DisplayFunction];
```



In this microscopic plot of the vicinity of {a, b}, the xy-grid on uv-paper is almost the same as the grid coming from the linearized versions of x[u, v] and y[u, v] at {a, b}.
Explain why:
You will see similar results no matter what the functions x[u, v] and y[u, v] are, and no matter what point {a, b} you go with (unless {a, b} is a singularity for x[u, v] or y[u, v]).

□Answer:

Way back, you remember learning that the linearized version of any function x[u, v] (respectively y[u, v]) at {a, b} is a superb approximation of x[u, v] (respectively y[u, v]) in the vicinity of {a, b}. That's why the grid coming from the linearized versions of x[u, v] and y[u, v] at {a, b} has to mimic the xy-grid so well near the point {a, b}. In fact, the closer you get to {a, b}, the better the approximation.

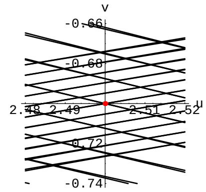
Watch it happen for new functions

x[u, v] = u Cos[v], and

y[u, v] = 2 u Sin[v],

and a new point {a, b} = {2.5, −0.7}:

```
{a, b} = {2.5, -0.7};
jump = 0.02;
Clear[u, v, h, x, y]
{x[u_, v_], y[u_, v_]} = {u Cos[v], 2 u Sin[v]};
xlevelcurves = ContourPlot[Evaluate[x[u, v]],
  {u, a - jump, a + jump}, {v, b - 2 jump, b + 2 jump},
  Contours → Table[x[a, b] + h, {h, -5 jump, 5 jump, jump}],
  ContourShading → False, DisplayFunction → Identity];
ylevelcurves = ContourPlot[Evaluate[y[u, v]],
  {u, a - jump, a + jump}, {v, b - 2 jump, b + 2 jump},
  Contours → Table[y[a, b] + h, {h, -5 jump, 5 jump, jump}],
  ContourShading → False, DisplayFunction → Identity];
xyGridonuvPaper = Show[xlevelcurves, ylevelcurves, Frame → False];
Clear[linearx, lineary, gradx, grady]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
linearx[u_, v_] = Expand[x[a, b] + gradx[a, b] . {u - a, v - b}];
lineary[u_, v_] = Expand[y[a, b] + grady[a, b] . {u - a, v - b}];
linearxlevelcurves = ContourPlot[Evaluate[linearx[u, v]],
  {u, a - jump, a + jump}, {v, b - 2 jump, b + 2 jump},
  Contours → Table[linearx[a, b] + h, {h, -5 jump, 5 jump, jump}],
  ContourShading → False, DisplayFunction → Identity];
linearylevelcurves = ContourPlot[Evaluate[lineary[u, v]],
  {u, a - jump, a + jump}, {v, b - 2 jump, b + 2 jump},
  Contours → Table[lineary[a, b] + h, {h, -5 jump, 5 jump, jump}],
  ContourShading → False, DisplayFunction → Identity];
linearxyGridonuvPaper =
 Show[linearxlevelcurves, linearylevelcurves, Frame → False];

Show[xyGridonuvPaper, linearxyGridonuvPaper,
 Graphics[{Red, PointSize[0.03], Point[{a, b}]}], Axes → True,
 AxesLabel → {"u", "v"}, DisplayFunction → $DisplayFunction];
```



Lookin' good; especially good near {a, b}.

Play with other functions x[u, v] and y[u, v], and other points {a, b}.

□B.2.b)

Summarize.

□Answer:

The gist:

→ Near a uv-paper point {a, b}, xy-paper plots look the same as plots on linearxlineary-paper. The closer you get to {a, b}, the better the linear grid approximates the curved grid.

→ The linearxlineary-grid on uv-paper is always a bunch of parallelograms in much the same way that the xy-grid on xy-paper is a bunch of rectangles.

B.3) Transforming 2D integrals: How you do it and why you do it

This assumes familarity with B.1) and B.2)

□B.3.a) The instantanteous area conversion factor $A_{xy}[u, v]$

Knowing that

$$\int_{x[a]}^{x[b]} f[x]\, dx = \int_a^b f[x[u]]\, x'[u]\, du$$

is a mark of calculus literacy.

But without the fudge function

$$\text{fudge}[u] = x'[u],$$

the transformation fails.

In the two-variable case, when you use functions x[u, v] and and y[u, v] to go from integrating on xy-paper to integrating on uv-paper, you've got to come up with the fudge function, fudge[u, v], that makes

$$\int\int_{R_{xy}} f[x, y]\, dx\, dy$$

$$= \int\int_{R_{uv}} f[x[u, v], y[u, v]]\, \text{fudge}[u, v]\, du\, dv$$

where R_{uv} is the uv-paper plot of the region R_{xy} orginally plotted on xy-paper.

What is the physical meaning of this function

fudge[u, v]?

□Answer:

The function fudge[u, v] has to satisfy

$$\int\int_{R_{xy}} f[x, y]\, dx\, dy$$

$$= \int\int_{R_{uv}} f[x[u, v], y[u, v]]\, \text{fudge}[u, v]\, du\, dv.$$

No matter what f[x, y] you have.

You get a pregnant clue to the physical meaning of fudge[u, v] by taking

$$f[x, y] = 1.$$

For this particular f[x, y], you get

$$\int\int_{R_{xy}} dx\, dy = \int\int_{R_{uv}} \text{fudge}[u, v]\, du\, dv.$$

Put Area[R_{xy}] equal to the area of R_{xy} measured on xy-paper and realize that this equation means that

$$\text{Area}[R_{xy}] = \int\int_{R_{xy}} dx\, dy = \int\int_{R_{uv}} \text{fudge}[u, v]\, du\, dv.$$

If you make the measurements on uv-paper, then

$$\text{Area}[R_{uv}] = \int\int_{R_{uv}} du\, dv.$$

So fudge[u, v] must be the instantaneous area conversion factor that you integrate to convert uv-paper area measurements into xy-paper area measurements.

You could say that the fudge[u, v] is the derivative of xy-paper area with repect to uv-paper area.

To dignify the fudge[u, v], give it a fancy look by writing

$$A_{xy}[u, v] = \text{fudge}[u, v].$$

Name it by calling $A_{xy}[u, v]$ the area conversion factor because it converts uv-paper area measurements into xy-paper area measurements.

□B.3.b) The formula for the area conversion factor $A_{xy}[u, v]$

Now go for the throat.

The formula for the area conversion factor $A_{xy}[u, v]$ that you integrate to convert uv-paper area measurements to xy-paper area measurements is:

```
Clear[x, y, u, v, gradx, grady, Axy]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = Abs[Det[{gradx[u, v], grady[u, v]}]]
```

$\text{Abs}[y^{(0,1)}[u, v]\, x^{(1,0)}[u, v] - x^{(0,1)}[u, v]\, y^{(1,0)}[u, v]]$

Some of the fancy folks call this the Jacobian determinant.

Where does this beauty come from?

□Answer:

Ultimately, it comes from the cross product.

Start with

$$x = x[u, v], \text{ and}$$

$$y = y[u, v].$$

At a fixed uv-paper point {a, b}, the linearized versions of x[u, v] and y[u, v] are calculated as follows:

```
Clear[x, y, u, v, a, b, gradx, grady, linearx, lineary]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
linearx[u_, v_] = x[a, b] + gradx[a, b] . {u - a, v - b}
```

$x[a, b] + (-b + v)\, x^{(0,1)}[a, b] + (-a + u)\, x^{(1,0)}[a, b]$

```
lineary[u_, v_] = y[a, b] + grady[a, b] . {u - a, v - b}
```

$y[a, b] + (-b + v)\, y^{(0,1)}[a, b] + (-a + u)\, y^{(1,0)}[a, b]$

Remember from B.2:

On uv-paper, the closer you are to {a, b}, the more spectacularly the xy-grid is approximated by the linearxlineary-grid. As a result, at the point {a, b}, the two area conversion factors are the same; in other words

$$A_{xy}[a, b] = A_{linearxlineray}[a, b].$$

Now calculate $A_{linearxlineray}[a, b]$:

To do this, remember that the linear grid is a bunch of parallelograms.

For this reason, the uv-paper square with corners at

{a, b}, {a + h, b}, {a + h, b + h} and {a, b + h}

plots out on linearxlineary-paper as the parallelogram with corners at:

Activate the next cell.

```
Clear[h]
basepoint = {linearx[a, b], lineary[a, b]};
corner1 = {linearx[a + h, b], lineary[a + h, b]};
corner2 = {linearx[a + h, b + h], lineary[a + h, b + h]};
corner3 = {linearx[a, b + h], lineary[a, b + h]};
```

The area of this parallelogram is given by the absolute value of:

If you want to refresh yourself about where the ingredients of this calculation come from, double click the box.

The area of the parallelogram determined by any vectors X and Y with their tails at a common base point is

||X|| ||Y|| |Sin[angle between]|

which is the same as the length of the base times the perpendicular height. To understand the calculation, put everything in three dimensions as follows:

```
Clear[X, x, Y, y]
X = {x[1], x[2]}
{x[1], x[2]}
X3D = {x[1], x[2], 0}
{x[1], x[2], 0}
Y = {y[1], y[2]}
{y[1], y[2]}
Y3D = {y[1], y[2], 0}
{y[1], y[2], 0}
```

Note:

```
√X.X == √X3D.X3D
True
√Y.Y == √Y3D.Y3D
True
```

This confirms that ||X|| = ||X3D|| and ||Y|| = ||Y3D||. So

area = ||X|| ||Y|| |Sin[angle between]|

= ||X3D|| ||Y3D|| |Sin[angle between]|.

Now look at:

```
X3DcrossY3D = Cross[X3D, Y3D]
{0, 0, -x[2] y[1] + x[1] y[2]}
Expand[(X3D.Y3D)^2 + X3DcrossY3D.X3DcrossY3D]
x[1]^2 y[1]^2 + x[2]^2 y[1]^2 + x[1]^2 y[2]^2 + x[2]^2 y[2]^2
Expand[X3D.X3D Y3D.Y3D]
x[1]^2 y[1]^2 + x[2]^2 y[1]^2 + x[1]^2 y[2]^2 + x[2]^2 y[2]^2
```

Remembering that

$(X3D \,.\, Y3D)^2 = ||X3D||^2\,||Y3D||^2\,\mathrm{Cos}[\text{angle between}]^2,$

you can see from the calculations immediately above that

$||X3D||^2\,||Y3D||^2\,\mathrm{Cos}[\text{angle between}]^2 + ||X3D \times Y3D||^2$

$= ||X3D||^2\,||Y3D||^2.$

So

$||X3D \times Y3D||^2 = ||X3D||^2\,||Y3D||^2\,(1 - \mathrm{Cos}[\text{angle between}]^2)$

$= ||X3D||^2\,||Y3D||^2\,\mathrm{Sin}[\text{angle between}]^2$

Consequently,

||X3D×Y3D|| = ||X3D|| ||Y3D|| |Sin[angle between]|

= area of parallelogram determined by X and Y.

End of explanation.

```
X = corner1 - basepoint;
Y = corner3 - basepoint;
X3D = Append[X, 0];
Y3D = Append[Y, 0];
X3DcrossY3D = Cross[X3D, Y3D];
AreaofParallelogram = √Collect[Expand[X3DcrossY3D.X3DcrossY3D], h^4]
√(h^4 (y^(0,1)[a, b]^2 x^(1,0)[a, b]^2 -
    2 x^(0,1)[a, b] y^(0,1)[a, b] x^(1,0)[a, b] y^(1,0)[a, b] +
    x^(0,1)[a, b]^2 y^(1,0)[a, b]^2))
```

The original uv-paper square with corners at

{a, b}, {a + h, b}, {a + h, b + h}, {a, b + h}

has area measuring out to h^2.

So the area conversion factor

$A_{xy}[a, b] = A_{linearxlineary}[a, b]$

is given by the absolute value of:

```
Simplify[AreaofParallelogram/h^2]
```

$$\frac{\sqrt{h^4\,(y^{(0,1)}[a,b]\,x^{(1,0)}[a,b] - x^{(0,1)}[a,b]\,y^{(1,0)}[a,b])^2}}{h^2}$$

Not so bad because this is the same as the absolute value of:

```
Det[{gradx[u, v], grady[u, v]}] /. {u → a, v → b}
y^(0,1)[a, b] x^(1,0)[a, b] - x^(0,1)[a, b] y^(1,0)[a, b]
```

The h's cancel out.

And now you see why the area conversion factor

$A_{xy}[u, v]$

is nothing more than the absolute value of:

```
Det[{gradx[u, v], grady[u, v]}]
y^(0,1)[u, v] x^(1,0)[u, v] - x^(0,1)[u, v] y^(1,0)[u, v]
```

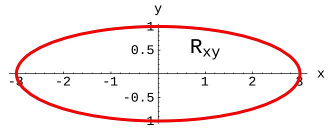
□B.3.c.i)

Now you get a chance to use this good stuff:
The region R_{xy} is everything inside and on the ellipse

$(\frac{x}{3})^2 + y^2 = 1$

on xy-paper:

```
Clear[x, y, t]
{x[t_], y[t_]} = {3 Cos[t], Sin[t]};
ParametricPlot[
 {x[t], y[t]}, {t, 0, 2 π}, PlotStyle → {{Thickness[0.01], Red}},
 AspectRatio → Automatic, AxesLabel → {"x", "y"},
 Epilog → Text[StyleForm["\!\(R\_xy\)", FontSize → 16], {1, 0.5}]];
```



Calculate

$\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy$

by transforming the integral to new variables u and v.

□Answer:

You can describe the ellipse and everything inside it by writing

{3 u Cos[v], u Sin[v]}

with $0 \le u \le 1$ and $0 \le v \le 2\pi$.

Put

x[u, v] = 3 u Cos[v], and

y[u, v] = u Sin[v]:

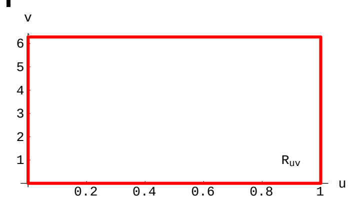
```
Clear[x, y, u, v]
{x[u_, v_], y[u_, v_]} = {3 u Cos[v], u Sin[v]}
{3 u Cos[v], u Sin[v]}
Clear[gradx, grady, Axy]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = TrigExpand[Det[{gradx[u, v], grady[u, v]}]]
3 u
```

Because $0 \le u \le 1$ and $0 \le v \le 2\pi$, the uv-paper plot, R_{uv}, of R_{xy} is the rectangle with corners at

{0, 0}, {1, 0}, {1, 2 π}, and {0, 2 π},

and everything inside it:

```
Show[Graphics[{Red, Thickness[0.01],
   Line[{{0, 0}, {1, 0}, {1, 2 π}, {0, 2 π}, {0, 0}}]}],
 Graphics[Text["\!\(R\_uv\)", {0.9, 1}]], Axes → True,
 AxesLabel → {"u", "v"}, AspectRatio → 1/2];
```



This gives

$\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy$

$= \int\int_{R_{uv}} (x[u, v]^2 + y[u, v]^2)\, A_{xy}[u, v]\, du\, dv$

$= \int_0^{2\pi}\int_0^1 (x[u, v]^2 + y[u, v]^2)\, A_{xy}[u, v]\, du\, dv:$

```
∫_0^(2π) ∫_0^1 (x[u, v]^2 + y[u, v]^2) Axy[u, v] ⅆu ⅆv
```

$\frac{15\pi}{2}$

So:

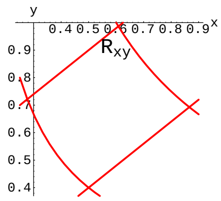
$$\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy = \frac{15\pi}{2},$$

and you're out of here without sweating and without a lot of miserable, irritating algebra.

□B.3.c.ii)

Look at the region R_{xy} inside the boundary described by the curves
$y = 0.8\,x$, $y = 0.8\,x + 0.5$,
$x\,y = 0.2$, and $x\,y = 0.6$:

```
Plot[{0.8 x, 0.8 x + 0.5, 0.2/x, 0.6/x},
 {x, 0.25, 0.9}, PlotStyle → {{Red, Thickness[0.01]}},
 PlotRange → {0.37, 1.0}, AxesLabel → {"x", "y"},
 Epilog → Text[StyleForm["\!\(R\_xy\)", FontSize → 16], {0.6, 0.9}]];
```



The region R_{xy} you are looking at is that stylish four-cornered figure you see above and everything inside it. Don't worry about the stray ends.

Calculate

$$\int\int_{R_{xy}} x\,y^2\, dx\, dy$$

without a lot of weeping, wailing, and gnashing of teeth by transforming the integral to new variables u and v.

□Answer:

This is a real bastard to calculate without leaving xy-paper because this integral will have to be broken into three integrals with lots of weeping, wailing, and gnashing of teeth.

But this is a great set-up for transforming to an integral easily set up on uv-paper.

The R_{xy} region has its boundary formed by the curves

$y = 0.8\,x$, $y = 0.8\,x + 0.5$,

$x\,y = 0.2$, and $x\,y = 0.6$.

Use

$u[x, y] = y - 0.8\,x$, and $v[x, y] = x\,y$.

```
Clear[x, y, u, v]
{u[x_, y_], v[x_, y_]} = {y - 0.8 x, x y}
```

{-0.8 x + y, x y}

Note

$y = 0.8\,x \longleftrightarrow u[x, y] = 0$

$y = 0.8\,x + 0.5 \longleftrightarrow u[x, y] = 0.5$

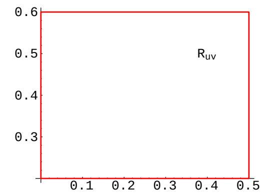
$x\,y = 0.2 \longleftrightarrow v[x, y] = 0.2$

$x\,y = 0.6 \longleftrightarrow v[x, y] = 0.6$.

After you stretch everything out so that the level curves of u[x, y] and v[x, y] become perpendicular straight lines on uv-paper, you see that the uv-paper plot, R_{uv}, of R_{xy} is the region inside the uv-paper rectangle with corners at

{0, 0.2}, {0.5, 0.2}, {0.5, 0.6}, and {0, 0.6}.

```
uvboundary = Show[Graphics[
   {Red, Line[{{0, 0.2}, {0.5, 0.2}, {0.5, 0.6}, {0, 0.6}, {0, 0.2}}]}],
  Graphics[Text["\!\(R\_uv\)", {0.4, 0.5}]], Axes → Automatic,
  Axes → Automatic];
```



Now

$$\int\int_{R_{xy}} x\,y^2\, dx\, dy$$

$$\int\int_{R_{uv}} x[u, v]\, y[u, v]^2\, A_{xy}[u, v]\, du\, dv$$

$$\int_{0.2}^{0.6}\int_0^{0.5} x[u, v]\, y[u, v]^2\, A_{xy}[u, v]\, du\, dv.$$

You can let Mathematica mop this up as soon as you find out what x[u, v], y[u, v], and $A_{xy}[u, v]$ are:

```
Solve[{u == u[x, y], v == v[x, y]}, {x, y}]
```

{{x → 0.25 (-2.5 u - 0.5 √(25. u² + 80. v)), y → 0.1 (5. u - 1. √(25. u² + 80. v))},
 {x → 0.25 (-2.5 u + 0.5 √(25. u² + 80. v)), y → 0.1 (5. u + √(25. u² + 80. v))}}

The original region R_{xy} consists of {x, y}'s with positive coordinates, so you use:

```
Clear[Axy, gradx, grady]
x[u_, v_] = 0.25 (-2.5 u + 0.5 √(25 u² + 80 v));
y[u_, v_] = 0.1 (5 u + √(25 u² + 80 v));
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = Det[{gradx[u, v], grady[u, v]}]
```

$-\frac{5.}{\sqrt{25\,u^2 + 80\,v}}$

This is negative; so throw in an extra minus sign to arrive at

$$\int\int_{R_{xy}} x\,y^2\, dx\, dy$$

$$= \int_{0.2}^{0.6}\int_0^{0.5} x[u, v]\, y[u, v]^2\, (-A_{xy[u,v]})\, du\, dv$$

```
integral =
 NIntegrate[x[u, v] y[u, v]^2 (-Axy[u, v]), {v, 0.2, 0.6}, {u, 0, 0.5}]
```

0.048339

There wasn't all that much to it.

Once you decided on a good uv-paper, then Mathematica ground it out without much trouble. The choice of uv-paper was more or less dictated by the set-up of the problem.

It saved a lot of miserable, irritating algebra.

□B.3.c.iii)

In calculating the last two integrals, what was the decisive advantage in switching from xy-paper to uv-paper?

□Answer:

Setting up the integrals on xy-paper would have been frustrating.

Setting up the integrals by transforming them to uv-paper was a breeze; it's the modern, up-to-date way of doing it.

VC.07 Transforming 2D Integrals Tutorials

T.1) Transforming $\int\int_{R_{xy}} f[x, y]\, dx\, dy$ when the boundary of R_{xy} is given by parametric formulas

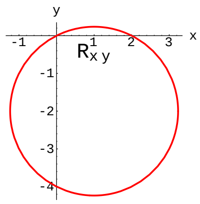
When the boundary of a region R_{xy} is plotted with parametric formulas, you often have a good shot at using the parametric formulas to come up with uv-paper on which R_{uv} is a rectangle. This is especially good because calculating 2D integrals over rectangles is usually very easy.

□T.1.a.i)

The region R_{xy} is everything inside and on the circle

$$(x - 1)^2 + (y + 2)^2 = 5:$$

```
Clear[x, y, t]
{x[t_], y[t_]} = {1, -2} + √5 {Cos[t], Sin[t]};
Rxyplot = ParametricPlot[{x[t], y[t]},
  {t, 0, 2 π}, PlotStyle → {{Thickness[0.01], Red}},
  AspectRatio → Automatic, AxesLabel → {"x", "y"}, Epilog →
   Text[StyleForm["\!\(R\_\( x y\)\)", FontSize → 16], {1, -0.5}]];
```



Calculate

$$\int\int_{R_{xy}} (3\,x^2 + 5\,y^4)\,dx\,dy$$

□Answer:

Look at the functions used to plot the boundary of R_{xy} :

```
{x[t], y[t]}
{1 + √5 Cos[t], -2 + √5 Sin[t]}
```

Put

```
Clear[x, y, u, t]
{x[u_, t_], y[u_, t_]} = {1, -2} + u {Cos[t], Sin[t]}
{1 + u Cos[t], -2 + u Sin[t]}
```

Realize this:

When you run u from 0 to $\sqrt{5}$, and you run t from 0 to 2π, then $\{x[u, t], y[u, t]\}$ runs through all of R_{xy}:

```
{{ulow = 0, uhigh = √5}, {tlow = 0, thigh = 2 π}}
{{0, √5}, {0, 2 π}}
```

The upshot:

R_{uv} is the rectangle

$$ulow \le u \le uhigh \text{ and } tlow \le t \le thigh$$

on ut-paper.

This, and the fact that

$$\int\int_{R_{xy}} f[x, y]\,dx\,dy$$

$$= \int\int_{R_{ut}} f[x[u, t], y[u, t]]\,A_{xy}[u, t]\,du\,dt$$

$$= \int_{tlow}^{thigh}\int_{ulow}^{uhigh} f[x[u, t], y[u, t]]\,A_{xy}[u, t]\,du\,dt,$$

tell you that you can turn everything over to the machine, right now:

```
Clear[gradx, grady, Axy]
gradx[u_, t_] = {D[x[u, t], u], D[x[u, t], t]};
grady[u_, t_] = {D[y[u, t], u], D[y[u, t], t]};
Axy[u_, t_] = TrigExpand[Det[{gradx[u, t], grady[u, t]}]]
u
```

Here comes the calculation of $\int\int_{R_{xy}} (3\,x^2 + 5\,y^4)\,dx\,dy$:

```
calculation =
 Simplify[∫_tlow^thigh ∫_ulow^uhigh (3 x[u, t]^2 + 5 y[u, t]^4) Axy[u, t] du dt]
10095 π
───────
   8
```

Nasty answer, but it wasn't hard to get.

□T.1.a.ii)

Could you have used the Gauss-Green formula to calculate the integral in part i)?

□Answer:

Yes.

Here is how it goes:

You want to calculate $\int\int_{R_{xy}} (3\,x^2 + 5\,y^4)\,dx\,dy$,

and you have a counterclockwise parameterization of the boundary of R_{xy}:

```
Clear[x, y, f, t]
f[x_, y_] = 3 x^2 + 5 y^4;
{x[t_], y[t_]} = {1, -2} + √5 {Cos[t], Sin[t]};
{a, b} = {0, 2 π}
{0, 2 π}
```

With $a \le t \le b$.

The Gauss-Green formula says

$$\int\int_{R_{xy}} (D[n[x, y], x] - D[m[x, y], y])\,dx\,dy$$

$$= \int_a^b m[x[t], y[t]]\,x'[t] + n[x[t], y[t]]\,y'[t]\,dt,$$

where $a = 0$ and $b = 2\pi$.

To calculate

$$\int\int_{R_{xy}} f[x, y]\,dx\,dy,$$

You just say

$$m[x, y] = 0, \text{ and}$$

$$n[x, y] = \int_0^x f[s, y]\,ds:$$

```
Clear[n, m, s, y]
m[x_, y_] = 0;
n[x_, y_] = ∫_0^x f[s, y] ds
x^3 + 5 x y^4
```

This gives you $f[x, y] = D[n[x, y], x] - D[m[x, y], y]$:

```
{D[n[x, y], x] - D[m[x, y], y], f[x, y]}
{3 x^2 + 5 y^4, 3 x^2 + 5 y^4}
```

Now you know that

$$\int\int_{R_{xy}} f[x, y]\,dx\,dy$$

$$\int\int_{R_{xy}} (D[n[x, y], x] - D[m[x, y], y])\,dx\,dy$$

$$= \int_a^b m[x[t], y[t]]\,x'[t] + n[x[t], y[t]]\,y'[t]\,dt:$$

The reason you re-enter $\{x[t], y[t]\}$ is that x and y were cleared when you calculated $m[x, y]$ and $n[x, y]$.

```
{x[t_], y[t_]} = {1, -2} + √5 {Cos[t], Sin[t]};
GGcalculation = ∫_a^b (m[x[t], y[t]] x′[t] + n[x[t], y[t]] y′[t]) dt
10095 π
───────
   8
```

As expected, this is the same nasty answer you got in part i).

□T.1.a.iii)

How do you decide whether to go with transformations as in part i), or with the Gauss-Green approach as in part ii)?

□Answer:

From the scientific point of view, it's a toss-up. Both methods work, so the decision is really a matter of personal choice.

Most folks prefer the approach using transformations in part i). But when you're putting on the ritz, you might want to go with the Gauss-Green approach.

Because you're already familiar with the Gauss-Green formula, this lesson will concentrate on the approach using transformations. You should do that too, because the approach using transformations has a clear extension to three dimensions. Versions of Gauss-Green in three dimensions are complicated.

□T.1.b.i)

Here's a plot of the part of the surface

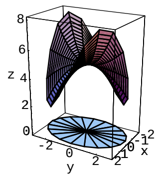
$z = x\,y + 5$

above the ellipse

$(\frac{x}{2})^2 + (\frac{y}{3})^2 = 1$

in the xy-plane:

```
Clear[x, y, r, t]
{x[r_, t_], y[r_, t_]} = {2 r Cos[t], 3 r Sin[t]};
{{rlow, rhigh}, {tlow, thigh}} = {{0, 1}, {0, 2 π}};
top = ParametricPlot3D[{x[r, t], y[r, t], x[r, t] y[r, t] + 5},
   {r, rlow, rhigh}, {t, tlow, thigh}, DisplayFunction → Identity];
base = ParametricPlot3D[
   {x[r, t], y[r, t], 0}, {r, rlow, rhigh}, {t, tlow, thigh},
   PlotPoints → {2, Automatic}, DisplayFunction → Identity];

Show[top, base, PlotRange → All, AxesLabel → {"x", "y", "z"},
 ViewPoint → CMView, DisplayFunction → $DisplayFunction];
```



Measure the volume under the plotted surface and above its base in the xy-plane.

□Answer:

It's duck soup.

The volume is measured by

$$\int\int_{R_{xy}} (x\,y + 5)\,dx\,dy,$$

where R_{xy} is everything inside and on the ellipse

$(\frac{x}{2})^2 + (\frac{y}{3})^2 = 1$

plotted in the xy-plane.

Go with rt-paper coming from:

```
{x[r, t], y[r, t]}
{2 r Cos[t], 3 r Sin[t]}
```

As you run r from rlow to rhigh, and as you run t from tlow to thigh,

{x[r, t], y[r, t]} sweeps out everything inside and on R_{xy}.

So R_{rt} is the rectangle rlow ≤ r ≤ rhigh and tlow ≤ t ≤ thigh, and

$$\text{volume} = \int\int_{R_{xy}} (x\,y + 5)\,dx\,dy$$

$$= \int\int_{R_{rt}} (x[r, t]\,y[r, t] + 5)\,A_{xy}[r, t]\,dr\,dt$$

$$= \int_{tlow}^{thigh}\int_{rlow}^{rhigh} (x[r, t]\,y[r, t] + 5)\,A_{xy}[r, t]\,dr\,dt:$$

```
Clear[gradx, grady, Axy]
gradx[r_, t_] = {D[x[r, t], r], D[x[r, t], t]};
grady[r_, t_] = {D[y[r, t], r], D[y[r, t], t]};
Axy[r_, t_] = TrigExpand[Det[{gradx[r, t], grady[r, t]}]]
6 r
```

Measure the volume:

```
∫_tlow^thigh ∫_rlow^rhigh (x[r, t] y[r, t] + 5) Axy[r, t] dr dt
30 π
```

Nice answer.

But if you hadn't gone to rt-paper to calculate the integral, getting this nice answer wouldn't have been so simple.

□T.1.b.ii)

Could you have measured the same volume using the Gauss-Green formula?

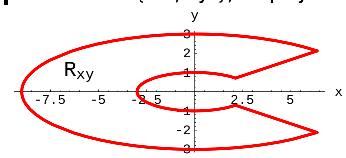
□Answer:

Yes.

□T.1.c.i)

Here's a plot of a region R_{xy} on xy-paper:

```
Clear[x, y, r, t]
{x[r_, t_], y[r_, t_]} = {3 r Cos[t], r Sin[t]};
{rlow, rhigh} = {1, 3};
{tlow, thigh} = {π/4, 7π/4};
twosides = ParametricPlot[{{x[rlow, t], y[rlow, t]},
    {x[rhigh, t], y[rhigh, t]}}, {t, tlow, thigh},
   PlotStyle → {{Thickness[0.01], Red}}, DisplayFunction → Identity];
twomoresides = ParametricPlot[{{x[r, tlow], y[r, tlow]},
    {x[r, thigh], y[r, thigh]}}, {r, rlow, rhigh},
   PlotStyle → {{Thickness[0.01], Red}}, DisplayFunction → Identity];
nametag =
  Graphics[Text[StyleForm["\!\(R\_xy\)", FontSize → 16], {-6, 1}]];

Show[twosides, twomoresides, nametag, AspectRatio → Automatic,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```



R_{xy} is everything inside and on this boundary.

Calculate

$$\int\int_{R_{xy}} x\,dx\,dy$$

with little thought, and with Mathematica doing the work.

□Answer:

Look at:

```
{x[r, t], y[r, t]}
{3 r Cos[t], r Sin[t]}
```

When you look at the plotting instructions above, then you see that when you run r from rlow to rhigh, and you run t from tlow to thigh, then {x[r, t], y[r, t]} describes R_{xy}. This tells you that on rt-paper R_{rt} is the rectangle

rlow ≤ r ≤ rhigh and tlow ≤ t ≤ thigh.

And your thinking is almost done.

Remembering that

$$\int\int_{R_{xy}} x\,dx\,dy = \int\int_{R_{rt}} x[r, t]\,A_{xy}[r, t]\,dr\,dt,$$

turn Mathematica loose:

```
Clear[gradx, grady, Axy]
gradx[r_, t_] = {D[x[r, t], r], D[x[r, t], t]};
grady[r_, t_] = {D[y[r, t], r], D[y[r, t], t]};
Axy[r_, t_] = Factor[TrigExpand[Det[{gradx[r, t], grady[r, t]}]]]
3 r
```

You can see that $A_{xy}[r, t]$ is never negative because r never goes negative. Now your thinking is done.

Now calculate

$$\int\int_{R_{xy}} x\,dx\,dy = \int\int_{R_{rt}} x[r, t]\,A_{xy}[r, t]\,dr\,dt$$

$$= \int_{tlow}^{thigh}\int_{rlow}^{rhigh} x[r, t]\,A_{xy}[r, t]\,dr\,dt:$$

```
∫_tlow^thigh ∫_rlow^rhigh x[r, t] Axy[r, t] dr dt
-78 √2
```

And you're out of here.

□T.1.c.ii)

Could you have calculated the same integral using the Gauss-Green formula?

□Answer:

In theory, yes.

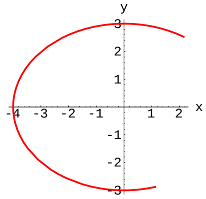
In practice, it would have been a very tedious job, because you would have to do a lot of bureaucratic work to come up with the required counterclockwise parameterization of the boundary. This would involve separate parameterizations for each of the four boundary segments.

Ugh.

□T.1.d.i) A constant-width ribbon.

Here's a curve on xy-paper:

```
Clear[x, y, r, t]
{x[t_], y[t_]} = {4 Cos[t], 3 Sin[t]};
{tlow, thigh} = {1, 5};

curveplot = ParametricPlot[{x[t], y[t]}, {t, tlow, thigh},
  PlotStyle → {{Thickness[0.01], Red}}, AspectRatio → Automatic,
  AxesLabel → {"x", "y"}];
```



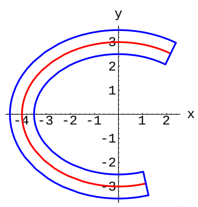
The outer normal to the curve at {x[t], y[t]} is:

```
Clear[unitnormal]
unitnormal[t_] = TrigExpand[{y'[t], -x'[t]} / Sqrt[x'[t]^2 + y'[t]^2]]
```

$$\left\{-\frac{3\sqrt{2}\,\mathrm{Cos}[t]\,\sqrt{25-7\,\mathrm{Cos}[2\,t]}}{-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2},\ -\frac{4\sqrt{2}\,\sqrt{25-7\,\mathrm{Cos}[2\,t]}\,\mathrm{Sin}[t]}{-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2}\right\}$$

Here's the boundary of a ribbon of constant width 1 centered on this curve:

```
Clear[r, outeribbon, innerribbon, oneend, otherend]
outerribbon[t_] = {x[t], y[t]} + 0.5 unitnormal[t];
innerribbon[t_] = {x[t], y[t]} - 0.5 unitnormal[t];
oneend[r_] = {x[tlow], y[tlow]} + r unitnormal[tlow];
otherend[r_] = {x[thigh], y[thigh]} + r unitnormal[thigh];
{rlow = -0.5, rhigh = 0.5};
twosides =
 ParametricPlot[{outerribbon[t], innerribbon[t]}, {t, tlow, thigh},
  PlotStyle → {{Thickness[0.01], Blue}}, DisplayFunction → Identity];
twomoresides =
 ParametricPlot[{oneend[r], otherend[r]}, {r, rlow, rhigh},
  PlotStyle → {{Thickness[0.01], Blue}}, DisplayFunction → Identity];

Show[curveplot, twosides, twomoresides, AspectRatio → Automatic,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```



Measure the area of this ribbon.

□**Answer:**

Call inside of the ribbon R_{xy}. The integral

$$\int\int_{R_{xy}} dx\,dy$$

measures the area of the ribbon.

Go to rt-paper with:

```
{rlow, rhigh} = {-0.5, 0.5};
Clear[x, y, r, t]
{x[r_, t_], y[r_, t_]} = {4 Cos[t], 3 Sin[t]} + r unitnormal[t]
```

$$\left\{4\,\mathrm{Cos}[t]-\frac{3\sqrt{2}\,r\,\mathrm{Cos}[t]\,\sqrt{25-7\,\mathrm{Cos}[2\,t]}}{-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2},\ 3\,\mathrm{Sin}[t]-\frac{4\sqrt{2}\,r\,\sqrt{25-7\,\mathrm{Cos}[2\,t]}\,\mathrm{Sin}[t]}{-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2}\right\}$$

When you look at the plotting instructions above, then you see that when you run r from rlow to rhigh, and you run t from tlow to thigh, then {x[r, t], y[r, t]} describes the whole ribbon R_{xy}. This tells you that on rt-paper R_{rt} is the rectangle

rlow ≤ r ≤ rhigh and tlow ≤ t ≤ thigh.

Remembering that

$$\text{area of ribbon} = \int\int_{R_{xy}} dx\,dy = \int\int_{R_{rt}} A_{xy}[r, t]\,dr\,dt,$$

turn Mathematica loose:

```
Clear[gradx, grady, Axy]
gradx[r_, t_] = {D[x[r, t], r], D[x[r, t], t]};
grady[r_, t_] = {D[y[r, t], r], D[y[r, t], t]};
Axy[r_, t_] = TrigExpand[Det[{gradx[r, t], grady[r, t]}]]
```

$$-\frac{24\,r}{-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2}-\frac{25\sqrt{25-7\,\mathrm{Cos}[2\,t]}}{\sqrt{2}\,(-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2)}+\frac{7\,\mathrm{Cos}[t]^2\sqrt{25-7\,\mathrm{Cos}[2\,t]}}{\sqrt{2}\,(-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2)}-\frac{7\sqrt{25-7\,\mathrm{Cos}[2\,t]}\,\mathrm{Sin}[t]^2}{\sqrt{2}\,(-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2)}$$

Analyzing this,it's hard to see whether this mess can ever go negative.

Take the easy way out and integrate its absolute value using NIntegrate.

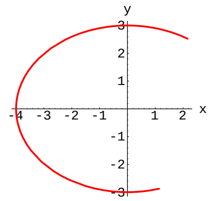
```
NIntegrate[Evaluate[Abs[Axy[r, t]]], {t, tlow, thigh},
 {r, rlow, rhigh}, AccuracyGoal → 2]
14.4267
```

About 14.4 square units.

□T.1.d.ii) A variable width ribbon

Go with the same base curve as in part i):

```
Clear[x, y, r, t]
{x[t_], y[t_]} = {4 Cos[t], 3 Sin[t]};
{tlow, thigh} = {1, 5};
curveplot = ParametricPlot[{x[t], y[t]}, {t, tlow, thigh},
  PlotStyle → {{Thickness[0.01], Red}}, AspectRatio → Automatic,
  AxesLabel → {"x", "y"}];
```

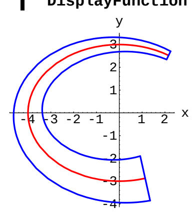


But this time make the width of the ribbon

0.4 t at {x[t], y[t]}.

Here's what you get:

```
Clear[unitnormal]
unitnormal[t_] = TrigExpand[{y'[t], -x'[t]} / Sqrt[x'[t]^2 + y'[t]^2]];

Clear[r, outeribbon, innerribbon, oneend, otherend, halfwidth]
halfwidth[t_] = 0.2 t;
outerribbon[t_] = {x[t], y[t]} + halfwidth[t] unitnormal[t];
innerribbon[t_] = {x[t], y[t]} - halfwidth[t] unitnormal[t];
tlowend[r_] = {x[tlow], y[tlow]} + r unitnormal[tlow];
thighend[r_] = {x[thigh], y[thigh]} + r unitnormal[thigh];
twosides =
 ParametricPlot[{outerribbon[t], innerribbon[t]}, {t, tlow, thigh},
  PlotStyle → {{Thickness[0.01], Blue}}, DisplayFunction → Identity];
onemoreside =
 ParametricPlot[tlowend[r], {r, -halfwidth[tlow], halfwidth[tlow]},
  PlotStyle → {{Thickness[0.01], Blue}}, DisplayFunction → Identity];
andonemoreside = ParametricPlot[
  thighend[r], {r, -halfwidth[thigh], halfwidth[thigh]},
  PlotStyle → {{Thickness[0.01], Blue}}, DisplayFunction → Identity];

Show[curveplot, twosides, onemoreside,
 andonemoreside, AspectRatio → Automatic, AxesLabel → {"x", "y"},
 DisplayFunction → $DisplayFunction];
```



Measure the area of this ribbon.

□**Answer:**

Call the plot of the ribbon above R_{xy}. The integral

$$\int\int_{R_{xy}} dx\,dy$$

measures the area of the ribbon.

Go to rt-paper with:

```
Clear[x, y, r, t]
{x[r_, t_], y[r_, t_]} =
 {4 Cos[t], 3 Sin[t]} + r halfwidth[t] unitnormal[t]
```

$$\left\{4\,\mathrm{Cos}[t]-\frac{0.848528\,r\,t\,\mathrm{Cos}[t]\sqrt{25-7\,\mathrm{Cos}[2\,t]}}{-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2},\ 3\,\mathrm{Sin}[t]-\frac{1.13137\,r\,t\sqrt{25-7\,\mathrm{Cos}[2\,t]}\,\mathrm{Sin}[t]}{-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2}\right\}$$

When you look at the plotting instructions above, then you see that when you run r from −1 to 1, and you run t from tlow to thigh, then {x[r, t], y[r, t]} describes R_{xy}. This tells you that on rt-paper, R_{rt} is the rectangle

$-1 \le r \le 1$ and tlow ≤ t ≤ thigh,

so

$$\int\int_{R_{xy}} dx\,dy = \int\int_{R_{rt}} A_{xy}[r, t]\,dr\,dt = \int_{tlow}^{thigh}\int_{-1}^{1} A_{xy}[r, t]\,dr\,dt.$$

```
Clear[gradx, grady, Axy]
gradx[r_, t_] = {D[x[r, t], r], D[x[r, t], t]};
grady[r_, t_] = {D[y[r, t], r], D[y[r, t], t]};
Axy[r_, t_] = Det[{gradx[r, t], grady[r, t]}]
```

$$\frac{24.\,r\,t^2\,\mathrm{Cos}[t]^2}{(-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2)^2}-\frac{6.72\,r\,t^2\,\mathrm{Cos}[t]^2\,\mathrm{Cos}[2\,t]}{(-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2)^2}+\frac{24.\,r\,t^2\,\mathrm{Sin}[t]^2}{(-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2)^2}-\frac{6.72\,r\,t^2\,\mathrm{Cos}[2\,t]\,\mathrm{Sin}[t]^2}{(-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2)^2}-\frac{2.54558\,t\,\mathrm{Cos}[t]^2\sqrt{25-7\,\mathrm{Cos}[2\,t]}}{-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2}-\frac{4.52548\,t\sqrt{25-7\,\mathrm{Cos}[2\,t]}\,\mathrm{Sin}[t]^2}{-25+7\,\mathrm{Cos}[t]^2-7\,\mathrm{Sin}[t]^2}$$

Analyzing this mess see whether it can ever go negative is a scary prospect. Take the easy way out and integrate its absolute value,using NIntegrate to calculate

$$\int\int_{R_{xy}} dx\,dy = \int\int_{R_{rt}} A_{xy}[r, t]\,dr\,dt = \int_{tlow}^{thigh}\int_{-1}^{1} A_{xy}[r, t]\,dr\,dt.$$

```
NIntegrate[Evaluate[Abs[Axy[r, t]]], {t, tlow, thigh}, {r, -1, 1},
 AccuracyGoal → 2]
```

17.2594

About 17.3 square units.

Pity those poor souls in the traditional calculus course. Most of them couldn't even dream of making this measurement.

And it's so easy because of rt -paper.

T.2) Transforming $\int\int_{R_{xy}} f[x, y]\,dx\,dy$ when the boundary of R_{xy} is not given with parametric formulas

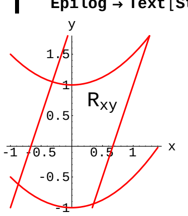
When the boundary of a region R_{xy} is plotted with nonparametric formulas, things are not always as simple as they were in T.1).
But even in this case, there are situations that allow you to inspect the boundary curves to help come up with favorable uv-paper.
Here's one such:

□T.2.a.i)

R_{xy} is the region plotted below which is bounded by the curves

$y = 0.5\,x^2 + 1$, $y = 0.5\,x^2 - 1$,
$y = 3\,x + 2$, and $y = 3\,x - 2$:

```
Clear[x]
Rxyplot = Plot[{0.5 x^2 + 1, 0.5 x^2 - 1, 3 x + 2, 3 x - 2},
  {x, -1, 1.4}, PlotStyle → {{Red, Thickness[0.01]}},
  AxesLabel → {"x", "y"}, PlotRange → {-1, 1.8},
  Epilog → Text[StyleForm["\!\(R\_xy\)", FontSize → 16], {0.5, 0.7}]];
```



The region R_{xy} under scrutiny is the four sided figure you see above and everything inside it.
Transform R_{xy} into a rectangle on uv-paper to help come up with a quick, easy calculation of

$$\int\int_{R_{xy}} (x^2 + y^2)\,dx\,dy.$$

□Answer:

Enter $f[x, y] = x^2 + y^2$:

```
Clear[f, x, y]
f[x_, y_] = x^2 + y^2
```

$x^2 + y^2$

Look at the formulas for the functions whose plots make up the boundary of R_{xy}. They are:

$y = 0.5\,x^2 + 1$, $y = 0.5\,x^2 - 1$,
$y = 3\,x + 2$, and $y = 3\,x - 2$:

Put

$u[x, y] = y - 0.5\,x^2$ and $v[x, y] = y - 3\,x$:

```
Clear[u, v, x, y]
{u[x_, y_], v[x_, y_]} = {y - 0.5 x^2, y - 3 x}
```

$\{-0.5\,x^2 + y,\ -3\,x + y\}$

The original boundary curves are level curves of u[x, y] and v[x, y]. In fact,

→ $y = 0.5\,x^2 + 1$ is the level curve u[x, y] = 1,
→ $y = 0.5\,x^2 - 1$ is the level curve u[x, y] = −1,
→ $y = 3\,x - 2$ is the level curve v[x, y] = 2, and
→ $y = 3\,x + 2$ is the level curve v[x, y] = −2.

This is very good news because this tells you that R_{uv} is the rectangle

$-1 \le u \le 1$ and $-2 \le v \le 2$.

The upshot:

$$\int\int_{R_{xy}} (x^2 + y^2)\,dx\,dy$$
$$\int\int_{R_{uv}} (x[u, v]^2 + y[u, v]^2)\,A_{xy}[u, v]\,du\,dv$$

$$\int_{-2}^{2}\int_{-1}^{1} (x[u, v]^2 + y[u, v]^2)\,A_{xy}[u, v]\,du\,dv.$$

First, you gotta to come up with formulas for x[u, v] and y[u, v]:

```
solutions = Solve[{u == u[x, y], v == v[x, y]}, {x, y}]
```

$\{\{y \to 1.\,(9. + v - 4.24264\sqrt{4.5 - 1.\,u + 1.\,v}),\ x \to 0.5\,(6. - 2.82843\sqrt{4.5 - 1.\,u + 1.\,v})\},$
$\{y \to 1.\,(9. + v + 4.24264\sqrt{4.5 - 1.\,u + 1.\,v}),\ x \to 0.5\,(6. + 2.82843\sqrt{4.5 - 1.\,u + 1.\,v})\}\}$

This gives two choices:

```
Clear[x1, y1, x2, y2]
{x1[u_, v_], y1[u_, v_]} =
  {0.5 (6 - Sqrt[36 - 4 (2 u - 2 v)]), 9 + v - 1.5 Sqrt[36 - 8 u + 8 v]}
```

$\{0.5\,(6 - \sqrt{36 - 4\,(2\,u - 2\,v)}),\ 9 + v - 1.5\sqrt{36 - 8\,u + 8\,v}\}$

```
{x2[u_, v_], y2[u_, v_]} =
  {0.5 (6 + Sqrt[36 - 4 (2 u - 2 v)]), 9 + v + 1.5 Sqrt[36 - 8 u + 8 v]}
```

$\{0.5\,(6 + \sqrt{36 - 4\,(2\,u - 2\,v)}),\ 9 + v + 1.5\sqrt{36 - 8\,u + 8\,v}\}$

You know that R_{uv} is the rectangle

$-1 \le u \le 1$ and $-2 \le v \le 2$.

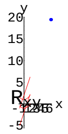
The uv-point {0.5, 1} is in this rectangle.

See which solution makes the uv-point {0.5, 1} plot out inside R_{xy} on xy-paper by seeing where {x1[0.5, 1], y1[0.5, 1]} and {x2[0.5, 1], y2[0.5, 1]} land on xy-paper:

```
Show[Rxyplot,
 Graphics[{Blue, PointSize[0.1], Point[{x1[0.5, 1], y1[0.5, 1]}]}],
 PlotRange → All];
```

Good; this lands inside R_{xy}. Just for the heck of it, try out the other pair of solutions:

```
Show[Rxyplot,
 Graphics[{Blue, PointSize[0.1], Point[{x2[0.5, 1], y2[0.5, 1]}]}],
 PlotRange → All];
```



Way outside R_{xy}. This means you definitely want to go with:

```
{x[u_, v_], y[u_, v_]} = {x1[u, v], y1[u, v]}
```

$\{0.5\,(6-\sqrt{36-4\,(2u-2v)}),\ 9+v-1.5\sqrt{36-8u+8v}\}$

Calculate the area conversion factor $A_{xy}[u, v]$:

```
Clear[Axy, gradx, grady]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = Det[{gradx[u, v], grady[u, v]}]
```

$\frac{2.\sqrt{36-4\,(2u-2v)}}{36-8u+8v}$

Calculate

$$\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy$$

$$\int\int_{R_{uv}} (x[u, v]^2 + y[u, v]^2)\, A_{xy}[u, v]\, du\, dv$$

$$\int_{-2}^{2}\int_{-1}^{1} (x[u, v]^2 + y[u, v]^2)\, A_{xy}[u, v]\, du\, dv.$$

```
NIntegrate[(x[u, v]^2 + y[u, v]^2) Axy[u, v], {v, -2, 2}, {u, -1, 1},
 AccuracyGoal → 2]
```

1.74016

Finished.

T.2.a.ii) The Achilles heel

When you have a set-up like the problem in part i), what can go wrong?

Answer:

In theory, nothing much can go wrong.

In practice, this technique can grind to a quick halt. The hitch is that you specify u[x, y] and v[x, y], and then you have to solve the simultaneous equations

u = u[x, y] and v = v[x, y]

for x and y to get the formulas for

x[u, v] and y[u, v].

Solving u = u[x, y] and v = v[x, y] for x and y is possible only in simple special situations.

Samples:

```
Clear[x, y, u, v]
{u[x_, y_], v[x_, y_]} = {y - Sin[x], y - x};
Solve[{u == u[x, y], v == v[x, y]}, {x, y}]
```

Solve[{u == y - Sin[x], v == -x + y}, {x, y}]

No dice. That transcendental function Sin[x] screws up the algebra.

```
Clear[x, y, u, v]
{u[x_, y_], v[x_, y_]} = {y - E^-x, y - x};
Solve[{u == u[x, y], v == v[x, y]}, {x, y}]
```

$\{\{x \to u - v + \text{ProductLog}[E^{-u+v}],\ y \to u + \text{ProductLog}[E^{-u+v}]\}\}$

No dice. That transcendental function e^{-x} screws up the algebra.

```
Clear[x, y, u, v]
{u[x_, y_], v[x_, y_]} = {y - 2 x, y + x};
ExpandAll[Solve[{u == u[x, y], v == v[x, y]}, {x, y}]]
```

$\{\{x \to -\frac{u}{3} + \frac{v}{3},\ y \to \frac{u}{3} + \frac{2v}{3}\}\}$

No sweat. This gives you

$x[u, v] = \frac{-u+v}{3}$ and $y[u, v] = \frac{u+2v}{3}$.

T.2.a.iii)

What is a transcendental function?

Answer:

```
This answer comes from Phillip Gillett's book
Calculus and Analytic Geometry (2nd edition),
D.C. Heath, Lexington, Massachusetts,1984, p.335.
```

Transcendental functions are those that transcend the ordinary processes of algebra.

The basic calculus functions Sin[x], Cos[x], and e^x are all transcendental.

That's why you are guaranteed to fail when you try simple things like:

```
Clear[x]
Solve[x == Sin[x], x]
```

Solve[x == Sin[x], x]

```
Clear[x]
Solve[x == Cos[x], x]
```

Solve[x == Cos[x], x]

```
Clear[x]
Solve[x == E^x, x]
```

{{x → -ProductLog[-1]}}

Line functions like f[x] = 3 x + 2 are not transcendental:

```
Clear[x]
Solve[x == 3 x + 2, x]
```

{{x → -1}}

Determining whether a given function is transcendental is part of the stuff of advanced mathematics.

T.3) The area conversion factor $A_{xy}[u, v]$

T.3.a) Expansion and compression

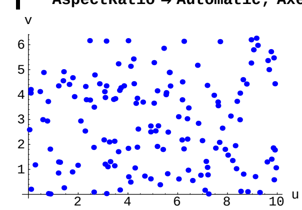
Polar coordinates {u, v} are related to xy-coordinates via

x = u Cos[v],

y = u Sin[v].

Here is a plot of a random bunch of points {u, v} on uv-paper with $0 \le u \le 10$ and $0 \le v \le 2\pi$:

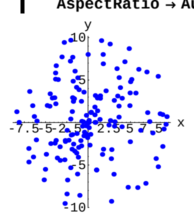
```
Clear[k]
uvpoints = Table[
  {Random[Real, {0, 10}], Random[Real, {0, N[2 π]}]}, {k, 1, 150}];
uvpaperplot = ListPlot[uvpoints, PlotStyle → {Blue, PointSize[0.02]},
  AspectRatio → Automatic, AxesLabel → {"u", "v"}];
```



Should be fairly well scattered. If not, then rerun.

Now look at the xy-paper plot of these uv-paper points:

```
Clear[x, y, u, v]
x[u_, v_] = u Cos[v];
y[u_, v_] = u Sin[v];
Clear[uvtoxy]
uvtoxy[{u_, v_}] = {x[u, v], y[u, v]};
xypoints = uvtoxy /@ uvpoints;

xypaperplot = ListPlot[xypoints, PlotStyle → {Blue, PointSize[0.03]},
  AspectRatio → Automatic, AxesLabel → {"x", "y"}];
```



On the xy-paper, why are the points so bunched up near the origin and sparsely scattered far from the origin?

Answer:

Look at the area conversion factor:

```
Clear[gradx, grady, Axy]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = TrigExpand[Det[{gradx[u, v], grady[u, v]}]]
```

```
u
```

At a point with uv-paper coordinates {u, v},

xy − paper area measurements .
= u times uv − paper area measurements

In this set-up, uv-paper coordinates are polar coordinates.

Look at:

```
{x[u, v], y[u, v]}
```

```
{u Cos[v], u Sin[v]}
```

The uv-paper coordinate, u, of the xy-paper point, {x[u, v], y[u, v]}, measures the distance from {0, 0} to {x[u, v], y[u, v]} on xy-paper.

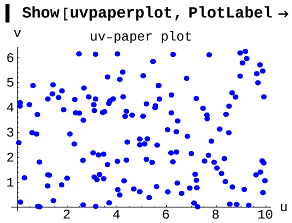
→ When A_{xy}[u, v] = u is big, then xy-paper area measurements are a lot bigger than the corresponding uv-area measurements, so uv-paper points with u big are flung apart when they are plotted on xy-paper.

→ When A_{xy}[u, v] = u is wee little, uv-paper points {u, v} with u small are compressed together when they are plotted on xy-paper.

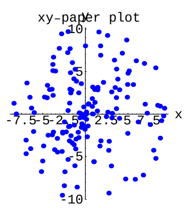
Since u is small for points near the origin, you see a pile-up near the origin on the xy-paper plot.

Take another look:

```
Show[uvpaperplot, PlotLabel → "uv-paper plot"];
```




```
Show[xypaperplot, PlotLabel → "xy-paper plot"];
```



Uh-huh.

□T.3.b.i) The sizer

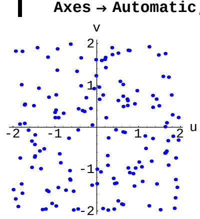
Take x[u, v] = $u^2 - v^2$ and y[u, v] = 2 u v and calculate the area conversion factor A_{xy}[u, v]:

```
Clear[x, y, u, v, gradx, grady, Axy]
x[u_, v_] = u^2 - v^2;
y[u_, v_] = 2 u v;
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = Det[{gradx[u, v], grady[u, v]}]
```

$4 u^2 + 4 v^2$

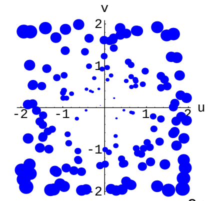
Here's a table of random points inside the uv-paper square with $-2 \le u \le 2$ and $-2 \le v \le 2$ plotted as dots on uv-paper:

```
Clear[k]
pointcount = 150;
uvpoints = Table[
  {Random[Real, {-2, 2}], Random[Real, {-2, 2}]}, {k, 1, pointcount}];

uvpointplot =
 Show[Table[Graphics[{Blue, PointSize[0.02], Point[uvpoints[[k]]]}],
   {k, 1, pointcount}], AspectRatio → Automatic,
  Axes → Automatic, AxesLabel → {"u", "v"}];
```



Should be fairly well scattered. If not, then rerun.
Here is the same plot with the size of each plotted point {u, v} adjusted by a factor proportional to $\sqrt{A_{xy}[u, v]}$.

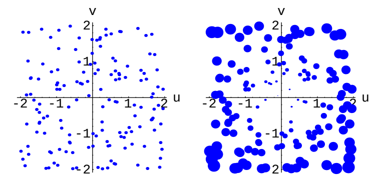
```
Clear[sizer]
sizer[u_, v_] = 0.015 √Axy[u, v];
sizeduvpointplot =
 Show[Graphics[Table[{Blue, PointSize[sizer @@ uvpoints[[k]]],
     Point[uvpoints[[k]]]}, {k, 1, pointcount}]],
  AspectRatio → Automatic, Axes → Automatic,
  AxesLabel → {"u", "v"}];
```



Grab both plots and animate them.

A humdinger.
See the plots side-by-side:

```
Show[GraphicsArray[{uvpointplot, sizeduvpointplot}]];
```



What information is conveyed by these plots?

□Answer:

The area conversion factor in going from {u, v} to {x[u, v], y[u, v]} is

A_{xy}[u, v].

The points are sized in proportion to $\sqrt{A_{xy}[u, v]}$.

Consequently, this second plot shows what the relative sizes of the plotted uv-points will be after they have been plotted on xy-paper.

Evidently as {u, v} gets farther and farther from {0, 0},

A_{xy}[u, v]

gets bigger and bigger. This fact is suggested by the plot, and is confirmed by the formula for A_{xy}[u, v]:

```
Axy[u, v]
```

$4 u^2 + 4 v^2$

If you wonder why the points are sized proportionally to

$\sqrt{A_{xy}[u, v]}$

instead of proportionally to

A_{xy}[u, v],

then click the little box.

In Mathematica, the PointSize instruction governs the radius of the plotted point. To make the area of the plotted point proportional to A_{xy}[u, v], you have to make the radius of the plotted point proportional to

$\sqrt{A_{xy}[u, v]}$,

because the area of a circle of radius r is proportional to r^2.

□T.3.b.ii)

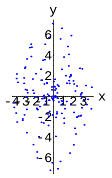
This is a continuation of part i) above.
Make sure all the active cells in part i) are executed.

Here is a plot of the same uv points as above in part i) but plotted on xy-paper with

x[u, v] = $u^2 - v^2$ and y[u, v] = 2 u v:

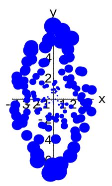
```
Clear[uvtoxy]
uvtoxy[{u_, v_}] = {x[u, v], y[u, v]};
xypoints = uvtoxy /@ uvpoints;

xypointplot = Show[Graphics[{Blue, PointSize[0.02],
```

```
Table[Point[xypoints⟦k⟧], {k, 1, pointcount}]}],
AspectRatio → Automatic, Axes → Automatic,
AxesLabel → {"x", "y"}];
```



And look at:

```
sizedxypointplot = Show[Table[Graphics[
    {Blue, PointSize[sizer@@xypoints⟦k⟧], Point[xypoints⟦k⟧]}],
   {k, 1, pointcount}], AspectRatio → Automatic,
  Axes → Automatic, AxesLabel → {"x", "y"}];
```



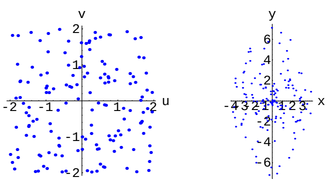
Grab both plots and animate

What information do these plots convey?

Answer:

Take a look at the original uv-points and the xy-points side-by-side:

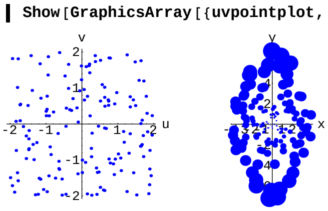
```
Show[GraphicsArray[{uvpointplot, xypointplot}]];
```



On the left you see the original points on uv-paper; on the right you see the same points plotted on xy-paper.

Now take a look at the original uv-points and the sized xy-points side-by-side:

```
Show[GraphicsArray[{uvpointplot, sizedxypointplot}]];
```



This is the final product.

On the left you see the original points on uv-paper; on the right you see the same points plotted on xy-paper sized according to their relative sizes on uv-paper.

T.4) Measurements of volume, mass, and density

T.4.a)

You make an object by distributing a substance over a certain region R on xy-paper. What does it mean to say that the density of the resulting object measures out at p[x, y] $\frac{\text{grams}}{\text{unit}^3}$ at location {x, y}?

Answer:

It means that the mass of the object is given by

$$\int\int_R p[x, y]\, dx\, dy$$

where R is the same region that you distibuted the substance over to begin with.

It also means that if R_1 is a region inside R, then the total mass of the substance that was spread over R_1 is

$$\int\int_{R_1} p[x, y]\, dx\, dy.$$

As a result $p[x_0, y_0]$ is the conversion factor that converts area at $\{x_0, y_0\}$ on xy-paper to mass of the object at $\{x_0, y_0\}$.

T.4.b.i)

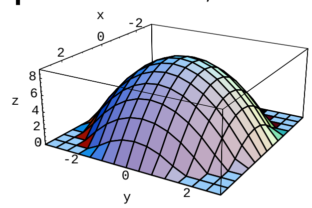
An object is made by forming a uniform substance that weighs 2 $\frac{\text{grams}}{\text{unit}^3}$ in the shape of the paraboloid

$$f[x, y] = 9 - x^2 - y^2$$

over the region inside and on the circle $x^2 + y^2 = 9$ on xy-paper.

Here's a look at it:

```
Clear[x, y]
Plot3D[9 - x^2 - y^2, {x, -3, 3}, {y, -3, 3}, PlotRange → {0, 9},
 ViewPoint → CMView, AxesLabel → {"x", "y", "z"}];
```



Measure the total mass of this object.

Measure the total volume of this object.

Answer:

The density of the object,p[x, y], at {x, y} is 2 f[x, y].

The total mass is given by

$$\int\int_R 2\, f[x, y]\, dx\, dy$$

where R is the region inside and on the circle $x^2 + y^2 = 9$ on xy-paper.

No one likes to integrate over circular regions, so calculate this by moving to uv-paper with

$x[u, v] = u\, \text{Cos}[v]$, and

$y[u, v] = u\, \text{Sin}[v]$ (polar coordinates).

On uv-paper, R plots out as the rectangle with corners at

$\{0, 0\}, \{3, 0\}, \{3, 2\pi\}$ and $\{0, 2\pi\}$

$(0 \leq u \leq 3$ and $0 \leq v \leq 2\pi)$.

The total weight is

$$\int\int_R 2\, f[x, y]\, dx\, dy$$
$$= \int_0^{2\pi}\int_0^3 2\, f[x[u, v], y[u, v]]\, A_{xy}[u, v]\, du\, dv:$$

```
Clear[f, u, v, x, y, gradx, grady, Axy]
f[x_, y_] = 9 - x^2 - y^2;
x[u_, v_] = u Cos[v];
y[u_, v_] = u Sin[v];
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = TrigExpand[Det[{gradx[u, v], grady[u, v]}]]
```

u

The total mass is:

```
mass = ∫_0^(2π) ∫_0^3 2 f[x[u, v], y[u, v]] Axy[u, v] du dv
```

81π

The total volume of this object is:

```
volume = ∫_0^(2π) ∫_0^3 f[x[u, v], y[u, v]] Axy[u, v] du dv
```

$\frac{81\pi}{2}$

Routine stuff.

□**T.4.b.ii)**

When you rubberized the xy-paper and stretched it out to make the uv-paper, you also deformed the original shape of the object in part i). The deformation affected measurements on the base, but had no effect on the height measurements.
What does the resulting deformed object look like when it is plotted on uv-paper for the functions

$x = x[u, v]$ and $y = y[u, v]$

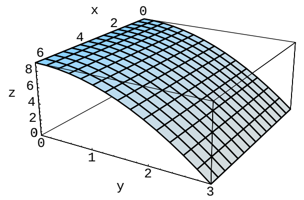
you used above in part i)?
What is the density of this deformed object at a point {u, v} inside the uv-paper rectangle $0 \le u \le 3$ and $0 \le v \le 2\pi$?

□**Answer:**

This is a continuation of the last part.
Please make sure that all instructions in the last part are alive on your machine.

Here is how it looks on uv-paper:

```
Plot3D[f[x[u, v], y[u, v]], {v, 0, 2π}, {u, 0, 3}, PlotRange → {0, 9},
 ViewPoint → CMView, AxesLabel → {"x", "y", "z"}];
```



Quite a change of shape.

You can figure out what the deformed object's density is at a point {u, v} within the uv-paper rectangle $0 \le u \le 3$ and $0 \le v \le 2\pi$. It is just what you integrate to calculate its mass:

```
TrigExpand[2 f[x[u, v], y[u, v]] Axy[u, v]]
```

$18 u - 2 u^3$

You've got to multiply by the area conversion factor because

$2 f[x[u, v], y[u, v]] A_{xy}[u, v]$

is what you integrate to calculate mass.

VC.07 Transforming 2D Integrals Give it a Try!

Experience with the starred problems will be useful for understanding developments later in the course.

G.1) Transforming 2D integrals*

G.1.a)

Here's a plot of the part of the surface

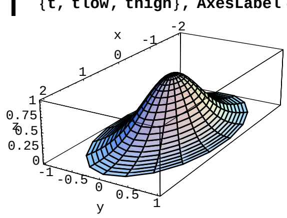
$z = e^{-(x^2 + 4 y^2)}$

above the ellipse

$(\frac{x}{2})^2 + y^2 = 1$

in the xy-plane:

```
Clear[x, y, r, t]
{x[r_, t_], y[r_, t_]} = {2 r Cos[t], r Sin[t]};
{{rlow, rhigh}, {tlow, thigh}} = {{0, 1}, {0, 2π}};
ParametricPlot3D[
 {x[r, t], y[r, t], E^-(x[r,t]^2+4 y[r,t]^2)}, {r, rlow, rhigh},
 {t, tlow, thigh}, AxesLabel → {"x", "y", "z"}, ViewPoint → CMView];
```

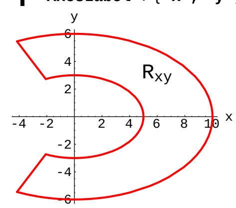


Measure the volume of the solid whose top skin is the surface plotted above and whose base is everything on xy-plane directly below this surface.

□**G.1.b)**

Here's a plot of a region R_{xy} on xy-paper:

```
Clear[x, y, r, t]
{x[r_, t_], y[r_, t_]} = {5 r Cos[t], 3 r Sin[t]};
{rlow, rhigh} = {1, 2};
{tlow, thigh} = {-2, 2};
twosides = ParametricPlot[{{x[rlow, t], y[rlow, t]},
   {x[rhigh, t], y[rhigh, t]}}, {t, tlow, thigh},
  PlotStyle → {{Thickness[0.01], Red}}, DisplayFunction → Identity];
tworeresides = ParametricPlot[{{x[r, tlow], y[r, tlow]},
   {x[r, thigh], y[r, thigh]}}, {r, rlow, rhigh},
  PlotStyle → {{Thickness[0.01], Red}}, DisplayFunction → Identity];
nametag =
 Graphics[Text[StyleForm["\!\(R\_xy\)", FontSize → 16], {6, 3}]];

Show[twosides, twomoresides, nametag, AspectRatio → Automatic,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```

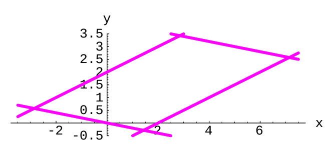


R_{xy} is everything inside and on this boundary.
Use a transformation to favorable uv-paper to measure the area of R_{xy} with little thought and with Mathematica doing the work.

□**G.1.c.i)**

Here's a parallelogram plotted on xy-paper

```
Plot[{0.5 x - 1, 0.5 x + 2, -0.2 x, -0.2 x + 4}, {x, -3.5, 7.5},
 PlotStyle → {{Magenta, Thickness[0.01]}}, PlotRange → {-0.5, 3.5},
 AspectRatio → Automatic, AxesLabel → {"x", "y"}];
```



Call R everything inside and on this parallelogram, and use a favorable transformation to help calculate

$\int\int_R e^{y - x}\, dx\, dy.$

□**G.1.c.ii)**

If $a \ne b$, $c < d$, and $r < s$, then you are guaranteed that the lines

$y = a x + c$, $y = a x + d$,

$y = b x + r$, and $y = b x + s$

define a parallelogram on xy-paper.
Reason:
Non-parallel lines cross each other.
Assume $b < a$, $c < d$, and $r < s$, and come up with a formula that measures the area of this parallelogram in terms of a, b, c, d, r, and s.

□**Tip:**

After you have your formula, make it look pretty by applying the Mathematica instruction Factor[Together[ ]].

After you do that, then you should get

$\frac{(d - c)(s - r)}{b - a}.$

□**G.1.d)**

Use a transformation to favorable uv-paper to calculate

$\int\int_{R_{xy}} (x + y)\, dx\, dy$

where R_{xy} is the region with $x \ge 0$ and $y \ge 0$ bounded by the curves

$x^2 - y^2 = 1$,

$x^2 - y^2 = 4$,

$x^2 + y^2 = 4$, and

$x^2 + y^2 = 9$.

□G.1.e)

Calculate

$$\int\int_{R_{xy}} e^{-x^2-y^2}\, dx\, dy$$

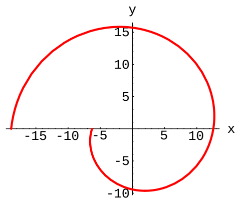
where R_{xy} is the region on xy-paper consisting of everything within and on the circle $x^2 + y^2 = 2$.

G.2) Ribbons*

□G.2.a)

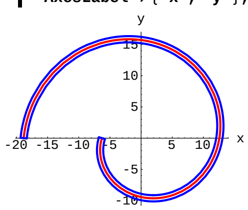
Here's a curve on xy-paper:

```
Clear[x, y, r, t]
{x[t_], y[t_]} = {2 t Cos[t], 2 t Sin[t]};
{tlow, thigh} = {π, 3 π};
curveplot = ParametricPlot[{x[t], y[t]}, {t, tlow, thigh},
  PlotStyle → {{Thickness[0.01], Red}}, AspectRatio → Automatic,
  PlotRange → All, AxesLabel → {"x", "y"}];
```



Here's the boundary of a ribbon of constant width 1 centered on this curve:

```
Clear[unitnormal]
unitnormal[t_] = TrigExpand[{y'[t], -x'[t]} / Sqrt[x'[t]^2 + y'[t]^2]];
Clear[r, outerribbon, innerribbon, oneend, otherend]
outerribbon[t_] = {x[t], y[t]} + 0.5 unitnormal[t];
innerribbon[t_] = {x[t], y[t]} - 0.5 unitnormal[t];
oneend[r_] = {x[tlow], y[tlow]} + r unitnormal[tlow];
otherend[r_] = {x[thigh], y[thigh]} + r unitnormal[thigh];
{rlow = -0.5, rhigh = 0.5};
twosides =
 ParametricPlot[{outerribbon[t], innerribbon[t]}, {t, tlow, thigh},
  PlotStyle → {{Thickness[0.01], Blue}}, DisplayFunction → Identity];
twomoresides =
 ParametricPlot[{oneend[r], otherend[r]}, {r, rlow, rhigh},
  PlotStyle → {{Thickness[0.01], Blue}}, DisplayFunction → Identity];

Show[curveplot, twosides, twomoresides, AspectRatio → Automatic,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```

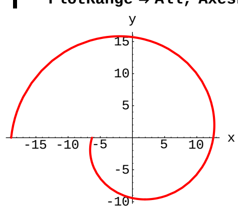


Change this ribbon to a new ribbon with constant width 2 centered on the given curve, and measure the area of the new ribbon.

□G.2.b) A variable width ribbon

Go with the same base curve as in part i).

```
Clear[x, y, r, t]
{x[t_], y[t_]} = {2 t Cos[t], 2 t Sin[t]};
{tlow, thigh} = {π, 3 π};
curveplot = ParametricPlot[{x[t], y[t]}, {t, tlow, thigh},
  PlotStyle → {{Thickness[0.01], Red}}, AspectRatio → Automatic,
  PlotRange → All, AxesLabel → {"x", "y"}];
```



But this time make the width of the ribbon

$2 + 4\,\mathrm{Sin}[2\,t]^2$ at {x[t], y[t]}.

Plot the resulting ribbon, and measure its area.

□Tip:

This ribbon is wider than the ribbon in part i). As a result, your area measurement of this ribbon should be bigger than your area measurment of the ribbon in part i).

□G.2.c)

Do something artistic with ribbons.
How about a real eye-catcher coming from your own mind?

G.3) Flow measurements*

□G.3.a)

To calculate the net flow of a vector field

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

across the boundary C of a region R, you have your choice:

→ You can go to the labor of parameterizing C, and then calculate

$$\oint_C -n[x, y]\, dx + m[x, y]\, dy.$$

→ Or if the field has no singularities inside R, you can put

divField[x, y] = D[m[x, y], x] + D[n[x, y], y]

and calculate the 2D integral

$$\int\int_R \mathrm{divField}[x, y]\, dx\, dy.$$

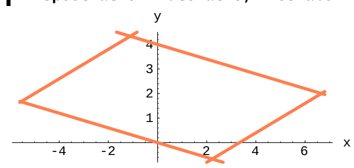
Here's a vector field:

```
Clear[x, y, m, n, Field]
{m[x_, y_], n[x_, y_]} = {x^3 + y, x + y^2}
```

$\{x^3 + y,\ x + y^2\}$

R is everything inside the parallelogram you see below:

```
Plot[{0.6 x - 2, 0.6 x + 5, -0.3 x, -0.3 x + 4}, {x, -5.6, 6.8},
 PlotStyle → {{Coral, Thickness[0.01]}}, PlotRange → {-0.8, 4.5},
 AspectRatio → Automatic, AxesLabel → {"x", "y"}];
```



Transform the 2D integral

$$\int\int_R \mathrm{divField}[x, y]\, dx\, dy$$

to favorable uv-paper to measure the net flow of this vector field across the parallelogram. Is the net flow of this vector field across this parallelogram from outside to inside or inside to outside?

□G.3.b)

To calculate the net flow of a vector field

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

along the boundary C of a region R, you have your choice:

→ You can go to the labor of parameterizing C, and then calculate

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

→ Or if the field has no singularities inside R, you can put

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

and calculate the 2D integral

$$\int\int_R \mathrm{rotField}[x, y]\, dx\, dy.$$

Here's a vector field:

```
Clear[x, y, m, n, Field]
{m[x_, y_], n[x_, y_]} = {2 Sin[y] + x, x + y};
```

R is everything inside the parallelogram plotted in part a).

Transform the 2D integral

$$\int\int_R \mathrm{rotField}[x, y]\, dx\, dy$$

to favorable uv-paper to measure the net flow of this vector field along the parallelogram. Is the net flow of this vector field along the parallelogram clockwise or counterclockwise?

G.4) Interpret the plots

□G.4.a)

All the plots below give information concerning the same phenomenon about what happens when you plot the region within the uv-paper square with corners at

$\{\frac{1}{2}, -2\}, \{2, -2\}, \{2, 2\}$ and $\{\frac{1}{2}, 2\}$

on xy-paper coming from

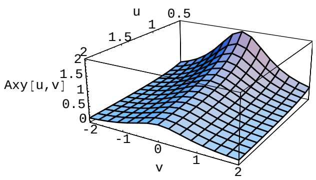
x[u, v] = Log[u] and y[u, v] = ArcTan[v].

Interpret the information conveyed by each plot.

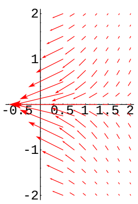
```
Clear[x, y, u, v, gradx, grady, Axy]
x[u_, v_] = Log[u];
y[u_, v_] = ArcTan[v];
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = Det[{gradx[u, v], grady[u, v]}]
```

$$\frac{1}{u\,(1+v^2)}$$

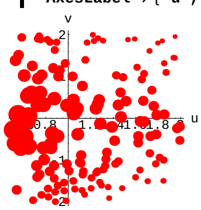
```
Plot3D[Axy[u, v], {u, 1/2, 2}, {v, -2, 2},
 AxesLabel -> {"u", "v", "Axy[u,v]"}, ViewPoint -> CMView];
```



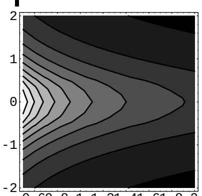

```
Clear[gradAxy]
gradAxy[u_, v_] = {D[Axy[u, v], u], D[Axy[u, v], v]};
scalefactor = 0.3;
Show[Table[Arrow[gradAxy[u, v], Tail -> {u, v}, VectorColor -> Red,
   ScaleFactor -> scalefactor], {u, 1/2, 2, 1/4}, {v, -2, 2, 1/4}],
 Axes -> Automatic];
```



```
Clear[sizer]
scalefactor = 0.08;
sizer[u_, v_] = scalefactor Sqrt[Axy[u, v]];
Clear[k]
uvpoints =
 Table[{Random[Real, {1/2, 2}], Random[Real, {-2, 2}]}, {k, 1, 150}];
Show[Table[Graphics[{PointSize[sizer @@ uvpoints[[k]]],
    Red, Point[uvpoints[[k]]]}], {k, 1, 150}],
 AspectRatio -> 1, Axes -> Automatic,
 AxesLabel -> {"u", "v"}];
```




```
ContourPlot[Axy[u, v], {u, 1/2, 2}, {v, -2, 2}];
```



The lighter the shading, the bigger $A_{xy}[u, v]$ is.

G.5) Semi-log paper and log-log paper

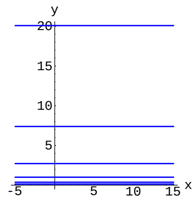
Semi-log and log paper are friends of every scientist because they make analysis of exponential and power functions very easy. Back at the beginning of Calculus&Mathematica, you used semi-log paper to some advantage.

Here are the lines

$y = e^{-2}$, $y = e^{-1}$, $y = e^{0}$, $y = e^{1}$, $y = e^{2}$, and $y = e^{3}$

plotted on xy-paper for $-5 \le x \le 15$:

```
Clear[t]
xylines =
 ParametricPlot[{{t, 1/E^2}, {t, 1/E}, {t, E^0}, {t, E^1}, {t, E^2}, {t, E^3}},
  {t, -5, 15}, PlotStyle -> {{Blue}}, AxesLabel -> {"x", "y"},
  AspectRatio -> Automatic];
```



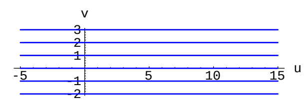
Semi-log paper is uv-paper for

u[x, y] = x, and

v[x, y] = Log[y].

Here are the same lines plotted on semi-log paper:

```
Clear[u, v, x, y]
u[x_, y_] = x;
v[x_, y_] = Log[y];
uvlines = ParametricPlot[{{u[t, 1/E^2], v[t, 1/E^2]},
   {u[t, 1/E], v[t, 1/E]}, {u[t, E^0], v[t, E^0]}, {u[t, E^1], v[t, E^1]},
   {u[t, E^2], v[t, E^2]}, {u[t, E^3], v[t, E^3]}}, {t, -5, 15},
  PlotStyle -> {Blue}, AxesLabel -> {"u", "v"}, AspectRatio -> Automatic];
```



□G.5.a)

Plot the xy-curves $y = 3\,e^{-0.75\,x}$ and $y = 2\,e^{1.5\,x}$ on semilog paper.

Describe what you see, and try to explain why you see it.

Why is it a good idea to plot xy-data on semi-log paper to reveal exponential relationships between the x-coordinate and the y-coordinate?

□G.5.b) Log-log paper

Log-log paper is uv-paper for

u[x, y] = Log[x] and

v[x, y] = Log[y].

Plot the power curves $y = 4\,x^{-0.6}$ and $y = 5\,x^{1.4}$ on xy-paper, and then plot them on log-log paper.

Why is the result not terribly surprising?

Why is it a good idea to plot xy-data on log-log paper to reveal power relationships between the x-coordinate and the y-coordinate?

G.6) What can happen when A_{xy}[u, v] is 0: What does the sign of Det[{gradx[u, v], grady[u, v]}] tell you?

G.6.a)

Ever wonder what can happen when $A_{xy}[u, v] = 0$?
The truth is that a uv-paper point $\{u_0, v_0\}$ with

$A_{xy}[u_0, v_0] = 0$

suffers the ultimate insult when it is plotted on xy-paper.
Here's why:
If $A_{xy}[u_0, v_0] = 0$, then the uv-paper point $\{u_0, v_0\}$ effectively disappears when it is plotted on xy-paper.
This sets up a situation in which crazy things can happen, because the xy-paper plot in the vicinity of one of these unlucky points is compressed so much.
Here's a sample:

```
Clear[u, v, x, y]
x[u_, v_] = u^2 - v^2
```
$u^2 - v^2$

```
y[u_, v_] = u v + 10
```
$10 + u v$

```
Clear[gradx, grady, Axy]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = Det[{gradx[u, v], grady[u, v]}]
```
$2 u^2 + 2 v^2$

At $\{u_0, v_0\} = \{0, 0\}$, the uv-paper to xy-paper area conversion factor

$A_{xy}[u, v]$

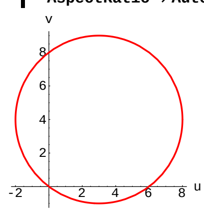
is 0, so {0, 0} is rubbed out on xy-paper. On xy-paper strange things should happen near the plot of the uv-paper point {0, 0}.
The uv-paper circle

$(u - 3)^2 + (v - 4)^2 = 5^2$

goes right through {0, 0}:

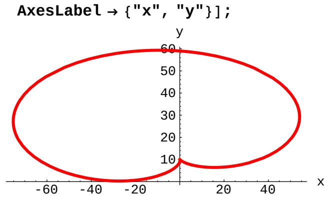
```
Clear[u, v, t]
{u[t_], v[t_]} = {3, 4} + 5 {Cos[t], Sin[t]};
```

```
ParametricPlot[
 {u[t], v[t]}, {t, 0, 2 π}, PlotStyle → {{Thickness[0.01], Red}},
 AspectRatio → Automatic, AxesLabel → {"u", "v"}];
```



Now look at the plot of this circle on xy-paper:

```
ParametricPlot[{x[u[t], v[t]], y[u[t], v[t]]}, {t, 0, 2 π},
 PlotStyle → {{Thickness[0.01], Red}}, AspectRatio → Automatic,
 AxesLabel → {"x", "y"}];
```



Sure enough; a dimple right at {0, 10}, which is the xy-plot of the uv-point {0, 0} because:

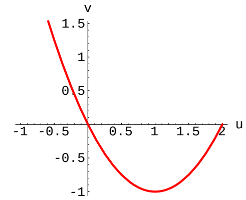
```
{x[0, 0], y[0, 0]}
```
{0, 10}

Now you try one.

G.6.a.i)

Here's a parametric plot of the parabola $v = u^2 - 2 u$ on uv-paper:

```
Clear[t]
ParametricPlot[
 {t, t^2 - 2 t}, {t, -1, 2}, PlotStyle → {{Thickness[0.01], Red}},
 AspectRatio → Automatic, AxesLabel → {"u", "v"}];
```

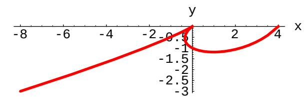


Go with:

```
Clear[u, v, x, y]
{x[u_, v_], y[u_, v_]} = {u^2 - v^2, u v}
```
$\{u^2 - v^2, u v\}$

Here is a plot of the same parabola on xy-paper:

```
ParametricPlot[{x[t, t^2 - 2 t], y[t, t^2 - 2 t]}, {t, -1, 2},
 PlotStyle → {{Thickness[0.01], Red}}, AspectRatio → Automatic,
 PlotRange → All, AxesLabel → {"x", "y"}];
```



Try to explain why you see what you see.

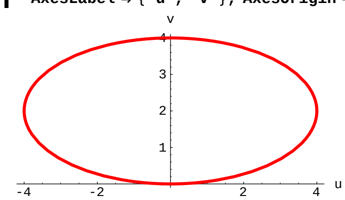
G.6.a.ii)

Here is a parametric plot of the ellipse

$$\left(\frac{u}{4}\right)^2 + \left(\frac{v-2}{2}\right)^2 = 1$$

on uv-paper:

```
Clear[t]
ParametricPlot[{4 Cos[t], 2 + 2 Sin[t]}, {t, 0, 2 π},
 PlotStyle → {{Thickness[0.01], Red}}, AspectRatio → Automatic,
 AxesLabel → {"u", "v"}, AxesOrigin → {0, 0}];
```

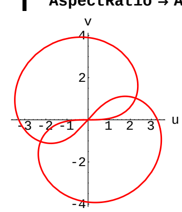


Go with polar paper:

```
Clear[u, v, x, y]
{x[u_, v_], y[u_, v_]} = {u Cos[v], u Sin[v]}
```
{u Cos[v], u Sin[v]}

Here's a plot of the same ellipse on xy-paper:

```
Clear[t]
ParametricPlot[{x[4 Cos[t], 2 + 2 Sin[t]], y[4 Cos[t], 2 + 2 Sin[t]]},
 {t, 0, 2 π}, PlotStyle → {{Thickness[0.01], Red}},
 AspectRatio → Automatic, AxesLabel → {"u", "v"}];
```

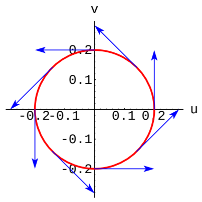


Try to explain why you see what you see.

G.6.b.i) The sign of Det[{gradx[u, v], grady[u, v]}]

Here is a rather small circle on uv-paper parameterized in the counterclockwise direction:

```
Clear[u, v, t, uvpaperplotter]
r = 0.2;
uvpaperplotter[t_] = {r Cos[t], r Sin[t]};
uvpaperplot = ParametricPlot[uvpaperplotter[t],
  {t, 0, 2 π}, PlotStyle → {{Thickness[0.01], Red}},
  AxesLabel → {"u", "v"}, DisplayFunction → Identity];
jump = π/4;
uvtangentvectors =
 Table[Arrow[uvpaperplotter'[t], Tail → uvpaperplotter[t],
   VectorColor → Blue], {t, 0, 2 π - jump, jump}];
Show[uvpaperplot, uvtangentvectors, AspectRatio → Automatic,
 AxesLabel → {"u", "v"}, DisplayFunction → $DisplayFunction];
```



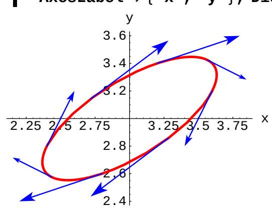
Now go to xy-paper with:

```
Clear[x, y, u, v]
{x[u_, v_], y[u_, v_]} = {3 u - v + 3, u - 2 v + 3}
```

```
{3 + 3 u - v, 3 + u - 2 v}
```

Here's the xy-paper plot of the same circle with tangent vectors coming from its resulting xy-parameterization:

```
Clear[xypaperplotter]
xypaperplotter[t_] = {x@@uvpaperplotter[t], y@@uvpaperplotter[t]};
xypaperplot = ParametricPlot[xypaperplotter[t],
  {t, 0, 2 π}, PlotStyle → {{Thickness[0.01], Red}},
  AxesLabel → {"u", "v"}, DisplayFunction → Identity];
jump = π/4;
xytangentvectors =
 Table[Arrow[xypaperplotter'[t], Tail → xypaperplotter[t],
   VectorColor → Blue], {t, 0, 2 π - jump, jump}];
Show[xypaperplot, xytangentvectors, AspectRatio → Automatic,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```



Even though the curve was parameterized in the counterclockwise direction on the original uv-paper, it is parameterized in the clockwise direction on xy-paper.

Now look at A_{xy}[u, v] with the absolute value dropped:

```
Clear[gradx, grady, Axy]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = Det[{gradx[u, v], grady[u, v]}]
```

```
-5
```

Negative.

Do you think this is an accident?

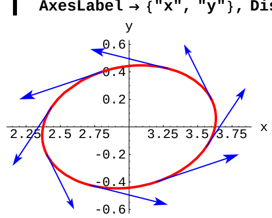
□G.6.b.ii)

Here's what happens on xy-paper to the same uv-circle when you go with

x[u, v] = 3 u − v + 3, and

y[u, v] = u + 2 v:

```
Clear[x, y, u, v, xypaperplotter]
{x[u_, v_], y[u_, v_]} = {3 u - v + 3, u + 2 v};
xypaperplotter[t_] = {x@@uvpaperplotter[t], y@@uvpaperplotter[t]};
xypaperplot = ParametricPlot[xypaperplotter[t],
  {t, 0, 2 π}, PlotStyle → {{Thickness[0.01], Red}},
  AxesLabel → {"u", "v"}, DisplayFunction → Identity];
jump = π/4;
xytangentvectors =
 Table[Arrow[xypaperplotter'[t], Tail → xypaperplotter[t],
   VectorColor → Blue], {t, 0, 2 π - jump, jump}];

Show[xypaperplot, xytangentvectors, AspectRatio → Automatic,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```



Counterclockwise on xy-paper.

Now look at A_{xy}[u, v] with the absolute value dropped:

```
Clear[gradx, grady, Axy]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = Det[{gradx[u, v], grady[u, v]}]
```

```
7
```

Are you surprised?

Why or why not?

□G.6.b.iii)

Fill in numbers

a, b, c, d, e, and f

of your own choice. Go with

x[u, v] = a u + b v + c, and

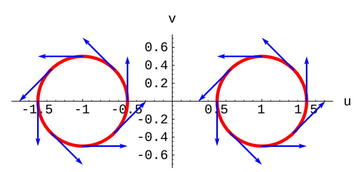
y[u, v] = d u + e v + f,

and investigate, as above, what happens to the same uv-circle on xy-paper, and how what happens is related to the sign of Det[{gradx[u, v], grady[u, v]}].

□G.6.b.iv)

Here are two small circles on uv-paper parameterized in the counterclockwise direction:

```
Clear[u, v, t, uvpaperplotter1, uvpaperplotter2]
r = 0.5;
uvpaperplotter1[t_] = {-1, 0} + {r Cos[t], r Sin[t]};
uvpaperplotter2[t_] = {1, 0} + {r Cos[t], r Sin[t]};
uvpaperplot =
 ParametricPlot[{uvpaperplotter1[t], uvpaperplotter2[t]},
  {t, 0, 2 π}, PlotStyle → {{Thickness[0.01], Red}},
  AxesLabel → {"u", "v"}, DisplayFunction → Identity];
jump = π/4;
uvtangentvectors =
 Table[{Arrow[uvpaperplotter1'[t], Tail → uvpaperplotter1[t],
    VectorColor → Blue], Arrow[uvpaperplotter2'[t],
    Tail → uvpaperplotter2[t], VectorColor → Blue]},
  {t, 0, 2 π - jump, jump}];

Show[uvpaperplot, uvtangentvectors, AspectRatio → Automatic,
 AxesLabel → {"u", "v"}, DisplayFunction → $DisplayFunction];
```

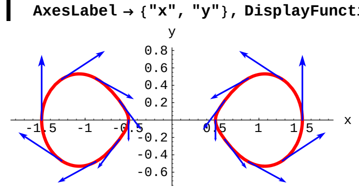


Here are the xy-paper plot of the same circles with tangent vectors coming from its own xy-parameterization when you go with

x[u, v] = u Cos[v], and

y[u, v] = u Sin[v]:

```
Clear[x, y, u, v]
{x[u_, v_], y[u_, v_]} = {u Cos[v], u Sin[v]};
Clear[xypaperplotter1, xypaperplotter2]
xypaperplotter1[t_] =
 {x@@uvpaperplotter1[t], y@@uvpaperplotter1[t]};
xypaperplotter2[t_] =
 {x@@uvpaperplotter2[t], y@@uvpaperplotter2[t]};
xypaperplot =
 ParametricPlot[{xypaperplotter1[t], xypaperplotter2[t]},
  {t, 0, 2 π}, PlotStyle → {{Thickness[0.01], Red}},
  AxesLabel → {"u", "v"}, DisplayFunction → Identity];
jump = π/4;
xytangentvectors =
 Table[{Arrow[xypaperplotter1'[t], Tail → xypaperplotter1[t],
    VectorColor → Blue], Arrow[xypaperplotter2'[t],
    Tail → xypaperplotter2[t], VectorColor → Blue]},
  {t, 0, 2 π - jump, jump}];

Show[xypaperplot, xytangentvectors, AspectRatio → Automatic,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```



Now look at:

```
Clear[gradx, grady, Axy]
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = TrigExpand[Det[{gradx[u, v], grady[u, v]}]]
```

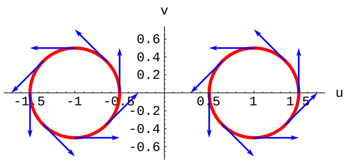
u

Say why you think one of xy-parameterizations was reversed from counterclockwise to clockwise, but the other remained counterclockwise.

□ Tip:

Look hard at the plots of the original circles on uv-paper:

```
Show[uvpaperplot, uvtangentvectors, AspectRatio → Automatic,
 AxesLabel → {"u", "v"}, DisplayFunction → $DisplayFunction];
```



And look hard at:

```
Axy[u, v]
```

u

G.7) Volume, mass, and density

□ G.7.a.i)

An object is made by forming a uniform substance that has density $1.71\ \frac{\text{grams}}{\text{unit}^3}$ in the shape of of the surface

$f[x, y] = \frac{3.14}{1 + x^2 + y^2}$

over the region inside and on the circle $x^2 + y^2 = 8$ on xy-paper.
Use polar coordinate paper with

$x[u, v] = u \operatorname{Cos}[v]$, and
$y[u, v] = u \operatorname{Sin}[v]$

to measure the total mass and volume of this object.

□ G.7.a.ii)

When you rubberized the xy-paper and stretched it out to make the uv-paper, you also deformed the original shape of the object in part i).
What does the resulting deformed object look like when it is plotted on uv-paper for the functions

$x[u, v] = u \operatorname{Cos}[v]$, and
$y[u, v] = u \operatorname{Sin}[v]$

used above in part i)?
What is the density of this deformed object at a point {u, v} inside the uv-paper rectangle $0 \le u \le \operatorname{Sqrt}[8]$ and $0 \le v \le 2\pi$?

G.8) Two recreational plots

If you want to find out why these plots turn out the way they do, take a complex variables course, and pay special attention to the part on conformal mapping.

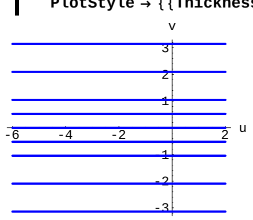
□ G.8.a) Flow out of an open pipe

Here is a uv-paper plot of the line segments

$v = -\pi,\ v = -\frac{2\pi}{3},\ v = -\frac{\pi}{3},\ v = -\frac{\pi}{6},$
$v = 0,\ v = \frac{\pi}{6},\ v = \frac{\pi}{3},\ v = \frac{2\pi}{3}$, and $v = \pi$

with $-6 \le u \le 2$:

```
Clear[t]
uvplot = ParametricPlot[{{t, -π}, {t, -2π/3}, {t, -π/3}, {t, -π/6},
  {t, 0}, {t, π/6}, {t, π/3}, {t, 2π/3}, {t, π}}, {t, -6, 2},
  PlotStyle → {{Thickness[0.01], Blue}}, AxesLabel → {"u", "v"}];
```



Plot these curves on xy-paper for

$x[u, v] = u + e^u \operatorname{Cos}[v]$, and
$y[u, v] = v + e^u \operatorname{Sin}[v]$.

If your xy-paper plot is correct, then it will depict streamlines of water flow out of an open pipe.

□ G.8.b) Airfoils

Ever wonder how to plot some airfoils (airplane wings)?
If so, then wait no longer.
Take

$x[u, v] = \frac{u^3 + u v^2 + u}{u^2 + v^2}$, and
$y[u, v] = \frac{v^3 + u^2 v - v}{u^2 + v^2}$,

and plot on xy-paper some uv-paper circles

$(u - a)^2 + v^2 = (1 + a)^2$

for several choices of a with $0 < a < 1$.

G.9) What went wrong?*

□ G.9.a)

Let R_{xy} be the region on xy-paper consisting of everything inside and on the circle

$x^2 + y^2 = 4.$

Polar cordinate paper is handy for calculating

$\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy$:

Go to polar coordinates via:

```
Clear[x, y, u, v, gradx, grady, Axy]
x[u_, v_] = u Cos[v];
y[u_, v_] = u Sin[v];
gradx[u_, v_] = {D[x[u, v], u], D[x[u, v], v]};
grady[u_, v_] = {D[y[u, v], u], D[y[u, v], v]};
Axy[u_, v_] = TrigExpand[Det[{gradx[u, v], grady[u, v]}]]
```

u

```
Clear[f]
f[x_, y_] = x^2 + y^2
```

$x^2 + y^2$

Here are four attempts at calculations of

$\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy.$

Identify the correct calculations, and determine what went wrong in the incorrect calculations.
There is always a possibility that a calculation produces a correct answer, but the method is wrong. Identify these as well.

□ Calculation 1:

The uv-paper rectangle $0 \le u \le 4$ and $0 \le v \le 2\pi$ plots out on xy-paper as the circle $x^2 + y^2 = 4$ and everything inside it.
So $\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy$ is given by:

```
∫_0^(2π) ∫_0^4 f[x[u, v], y[u, v]] Axy[u, v] du dv
```

128π

□ Calculation 2:

The uv-paper rectangle $0 \le u \le 4$ and $0 \le v \le 2\pi$ plots out on xy-paper as the circle $x^2 + y^2 = 4$ and everything inside it.
So $\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy$ is given by:

```
∫_0^(2π) ∫_0^2 f[x[u, v], y[u, v]] Axy[u, v] du dv
```

8π

□ Calculation 3:

The uv-paper rectangle $0 \le u \le 4$ and $0 \le v \le 2\pi$ plots out on xy-paper as the circle $x^2 + y^2 = 4$ and everything inside it.
So $\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy$ is given by:

```
∫_0^(4π) ∫_0^2 f[x[u, v], y[u, v]] Axy[u, v] du dv
```

16π

□Calculation 4:

The uv-paper rectangle $0 \le u \le 4$ and $0 \le v \le 2\pi$ plots out on xy-paper as the circle $x^2 + y^2 = 4$ and everything inside it.
So $\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy$ is given by:

$$\int_{-2}^{0}\int_{0}^{2\pi} f[x[u, v], y[u, v]]\, Axy[u, v]\, dv\, du$$

-8π

□Calculation 5:

The uv-paper rectangle $0 \le u \le 4$ and $0 \le v \le 2\pi$ plots out on xy-paper as the circle $x^2 + y^2 = 4$ and everything inside it.
So $\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy$ is given by:

$$\int_{0}^{2}\int_{-\pi}^{\pi} f[x[u, v], y[u, v]]\, Axy[u, v]\, dv\, du$$

8π

□Calculation 6:

The uv-paper rectangle $0 \le u \le 4$ and $0 \le v \le 2\pi$ plots out on xy-paper as the circle $x^2 + y^2 = 4$ and everything inside it.
So $\int\int_{R_{xy}} (x^2 + y^2)\, dx\, dy$ is given by:

$$\int_{0}^{2}\int_{10\pi}^{12\pi} f[x[u, v], y[u, v]]\, Axy[u, v]\, dv\, du$$

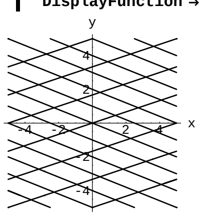
8π

G.10) Linear equations and area conversion factors

When you go from xy-paper to uv-paper through formulas
  u[x, y] = a x + b y, and
  v[x, y] = c x + d y,
for fixed numbers a, b, c, and d, things get pictorial. For one thing, the uv-grid on xy-paper comes from straight lines:

```
{a, b, c, d} = {2, 5, -1, 3};
Clear[u, v, x, y]
u[x_, y_] = a x + b y;
v[x_, y_] = c x + d y;
{{xlow, xhigh}, {ylow, yhigh}} = {{-5, 5}, {-5, 5}};
```

```
ulevelcurves =
 ContourPlot[Evaluate[u[x, y]], {x, xlow, xhigh}, {y, ylow, yhigh},
  Contours → {-30, -25, -20, -15, -10, -5, 0, 5, 10, 15, 20, 25, 30},
  ContourShading → False, DisplayFunction → Identity];
vlevelcurves = ContourPlot[Evaluate[v[x, y]], {x, xlow, xhigh},
  {y, ylow, yhigh}, Contours → {-15, -10, -5, 0, 5, 10, 15, 20, 25, 30},
  ContourShading → False, DisplayFunction → Identity];
uvGridonxyPaper = Show[ulevelcurves, vlevelcurves, Frame → False];

Show[uvGridonxyPaper, Axes → True, AxesLabel → {"x", "y"},
 DisplayFunction → $DisplayFunction];
```



This is one reason that lots of folks call the transformation from xy-paper to uv-paper coming from
  u[x, y] = a x + b y, and
  v[x, y] = c x + d y
by the name linear transformation. When you take a course called linear algebra, you will hear a lot about linear transformations. In this problem, you get to see some stuff about linear transformations that you won't see in a standard linear algebra course.

□G.10.a.i) Tripping

Go with:

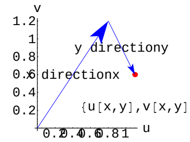
```
Clear[x, y, u, v]
{a, b, c, d} = {2, -1, 3, 2};
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y}
{2 x - y, 3 x + 2 y}
```

Make two vectors:

```
directionx = {a, c}
{2, 3}
directiony = {b, d}
{-1, 2}
```

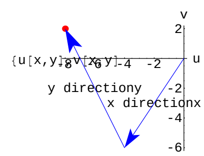
Look at these uv-paper plots of
  x directionx with its tail at {0, 0} and
  y directiony with its tail at the tip of x directionx
and the point
  {u[x, y], v[x, y]} for x = 0.4 and y = −0.3:

```
{x, y} = {0.4, -0.3};
uvpoint = Graphics[{Red, PointSize[0.05], Point[{u[x, y], v[x, y]}]}];
trip = {Arrow[x directionx, Tail → {0, 0}],
  Arrow[y directiony, Tail → x directionx]};
labels = {Graphics[Text["x directionx", x directionx/2]],
  Graphics[Text["y directiony", x directionx + y directiony/2]],
  Graphics[Text["{u[x,y],v[x,y]", {u[x, y], v[x, y]}, {0, 4}]]};

Show[uvpoint, labels, trip, PlotRange → All, Axes → True,
 AxesLabel → {"u", "v"}];
```



Go again with the same u[x, y] and v[x, y] but this time take x = −2 and y = 4:

```
{x, y} = {-2, 4};
uvpoint = Graphics[{Red, PointSize[0.05], Point[{u[x, y], v[x, y]}]}];
trip = {Arrow[x directionx, Tail → {0, 0}],
  Arrow[y directiony, Tail → x directionx]};
labels = {Graphics[Text["x directionx", x directionx/2]],
  Graphics[Text["y directiony", x directionx + y directiony/2]],
  Graphics[Text["{u[x,y],v[x,y]", {u[x, y], v[x, y]}, {0, 4}]]};

Show[uvpoint, labels, trip, PlotRange → All, Axes → True,
 AxesLabel → {"u", "v"}];
```



Read the code, edit it and and run a few more if you like. Once you are certain what's happening, explain the physical process of going with an xy-paper point {x, y} and the vectors directionx and directiony to take a trip starting at {0, 0} and ending at the uv-paper plot {u[x, y], v[x, y]} of the original xy-paper point {x, y}.

□Tip:

What you saw was the geometry of the trip.

Here's the algebra:

```
Clear[x, y, u, v, a, b, c, d];
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y}
{a x + b y, c x + d y}
directionx = {a, c}
{a, c}
directiony = {b, d}
{b, d}
{u[x, y], v[x, y]} == x directionx + y directiony
True
```

□G.10.a.ii) Tripping when directionx and directiony are either parallel or point in opposite directions

Go with:

```
Clear[x, y, u, v]
{a, b, c, d} = {4, 6, 2, 3};
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y}
{4 x + 6 y, 2 x + 3 y}
```

Make two the vectors:

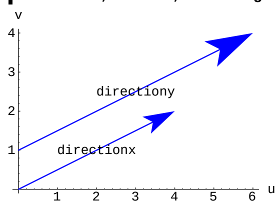
```
directionx = {a, c}
{4, 2}
```

```
directiony = {b, d}
```

{6, 3}

Note that the vectors directionx and directiony are parallel:

```
vectors = {Arrow[directionx, Tail → {0, 0}, VectorColor → Blue],
  Arrow[directiony, Tail → {0, 1}, VectorColor → Blue]};
labels = {Graphics[Text["directionx", directionx/2]],
  Graphics[Text["directiony", {0, 1} + directiony/2]]};

Show[
 vectors, labels, PlotRange → All, Axes → True, AxesLabel → {"u", "v"}];
```



Look at these uv-paper plots of

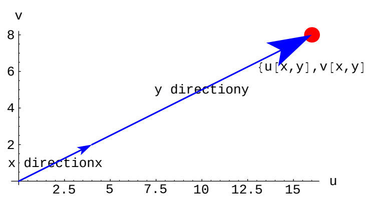
x directionx with its tail at {0, 0}, and

y directiony with its tail at the tip of x directionx,

and the point

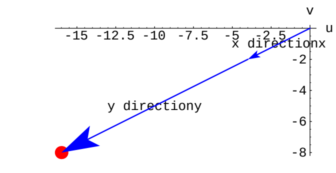
{u[x, y], v[x, y]} for x = 1 and y = 2:

```
{x, y} = {1, 2};
uvpoint = Graphics[{Red, PointSize[0.05], Point[{u[x, y], v[x, y]}]}];
trip = {Arrow[x directionx, Tail → {0, 0}, VectorColor → Blue],
  Arrow[y directiony, Tail → x directionx, VectorColor → Blue]};
labels = {Graphics[Text["x directionx", x directionx/2]],
  Graphics[Text["y directiony", x directionx + y directiony/2]],
  Graphics[Text["{u[x,y],v[x,y]", {u[x, y], v[x, y]}, {0, 4}]]};

Show[uvpoint, labels, trip, PlotRange → All, Axes → True,
 AxesLabel → {"u", "v"}];
```



Go again with the same u[x, y] and v[x, y], but this time take x = −1 and y = −2:

```
{x, y} = {-1, -2};
uvpoint = Graphics[{Red, PointSize[0.05], Point[{u[x, y], v[x, y]}]}];
trip = {Arrow[x directionx, Tail → {0, 0}, VectorColor → Blue],
  Arrow[y directiony, Tail → x directionx, VectorColor → Blue]};
labels = {Graphics[Text["x directionx", x directionx/2]],
  Graphics[Text["y directiony", x directionx + y directiony/2]],
  Graphics[Text["{u[x,y],v[x,y]", {u[x, y], v[x, y]}, {0, 4}]]};

Show[uvpoint, labels, trip, PlotRange → All, Axes → True,
 AxesLabel → {"u", "v"}];
```



Edit and run a few more if you like.

Once you smell the truth, explain where the points {u[x, y], v[x, y]} must plot out as x and y vary.

□G.10.a.iii)

Continue to go with parallel vectors directionx and directiony as above.

```
Clear[x, y, u, v]
{a, b, c, d} = {4, 6, 2, 3};
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y}
```

{4 x + 6 y, 2 x + 3 y}

Make two the vectors:

```
directionx = {a, c}
```

{4, 2}

```
directiony = {b, d}
```

{6, 3}

Remember that:

```
{u[x, y], v[x, y]} == x directionx + y directiony
```

True

Look at the following attempt to solve:

```
Solve[{0 == a x + b y, 2 == c x + d y}, {x, y}]
```

{}

Use the fact that

{u[x, y], v[x, y]}

= {a x + b y, c x + d y}

= x directionx + y directiony

to explain why there was never any hope of solving the simultaneous equations

0 = a x + b y,

2 = c x + d y

for x and y. Then look at:

```
directionx
```

{4, 2}

```
directiony
```

{6, 3}

Say how the numbers u and v must be related to guarantee that the simultaneous equations

u = a x + b y

v = c x + d y

can be solved for x and y.

□G.10.a.iv)

Continue to go with parallel vectors directionx and directiony as above.

```
Clear[x, y, u, v]
{a, b, c, d} = {4, 6, 2, 3};
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y}
```

{4 x + 6 y, 2 x + 3 y}

This time, look at the xy-to-uv-area conversion factor:

```
Clear[gradu, gradv, Auv]
gradu[x_, y_] = {D[u[x, y], x], D[u[x, y], y]};
gradv[x_, v_] = {D[v[x, y], x], D[v[x, y], y]};
Auv[x_, y_] = Det[{gradu[x, y], gradv[x, y]}]
```

0

What happens to the xy-paper when it is stretched and compressed to make uv-paper?

How does this explain why when you go with

a = 4, b = 6, c = 2, and d = 3

as above, then for all but very special choices of u and v, it will be impossible to solve

u = a x + b y

v = c x + d y

for x and y?

□G.10.b)

This time go with cleared values of a, b, c, and d:

```
Clear[x, y, u, v, a, b, c, d]
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y}
```

{a x + b y, c x + d y}

```
directionx = {a, c}
```

{a, c}

```
directiony = {b, d}
```

{b, d}

Look at the xy-to-uv-area conversion factor:

```
Clear[gradu, gradv, Auv]
gradu[x_, y_] = {D[u[x, y], x], D[u[x, y], y]};
gradv[x_, y_] = {D[v[x, y], x], D[v[x, y], y]};
Auv[x_, y_] = Det[{gradu[x, y], gradv[x, y]}]
```

```
-b c + a d
```

Explain:
Saying that the vectors directionx and directiony are either parallel or point in opposite directions is the same as saying that the area conversion factor $A_{uv}[x, y] = 0$.
And explain:
If you go with numbers a, b, c, and d that make

$A_{uv}[x, y] = 0$,

then for all but very special choices of numbers u and v, it will be impossible to solve

$u = a x + b y$
$v = c x + d y$

for x and y.

□G.10.b.ii)

Look at this:

```
Clear[x, y, u, v, a, b, c, d, t];
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y}
{a x + b y, c x + d y}
Clear[gradu, gradv, Auv]
gradu[x_, y_] = {D[u[x, y], x], D[u[x, y], y]};
gradv[x_, y_] = {D[v[x, y], x], D[v[x, y], y]};
Auv[x_, y_] = Det[{gradu[x, y], gradv[x, y]}]
-b c + a d
```

Now look at what most folks call the coefficient matrix:

```
coefficientmatrix = {{a, b}, {c, d}};
MatrixForm[coefficientmatrix]
```

$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

And its determinant:

```
Det[coefficientmatrix]
-b c + a d
```

Compare the xy-to-uv-area conversion factor to the determinant of the coefficient matrix:

```
Det[coefficientmatrix] == Auv[x, y]
True
```

Explain:
When you go with numbers a, b, c, and d that make

determinant of the coefficient matrix = 0,

then for all but very special choices of u and v, it will be impossible to solve

$u = a x + b y$
$v = c x + d y$

for x and y.
And explain:
When you go with numbers a, b, c, and d that make

determinant of the coefficient matrix ≠ 0,

then, no matter what choice of u and v you make, it will be possible to solve

$u = a x + b y$
$v = c x + d y$

for x and y.

G.11) Eigenvalues and eigenvectors

One good way to extend your mathematical horizons is to look up some unfamiliar Mathematica instructions and try them out. This is what you will do here with a little coaching from your friends at Calculus&Mathematica.
Take four numbers a, b, c, and d with c = b like this:

```
{a, b, d} = {3.5, 1.0, 4.2};
{{a, b}, {c, d}} = {{a, b}, {b, d}};
matrix = {{a, b}, {c, d}};
MatrixForm[matrix]
```

$\begin{pmatrix} 3.5 & 1. \\ 1. & 4.2 \end{pmatrix}$

Go from xy-paper to uv-paper with

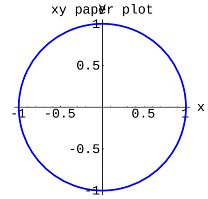
```
Clear[u, v, x, y]
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y}
{3.5 x + 1. y, 1. x + 4.2 y}
```

Here is the xy-paper circle

$x^2 + y^2 = 1$

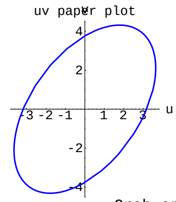
plotted on xy-paper:

```
xyplot = ParametricPlot[{Cos[t], Sin[t]}, {t, 0, 2 π},
  AspectRatio → Automatic, PlotStyle → {{Blue, Thickness[0.01]}},
  AxesLabel → {"x", "y"}, PlotLabel → "xy paper plot"];
```



Here is how this circle plots out on uv-paper in true scale:

```
uvplot =
 ParametricPlot[{u[Cos[t], Sin[t]], v[Cos[t], Sin[t]]}, {t, 0, 2 π},
  AspectRatio → Automatic, PlotStyle → {{Blue, Thickness[0.01]}},
  AxesLabel → {"u", "v"}, PlotLabel → "uv paper plot"];
```



Grab and animate these two plots and run slowly.

You guessed right!
This is a tilted ellipse centered at {0, 0}.
Here comes the unfamiliar Mathematica instruction:

```
eigens = Eigensystem[matrix]
{{4.90948, 2.79052}, {{-0.57864, -0.815583}, {-0.815583, 0.57864}}}
```

The eigenvalues of the matrix are:

```
{eigenvalue1, eigenvalue2} = eigens[[1]]
{4.90948, 2.79052}
```

The eigenvectors of the matrix are:

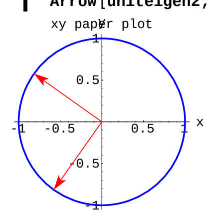
```
{eigenvector1, eigenvector2} = eigens[[2]]
{{-0.57864, -0.815583}, {-0.815583, 0.57864}}
```

The unit eigenvectors of the matrix are:

```
{uniteigen1, uniteigen2} =
 { eigenvector1 / Sqrt[eigenvector1 . eigenvector1], eigenvector2 / Sqrt[eigenvector2 . eigenvector2] }
{{-0.57864, -0.815583}, {-0.815583, 0.57864}}
```

Here are the unit eigenvectors of this matrix together with a plot of the circle $x^2 + y^2 = 1$:

```
Show[xyplot, Arrow[uniteigen1, Tail → {0, 0}, VectorColor → Red],
  Arrow[uniteigen2, Tail → {0, 0}, VectorColor → Red]];
```

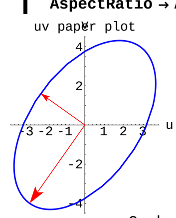


Now put:

```
{amazing1, amazing2} = {eigenvalue1 uniteigen1, eigenvalue2 uniteigen2}
{{-2.84082, -4.00409}, {-2.2759, 1.61471}}
```

And look at this plot:

```
Show[uvplot, Arrow[amazing1, Tail → {0, 0}, VectorColor → Red],
  Arrow[amazing2, Tail → {0, 0}, VectorColor → Red],
  AspectRatio → Automatic];
```



Grab and animate these plots, running slowly.

Bingo.
Just for good measure, look at:

```
Clear[gradu, gradv, Auv]
gradu[x_, y_] = {D[u[x, y], x], D[u[x, y], y]};
gradv[x_, y_] = {D[v[x, y], x], D[v[x, y], y]};
Auv[x_, y_] = Det[{gradu[x, y], gradv[x, y]}];
{eigenvalue1 eigenvalue2, Auv[x, y]}
```

```
{13.7, 13.7}
```

Cowabunga.
The xy-to-uv-area conversion factor is the same as the product of the two eigenvalues.

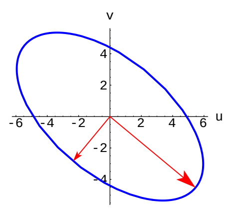
□G.11.a)

Remembering that the ellipse above is the uv-paper plot of the xy-paper circle $x^2 + y^2 = 1$, write down the uv-paper measurement of the area enclosed by the ellipse plotted above.

□G.11.b)

Do the it again for another choice of
a, b, c, and d with c = b:

```
{a, b, d} = {2.7, -5.3, 0.7};
{{a, b}, {c, d}} = {{a, b}, {b, d}};
matrix = {{a, b}, {c, d}};
Clear[u, v, x, y]
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y};
uvplot =
 ParametricPlot[{u[Cos[t], Sin[t]], v[Cos[t], Sin[t]]}, {t, 0, 2 π},
  AspectRatio → Automatic, PlotStyle → {{Blue, Thickness[0.01]}},
  AxesLabel → {"u", "v"}, DisplayFunction → Identity];
eigens = Eigensystem[matrix];
{eigenvalue1, eigenvalue2} = eigens[[1]];
{eigenvector1, eigenvector2} = eigens[[2]];
{uniteigen1, uniteigen2} =
 {eigenvector1/√(eigenvector1 . eigenvector1), eigenvector2/√(eigenvector2 . eigenvector2)};
{amazing1, amazing2} =
 {eigenvalue1 uniteigen1, eigenvalue2 uniteigen2};

Show[uvplot, Arrow[amazing1, Tail → {0, 0}, VectorColor → Red],
 Arrow[amazing2, Tail → {0, 0}, VectorColor → Red],
 DisplayFunction → $DisplayFunction];
```



Bingo again.
Again just for good measure, look at:

```
Clear[gradu, gradv, Auv]
gradu[x_, y_] = {D[u[x, y], x], D[u[x, y], y]};
gradv[x_, y_] = {D[v[x, y], x], D[v[x, y], y]};
Auv[x_, y_] = Det[{gradu[x, y], gradv[x, y]}];
{eigenvalue1 eigenvalue2, Auv[x, y]}
```

```
{-26.2, -26.2}
```

Totally cool.
Play with some more choices of a, b, c, and d with b = c until you get enough experience to form your own opinions about the anwers to the questions:
→ What do you think unit eigenvectors are?
→ What do you think eigenvalues are?
→ Why are lots of folks delighted with eigenvectors and eigenvalues?

□Tip:

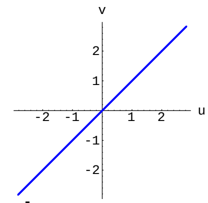
Don't look up the answer in a linear algebra book because most linear algebra books don't look at eigenvector and eigenvalues from this visual perspective. Instead, they make eigenvector and eigenvalues the final products of some dreary algebra.

How sad.

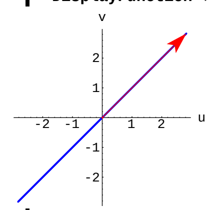
□G.11.c)

Look at this one:

```
{a, b, d} = {2, 2, 2};
{{a, b}, {c, d}} = {{a, b}, {b, d}};
matrix = {{a, b}, {c, d}};
Clear[u, v, x, y]
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y};
uvplot =
 ParametricPlot[{u[Cos[t], Sin[t]], v[Cos[t], Sin[t]]}, {t, 0, 2 π},
  AspectRatio → Automatic, PlotStyle → {{Blue, Thickness[0.01]}},
  AxesLabel → {"u", "v"}, DisplayFunction → Identity];
eigens = Eigensystem[matrix];
{eigenvalue1, eigenvalue2} = eigens[[1]];
{eigenvector1, eigenvector2} = eigens[[2]];
{uniteigen1, uniteigen2} =
 {eigenvector1/√(eigenvector1 . eigenvector1), eigenvector2/√(eigenvector2 . eigenvector2)};
{amazing1, amazing2} =
 {eigenvalue1 uniteigen1, eigenvalue2 uniteigen2};
Show[uvplot, DisplayFunction → $DisplayFunction];
```



```
Show[uvplot, Arrow[amazing1, Tail → {0, 0}, VectorColor → Red],
 Arrow[amazing2, Tail → {0, 0}, VectorColor → Red],
 DisplayFunction → $DisplayFunction];
```



```
{eigenvalue1, eigenvalue2}
```

```
{0, 4}
```

```
Clear[gradu, gradv, Auv]
gradu[x_, y_] = {D[u[x, y], x], D[u[x, y], y]};
gradv[x_, y_] = {D[v[x, y], x], D[v[x, y], y]};
Auv[x_, y_] = Det[{gradu[x, y], gradv[x, y]}];
{eigenvalue1 eigenvalue2, Auv[x, y]}
```

```
{0, 0}
```

What do you think happened to the ellipse?

□G.11.d)

If what you've seen so far hasn't blown your mind, look at these:

```
{a, b, d} = {9, -4, 2};
{{a, b}, {c, d}} = {{a, b}, {b, d}};
matrix = {{a, b}, {c, d}};
Clear[u, v, x, y]
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y};
eigens = Eigensystem[matrix];
{eigenvalue1, eigenvalue2} = eigens[[1]];
{eigenvector1, eigenvector2} = eigens[[2]];
{uniteigen1, uniteigen2} =
 {eigenvector1/√(eigenvector1 . eigenvector1), eigenvector2/√(eigenvector2 . eigenvector2)};
{amazing1, amazing2} =
 {eigenvalue1 uniteigen1, eigenvalue2 uniteigen2};
{u[x, y], v[x, y]} == ExpandAll[
 Together[{x, y} . uniteigen1 amazing1 + {x, y} . uniteigen2 amazing2]]
```

```
True
```

It works for any a, b, c, and d you feed in as long as b = c:

```
{a, b, d} = {Random[Real, {-10, 10}],
 Random[Real, {-10, 10}], Random[Real, {-10, 10}]};
{{a, b}, {c, d}} = {{a, b}, {b, d}};
matrix = {{a, b}, {c, d}};
Clear[u, v, x, y]
{u[x_, y_], v[x_, y_]} = {a x + b y, c x + d y};
eigens = Eigensystem[matrix];
{eigenvalue1, eigenvalue2} = eigens[[1]];
{eigenvector1, eigenvector2} = eigens[[2]];
{uniteigen1, uniteigen2} =
 {eigenvector1/√(eigenvector1 . eigenvector1), eigenvector2/√(eigenvector2 . eigenvector2)};
{amazing1, amazing2} =
 {eigenvalue1 uniteigen1, eigenvalue2 uniteigen2};
{u[x, y], v[x, y]} == ExpandAll[
 Together[{x, y} . uniteigen1 amazing1 + {x, y} . uniteigen2 amazing2]]
```

```
True
```

In other words when you go with any numbers a, b, c, and d with b = c, and you put
u[x, y] = a x + b y, and

v[x, y] = c x + d y,

then you can count on having

{u[x, y], v[x, y]}

= ({x, y} . uniteigen1) amazing1 + ({x, y} . uniteigen2) amazing2.

This resolves {u[x, y], v[x, y]} into two perpendicular components in the directions of the two eigenvectors.

Fancy folks call this the "spectral theorem" and make this the centerpiece of a good linear algebra course. If you did the problem G.10) above, then some interesting plots might come to your mind.

But you are not asked to write anything about what all this means; so sit back and enjoy.

On the other hand, if you've got something to say, then go ahead and indulge yourself.