# Work in Conservative and Non-Conservative Force Fields

*Note: You may notice differences between this Maple worksheet and the equivalent Mathematica notebook. These differences were introduced to preserve the content of these modules and were necessary because of major functional differences between Maple and Mathematica.*

## Introduction

OBJECTIVE: Visualize and evaluate work integrals along different paths, and observe the effect of following different paths through conservative and non-conservative force fields.

You will explore integration over vector fields and experiment with both conservative and non-conservative force functions and different paths. These explorations should help you understand line integrals, as well as better appreciate situations when the work done is independent of the path taken.

## Technology Guidelines

NOTE: If you have just finished a worksheet, **restart** *Maple* before executing a new worksheet.
TO OPEN SECTIONS,
Click on the **PLUS** sign at the left hand side of the screen *or* select **Expand All Sections** from the **View** drop down menu.

TO STOP AN EXECUTION
Click on **STOP** button from the toolbar.

ORDER OF EXECUTION
Execute commands in the order given. Do not skip any *Maple* Input lines within a given worksheet

Alternatively, you can execute the entire worksheet by selecting the **Execute Worksheet** command from the **Edit** drop down menu.

SAVING WORKSHEETS.

You can save anytime to any directory you choose, and it is wise to save often.
EXPERIENCING MAJOR PROBLEMS
Save if appropriate, and then shut down *Maple* and start it up again.

## Part I: Examples in Two Dimensions

### Conservative Force

#### Verifying that the Force is Conservative

Consider the following force defined by $[-x\cos(2y), x^2\sin(2Y)]$ , and verify that it is conservative.

```
> force:=[-x*cos(2*y), x^2*sin(2*y)];
  my:=diff(force[1],y);
  nx:=diff(force[2],x);
  if (my=nx) then print(`The force is conservative.`) else print(`The force is not c
```

$$force := [-x\cos(2y), x^2\sin(2y)]$$

$$my := 2\,x\sin(2y)$$

$$nx := 2\,x\sin(2y)$$

*The force is conservative.*

# Visualizing the Force Field and Different Paths

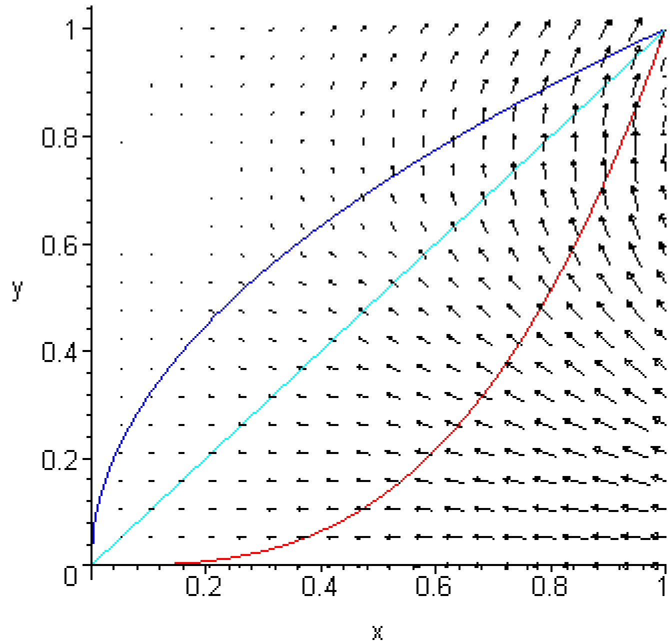
To visualize the force field, you need to first load the plots package.

```
> with(plots):
```

Warning, the name changecoords has been redefined

The force field is plotted together with three paths between (0, 0) and (1, 1): $y = x$, $\sqrt{x}$ and $x^3$ .

```
> pv:=fieldplot(force,x=0..1,y=0..1,arrows=SLIM):
  pc:=plot({x,sqrt(x),x^3}, x=0..1, color=[red,cyan,blue]):
  print(display({pv,pc}));
```




# Writing Parametrizations and Computing Work Integrals

Find the work done in traveling along the straight line, $y = x$. Choose an appropriate parameterization. We write the position and velocity vector to assist in computing the work integral. Note that by setting $x$ and $y$ equal to particular functions of $t$, the force function will reflect that parametrization when it appears in the line integral.

```
> x:=t:
  y:=t:
  r1:=[x,y]:
  v1:=diff(r1,t):print(`velocity = `, v1);
  print(`force function along curve = `,force);
  w1:=int(linalg[dotprod](force,v1), t=0..1):
  print(`work done along path = `, evalf(w1));
```

$$velocity = , [1, 1]$$

$$force\ function\ along\ curve = , [-t\cos(2t), t^2\sin(2t)]$$

$$work\ done\ along\ path = , 0.2080734182$$

Now find the work done in traveling along the lower curve $y = x^3$. Choose an appropriate parametrization.

```
> x:=t:
  y:=t^3:
```

```
r2:=[x,y]:
v2:=diff(r2,t):
print(`velocity = `, v2);
print(`force function along curve = `,force);
w2:=int(linalg[dotprod](force,v2), t=0..1):
print(`work done along path = `, evalf(w2));
```

$$\mathit{velocity} = , [1, 3\,t^2]$$

$$\mathit{force\ function\ along\ curve} = , [-t\cos(2\,t^3), t^2\sin(2\,t^3)]$$

$$\mathit{work\ done\ along\ path} = , 0.2080734183$$

Now find the work done in traveling along the lower curve $y = \sqrt{x}$ . As before, choose an appropriate parametrization.

```
> x:=t:
  y:=sqrt(t):
  r3:=[x,y]:
  v3:=diff(r3,t):
  print(`velocity = `, v3);
  print(`force function along curve = `,force);
  w3:=int(linalg[dotprod](force,v3), t=0..1):
  print(`work done along path = `, evalf(w3));
```

$$\mathit{velocity} = , \left[1, \frac{1}{2\sqrt{t}}\right]$$

$$\mathit{force\ function\ along\ curve} = , [-t\cos(2\sqrt{t}), t^2\sin(2\sqrt{t})]$$

$$\mathit{work\ done\ along\ path} = , 0.2080734182$$

Was the work done along each path the same?

# Non-Conservative Force

Consider the a spinning force, and verify that it is not conservative.

```
> unassign('x','y','z','force');
  force:=[-y/sqrt(x^2+y^2),x/sqrt(x^2+y^2)]:
  my:=simplify(diff(force[1],y));
  nx:=simplify(diff(force[2],x));
  simplify(my=nx);
```

$$my := -\frac{x^2}{(x^2+y^2)^{\left(\frac{3}{2}\right)}}$$

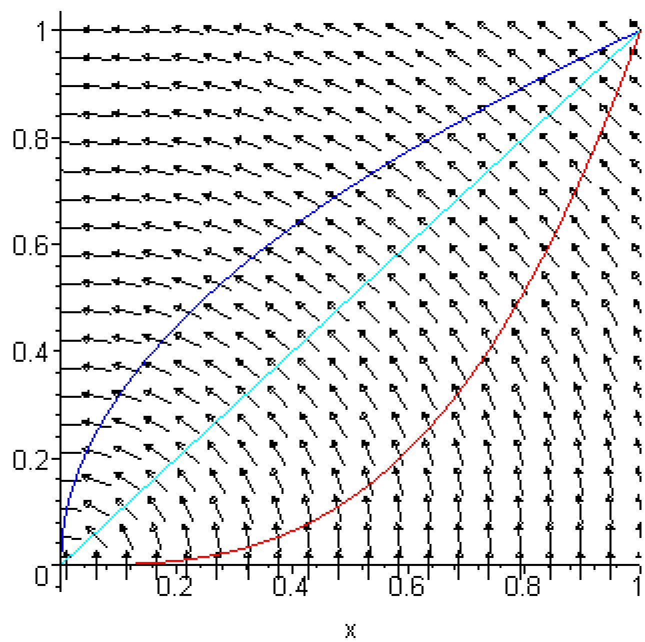
$$nx := \frac{y^2}{(x^2+y^2)^{\left(\frac{3}{2}\right)}}$$

$$-\frac{x^2}{(x^2+y^2)^{\left(\frac{3}{2}\right)}} = \frac{y^2}{(x^2+y^2)^{\left(\frac{3}{2}\right)}}$$

You can see that for arbitrary values of *x* and *y*, this last equation will not be true. Therefore, by our definition, the force is not conservative.

# Defining and Visualizing the Force Field

```
> pv:=fieldplot(force, x=0.01..1, y=0..1,arrows=SLIM):
  pc:=plot({x,sqrt(x), x^3},x=0..1, color=[red,cyan,blue]):
  print(display({pc,pv}));
```




# Writing Parametrizations and Computing Work

# Integrals

As you have learned from hand computation, line integrals can be difficult to evaluate. Even *Maple* has problems with some line integrals. When you experiment with your own force or path functions, sometimes you may have to use the **integration type=numerical** option, and sometimes you may have to use symbolic integration (default). You will also get error messages at times, warning you about problems associated with the convergence of the integration technique. In these cases, you will frequently, but not always, be given an answer that is reasonably accurate. In this example, we have a problem at $t = 0$, because the force function has a 0 denominator for the parametrizations given.

Find the work done in traveling along the straight line $y = x$. Choose an appropriate parametrization. We write the position and velocity vector to assist in computing the work integral. Note that by setting $x$ and $y$ equal to particular functions of $t$, the force function will reflect that parametrization when it appears in the line integral.

```
> x:=t:
  y:=t:
  r1:=[x,y]:
  v1:=diff(r1,t):
  print(`velocity = `, v1);
  print(`force function along curve = `,force);
  w1:=int(linalg[dotprod](force,v1), t=0..1):
  print(`work done along path = `, evalf(w1));
```

$$velocity = , [1, 1]$$

$$force\ function\ along\ curve = , \left[-\frac{1}{2}\frac{t\sqrt{2}}{\sqrt{t^2}}, \frac{1}{2}\frac{t\sqrt{2}}{\sqrt{t^2}}\right]$$

$$work\ done\ along\ path = , 0.$$

Could you have predicted this answer by looking at the graph?

Now find the work done in traveling along the lower curve $y = x^3$ . Choose an appropriate parametrization.

```
> x:=t:
  y:=t^3:
  r2:=[x,y]:
  v2:=diff(r2,t):
  print(`velocity = `, v2);
  print(`force function along curve = `,force);
  w2:=int(linalg[dotprod](force,v2), t=0..1):
  print(`work done along path = `, evalf(w2));
```

$$velocity = , [1, 3t^2]$$

$$\textit{force function along curve} = , \left[ -\frac{t^3}{\sqrt{t^2+t^6}}, \frac{t}{\sqrt{t^2+t^6}} \right]$$

$$\textit{work done along path} = , 0.5670004769 - 0.1676883352\ 10^{-9}\ I$$

You can ignore the imaginary portion of the answer; it is near 0 and due to round off error. The result is very different from our previous value. Now find the work done in traveling along the upper curve $y = \sqrt{x}$ . As before, choose an appropriate parametrization.

```
> x:=t:
  y:=sqrt(t):
  r3:=[x,y]:
  v3:=diff(r3,t):
  print(`velocity = `, v3);
  print(`force function along curve = `,force);
  w3:=int(linalg[dotprod](force,v3), t=0..1):
  print(`work done along path = `, evalf(w3));
```

$$\textit{velocity} = , \left[ 1, \frac{1}{2\sqrt{t}} \right]$$

$$\textit{force function along curve} = , \left[ -\frac{\sqrt{t}}{\sqrt{t^2+t}}, \frac{t}{\sqrt{t^2+t}} \right]$$

$$\textit{work done along path} = , -0.414213562$$

What does it mean when our work done went from 0 on the first path to a positive number on the second path and now to a negative value? Could you have predicted that from the graph showing the force field? This example demonstrates how for non-conservative forces, the work done in getting from one point to another is not independent of the path taken.

# You Try It: Part I

Try a different path in going from (0, 0) to (1, 1). Suppose you go from (0, 0) to (1, 0) and then to (1, 1), all along parallel and perpendicular lines. The following commands specify appropriate parameterizations. You can execute them for any two-dimensional force. Begin by entering any force you wish by replacing the terms in **newforce**.

```
> unassign('x','y','t');
  newforce:=[cos(5*x),-3*x*y]:
```

First, go from (0,0) to (1,0).

```
> x:=t:
  y:=0:
  r1:=[x,y]:
  dr1:=diff(r1,t):
  w4a:=int(linalg[dotprod](newforce,dr1), t=0..1):
  print(`work done along path =`, evalf(w4a));
```

*work done along path* =, -0.1917848549

Next, go from (1,0) to (1,1).

```
> x:=1:
  y:=t:
  r2:=[x,y]:
  dr2:=diff(r2,t):
  w4b:=int(linalg[dotprod](newforce,dr2), t=0..1):
  print(`work done along path =`, evalf(w4b));
```

*work done along path* =, -1.500000000

Add your results, and contrast them to what you would get with the paths used earlier. Do this for both conservative and non-conservative forces.

```
> w4:=evalf(w4a+w4b);
```

$w4 := -1.691784855$

Compute the work done along the line $y = x$.

```
> x:=t:
  y:=t:
  r1:=[x,y]:
  v1:=diff(r1,t):
  print(`velocity = `, v1);
  print(`force function along curve = `,newforce);
  w1:=int(linalg[dotprod](newforce,v1), t=0..1):
  print(`work done along path = `, evalf(w1));
```

*velocity* = , [1, 1]

*force function along curve* = , $[\cos(5\,t), -3\,t^2]$

*work done along path* = , -1.191784855

Compute the work done along the curve $y = x^3$ .

```
>
```

```
x:=t:
y:=t^3:
r2:=[x,y]:
v2:=diff(r2,t):
print(`velocity = `, v2);
print(`force function along curve = `,newforce);
w2:=int(linalg[dotprod](newforce,v2), t=0..1):
print(`work done along path = `, evalf(w2));
```

$$velocity = , [1, 3t^2]$$

$$force\ function\ along\ curve = , [\cos(5t), -3t^4]$$

$$work\ done\ along\ path = , -1.477499141$$

Compute the work done along the curve $y = \sqrt{x}$ .

```
> x:=t:
y:=sqrt(t):
r3:=[x,y]:
v3:=diff(r3,t):
print(`velocity = `, v3);
print(`force function along curve = `,newforce);
w3:=int(linalg[dotprod](newforce,v3), t=0..1):
print(`work done along path = `, evalf(w3));
```

$$velocity = , \left[1, \frac{1}{2\sqrt{t}}\right]$$

$$force\ function\ along\ curve = , \left[\cos(5t), -3t^{\left(\frac{3}{2}\right)}\right]$$

$$work\ done\ along\ path = , -0.9417848549$$

Is the work done in going from (0, 0) to (1, 1) the same for all the paths?

```
> evalf([w1,w2,w3,w4]);
```

$$[-1.191784855, -1.477499141, -0.9417848549, -1.691784855]$$

```
> if (evalf(w1)=evalf(w2) and evalf(w3) =evalf(w4)) then print(`true`) else print(`false`) fi;
```

$$false$$

# Part II: Example in Three Dimensions

**Section 16.3, Exercise 30**

## Conservative Force

Given the force with components: { $e^{yz}$ , $x z e^{yz} + z\cos(y)$ , $x y e^{yz} + \sin(y)$ }, find

the work done in going from (1, 0, 1) to (1, $\frac{\pi}{2}$ , 0) by traveling along three different

paths. The parameterizations for each of these paths are detailed below and the work is computed for each. The results from the three paths are then compared and visualized.

```
> unassign('x','y','z'):
  force:=[exp(y*z), x*z*exp(y*z)+z*cos(y), x*y*exp(y*z)+sin(y)];
```

$$force := [e^{(y z)}, x z e^{(y z)} + z\cos(y), x y e^{(y z)} + \sin(y)]$$

## Verifying that the Force is Conservative

By computing and comparing the appropriate first partial derivatives, verify that the force defined is conservative.

```
> my:=diff(force[1],y);
  nx:=diff(force[2],x);
  mz:=diff(force[1],z);
  px:=diff(force[3],x);
  nz:=diff(force[2],z);
  py:=diff(force[3],y);
  if (my=nx) then print(`true`) else print(`false`) fi;
  if (mz=px) then print(`true`) else print(`false`) fi;
  if (nz=py) then print(`true`) else print(`false`) fi;
```

$$my := z e^{(y z)}$$

$$nx := z e^{(y z)}$$

$$mz := y e^{(y z)}$$

$$px := y e^{(y z)}$$

$$nz := x\,\mathbf{e}^{(y\,z)} + x\,z\,y\,\mathbf{e}^{(y\,z)} + \cos(y)$$

$$py := x\,\mathbf{e}^{(y\,z)} + x\,z\,y\,\mathbf{e}^{(y\,z)} + \cos(y)$$

*true*

*true*

*true*

# Writing the Parameterization and Computing the Work Integral for 16.3, Exercise 30a

Find the work done in traveling along the straight line from (1, 0, 1) to (1, $\frac{\pi}{2}$ , 0). Choose an appropriate parameterization. Write the position and velocity vector to assist in computing the work integral. Set *x*, *y*, and *z* equal to particular functions of *t*, the force function will reflect that parametrization when it appears in the line integral.

```
> x:=1:
  y:=Pi*t/2:
  z:=1-t:
  r1:=[x,y,z]:
  v1:=diff(r1,t):
  print(`velocity`, v1);
  print(`force function along curve =`, force);
  w1:=int(linalg[dotprod](force, v1), t=0..1):
  print(`work done along path = `,evalf(w1));
```

$$velocity, \left[0, \frac{\pi}{2}, -1\right]$$

*force function along curve =,*

$$\left[\mathbf{e}^{\left(\frac{\pi\,t\,(1-t)}{2}\right)}, (1-t)\,\mathbf{e}^{\left(\frac{\pi\,t\,(1-t)}{2}\right)} + (1-t)\cos\left(\frac{\pi\,t}{2}\right), \frac{1}{2}\pi\,t\,\mathbf{e}^{\left(\frac{\pi\,t\,(1-t)}{2}\right)} + \sin\left(\frac{\pi\,t}{2}\right.\right.$$

$$work\ done\ along\ path =, 0.$$

# Writing Parameterizations and Computing Work Integrals for 16.3, Exercise 30b

Find the work done in the two straight line paths. Choose appropriate parametrizations.

First, go from (1, 0, 1) to the origin.

```
> x:=1-t:
  y:=0:
  z:=1-t:
  r21:=[x,y,z]:
  v2:=diff(r21,t):
  print(`velocity`, v2);
  print(`force function along curve =`, force);
  w21:=int(linalg[dotprod](force, v2), t=0..1):
  print(`work done along path = `,evalf(w21));
```

$$velocity, [-1, 0, -1]$$

$$force\ function\ along\ curve =, [1, (1-t)^2+1-t, 0]$$

$$work\ done\ along\ path =, -1.$$

Next, go from the origin to (1, $\frac{\pi}{2}$ , 0).

```
> x:=t:
  y:=Pi*t/2:
  z:=0:
  r22:=[x,y,z]:
  v2:=diff(r22,t):
  print(`velocity`, v2);
  print(`force function along curve =`, force);
  w22:=int(linalg[dotprod](force, v2), t=0..1):
  print(`work done along path = `,evalf(w22));
```

$$velocity, \left[1, \frac{\pi}{2}, 0\right]$$

$$force\ function\ along\ curve =, \left[1, 0, \frac{t^2\pi}{2}+\sin\left(\frac{\pi t}{2}\right)\right]$$

$$work\ done\ along\ path =, 1.$$

So the total work done is as follows.

```
> w2:=w21+w22;
```

$$w2 := 0$$

# Writing Parameterizations and Computing Work Integrals for 16.3, Exercise 30c

Find the work done for the two straight line paths plus the parabolic path. Choose appropriate parametrizations. First, we will go from (1, 0, 1) to (1, 0, 0).

```
> x:=1:
  y:=0:
  z:=1-t:
  r31:=[x,y,z]:
  v3:=diff(r32,t):
  print(`velocity`, v3);
  print(`force function along curve =`, force);
  w31:=int(linalg[dotprod](force, v3), t=0..1):
  print(`work done along path = `,evalf(w31));
```

$$velocity, 0$$

$$force\ function\ along\ curve\ =, [1, 2-2t, 0]$$

$$work\ done\ along\ path\ =\ , 0.$$

Now follow the *x* axis in going from (1,0,0) to the origin.

```
> x:=1-t:
  y:=0:
  z:=0:
  r32:=[x,y,z]:
  v3:=diff(r32,t):
  print(`velocity`, v3);
  print(`force function along curve =`, force);
  w32:=int(linalg[dotprod](force, v3), t=0..1):
  print(`work done along path = `,evalf(w32));
```

$$velocity, [-1, 0, 0]$$

$$force\ function\ along\ curve\ =, [1, 0, 0]$$

$$work\ done\ along\ path\ =\ , -1.$$

Now follow a parabola from the origin to $(1, \frac{\pi}{2}, 0)$.

```
> x:=t:
  y:=Pi*t^2/2:
  z:=0:
```

```
r33:=[x,y,z]:
v3:=diff(r33,t):
print(`velocity`, v3);
print(`force function along curve =`, force);
w33:=int(linalg[dotprod](force, v3), t=0..1):
print(`work done along path = `,evalf(w33));
```

$$velocity, [1, \pi t, 0]$$

$$force\ function\ along\ curve\ =, \left[1, 0, \frac{t^3 \pi}{2} + \sin\left(\frac{t^2 \pi}{2}\right)\right]$$

$$work\ done\ along\ path\ = , 1.$$

So the total work done is as follows.

```
> w3:=w31+w32+w33;
```

$$w3 := 0$$

Was the work done along each path the same?

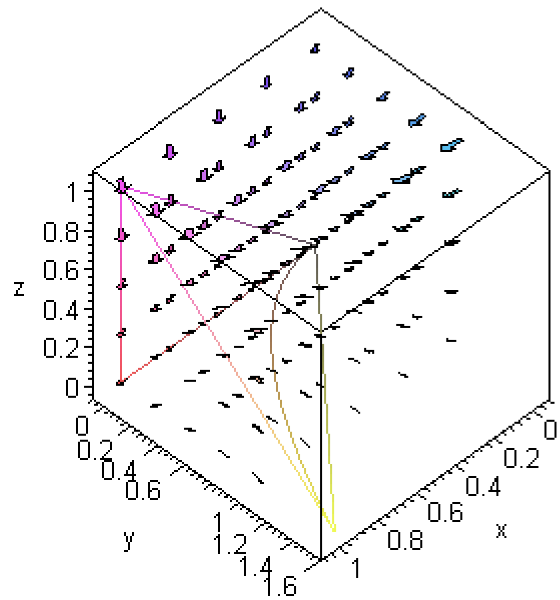
```
> if (w1=w2 and w2=w3) then print(`true`) else print(`false`) fi;
```

$$true$$

# Visualizing the Force Field and Different Paths

Once the graph is displayed, view it from different angles by clicking and dragging the plot.

```
> pp1:=spacecurve(r1, t=0..1):
pp21:=spacecurve(r21, t=0..1):
pp22:=spacecurve(r22, t=0..1):
pp31:=spacecurve(r31, t=0..1):
pp32:=spacecurve(r32, t=0..1):
pp33:=spacecurve(r33, t=0..1):
unassign('x','y','z');
forceplot:=fieldplot3d(force,x=0..1, y=0..1,z=0..1, axes=BOXED,arrows=THICK, gri
print(display({pp1,pp21,pp22,pp31,pp32,pp33, forceplot}));
```




# You Try It: Part II

Redefine your force in the previous example, and re-execute all the cells. Only the terms in the force function need to be changed. This time try a non-conservative force. You could begin by checking out the function given; we have followed the same paths as in Part II. Be careful to use correct terminolgy.

```
> unassign('x','y','z');
  force:=[x^2*cos(y), exp(y*z)*(1-x), x*y*sin(z)];
```

$$force := [x^2 \cos(y), e^{(y z)} (1-x), x y \sin(z)]$$

## Is the Force Conservative or Not?

```
> my:=diff(force[1],y):
  nx:=diff(force[2],x):
  mz:=diff(force[1],z):
  px:=diff(force[3],x):
  nz:=diff(force[2],z):
  py:=diff(force[3],y):
  my=nx;
  mz=px;
  nz=py;
```

$$-x^2 \sin(y) = -e^{(y z)}$$

$$0 = y \sin(z)$$

$$y\,e^{(y z)}(1-x)=x\sin(z)$$

Are the corresponding partial derivatives equal to one another?

# Writing the Parameterization and Computing the Work Integral for 16.3, Exercise 30a

Find the work done in traveling along the straight line from (1, 0, 1) to (1, $\frac{\pi}{2}$, 0). Choose an appropriate parameterization. Write the position and velocity vector to assist in computing the work integral. When you set *x, y*, and *z* equal to particular functions of *t,* the force function will reflect that parametrization when it appears in the line integral.

```
> x:=1:
  y:=Pi*t/2:
  z:=1-t:
  r1:=[x,y,z]:
  v1:=diff(r1,t):
  print(`velocity`, v1);
  print(`force function along curve =`, force);
  w1:=int(linalg[dotprod](force, v1), t=0..1):
  print(`work done along path = `,evalf(w1));
```

$$velocity, \left[0, \frac{\pi}{2}, \text{-}1\right]$$

$$force\ function\ along\ curve =, \left[\cos\left(\frac{\pi t}{2}\right), 0, -\frac{1}{2}\pi t\sin(-1+t)\right]$$

$$work\ done\ along\ path =, \text{-}0.249016795$$

# Writing Parameterizations and Computing Work Integrals for 16.3, Exercise 30b

Now find the work done in the two straight line paths specified in part b). Choose appropriate parametrizations. First, we go from (1, 0, 1) to the origin.

```
> x:=1-t:
  y:=0:
  z:=1-t:
  r21:=[x,y,z]:
  v2:=diff(r21,t):
  print(`velocity`, v2);
  print(`force function along curve =`, force);
```

```
w21:=int(linalg[dotprod](force, v2), t=0..1):
print(`work done along path = `,evalf(w21));
```

$$velocity, [-1, 0, -1]$$

$$force\ function\ along\ curve =, [(1-t)^2, t, 0]$$

$$work\ done\ along\ path = , -0.3333333333$$

Next, go from the origin to (1, $\frac{\pi}{2}$ , 0).

```
> x:=t:
  y:=Pi*t/2:
  z:=0:
  r22:=[x,y,z]:
  v2:=diff(r22,t):
  print(`velocity`, v2);
  print(`force function along curve =`, force);
  w22:=int(linalg[dotprod](force, v2), t=0..1):
  print(`work done along path = `,evalf(w22));
```

$$velocity, \left[1, \frac{\pi}{2}, 0\right]$$

$$force\ function\ along\ curve =, \left[t^2 \cos\left(\frac{\pi t}{2}\right), 1-t, 0\right]$$

$$work\ done\ along\ path = , 0.9059933850$$

So the total work done is as follows.

```
> w2:=evalf(w21+w22);
```

$$w2 := 0.5726600517$$

# Writing Parameterizations and Computing Work Integrals for 16.3, Exercise 30c

Find the work done for the two straight line paths plus the parabolic path. Choose appropriate parametrizations. First, go from (1, 0, 1) to (1, 0, 0).

```
> x:=1:
  y:=0:
  z:=1-t:
```

```
r31:=[x,y,z]:
v3:=diff(r31,t):
print(`velocity`, v3);
print(`force function along curve =`, force);
w31:=int(linalg[dotprod](force, v3), t=0..1):
print(`work done along path = `,evalf(w31));
```

$$velocity, [0, 0, -1]$$

$$force\ function\ along\ curve =, [1, 0, 0]$$

$$work\ done\ along\ path = , 0.$$

Now follow the *x* axis in going from (1,0,0) to the origin.

```
> x:=1-t:
  y:=0:
  z:=0:
  r32:=[x,y,z]:
  v3:=diff(r32,t):
  print(`velocity`, v3);
  print(`force function along curve =`, force);
  w32:=int(linalg[dotprod](force, v3), t=0..1):
  print(`work done along path = `,evalf(w32));
```

$$velocity, [-1, 0, 0]$$

$$force\ function\ along\ curve =, [(1-t)^2, t, 0]$$

$$work\ done\ along\ path = , -0.3333333333$$

Now follow a parabola from the origin to (1, $\frac{\pi}{2}$ , 0,).

```
> x:=t:
  y:=Pi*t^2/2:
  z:=0:
  r33:=[x,y,z]:
  v3:=diff(r33,t):
  print(`velocity`, v3);
  print(`force function along curve =`, force);
  w33:=int(linalg[dotprod](force, v3), t=0..1):
  print(`work done along path = `,evalf(w33));
```

$$velocity, [1, \pi t, 0]$$

$$\textit{force function along curve} =, \left[t^2 \cos\left(\frac{t^2 \pi}{2}\right), 1 - t, 0\right]$$

$$\textit{work done along path} = , 0.7024064426$$

So the total work done is as follows:

```
> w3:=evalf(w31+w32+w33);
```

$$w3 := 0.3690731093$$

Was the work done along each path the same?

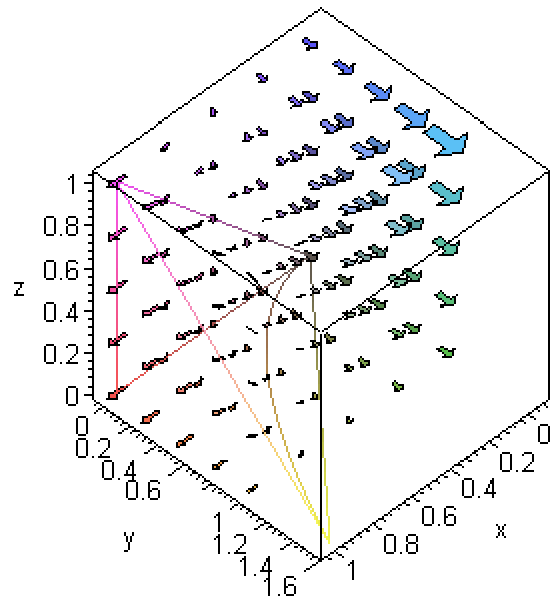
```
> if (w1=w2 and w2=w3) then print(`true`) else print(`false`) fi;
```

*false*

# Visualizing the Force Field and Different Paths

Once the graph is displayed, view it from different angles by clicking and dragging the plot.

```
> pp1:=spacecurve(r1, t=0..1):
  pp21:=spacecurve(r21, t=0..1):
  pp22:=spacecurve(r22, t=0..1):
  pp31:=spacecurve(r31, t=0..1):
  pp32:=spacecurve(r32, t=0..1):
  pp33:=spacecurve(r33, t=0..1):
  unassign('x','y','z');
  forceplot:=fieldplot3d(force,x=0..1, y=0..1,z=0..1, axes=BOXED,arrows=THICK, gri
  print(display({pp1,pp21,pp22,pp31,pp32,pp33, forceplot}));
```


1
0.8
0.6
z
0.4
0.2
0
0
0.2
0.4
0.6
y
1
1.2
1.4
1.6
0
0.2
0.4
0.6
x
0.8
1

>