Philosophy: The philosophy is this worksheet is to first **solidify the students' understanding of the structure of a parameterization of a line and the vector equation of a plane**. For this reason, "unenlightening" names have been used for various quantities in the code -- so as not to give the students a big hint. We welcome comments on whether the students would be better served to have such a hint and how to strike a balance (as well as any other comments, of course). Please send comments to symington_mf(at)mercer.edu.

Lines and planes

by Curtis Herink, Jawahar Pathak, and Margaret Symington

Lines

The most convenient way to define a line in 3-space is parametrically -- with a nice parameterization! Think of starting at a point and then moving along a line. You can move forward and backward from that point in the direction of a vector.

This leads to the parametrization **<A,B,C> + t* <a,b,c>** ,or **v1 + t *v2** with **v1 = <A,B,C>** and **v2= <a,b,c>**.

Other ways of expressing this parameterization are:

< A+a*t , B+b*t , C+c*t > or
x = A+a*t, y = B+b*t, z = C+c*t .

Run the following code with a variety of choices for the values of the constants A,B,C and a,b,c until you understand what they mean.

```
> restart;
> with(plots); with(Student[VectorCalculus])
```

[*Interactive, animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot, densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, replot, rootlocus, semilogplot, setoptions, setoptions3d, spacecurve, sparsematrixplot, sphereplot, surfdata, textplot, textplot3d, tubeplot*]

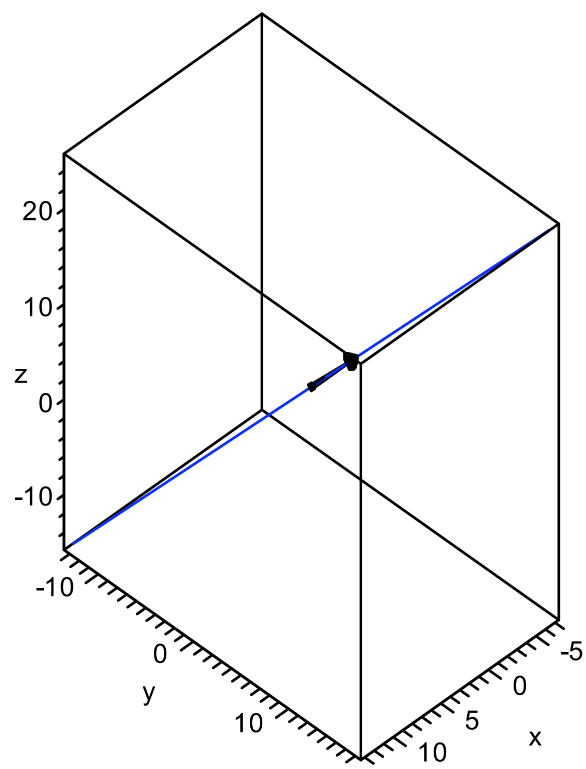
```
Warning, inserted missing semicolon at end of statement
```

[&x, *, +, -, ., <,>, <|>, *BasisFormat, Binormal, CrossProduct, Curl, Curvature*, D, *Del*, (1) *Divergence, DotProduct, FlowLine, Flux, GetCoordinates, Gradient, Laplacian, LineInt, MapToBasis, Nabla, Norm, Normalize, PathInt, PrincipalNormal, RadiusOfCurvature, ScalarPotential, SetCoordinates, SpaceCurve, SpaceCurveTutor, SurfaceInt, TNBFrame, TangentVector, Torsion, Vector, VectorField, VectorFieldTutor, VectorPotential, diff, evalVF, int, limit, series*]

```
> v1 := < A, B, C > : v2 := < a, b, c > :
> A := 3 : B := 2 : C := 5 : a := -2 : b := 3 : c := 4 :
> line1 := spacecurve(v1+t*v2, t = -5 .. 5, color = blue)
Warning, inserted missing semicolon at end of statement
```

line1 := *PLOT3D*(...) (2)

```
> vect1 := arrow(v1,v2,width = 0.4, head_width=1.5, color = black):
> point1:=pointplot3d(v1,color=red,symbol=circle):
> display3d({vect1,line1,point1},axes=boxed,scaling=CONSTRAINED,
  labels=[x,y,z]);
```




```
>
```

1. Plot a line with tangent vector <6,-1,3> that passes through the point (4,4,-7). Show the tangent vector on your plot.

2. Graph a line through the points (4,-5,1) and (6,1,-2) by modifying the above code slightly.

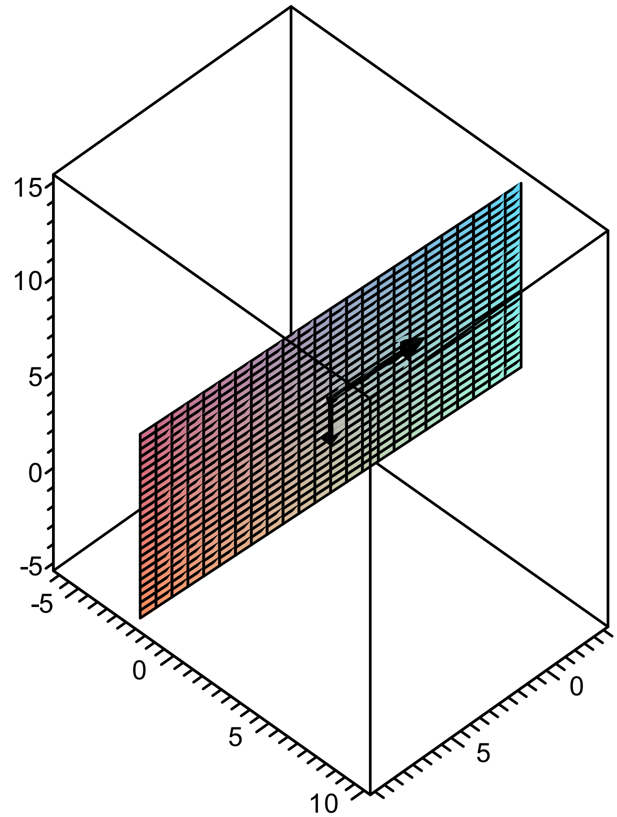
3. Describe, in your own words, the meaning of the vectors **v1=<A,B,C>** and **v2=<a,b,c>**.

Planes

There is are a few convenient ways to define a plane in 3-space. First, we can generalize our approach to lines. The simplest way to parameterize a plane is as:
<A,B,C> + s*<a,b,c> + t*<e,f,g>.

It is crucial to understand the meaning of the constants that appear in the above expression. Explore the effect of changing their values in the following code.

```
> v1 :=< A, B, C >: v2 :=< a, b, c >: v3 :=< e, f, g >:
> A := 3 : B := 2 : C := 5 : a := -2 : b := 3 : c := 4 : e := 1 : f := 1 : g := -1 :
> point1:=pointplot3d(v1,color=red,symbol=circle):
> param1:=v1+s*v2+t*v3:
> plane1:=plot3d(param1,s=-2..2,t=-2..2):
> vect1:=arrow(v1,v2,width = 0.25, head_width=0.8,color=black):
> vect2 := arrow(v1,v3,width = 0.25, head_width=0.8, color = black):
> display([vect1,vect2,plane1,point1],axes=boxed,scaling=
  CONSTRAINED);
```



4. Plot the graph of a plane with tangent vectors <-5,1,6> and <3,1,-1> that passes through the point (0, 4,-5). Show the vectors in your plot.

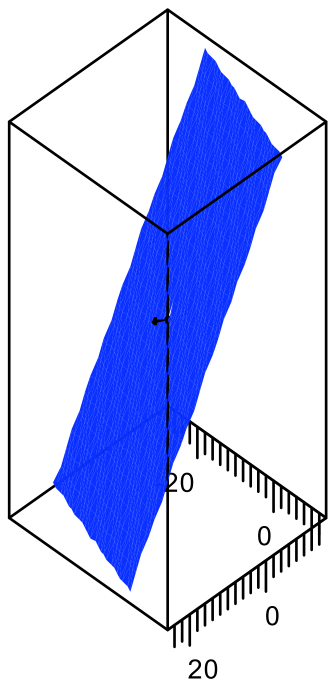
5. Plot the graph of a plane that passes through the points (0,1,2), (4,-1,1), and (-3,0,2). On your plot, show two vectors in the plane that share a base point AND show another vector IN the plane.

The simplest expression that defines a plane is
Ax + By + Cz = D,
but the easiest to work with when juggling lines and planes is
<n1,n2,n3> . (<x,y,z> - <p1,p2,p3>)=0.
This is the form we use in the following code. Here, **<n1,n2,n3>** is a vector normal to the plane and **(p1,p2,p3)** is a point in the plane.

6. Explain why any vector <**x,y,z**> that satisfies the vector equation
<n1,n2,n3> . (<x,y,z> - <p1,p2,p3>)=0
must be the position vector for a point in the plane with normal vector **<n1,n2,n3>** that passes through the point **(p1,p2,p3)**.

```
> v3:=<n1,n2,n3>: v4:=<p1,p2,p3>:
```

```
> n1:=3: n2:=-1: n3:=1: p1:=5: p2:=-7: p3:=2:
> v5:=<-n2,n1,0>: v6:=CrossProduct(v3,v5):
> plane2:=plot3d([v4[1]+s*v5[1]+t*v6[1],v4[2]+s*v5[2]+t*v6[2],v4[3]+
  s*v5[3]+t*v6[3]], s=-5..5,t=-5..5,
     transparency=.5, color=blue, style=patchnogrid):
> vect2:=arrow(v4,v3,width = 0.4, head_width=1.5, color = black):
> point2:=pointplot3d(v4,color=red,symbol=circle):
> display([plane2,vect2,point2],axes=boxed,scaling=CONSTRAINED);
```




```
>
```

7. Plot a plane with normal vector <1,-2,1> that passes through the point (3,-5,10). Show the normal vector on your plot.

8. Find an expression for the line that is the intersection of two planes, one with normal vector <4,-5, 1> and passing through the point (6,1,-2), and the other with normal vector <3,2,1> and passing through (1,-1,0). Make a plot that shows both planes and the line. (The line should be drawn in its own color to stand out.)