3D Vector Modeling

OBJECTIVES

In this course you will construct computer models to:

- Visualize motion in 3D using a programming environment called VPython, which is the widely-used Python programming language (python.org) plus a 3D graphics module, visual
- Visualize vector quantities like position, momentum, and force in 3D
- Do calculations based on fundamental principles to predict the motion of interacting objects
- Animate the predicted motions in 3D

In this lab you will apply what you learned in the introductory exercise to:

- Create and visualize several given 3D spherical objects
- Connect them in a logical way using the concept of relative position vectors

OVERVIEW OF A COMPUTER PROGRAM

- A computer program consists of a sequence of instructions.
- The computer carries out the instructions one by one, in the order in which they appear, and stops when it reaches the end.
- Each instruction must be entered exactly correctly (as if it were an instruction to your calculator).
- If the computer encounters an error in an instruction (such as a typing error), it will stop running and print a red error message.

A typical program has four sections:

- Setup statements
- Definitions of constants (if needed)
- Creation of objects and specification of initial conditions
- Calculations to predict motion or move objects (done repetitively in a loop)

1 Setup statements

A minimal working program has been provided for you on eLC. Download and save it to your computer. (By default, your Web browser will put the file in the Downloads folder.) Rename the file to something descriptive; some suggestions are "vector_model.py", or "vectorlab_*your names*.py". Remember to give it the extension ".py".

• **Open the VIDLE program editor by clicking on its icon in the Dock. From there, use "File:Open" to open the program you downloaded.** Do *not* simply double-click on the file in Finder; that will not open it correctly.

• **Add a descriptive comment header to the top of the file.**
This should contain your name(s) and a description of the program.

• **Run this "skeleton" program. Make sure you understand the purpose of each program statement.**

2 Adding Objects

Next, you'll be adding objects to your 3-D environment.

Create spheres at the following locations:

Name	Position
Alioth	$\langle -9.1, 58.0, -38.1 \rangle$
Alkaid	$\langle -61.8, 159.2, -122.2 \rangle$
Dubhe	$\langle 12.1, 92.5, -48.2 \rangle$
Megrez	$\langle -2.4, 54.5, -35.3 \rangle$
Merak	$\langle 11.1, 66.6, -42.9 \rangle$
Mizar	$\langle -18.1, 72.0, -47.2 \rangle$
Phecda	$\langle 1.4, 72.5, -53.3 \rangle$

Give each sphere a radius of 1, and give them whatever color(s) you like.

Run your program. Does this collection of points look familiar? Feel free to change the size and/or color of the spheres and re-run your program if you think it will help you to visualize them better.

3 Connect the Dots

Your next task is to "connect the dots" in a logical manner to complete your diagram. You can make "lines" in your diagram using thin cylinders, similar to the coordinate axes already drawn:

```
ThisToThat = cylinder(pos = ..., axis = ...)
```

Here `pos` is the position of one end of the cylinder, and `axis` is a vector that points from this end of the cylinder to the other. Adjust the thickness of your lines by using the `radius` attribute, and feel free to give your lines some `color` as well.

When you're satisfied with your program, compare your results to that of another group.

Zoom in and out, and rotate the display viewpoint. What do you notice about how these points look from different angles?

4 Travel Problems

Do the following calculations on a sheet of paper, and hand them in to your TA.

It will help to recall the position update formula, AKA definition of average velocity: $\vec{\mathbf{r}}_f = \vec{\mathbf{r}}_i + \vec{\mathbf{v}}_{\text{avg}}\,\Delta t$.

1. How long does it take light to travel from Alkaid to Dubhe? (The position vectors were given to you in units of light-years.)
2. A ship has been sent from Alkaid to Dubhe. It is scheduled to arrive 372 years after its launch. Calculate the ship's velocity vector *and* its speed.
3. What is the direction of the ship's travel? (i.e., what is the unit vector pointing in the direction of motion?)
4. Where will be the ship's coordinates after 100 years of travel? **Place an object at these coordinates.** Be sure to distinguish it from the other objects in some way. (A different color and/or size of `sphere`, for example; or you might want to create a `box` or `cone` for your ship.)