$3,0\rangle$ kg·m/s
$4,0\rangle$ kg·m/s
...e change in the ball's momen...
...acent locations. **(b)** On a copy of th...
...presenting each $\Delta\vec{p}$ you calculated in pa...
...ich two locations is the magnitude of the change
greatest?

...s the velocity of a 3 kg object when its momentum is ... $-30\rangle$ kg·m/s?

•**P58** A 1500 kg car located at $\langle 300,0,0\rangle$ m has a momentum of $\langle 45000,0,0\rangle$ kg·m/s. What is its location 10 s later?

••**P59** An ice hockey puck of mass 170 g enters the goal with a momentum of $\langle 0,0,-6.3\rangle$ kg·m/s, crossing the goal line at location $\langle 0,0,-26\rangle$ m relative to an origin in the center of the rink. The puck had been hit by a player 0.4 s before reaching the goal. What was the location of the puck when it was hit by the player, assuming negligible friction between the puck and the ice? (Note that the ice surface lies in the xz plane.)

••**P60** A space probe of mass 400 kg drifts past location $\langle 0,3\times 10^4,-6\times 10^4\rangle$ m with momentum $\langle 6\times 10^3,0,-3.6\times 10^5\rangle$ kg·m/s. Assuming the momentum of the probe does not change, what will be its position 2 minutes later?

Section 1.10

•**P61** A proton in an accelerator attains a speed of $0.88c$. What is the magnitude of the momentum of the proton?

•**P62** An electron with a speed of $0.95c$ is emitted by a supernova, where c is the speed of light. What is the magnitude of the momentum of this electron?

...A "cosmic-ray" proton hits the upper atmosphere with ...$0.9999c$, where c is the speed of light. What is the ...he momentum of this proton? Note that $|\vec{v}|/c =$ ... ...tually need to calculate the speed $|\vec{v}|$.

...ticle accelerator is traveling at a speed ... ...cles are given on the inside back cover of the ... ...you use the approximate nonrelativistic equation for the ...gnitude of momentum of the proton, what answer do you get? **(b)** What is the magnitude of the correct relativistic momentum of the proton? **(c)** The approximate value (the answer to part a) is significantly too low. What is the ratio of magnitudes you calculated (correct/approximate)?

•**P65** When the speed of a particle is close to the speed of light, the factor γ, the ratio of the correct relativistic momentum $\gamma m\vec{v}$ to the approximate nonrelativistic momentum $m\vec{v}$, is quite large. Such speeds are attained in particle accelerators, and at these speeds the approximate nonrelativistic equation for momentum is a very poor approximation. Calculate γ for the case where $|\vec{v}|/c = 0.9996$.

•**P66** An electron travels at speed $|\vec{v}| = 0.996c$, where $c = 3\times 10^8$ m/s is the speed of light. The electron travels in the direction given by the unit vector $\hat{v} = \langle 0.655,-0.492,-0.573\rangle$. The mass of an electron is 9×10^{-31} kg. **(a)** What is the value of γ? You can simplify the calculation if you notice that $(|\vec{v}|/c)^2 = (0.996)^2$. **(b)** What is the speed of the electron? **(c)** What is the magnitude of the electron's momentum? **(d)** What is the vector momentum of the electron? Remember that any vector can be "factored" into its magnitude times its unit vector, so that $\vec{v} = |\vec{v}|\hat{v}$.

•**P67** If $|\vec{p}|/m$ is $0.85c$, what is $|\vec{v}|$ in terms of c?

COMPUTATIONAL PROBLEMS

These problems are intended to introduce you to using a computer to model matter, interactions, and motion in 3D. You do not need to know how to program; you will learn what you need to know by doing these problems. In later chapters you will build on these small calculations to build models of physical systems.

To install the free 3D programming environment VPython, go to http://vpython.org and (carefully) follow the instructions for your operating system (Windows, MacOS, or Linux). Note the instructions given there on how to zoom and rotate the "camera" when viewing a 3D scene you have created.

More detailed and extended versions of some of these computational modeling problems may be found in the lab activities included in the *Matter & Interactions 4th Edition* resources for instructors.

•**P68** Watch the first introductory VPython video, *VPython Instructional Videos 1: 3D Objects*, at vpython.org/video01.html and complete the challenge activity at the end of the video.

•**P69** **(a)** Write a VPython program that creates eight spheres, each placed at one corner of a cube centered on the origin. The length of a side of the cube should be 6 units, and the radius of each sphere should be 0.5. Use at least two different colors for the spheres. **(b)** Add to the program an arrow whose tail is at one corner of the cube and whose tip is at the corner diagonally opposite. Figure 1.62 shows the display from one possible solution to this problem.

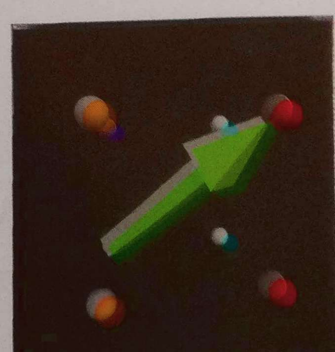

Figure 1.62

•**P70** Write a VPython program that represents the x, y, and z axes by three cylinders of different colors. The display from one possible solution is shown in Figure 1.63.

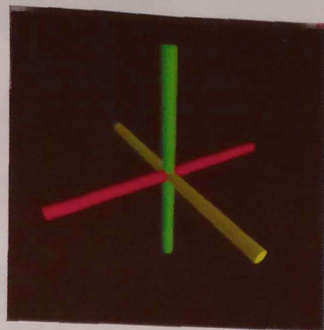

Figure 1.63

•**P71** Write a VPython program that represents the x, y, and z axes by three boxes (rectangular solids) of different colors. The display from one possible solution is shown in Figure 1.64.

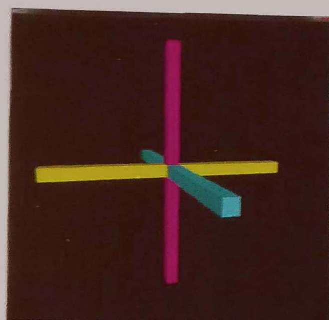

Figure 1.64

•**P72** Watch the second introductory VPython video, *VPython Instructional Videos 2: Variable Assignment*, at vpython.org/video02.html and complete the challenge activity at the end of the video.

•**P73** Based on Problem P72, write a program that shows an arrow pointing from one small box to another in such a way that when you change only the position of the first box, making no other changes, the arrow and the other box move too, so that the two boxes remain linked by the arrow.

•**P74** Watch the third introductory VPython video, *VPython Instructional Videos 3: Beginning Loops*, at vpython.org/video03.html and complete the challenge activity at the end of the video.

••**P75** **(a)** Write a VPython program that uses three while loops to create a display in which each of the axes (x, y, z) is represented by a linear array of boxes, with spaces between the boxes. Figure 1.65 shows a possible example. **(b)** (Optional) Rewrite your program to produce the same display using only one while loop.

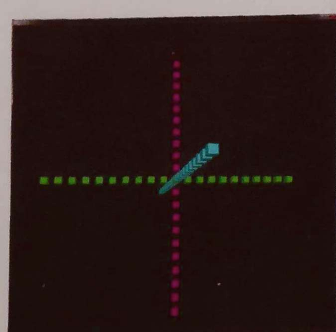

Figure 1.65

••**P76** Write a VPython program that creates three circles of spheres: one in the xy plane, one in the yz plane, and one in the xz plane. Each ring should be centered on the origin. Use either three while loops or one while loop. A possible solution is shown in Figure 1.66.

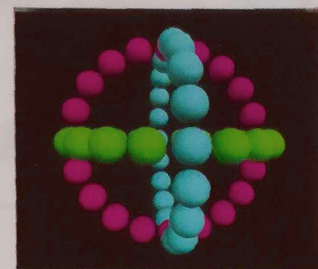

Figure 1.66

•**P77** Watch the fourth introductory VPython video, *VPython Instructional Videos 4: Loops and Animation*, at vpython.org/video04.html and complete the challenge activity at the end of the video.

••**P78** Consider the VPython program shown below. **(a)** An important skill is being able to read and understand an existing program, in order to be able to make useful modifications. *Before running the program*, study the program carefully line by line, then answer the following questions: (1) What is the initial velocity of the particle? (2) Is the particle initially located in front of the box or behind it? (3) In which line of code is the position of the particle updated? (4) What is the value of the time step Δt? (5) Will the particle bounce off of the red box, or travel through it? **(b)** Now run the program, and see if your answers were correct. **(c)** Modify the program to start the particle from an initial position on the $+x$ axis, to the right of and in front of the red box. Give the particle a velocity that will make it travel to the left, along the x axis, passing in front of the box.

```
from visual import *
box(pos=vector(0,0,-1),
    size=(5,5,0.5),
    color=color.red,
    opacity = 0.4)
particle = sphere(pos=vector(-5,0,-5),
              radius=0.3,
              color=color.cyan,
              make_trail = True)
v = vector(0.5,0,0.5)
delta_t = 0.05
t = 0
while t < 20:
    rate(100)
    particle.pos = particle.pos + v * delta_t
    t = t + delta_t
```

•••**P79** Modify the program shown above to make the particle bounce off the red box instead of passing through it. On the Help menu available in IDLE (the VPython editor), choose "Python docs" or search the web for "Python if" to find out how to use an `if` statement. You may also find it helpful to look at the example program `bounce2.py`, included in the examples installed with VPython.