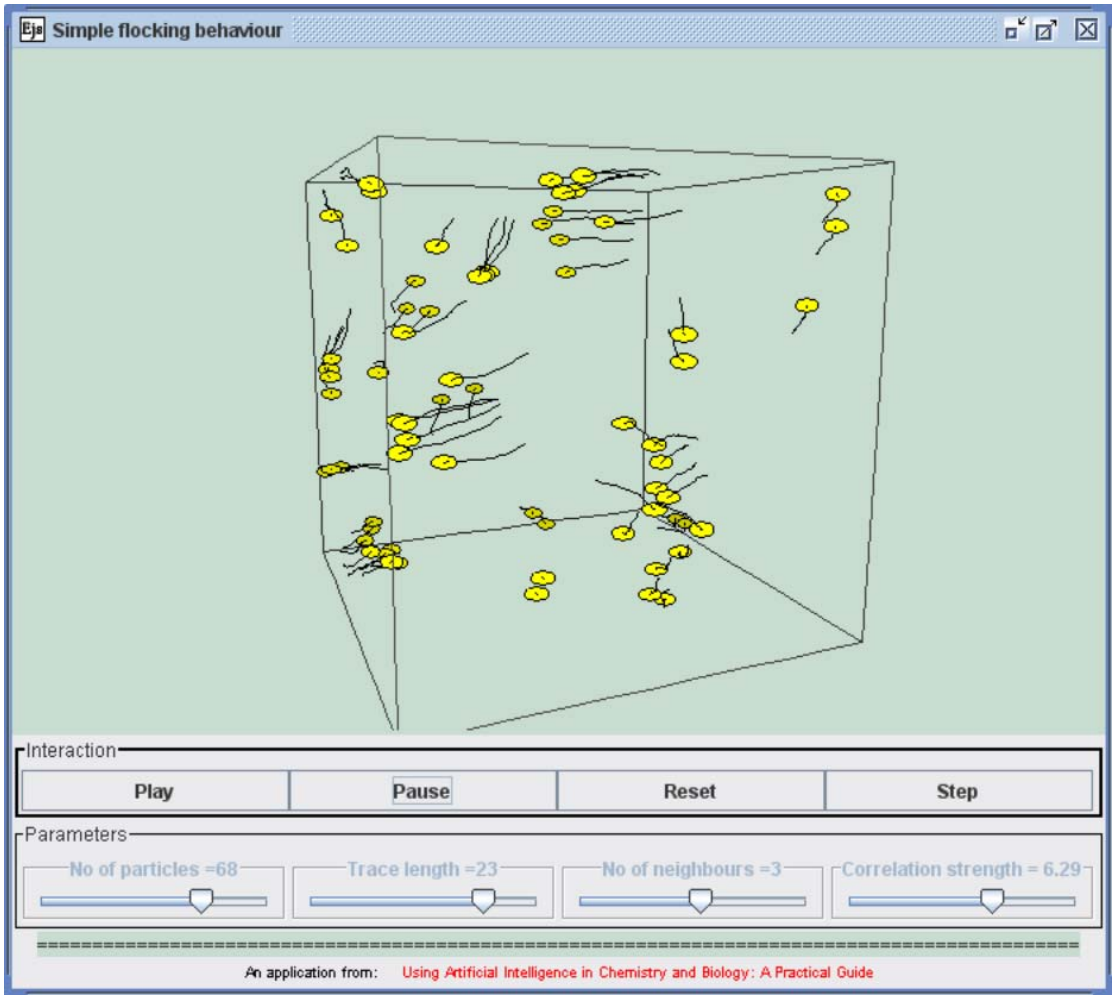
Simple flocking program

The flocking program is particularly simple, emulating the behaviour of a group of particles that show crude flocking behaviour.



A number of particles are placed in the simulation box with randomly-selected positions and velocities:

```
for (i=0; i<96; i++)
{
    x[i]=100.0*Math.random();
    y[i]=100.0*Math.random();
    z[i]=100.0*Math.random();

    vx[i]=(0.5-Math.random());
    vy[i]=(0.5-Math.random());
    vz[i]=(0.5-Math.random());
}
```

The distance between each pair of particles is found:

```
for (i=0; i<np-1; i++)
{
    for (j=i; j<np; j++)
    {
        d2[i][j]=Math.pow(x[i]-
x[j],2.0)+Math.pow(y[i]-y[j],2)+Math.pow(z[i]-
z[j],2);
        d2[j][i]=d2[i][j];
    }
}
```

and the velocities of particles that are found to be neighbours are adjusted so that they become slightly more alike.

```
for (k=0; k<nclose; k++)
    {
        j=neighbour[i][k];
        newvx[i]=newvx[i]+correlation*vx[j]/100.0;
        newvy[i]=newvy[i]+correlation*vy[j]/100.0;
        newvz[i]=newvz[i]+correlation*vz[j]/100.0;
    }
```

This results in the movement of nearby particles becoming loosely or more strongly alike, depending upon the value of `correlation`.

BUTTONS

Button	Function
Play	Restart the program if it has previously been paused.
Pause	Temporarily halt execution.
Reset	Restart the program from scratch; all parameters are reset to their default values.
Step	Execute a single cycle of the program.

SLIDERS

Slider	Default	Comment
No. of particles	2	Number of particles to be displayed.
Trace length	12	The length of the "tail" that each particle leaves as it moves. Note (a) a trace length of zero is not permitted by the program, but this may easily be altered by changing the lower limit for this slider in the View section of the program. A trace length of zero gives a trace which grows without limit, which slows the program; (b) there appears to be a bug in the method Ejs uses to draw the traces, since a change in trace length is sometimes ignored until the number of particles is changed.
Number of neighbors	0	The number of particles to be considered as neighbors to any particle. There is no limit to the distance at which neighbors may be found. In the program as written, the strength of interaction does not depend upon the distance, so the degree of interaction is the same, whether the particles are close or far apart (see exercise 2 below).
Correlation strength	4	This determines the strength of the interaction between a particle and its neighbors.

Investigations and Exercises

1. Default parameters

Run the program and add a few particles using the appropriate slider. Check that the movement of all particles is completely uncorrelated.

Now try a non-zero number of neighbors and note the immediate formation of small groups of particles whose behaviour is partly correlated.

(The code

```
newvx[i]=vx[i]*(1.0-lowerv)+noise*(Math.random()-0.5);
newvy[i]=vy[i]*(1.0-lowerv)+noise*(Math.random()-0.5);
newvz[i]=vz[i]*(1.0-lowerv)+noise*(Math.random()-0.5);
```

introduces a little random noise to ensure the particles do not align their velocities perfectly.)

2. Distance-dependent interactions

Add a piece of code to make the interactions distance-dependent (dependent upon the inverse of the inter-particle distance for example, but beware of overflow in the calculations, as there is nothing in the current code to prevent two particles arriving at the same point and therefore being no distance apart!); check that your code gives the sort of behaviour you would expect.

3. Slightly more complicated flocking

The book refers to a marginally more complicated (but still simple) flocking algorithm, in which particles not only align their velocities, but also take note of the direction in which they are moving and try to avoid crashing into their neighbors. Write a program to incorporate this; assess qualitatively whether the behaviour of the particles is substantially different from that shown by the present model.