writing and running a program, to do a quick estimate of how long you expect your calculation to take. If it's going to take a year then it's not worth it: you need to find a faster way to do the calculation, or settle for a quicker but less accurate answer. The simplest way to estimate running time is to make a rough count of the number of mathematical operations the calculation will involve; if the number is significantly greater than a billion, you have a problem.

EXAMPLE 4.3: MATRIX MULTIPLICATION

Suppose we have two $N \times N$ matrices represented as arrays `A` and `B` on the computer and we want to multiply them together to calculate their matrix product. Here is a fragment of code to do the multiplication and place the result in a new array called `C`:

```
from numpy import zeros
N = 1000
C = zeros([N,N],float)

for i in range(N):
    for j in range(N):
        for k in range(N):
            C[i,j] += A[i,k]*B[k,j]
```

We could use this code, for example, as the basis for a user-defined function to multiply arrays together. (As we saw in Section 2.4.4, Python already provides the function "`dot`" for calculating matrix products, but it's a useful exercise to write our own code for the calculation. Among other things, it helps us understand how many operations are involved in calculating such a product.)

How large a pair of matrices could we multiply together in this way if the calculation is to take a reasonable amount of time? The program has three nested for loops in it. The innermost loop, which runs through values of the variable `k`, goes around N times doing one multiplication operation each time and one addition, for a total of $2N$ operations. That whole loop is itself executed N times, once for each value of `j` in the middle loop, giving $2N^2$ operations. And those $2N^2$ operations are themselves performed N times as we go through the values of `i` in the outermost loop. The end result is that the matrix multiplication takes $2N^3$ operations overall. Thus if $N = 1000$, as above, the whole calculation would involve two billion operations, which is feasible in a few minutes of running time. Larger values of N, however, will rapidly become intractable. For $N = 2000$, for instance, we would have 16 billion op-