Scientific Computing for DPhil Students I
Assignment 1 Solutions

1. Here's a Matlab session that answers the questions.

```
>> A = [1 1 3; 1 3 1; 0 1 3]
A =
     1     1     3
     1     3     1
     0     1     3

>> A^2
ans =
     2     7    13
     4    11     9
     1     6    10

>> inv(A)
ans =
    1.0000         0   -1.0000
   -0.3750    0.3750    0.2500
    0.1250   -0.1250    0.2500

>> det(A)
ans =  8

>> eig(A)
ans =
   1.2442 + 0.4745i
   1.2442 - 0.4745i
   4.5115 + 0.0000i

>> prod(ans)
ans =  8.0000
```

2. The Matlab code

```
expmA = expm(A)
k = 0; Ak = eye(size(A)); expA = Ak; termk = Ak;
while norm(termk) > 1e-20
  k = k+1;
  Ak = A*Ak;
  termk = Ak/factorial(k);
  expA = expA + termk;
end
expA, diff = expmA - expA
```

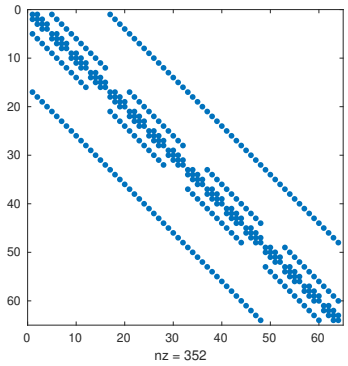
gives the results

```
expmA =
   11.3532   35.2345   49.1500
   14.0269   46.4762   49.3730
    7.0692   28.1653   39.4070
expA =
   11.3532   35.2345   49.1500
   14.0269   46.4762   49.3730
    7.0692   28.1653   39.4070
diff =
   1.0e-12 *
    0.0444    0.1705    0.1847
    0.0604    0.1990    0.2487
    0.0266    0.0995    0.1279
```

The computations differ by around 1e-13, which is 14–15 digits below the order of magnitude of the answers. This is not too bad; it is close to the machine precision of $2^{-52} \approx 2.22 \times 10^{-16}$.

3. The dimension of A is N^3, which gets big fast, but fortunately, A is sparse. For $N = 2$ the command `full(A(2))` gives this:

```
 6    -1    -1     0    -1     0     0     0
-1     6     0    -1     0    -1     0     0
-1     0     6    -1     0     0    -1     0
 0    -1    -1     6     0     0     0    -1
-1     0     0     0     6    -1    -1     0
 0    -1     0     0    -1     6     0    -1
 0     0    -1     0    -1     0     6    -1
 0     0     0    -1     0    -1    -1     6
```



Executing `spy(A(4))` gives this picture:

For timings, I ran this code:

```
D = @(N) sparse(toeplitz([2 -1 zeros(1,N-2)]));
I = @(N) speye(N);
A = @(N) kron(I(N),kron(I(N),D(N))) + kron(I(N),kron(D(N),I(N))) ...
        + kron(D(N),kron(I(N),I(N)));
b = @(N) ones(N^3,1);

tt = []; NN = [];
for N = 10:10:80
    NN = [NN; N];
    tic, A(N)\b(N); t = toc;
    fprintf('N = %d   t = %6.3f\n', N, t)
    tt = [tt; t];
end
loglog(NN, tt, '.-', 'markersize', 12)
grid on, hold on
loglog(NN, 1e-9*NN.^6, '--r'), hold off
text(40, 20, 'N^6', 'color', 'r', 'fontsize', 20)
xlabel('N')
ylabel('time (secs.)')
```

The table and plot suggest that the work scales as $O(N^6)$. For a dense matrix of dimensions $N^3 \times N^3$, the work would be $O(N^9)$, and the matrix would have N^6 entries, so it wouldn't fit in memory for values much bigger than $N = 10$.

```
N = 10   t =  0.009
N = 20   t =  0.028
N = 30   t =  0.159
N = 40   t =  0.846
N = 50   t =  2.850
N = 60   t =  9.113
N = 70   t = 23.630
N = 80   t = 48.886
```