Computational Science in MATLAB
COSC1001 (Normal) & COSC1901 (Advanced)

Exam Solutions and Marking Scheme

This exam assesses the entire course, with more emphasis for COSC 1001 placed on the early part of the course (basic MATLAB) with question 3, and more emphasis for COSC 1901 placed on the final two lectures with question 4.

1. Measuring acceleration due to gravity

This question assesses week 8 ("Review the application of linear algebra to physical problems", "Understand the built-in matrix manipulation functions" and "Apply matrix methods to problems in linear algebra...") , week 3 ("Write user-defined functions"), week 4 ("Understand the matrix datatype") and week 7 ("Calculate the accuracy of statistics drawn from limited populations")

(a) **3 marks**

\[
\begin{align*}
0.13^2 g + 0.13 b + c & = 0 \\
0.57^2 g + 0.57 b + c & = 2 \\
0.76^2 g + 0.76 b + c & = 4
\end{align*}
\]

\[
\begin{pmatrix}
0.13^2 & 0.13 & 1 \\
0.57^2 & 0.57 & 1 \\
0.76^2 & 0.76 & 1
\end{pmatrix}
\begin{pmatrix}
g \\
b \\
c
\end{pmatrix}
=
\begin{pmatrix}
0 \\
2 \\
4
\end{pmatrix}
\]

(b) **3 marks**

```
>> A=[0.13^2 0.13 1; 0.57^2 0.57 1; 0.76^2 0.76 1];
>> y = [0; 2; 4];
>> gbc = inv(A)*y
```

(c) **2 marks**

```matlab
function g = calcg(t)
    % This function computes the acceleration due to gravity, based on
    % an experiment that includes sensors placed at 0, 1 and 2 m
    A=[t(1)^2 t(1) 1; t(2)^2 t(2) 1; t(3)^2 t(3) 1];
    gbc = A\[0; 2; 4];
    g=gbc(1);
end
```

(d) **2 marks**

```matlab
display(['g = ' num2str(mean(g)) ' +/- ' num2str(std(g)/length(g))])
```

2. Random Numbers

This question assesses week 7 (all remaining points) and week 6 ("generate a normal distribution from uniform deviates").

(a) **2 marks**

(Directly from lecture notes, week 7) We determine the area under the curve and ensure that is equal to unity. Evaluation of the integral \( \int_0^\infty P(x) \, dx \) yields

\[
\int_0^\infty e^{-x/\lambda} \, dx = \left[ -\lambda e^{-x/\lambda} \right]_0^\infty = 0 + \lambda = \lambda
\]
and thus we divide by $\lambda$ (i.e. the area under the curve) to obtain the normalised expression for $P(x)$. Using this expression for $P(x)$, which takes the form

$$P(x) = \frac{1}{\lambda} e^{-x/\lambda}$$

(b) **2 marks**

(Directly from lecture notes, week 7) We now compute the area under the curve between 0 and some fixed value $t$. This area is, by construction, less than one, and is given by the expression

$$\int_0^t \frac{1}{\lambda} e^{-x/\lambda} \, dx = \left[ -e^{-x/\lambda} \right]_0^t = -e^{-t/\lambda} + 1$$

We can now use `rand` to generate a random area between 0 and 1, and then invert the equation above to solve for $t$.

$$\text{Area} = 1 - e^{-t/\lambda}$$

$$e^{-t/\lambda} = 1 - \text{Area}$$

$$-t/\lambda = \log(1 - \text{Area})$$

$$t = -\lambda \log(1 - \text{Area})$$

(c) **2 marks**

```matlab
lambda=1;
t=-lambda*(1-rand);
```

The histogram plot should look like a decaying exponential with several bins.

(d) **3 marks**

The best code is:

```matlab
function times=pulses(N)
times=sum(-log(1-rand(1000,N)));
```

Acceptable code (for 2 marks) is:

```matlab
function times=pulses(N)
for i=1:1000;
t=-log(1-rand(1,N));
times(i)=sum(t);
end
```

(e) **1 mark**

`pulses(1000)` should have a Normal distribution by the Central Limit Theorem.

3. **(COSC 1001) Basic MATLAB**

This question assesses most of weeks 1-4, with the exception of those parts assessed in question 1.

(a) **4 marks**
t=0:0.01:3;
x=10-5*t-t.^2;
y=4*sin(t);
d=sqrt(x.^2+y.^2);
mind=min(d);
t(d==mind)

Although this would be neater with e.g. `num2str` and `disp` commands, we will not give extra marks for this sophistication.

(b) i. 2 marks

4 5 6

ii. 2 marks

7

7

iii. 2 marks

2 5 9

4. (COSC 1901) Oscillations in a hanging rope/chain

This question principally assesses week 10 and week 9, although knowledge of material from weeks 1-4 is still essential.

(a) 2 marks

The eigenvalues and eigenvectors describe the periodic solutions to this problem, called the modes of the system. For each mode, the eigenvalue $k^2$ determines the period of the mode by equation 4 and the eigenvector determines the shape of the mode, which is the shape of the rope at the point in its oscillation when it has zero velocity.

(b) 2 marks

```matlab
function A=ChainMat(N)
A = diag([1:2:2*N-1]) - diag([1:N-1],1) - diag([1:N-1],-1);
```

Note that if a student can not remember how to use `diag` but successfully writes a function using e.g. a for loop, we can still give them full marks.

(c) 1 mark

```
   1          2          3          4
```

(d) 2 marks

```matlab
function pendulum(N,M)
A = ChainMat(N);
[v,e]=eig(A);
```
disp(e(M,M))
plot([v(:,M); 0],0:N)
xlabel('Displacement')
ylabel('Link Number')

NB Marks will not be deducted for omitting labels on axes. Note that a student can still get full marks for this question if they can’t figure out part (b): they just have to assume that a function ChainMat is written.

(c) 3 marks

N=50;
for i=1:N
    A = ChainMat(i);
    e=eig(A)
    t(i)=2*pi/sqrt(e(1))*sqrt(1/9.8/i);
end
plot(t,'o')
xlabel('N');
ylabel('Period (s)')

The oscillation period of a chain is close to the $N = 50$ period, or 1.68 s, and the oscillation period of a simple pendulum has an oscillation period of $\sim2.01$ s, so the ratio is about 0.84.

NB Again, no marks should be deducted if the labelling and plotting symbols are omitted from the program.