# Question 1:

*Please refer to MATLAB script on page 4**for the working.*

### **What is the error in each of these fits?**

The average residual errors of each degree are as follows:

Degree 1: **0.1843 s**

Degree 2: **0.1697 s**

Degree 3: **0.1683 s**

### **In the case of the cubic fit, MATLAB issues a warning. Why does MATLAB do this?**

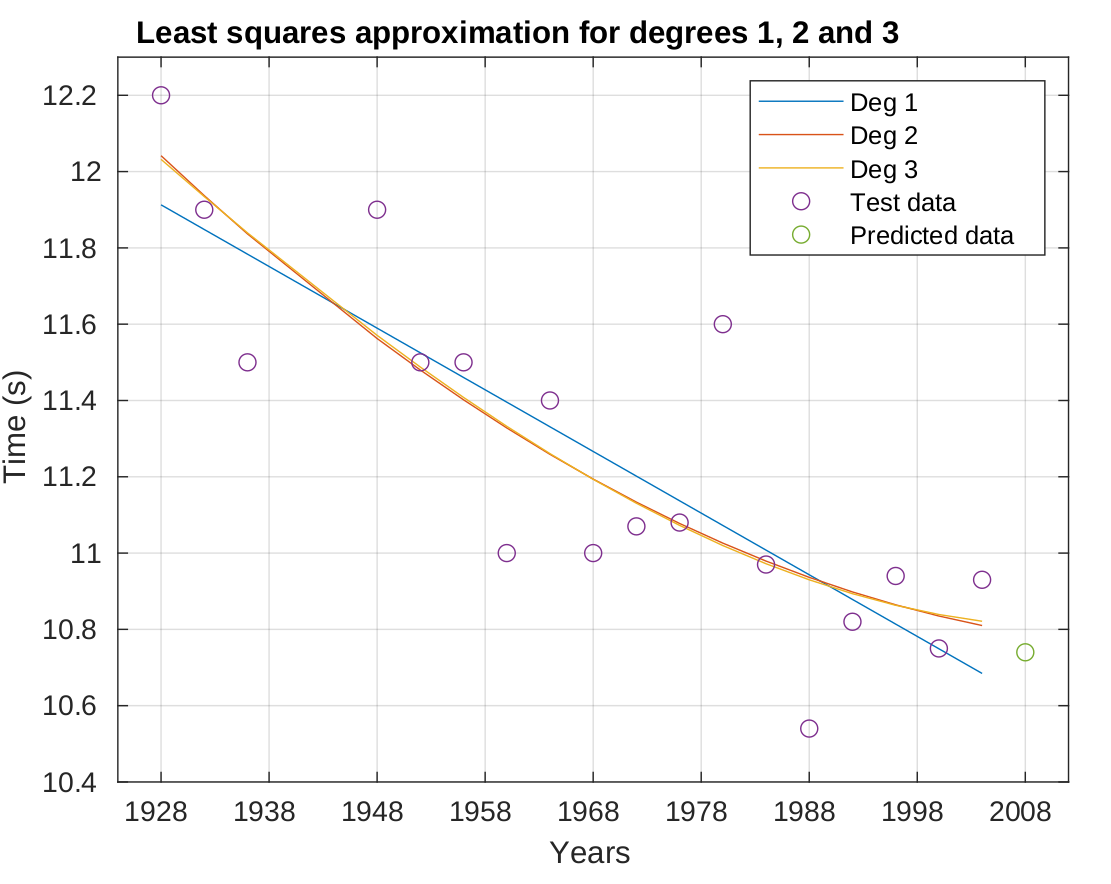
When the dates on the x-axis are set as the year values (1928 – 2004), the cubed values in matrix A3 (for polynomial of degree 3) become much larger in proportion to the first two columns. This difference in size is too large for MATLAB, hence the issuing of a warning.

### **What can be done to avoid this (potential) problem?**

Scaling the dates down by subtracting 1928 off each value prevents the years from growing out of proportion. Another option was to divide the value by 100 after subtracting 1928 as this would cause each value to be within 0 and 1. However, this was avoided for similar reasons as the small values, once cubed, would become much smaller in proportion to the column of ones.

### **Use each of these fits to predict the winning time for the Beijing Olympic Games in 2008.**

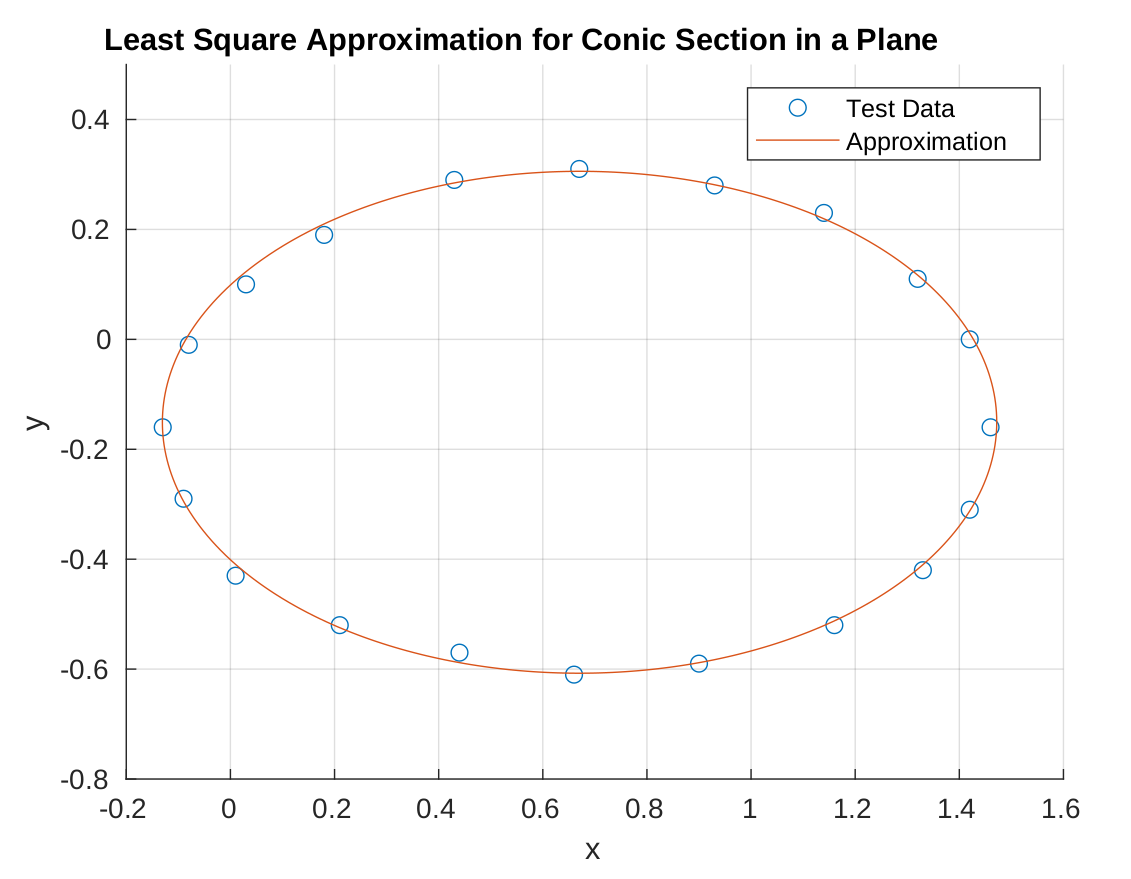
The predicted value was found to be **10.74 s** by substituting 80 (from 2008 – 1928) into each equation to find the average prediction from each degree as plotted in Figure 1. Using the degree 1 fit would is ill-advised as there must be a limiting factor in a human’s ability to run 100 m and the linear trend would continue downwards, eventually crossing the x-axis which is an impossible result. Similarly, a polynomial of degree 2 or 3 will eventually show a trend that is impossible for real world results to follow. An equation resembling may be a more suitable line of best fit as it will reach a limiting factor, therefore more accurately representing this real-world scenario.



*Figure 1.* Least squares approximation for degrees 1, 2 and 3.

# Question 2:

See Figure 2 for the best conic section going through the provided points.



*Figure 2.* Least squares approximation for conic section in a plane.

# Question 3:

(a) The largest singular value is **88962** and the smallest is **5.7696**.

(b) See figure 3a - 3h for the different image outputs.

(c) The storage required as a percentage of the original image for the k largest singular values is found using which simplifies to where n and m are the dimensions of the matrix.

(d) The storage required for each image from part (b) are listed below, rounded to 2 dp:

2 Singular Values: **0.42%**

6 Singular Values: **1.25%**

10 Singular Values: **2.08%**

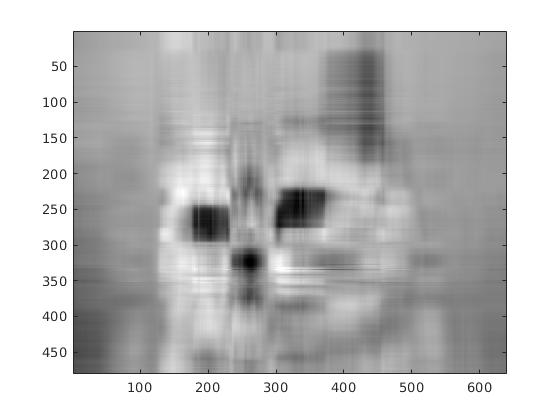
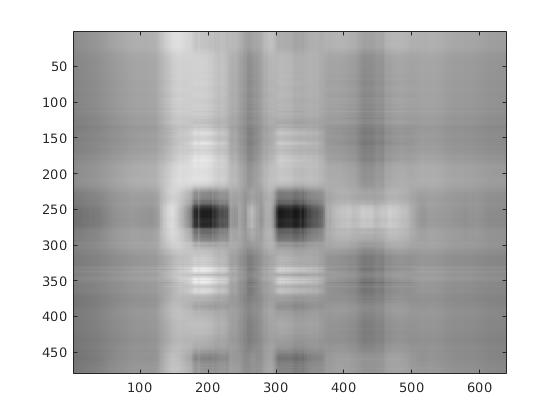
15Singular Values: **3.13%**

20 Singular Values: **4.17%**

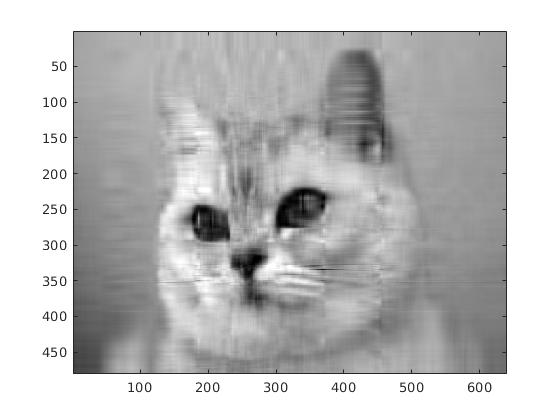
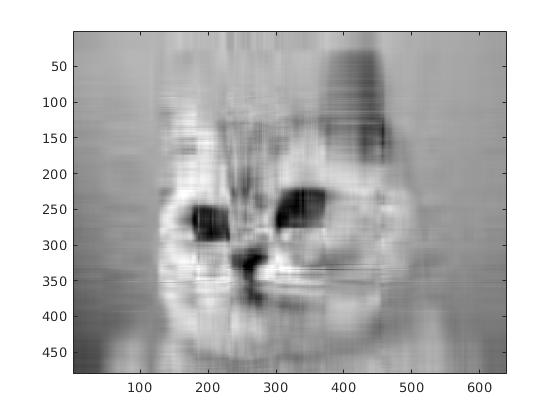
30 Singular Values: **6.25%**

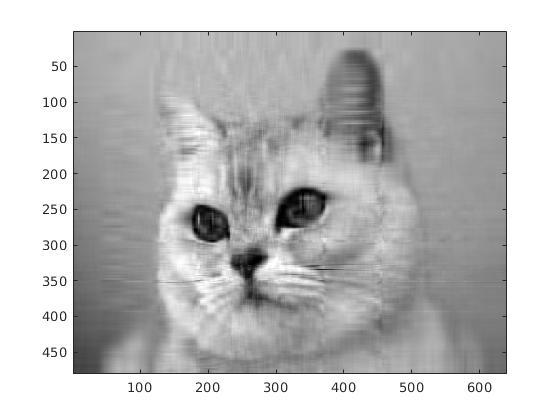
50 Singular Values: **10.42%**

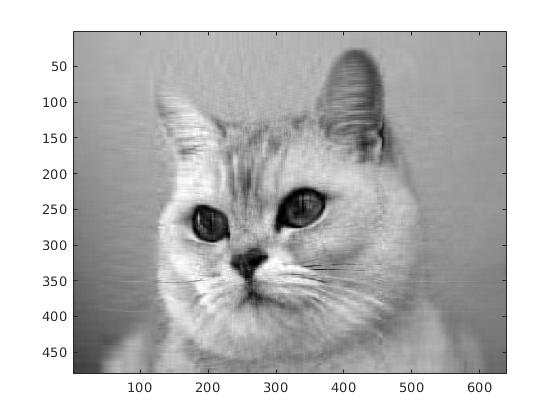
100 Singular Values: **20.83%**

**

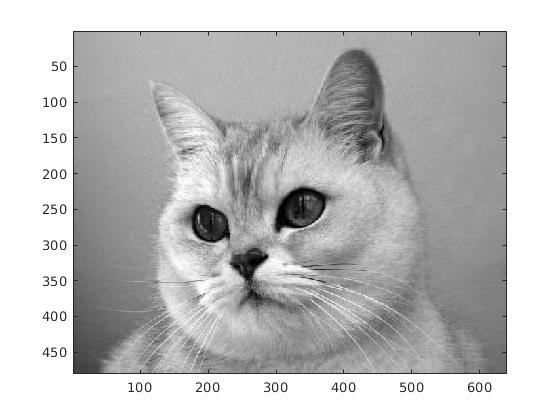
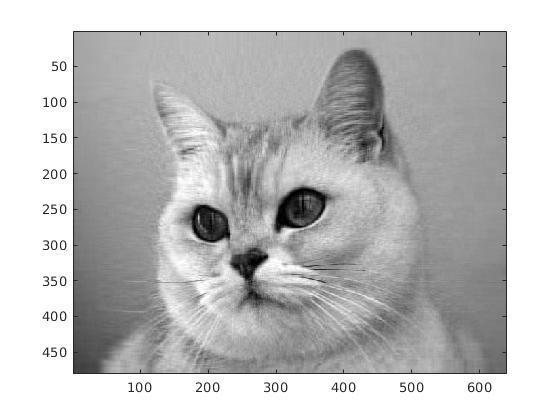
*Figure 3a.* 2 singular values. *Figure 3b. 6* singular values.

**

 *Figure 3c. 10* singular values. *Figure 3d. 15* singular values.

**

*Figure 3e. 20* singular values. *Figure 3f. 30* singular values.

**

*Figure 3g. 50* singular values. *Figure 3h. 100* singular values.