M4N9 PROJECT 1

Isa Majothi
CID: 00950286
The contexts of this report and associated codes are my own work unless otherwise stated

Autumn 2017

QUESTION 1

Given a set of data points $\Omega = \{(x_i, h_i)\}_{i=1}^{N}$ for some $N \in \mathbb{N}$, it is possible to fit a polynomial h_n of degree n which is the polynomial of degree at most n such that h_n "best fits" the data Ω, namely the sum

$$S = \sum_{i=1}^{N} |h_n(x_i) - h_i|^2 \tag{1}$$

is minimised; this is known as the method of least squares (LS) whereby h_n is the LS solution of degree n. For the given data, namely the height of the Moon's surface along a line through the centre of one of its craters Tycho, I have computed the LS solutions h_n for $n = 5$, 10 and 20 and plotted these fitted polynomials alongside the original data in Figures 1, 2 and 3 respectively.

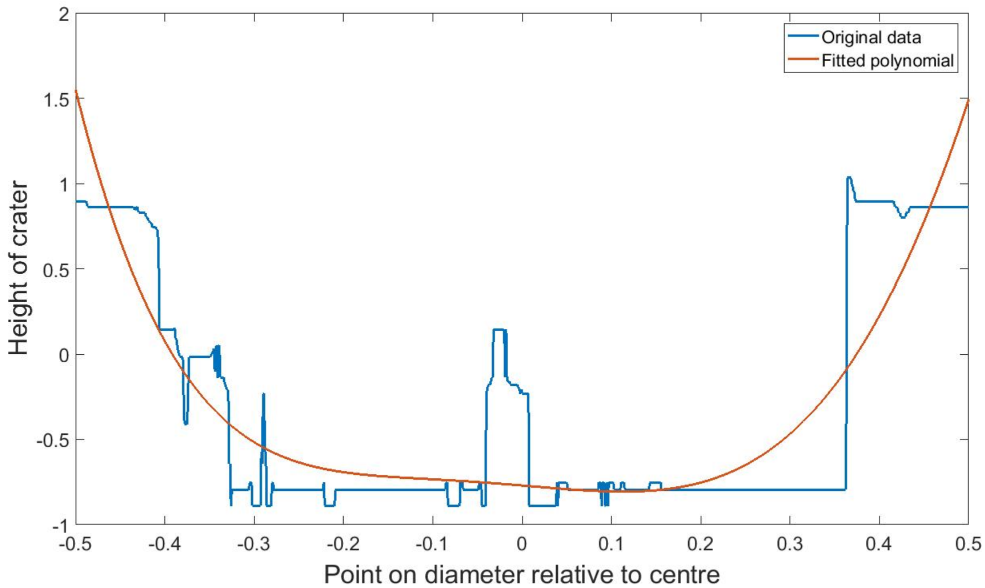

Figure 1: Plot of h_5 against the original data h_i

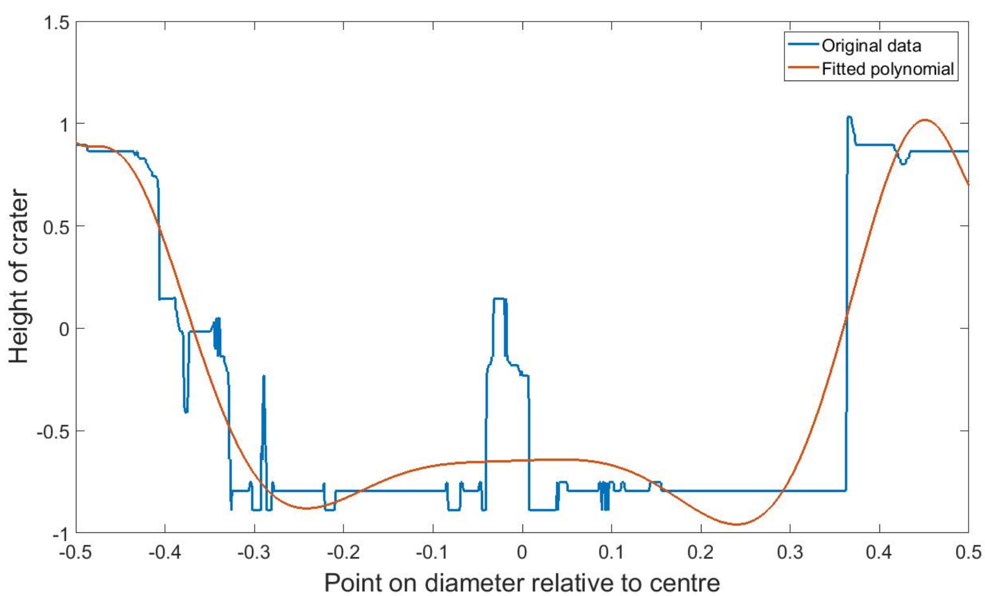

Figure 2: Plot of h_{10} against the original data h_i

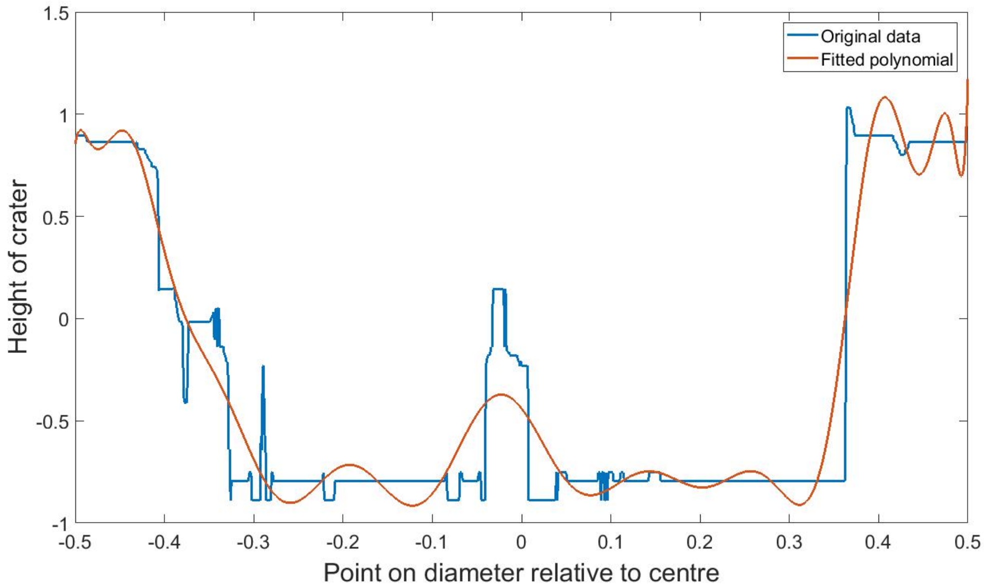

Figure 3: Plot of h_{20} against the original data h_i

For each value of n, I have included the first six monomial coefficients to 4 decimal places (d.p) in Table 1 , where the reported values are in standard form.

Term of $h_n(x)$	$n = 5$	$n = 10$	$n = 20$
x^0	-7.7026×10^{-1}	-6.4603×10^{-1}	-4.4001×10^{-1}
x^1	-4.3420×10^{-1}	1.5433×10^{-1}	-5.6094×10^{0}
x^2	-3.2875×10^{-1}	2.4913×10^{-1}	-9.4455×10^{1}
x^3	8.1133×10^{0}	-1.5936×10^{1}	1.1296×10^{3}
x^4	3.7996×10^{1}	-3.2911×10^{2}	8.0371×10^{3}
x^5	-2.6439×10^{1}	2.3578×10^{2}	-7.6082×10^{4}

Table 1: Monomial coefficients of the six lowest degree terms for $n = 5, 10$ and 20

The table below contains the values of $||h(x_i) - h_i||_2$ rounded to 4 d.p in standard form for each value of n mentioned above, where we have

$$||h(x_i) - h_i||_2 = \left(\sum_{i=1}^{N} |h(x_i) - h_i|^2 \right)^{\frac{1}{2}} \tag{2}$$

Note that $h(x_i)$ represents the value of our LS polynomial evaluated at the point x_i, and h_i is the actual value of h at that corresponding point x_i in the data. These values of $||h(x_i) - h_i||_2$ are stored in the vector *sumResidualSquares1* in my Matlab code.

n	$\|\|h(x_i) - h_i\|\|_2$
5	1.0505×10^1
10	7.7858×10^0
20	5.8983×10^0

Table 2: Residual sum of squares for $n = 5, 10$ and 20

It is evident from looking at Figures 1, 2 and 3 that as we increase the value of n the fits become better at representing certain features of the original data, namely the sudden central peak as well as both gradual and sudden changes in elevation, particularly sudden changes at the sides of the crater. For example, the original data clearly demonstrates a sudden peak at the centre of the crater, and it is only when $n = 20$ that we really see this characteristic demonstrated in our polynomial as Figure 3 shows. However, Figures 1 and 2 show that in the cases where $n = 5$ and $n = 10$ respectively we do not see this feature being exhibited in our polynomials, rather the polynomials show a gradual decrease in depth before a gradual increase once we pass the centre; it is worth noting that h_{10} increases very slightly in the interval $x \in [-0.2, 0.1]$, but it does not reflect the peak to the same effect as h_{20}.

In addition to the centralised peak of the crater, increasing the value of n also seems to better characterise the rigid behaviour of the depth at the sides of the crater. In the case where $n = 5$, Figure 1 shows us that from $x = -0.5$ all we see is a smooth decrease in h_5 until around $x = 0.15$, after which we see a similar gradual rise. Also, whilst Figure 2 may show that h_{10} appears relatively constant in the interval $x \in [-0.5, -0.45]$ the subsequent decrease does not resemble a steep drop particularly well. In addition to this, the sudden increase in depth as x increases around $x = 0.36$ is very poorly fitted by both h_5 and h_{10}. However, in the case of $n = 20$ the sudden decrease in depth around $x = -0.43$ is shown much better, as is the relatively constant depth prior to this via three stationary points in this interval (see Figure 3). The relatively constant depth either side of the crater's centre is also fitted much better, again by stationary points, as is the central peak (see previous paragraph). Finally, the sudden increase in depth around $x = 0.36$ is fitted much better because Figure 3 shows a sharp increase in h_{20} around this point, and the four further stationary points after the sudden rise help to model the depth better than in the cases when $n = 5$ or $n = 10$.

Table 2 which shows $||h(x_i) - h_i||_2$ for each value of n is also further proof that, based on our results, increasing n will improve the fit given by the LS method. We clearly see from the table that $||h(x_i) - h_i||_2$ decreases as n increases, meaning that the cumulative difference between the fitted polynomial h_n evaluated at the points $\{x_i\}$ and the original data is lessened, indicating a better fit.

Taking all of this into account, namely assessing the polynomial fits qualitatively when looking at Figures 1, 2 and 3 and quantitatively via analysing the change in $||h(x_i) - h_i||_2$, using these results we can conclude that as we increase n the fit of the polynomial given by the LS method does improve.

QUESTION 2

Recall that in finding the LS solution, we are seeking the polynomial h_n that minimises

$$S = \sum_{i=1}^{N} |h_n(x_i) - h_i|^2$$

This can be thought of as finding the "best-fit" solution $x \in \mathbb{C}^{n+1}$ to a linear system of equations $Ax = b$, where $A \in \mathbb{C}^{N\times(n+1)}$ with $N \geq (n+1)$ and $b \in \mathbb{C}^N$ i.e. we have an over-determined system. In using the LS method in Question 1, I set the matrix A to be the Vandermonde matrix with the appropriate number of columns depending on n, namely $n+1$ columns. The LS algorithm was then applied to the system where b was the vector h containing the data for the depths of the crater at different points, and using the reduced QR factorisation of A the solution $x \in \mathbb{C}^{n+1}$ was a vector consisting of the coefficients of the polynomial of degree n giving the LS solution.

Now given any point x, we can apply the technique of moving least squares (MLS) to fit the original data by minimising the following sum at any given point x

$$S(x) = \min_{p\in P_n} \sum_{i=1}^{N} \theta(|x - x_i|)\, |p(x_i) - h_i|^2 \tag{3}$$

where x_i and h_i correspond to the points in the data set and $\theta(r)$ is a weight function that decays as $|\mathbf{x} - \mathbf{x}_i| \to \infty$ and $\theta(r) > 0$ for all r.

Since we have fixed the value of x and the x_i's are fixed by the data, this means that $\theta(|x - x_i|)$ is fixed for $i = 1, \ldots, N$, and by comparison with (1) we can implement the MLS method by applying the LS method to a slightly modified linear system which incorporates the weight function values for $i = 1, \dots, N$ (this is because (3) also involves minimising a summation over all polynomials of degree at most n).

For our given data about the crater Tycho, to compute the LS solution of degree n in Question 1 I found the vector $\mathbf{c} = (c_0, c_1, \dots, c_n)^T \in \mathbb{C}^{n+1}$ which minimised (1) by consideing the following linear system of equations

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_N \end{pmatrix} \tag{4}$$

Comparing this to solving $Ax = b$, this gives $A \in \mathbb{C}^{N\times(n+1)}$, $x \in \mathbb{C}^{n+1}$ and $b \in \mathbb{C}^N$ and this gave the LS solution in the form of a polynomial of degree n whose coefficients were $\{c_i\}_{i=0}^n$.

Again, comparing this to the MLS algorithm in (3), observe that we need to multiply $|p(x_i) - h_i|^2$ by $\theta(|x - x_i|)$, which gives the linear system

$$\begin{pmatrix} \theta_1 & \theta_1 x_1 & \theta_1 x_1^2 & \dots & \theta_1 x_1^n \\ \theta_2 & \theta_2 x_2 & \theta_2 x_2^2 & \dots & \theta_2 x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_N & \theta_N x_N & \theta_N x_N^2 & \dots & \theta_N x_N^n \end{pmatrix} \begin{pmatrix} \bar{c}_0 \\ \bar{c}_1 \\ \vdots \\ \bar{c}_n \end{pmatrix} = \begin{pmatrix} \theta_1 h_1 \\ \theta_2 h_2 \\ \vdots \\ \theta_N h_N \end{pmatrix} \tag{5}$$

where θ_i is equal to $\theta(|x_i - x|)$ for $i = 1, \dots N$.

To find the MLS solution, at each point x in our domain we find the LS solution the linear system above to obtain the polynomial $h \in P_n$ which minimises (3) and whose coefficients are $\{\bar{c}_i\}_{i=0}^n$. Using this polynomial, we then compute the value of this polynomial evaluated at the point x, giving the MLS solution at x to be

$$h(x) = \sum_{i=0}^{n} \bar{c}_i x^i$$

Note that by the continuity of the weight function $\theta(r)$, applying the MLS method over the domain will result in a continuous function $h(x)$ despite inevitably obtaining different LS solutions at every point, so by repeatedly applying this procedure we will obtain the MLS solution h.

To concisely highlight how the standard LS algorithm can be adapted to perform the MLS method, I have included step-by-step algorithms for each method below. Note that in both cases, $A \in C^{m\times n}$ with $m \geq n$, $b \in C^m$ and we are solving for $x \in \mathbb{C}^n$.

Standard Least Squares Algorithm

1. Compute the QR factorisation of A, where A is the Vandermonde matrix with $n+1$ columns if we're seeking the LS solution of degree n. In this project, this was done via Householder reflections using the provided Matlab function *house*.
2. Using the fact that $R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}$, where $\hat{R}$ is an upper-triangular square matrix, as well as the fact that Q is unitary, reduce the system $Ax = b$ to $\hat{R}x = \hat{Q}^*b$.
3. Use backwards substitution to solve $\hat{R}x = \hat{Q}^*b$ for x to obtain the LS solution coefficients in x, making use of fact that $\hat{R}$ is upper-triangular.

Moving Least Squares Algorithm

1. Given our Vandermonde matrix A, multiply each entry in row i by θ_i, where θ_i is as previously defined - call this matrix A'.
2. Multiply the $i-th$ entry of b by θ_i for a given point - call this vector b'.
3. Compute the QR factorisation of A'. Again, in this project this was done via Householder reflections using the given Matlab function *house*.
4. Using the fact that $R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}$, where $\hat{R}$ is an upper-triangular square matrix, as well as the fact that Q is unitary, we reduce the system $A'x = b'$ to $\hat{R}x = \hat{Q}^*b'$.
5. Use backwards substitution to solve $\hat{R}x = \hat{Q}^*b$ for x to obtain the LS solution *at this particular point* - repeat Steps 1-5 over the domain to obtain the entire MLS solution.

QUESTION 3

Following on from the concepts explained in Question 2, a slight modification to the Matlab code which implements the LS method will give the MLS solution. Using the given weight function $\theta(r) = e^{-r^2/\eta}$, I have modified my Matlab code from Question 1 to carry out the *accelerated* MLS method, whereby I use the modified weight function

$$\tilde{\theta}(r) = \begin{cases} \theta(r) & \text{for} \quad \theta(r) \geq \epsilon \\ 0 & \text{for} \quad \theta(r) < \epsilon \end{cases}$$

Note that since we are evaluating $\tilde{\theta}(|x-x_i|)$, we have that $\tilde{\theta}(|x-x_i|)$ is a decreasing function due to the fact that $\theta(r) = e^{-r^2/\eta}$ is monotonically decreasing for $r \geq 0$ and clearly $|x - x_i| \geq 0$. As a result, in the linear system (5) with θ_i replaced with $\tilde{\theta}_i$ for $i = 1, \ldots, N$ we get

$$\begin{pmatrix} \tilde{\theta}_1 & \tilde{\theta}_1 x_1 & \tilde{\theta}_1 x_1^2 & \ldots & \tilde{\theta}_1 x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{\theta}_p & \tilde{\theta}_p x_p & \tilde{\theta}_p x_p^2 & \ldots & \tilde{\theta}_p x_p^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{\theta}_q & \tilde{\theta}_q x_q & \tilde{\theta}_q x_q^2 & \ldots & \tilde{\theta}_q x_q^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{\theta}_N & \tilde{\theta}_N x_N & \tilde{\theta}_N x_N^2 & \ldots & \tilde{\theta}_N x_N^n \end{pmatrix} \begin{pmatrix} \bar{c}_0 \\ \bar{c}_1 \\ \vdots \\ \bar{c}_n \end{pmatrix} = \begin{pmatrix} \tilde{\theta}_1 h_1 \\ \vdots \\ \tilde{\theta}_p h_p \\ \vdots \\ \tilde{\theta}_q h_q \\ \vdots \\ \tilde{\theta}_N h_N \end{pmatrix} \tag{6}$$

where $\tilde{\theta}_i = 0$ for $i < p$ and $i > q$, and $1 \leq p \leq q \leq N$ with at least one strict inequality. As a result, the system of linear equations in (6) which has N equations and $n+1$ unknowns can be reduced to a linear system with $q-p+1$ equations since we will have entire rows of zeros in our matrix and also the vector on the right-hand side in (6); this results in the needing to find the LS solution of the following linear system of equations

$$\begin{pmatrix} \tilde{\theta}_p & \tilde{\theta}_p x_p & \tilde{\theta}_p x_p^2 & \ldots & \tilde{\theta}_p x_p^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{\theta}_q & \tilde{\theta}_q x_q & \tilde{\theta}_q x_q^2 & \ldots & \tilde{\theta}_q x_q^n \end{pmatrix} \begin{pmatrix} \bar{c}_0 \\ \bar{c}_1 \\ \vdots \\ \bar{c}_n \end{pmatrix} = \begin{pmatrix} \tilde{\theta}_p h_p \\ \vdots \\ \tilde{\theta}_q h_q \end{pmatrix} \tag{7}$$

In implementing the accelerated MLS method, this clearly reduces the number of operations required than the normal MLS method because the linear system upon which we find the LS solution is evidently smaller in (7) than it is in (5). Given a point x, for relatively large values of $|x - x_i|$ the value of $\theta(|x - x_i|)$ will be negligible due to the decreasing nature of $\theta(r)$, and since $\tilde{\theta}(r)$ reduces these small values to zero this allows for massive sections of the matrix to be removed to give a significantly smaller linear system of equations to solve. In terms of then finding the MLS solution, the LS solution of this reduced system is found by applying Steps 3 - 5 in the MLS algorithm (see Question 2).

It is also worth noting that implementing the MLS method with this modified weight function will not noticeably affect the obtained solution, because the rows which are removed from the matrix will be negligible in size, so it makes sense from a computational viewpoint to remove them as it greatly reduces the number of operations required without significantly affecting the solution.

QUESTION 4

After having modified my code for the standard LS algorithm to implement the accelerated MLS algorithm, the results of using the modified weight function

$$\tilde{\theta}(r) = \begin{cases} \theta(r) & \text{for} \quad \theta(r) \geq \epsilon \\ 0 & \text{for} \quad \theta(r) < \epsilon \end{cases}$$

for different values of η and n can be found below. This is so the effect of changing η and n on the goodness of the fit when using the MLS method can be investigated, as well as seeing how the MLS method compares to he standard LS method. Note that when obtaining these results I used my accelerated MLS function, setting $\epsilon = 10^{-6}$.

Part (a): $\eta = 10^{-2}, 10^{-3}$ and 10^{-4} with $n = 3$

I have inserted plots for the solutions of the MLS method where $n = 3$ is fixed and $\eta = 10^{-2}, 10^{-3}$ and 10^{-4} in Figures 4, 5 and 6 respectively. I also included a table of the values of $||h(x_i) - h_i||_2$ rounded to 4 d.p in standard form for each value of η, which are stored in the vector *sumResidualSquares4a* in my Matlab code.

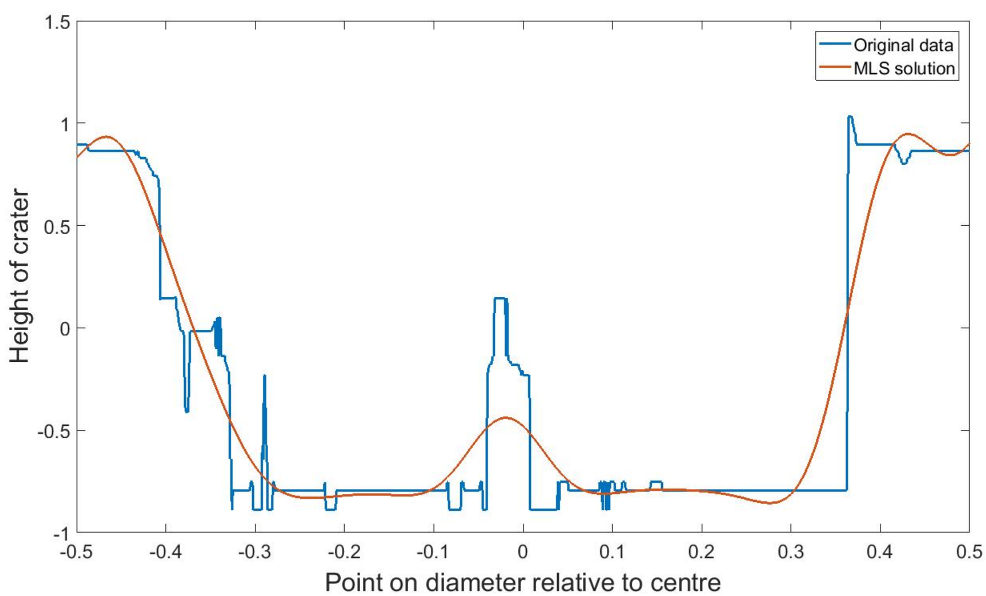

Figure 4: Plot of MLS solution h against the original data h_i, with $n = 3$ and $\eta = 10^{-2}$

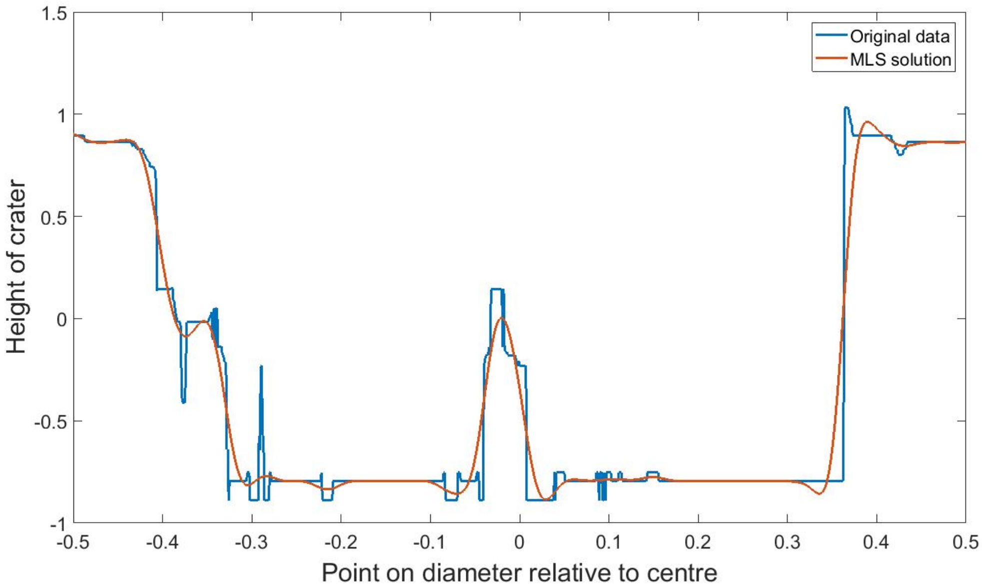

Figure 5: Plot of MLS solution h against the original data h_i, with $n = 3$ and $\eta = 10^{-3}$

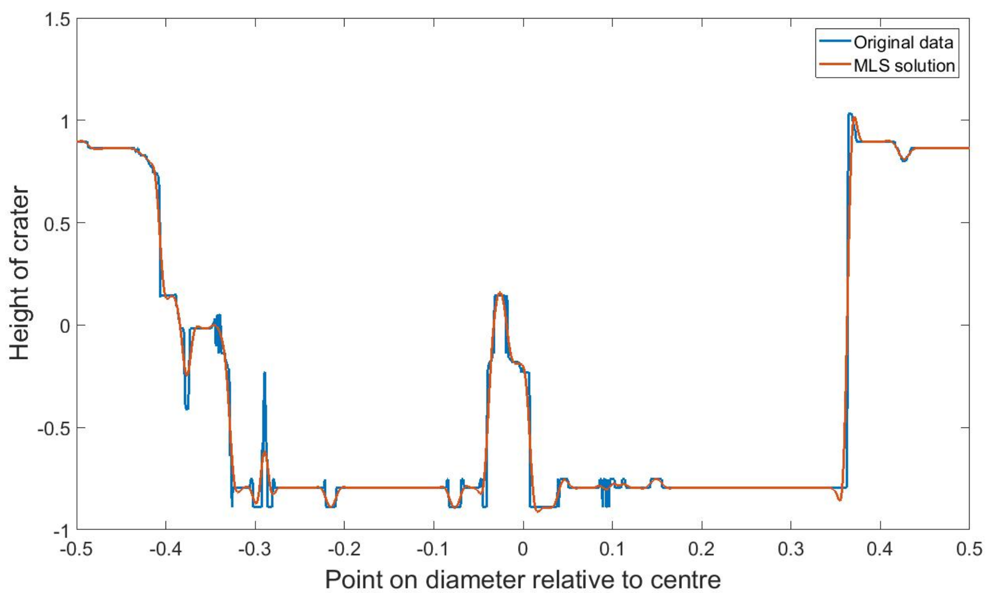

Figure 6: Plot of MLS solution h against the original data h_i, with $n = 3$ and $\eta = 10^{-4}$

η	$\|\|h(x_i) - h_i\|\|_2$
10^{-2}	6.4977
10^{-3}	3.6557
10^{-4}	2.0390

Table 3: Residual sum of squares for $\eta = 10^{-2}, 10^{-3}$ and 10^{-4} with $n = 3$

Part (b): $n = 1, 3$ and 10 with $\eta = 10^{-3}$

I have inserted plots for the solutions of the MLS method where $\eta = 10^{-3}$ is fixed and $n = 1, 3$ and 10 in Figures 7, 8 and 9 respectively. I also included a table of the values of $||h(x_i) - h_i||_2$ rounded to 4 d.p in standard form for each value of n, which are stored in the vector *sumResidualSquares4b* in my Matlab code.

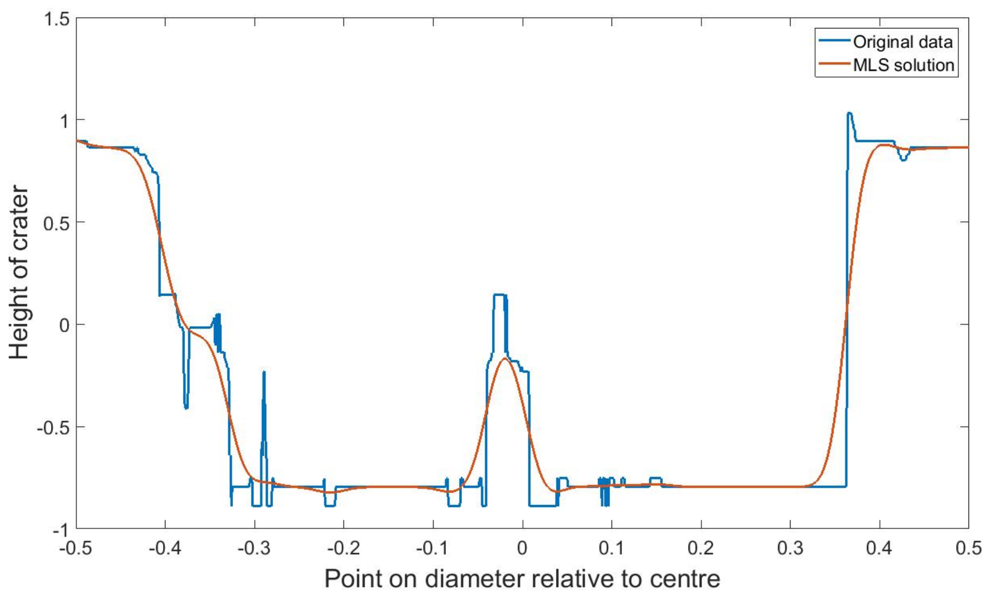

Figure 7: Plot of MLS solution h against the original data h_i, with $n = 1$ and $\eta = 10^{-3}$

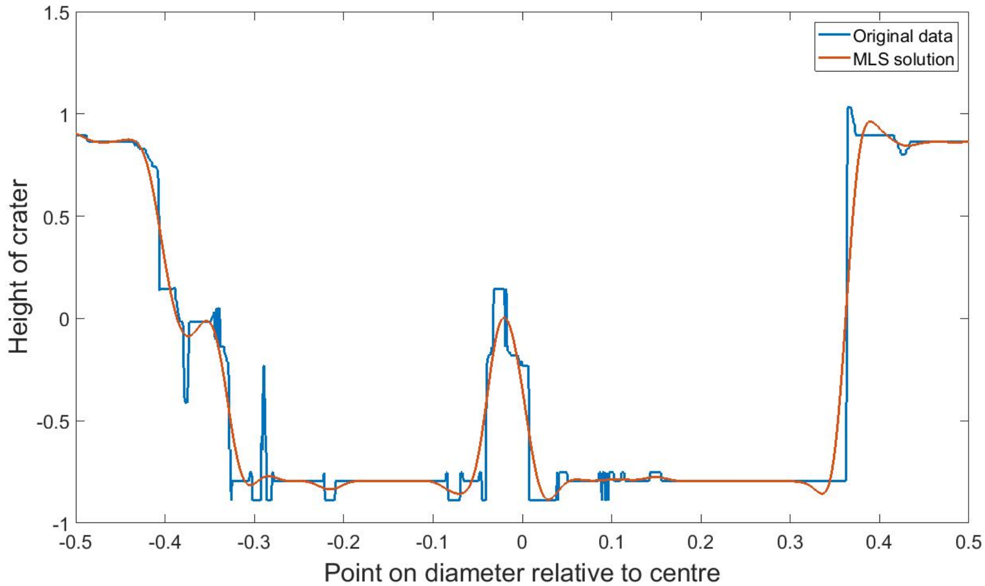

Figure 8: Plot of MLS solution h against the original data h_i, with $n = 3$ and $\eta = 10^{-3}$

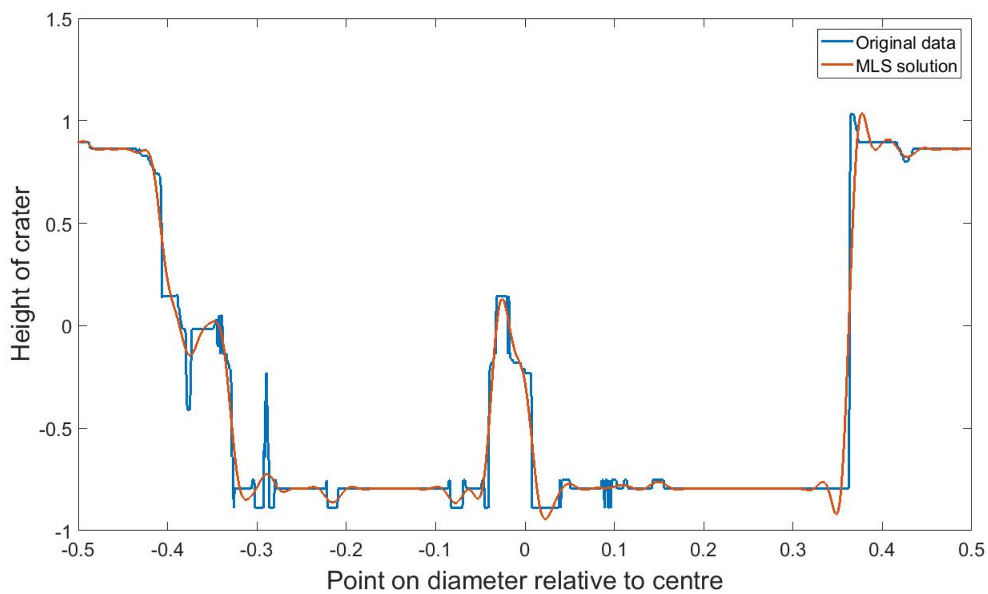

Figure 9: Plot of MLS solution h against the original data h_i, with $n = 10$ and $\eta = 10^{-3}$

n	$\lvert\lvert h(x_i) - h_i \rvert\rvert_2$
1	4.5577
3	3.6557
10	2.8026

Table 4: Residual sum of squares for $n = 1, 3$ and 10 with $\eta = 10^{-3}$

Analysis and Interpretation

It is evident from the results in Part (a) that deceasing the value of η for a fixed value of n leads to an improvement in the solution obtained by applying the MLS method. After assessing Figures 4, 5 and 6, it is clear that as η is changed by a factor of 10^{-1} we observe an ever-increasing improvement to the fit. Figure 4 shows that the MLS solution obtained when $n = 3$ and $\eta = 10^{-2}$ still resembles a polynomial of some sorts as there are no sudden changes in h at the sides of the crater nor in the centre. It is worth noting that Figure 5 shows a stark improvement when $\eta = 10^{-3}$ for $n = 3$, as the sudden changes in the depth of the crater are characterised much better than in Figure 4. However, whilst Figure 5 shows a somewhat satisfactory fit for the data, setting $\eta = 10^{-4}$ produced an exceptionally good fit to the original data, as is shown in Figure 6.

An explanation as to why this is the case can be obtained by referring back to our weight function

$$\tilde{\theta}(r) = \begin{cases} \theta(r) & \text{for} \quad \theta(r) \geq \epsilon \\ 0 & \text{for} \quad \theta(r) < \epsilon \end{cases}$$

Notice that decreasing the value of η for this given $\theta(r)$ causes the value of the function to decrease extremely fast (since it is an exponential function), meaning that when evaluating $\tilde{\theta}(|x - x_i|)$ for a given point x we observe that many of them will be equal to zero i.e. when $|x - x_i|$ is relatively large. This means that only the data points corresponding to x_i which are very close to x will be considered when computing the MLS solution $h(x)$ using the accelerated MLS method, hence the value of $h(x)$ obtained is more likely to be closer to the h_i around it, even if there are steep rises or drops in the depth of the crater at any point.

This conclusion is also supported by the numerical evidence shown in Table 3, because as η is decreased we see that $||h(x_i) - h_i||_2$ decreases as well. This means that the absolute difference between the original data and the solution obtained by the MLS method at the points x_i for $i = 1, \ldots, N$ is smaller for smaller values of η given fixed n, as is clearly shown graphically by Figures 4, 5 and 6.

The results from Part (b) indicate that the performance of the MLS method improves as n is increased for a fixed value of η. Whilst $\eta = 10^{-3}$, setting $n = 1$ and performing the MLS method produced a solution which did demonstrate the features of the original data to some degree, as Figure 7 shows the sharp changes in elevation at the edges of the crater, as well as a peak in the centre. Figure 8 also shows these features exhibited in our MLS solution for when $n = 3$, and we observe that aside from the sudden changes in depth at the sides and centre we also see more subtle changes in the original data are reflected in our solution. But again, increasing n further such that $n = 10$ fitted the data even better as Figure 9 shows, because based on the solution and the original data we observe very little difference between the two, especially as the sudden changes in elevation are fitted very well, whereas this wasn't the case for $n = 3$ to the same extent. Table 4 provides further justification of this, since the values of $||h(x_i) - h_i||_2$ decreased as n increased.

It is worth stating that based on these results, I believe that η has a greater influence on the performance of the MLS method than n does, even though there is clear evidence to suggest that both have an impact on the performance of the MLS method. This is because even though performing the LS method at a particular point with a greater value of n will undoubtedly produced a better polynomial fit, Figure 7 shows that when $n = 1$ and $\eta = 10^{-3}$ the MLS solution was much better at representing the original data than when $n = 3$ and $\eta = 10^{-2}$ (see Figure 4), even though the latter involves a greater value of n. This is most likely due to the fact that our weight function decreased exponentially whereas changing n only affects the fit in a polynomial way, and it is common knowledge that exponentials change at a much faster rate than polynomials; thus the effect of changing η which appears in the exponent will have a greater influence on the quality of the MLS solution than n. This is also supported by the information in Tables 3 and 4, because we observe that for $n = 3$ decreasing the value of η caused a relatively greater decrease in $||h(x_i)-h_i||_2$ than was the case for $\eta = 10^{-3}$ and increasing the value of n.

In terms of the quality of the fit given by their respective solutions, the plots in Figures 1-9 are resounding evidence that the MLS method produces a much better fit than the standard LS method. In the LS solutions seen in Figures 1, 2 and 3, it is only in the case when $n = 20$ that we start to see any real representation of the characteristics of the original data in the solution, and even then the fit only captures certain characteristics of the data to an extent. However, with the exceptions of Figures 4 and 7 the solutions of using the MLS method are conclusive evidence that the MLS method is better at fitting the original data, because the sudden changes in depth are all demonstrated in these MLS solutions, regardless of the point we are looking at. In addition, at face value the MLS solution in Figure 7 does appear to fit the original data better than the LS solution in Figure 3, whilst it is only in Figure 4 that we could argue that the MLS solution given when $n = 3$ and $\eta = 10^{-2}$ is not as good at fitting the original data as the LS solution in Figure 3 is.

Further evidence to support this is the values of $||h(x_i) - h_i||_2$ in Tables 2, 3 and 4; when comparing the lowest value of $||h(x_i) - h_i||_2$ given by the LS solutions in Table 2 i.e. when $n = 20$, we observe that the only MLS solution that gives a greater value of $||h(x_i) - h_i||_2$ is the case when $n = 3$ and $\eta = 10^{-2}$ (see Table 3), whereas all other values in Tables 3 and 4 are smaller than $||h(x_i) - h_i||_2$ for the LS solution when $n = 20$. This shows that the overall deviation of the fits from the original data was less in all but one case of the MLS solutions when compared to the LS solutions.

Whilst the MLS solutions do produce significantly improved fits for the original data than the LS solutions, this obviously comes at an operational cost. Despite using the accelerated MLS method to find the MLS solutions, which drastically reduces the size of the system to which the standard LS method is applied to (see Question 3), in actually computing the MLS solution we are applying the LS algorithm at every point in our domain; note that in my Matlab code I computed the MLS solution at each data point x_i and since there were 1001 data points this meant applying the LS method 1001 times. However, in the standard LS method it is only applied once and from this the polynomial of best fit for any given degree n is determined considering the data as a whole, whereas the use of a weight function in the MLS method helps to put more emphasis on the data which is closer to the point x in question. So clearly, whilst the MLS method may be much better at fitting the original data, ensuring the quality of this fit comes at a potentially huge operational cost compared to the standard LS method.

This is reflected in the run times of carrying out each method. For example, applying the LS method used in Question 1 with $n = 10$ gave a run time of 0.308133 seconds, whereas the accelerated MLS method with $\eta = 10^{-4}$, $n = 10$ and $\epsilon = 10^{-6}$ gave a run time of 0.909231 seconds, which is about three times longer. Furthermore, applying the non-accelerated MLS method (by setting $\epsilon = 0$) with these same values of η and n produced a run time of 1.373748 seconds. All of this information provides further evidence that the much-improved fit of the MLS method comes at an operational cost, but also that the accelerated MLS produces a substantial increase in speed compared to the non-accelerated MLS method, which was a speed-up of factor 1.51 to 2 d.p in the example above. Table 5 highlights the speed-up of using the accelerated MLS method for the values of n and η used earlier in Question 4.

n	η	Non-Accelerated MLS Run Time (seconds)	Accelerated MLS Run Time (seconds)	Speed-Up
1	10^{-3}	0.661412	0.620544	1.07
3	10^{-2}	0.731465	0.641011	1.14
3	10^{-3}	0.763849	0.650526	1.17
3	10^{-4}	0.723404	0.629304	1.15
10	10^{-3}	1.373748	0.909231	1.51

Table 5: Run times for the non-accelerated and accelerated MLS methods

QUESTION 5

It is evident from Question 4 that computing the MLS solution results in an increased operational cost compared to the standard LS method. In order to reduce this cost, I have reduced the number of data points which I used to generate the MLS solution below in order to investigate how this affects the fit of the MLS solution to the original data. I did this in two ways: one being systematically by including the first data point (namely $x = -0.5$) and then taking every $k-th$ point after this, where k is an integer greater than 1; the other by taking a random subset of the original data, whereby I took a small random sample of the data (5% of the original data) to see how the distribution of these random points affected the quality of the MLS solution. Note that in obtaining the results below, I fixed $n = 3$ and $\eta = 10^{-3}$, and since I used the accelerated MLS method I set $\epsilon = 10^{-6}$ as before.

k	$\|\|h(x_i) - h_i\|\|_2$
2	3.6494e+00
3	3.6764e+00
4	3.6890e+00
5	3.6571e+00
10	3.8508e+00
20	4.5533e+00
30	5.5122e+00

Table 6: Residual sum of squares after reducing the number of data points used to generate the MLS solution by taking every $k - th$ point

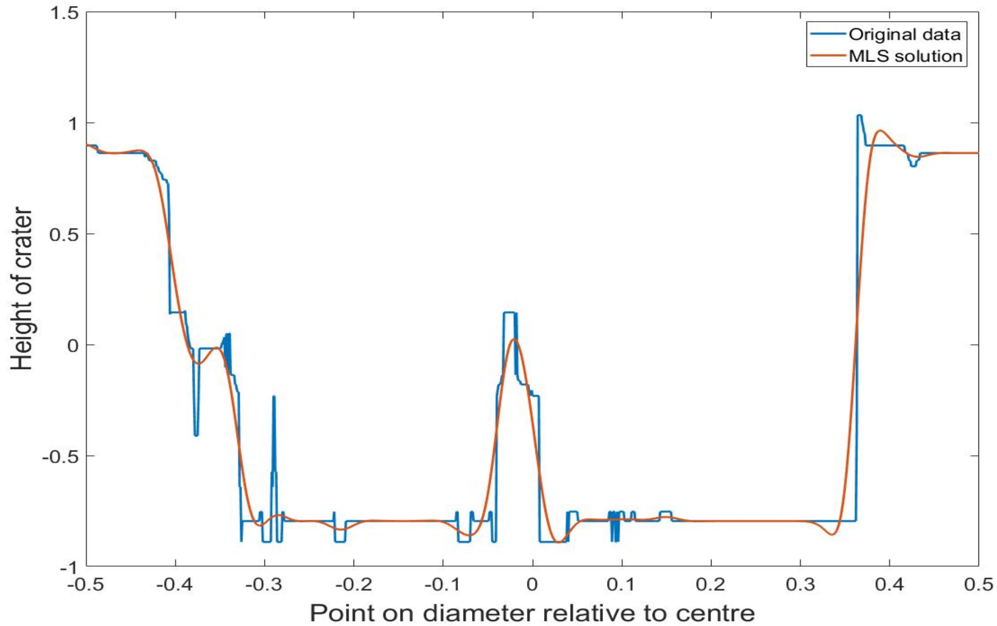

Figure 10: Plot of MLS solution h against the original data h_i, with $k = 2$

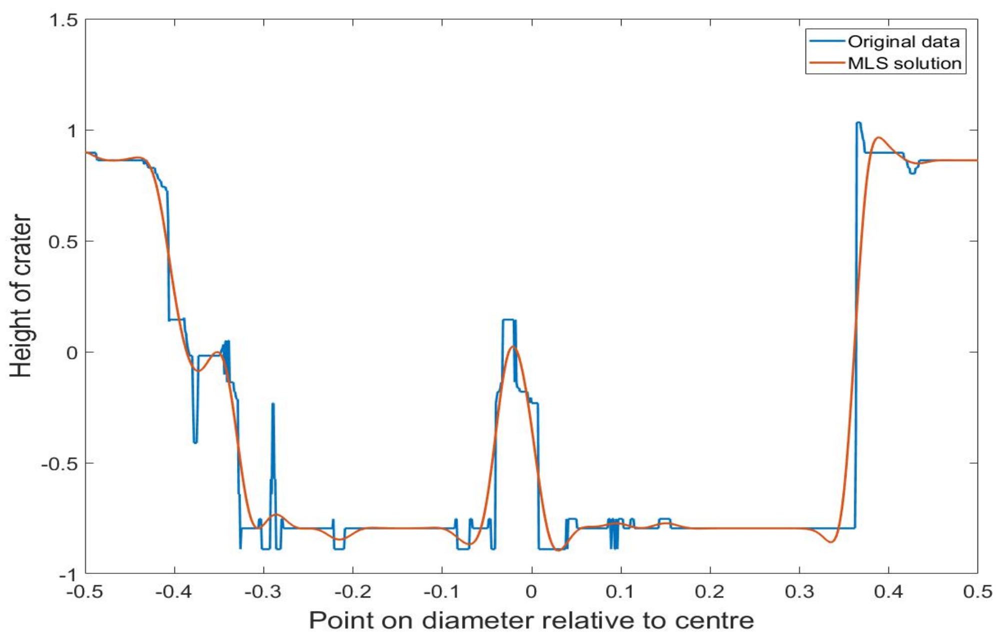

Figure 11: Plot of MLS solution h against the original data h_i, with $k = 5$

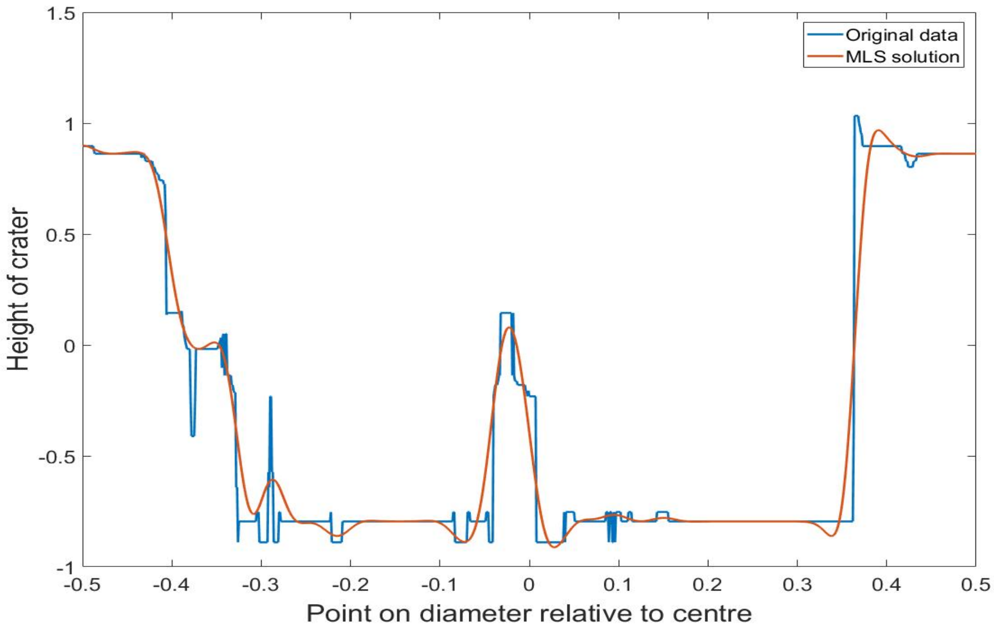

Figure 12: Plot of MLS solution h against the original data h_i, with $k = 10$

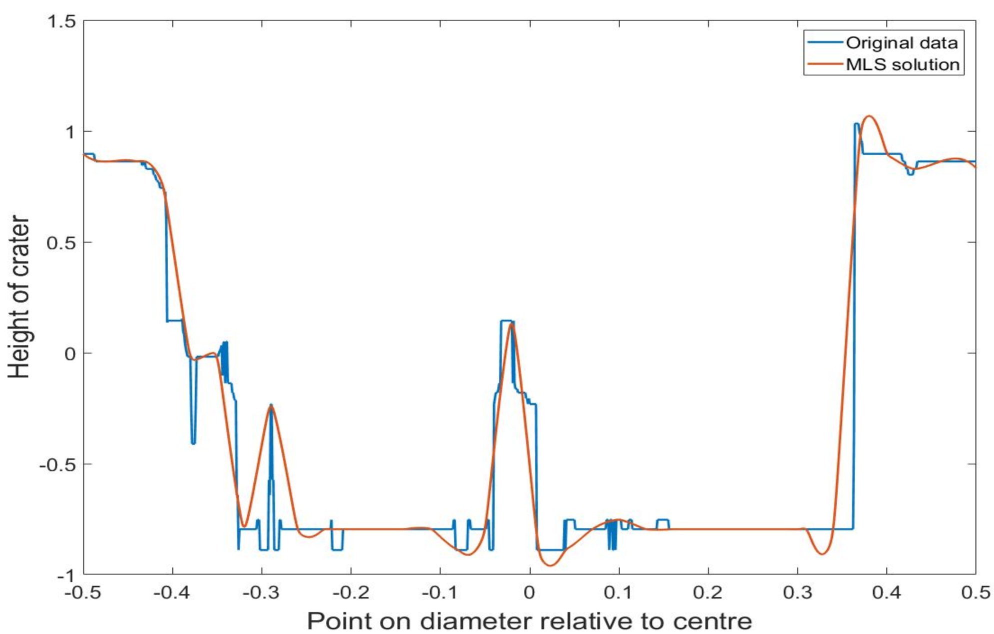

Figure 13: Plot of MLS solution h against the original data h_i, with $k = 30$

After looking at the values of $||h(x_i) - h_i||_2$ in Table 6, and comparing the quality of the fits shown in Figures 10-13, it is clear that decreasing the number of data points used from the original data reduces the quality of the fit of the MLS solution. Whilst all of Figures 10-13 show relatively good fits that are better than all of the LS solutions in Question 1, Table 6 shows that as we increase the step size when systematically taking every $k-th$ data point that this increases the deviation of the fitted MLS solution from our original data, because beyond $k \geq 4$ we observe that $||h(x_i) - h_i||_2$ only increases as k increases.

However, it is worth noting that $||h(x_i) - h_i||_2$ does not increase by a substantial amount, so from a computational angle it may be beneficial to reduce the number of data points used in this systematic way since the resulting fit is not affected dramatically. This is most likely due to the fact that we wouldn't expect the elevation of the crater to vary too drastically over a small interval, so it is sufficient to take a subset in such a systematic because within this subset the characteristics of the original data will not be lost on the whole.

When taking a relatively large random subset of the original data, we would not expect there to be much difference between the two respective MLS solutions since our subset will more likely than not still include the features of the original data. More interesting is the quality of the fit when a randomly selected subset of the data only includes a small percentage of the original data, because since this subset is so relatively small if it is distributed in such a way that certain intervals of the domain are not represented appropriately then the MLS solution will not produce a particularly good fit.

In order to investigate how the distribution of such a randomly selected subset can affect the quality of the MLS solution, I have included the results of three MLS solutions given when the subset consists of 5% of the original data, which is roughly equivalent to setting $k = 20$ in the systematic approach with the difference that these points will not be equispaced. The resulting fits of these three trials are included in Figures 14, 15 and 16.

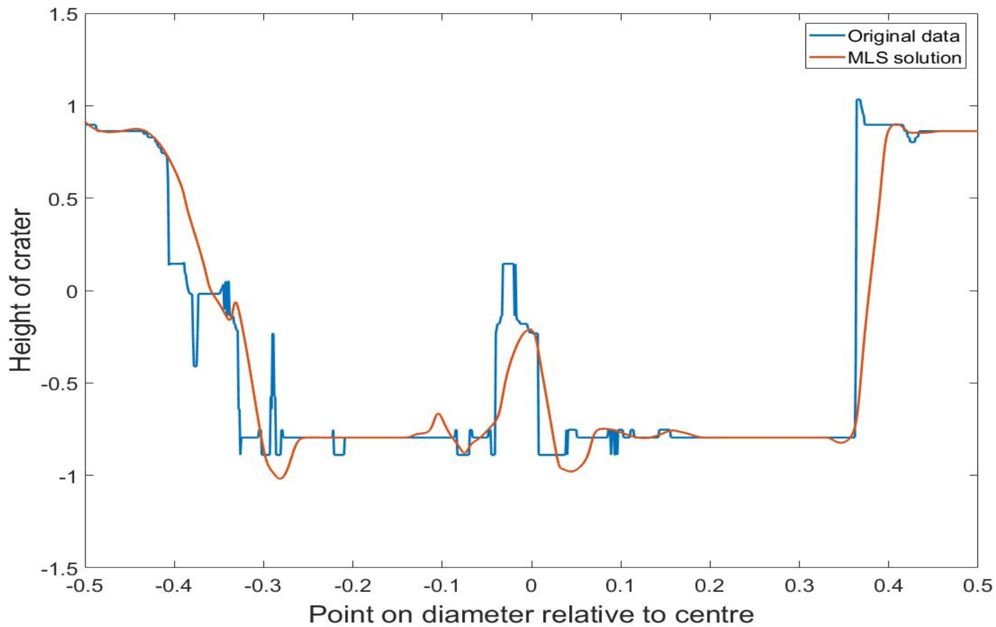

Figure 14: Trial 1 - Plot of MLS solution h against the original data h_i, in which $||h(x_i) - h_i||_2 = 7.6424$ to 4 d.p

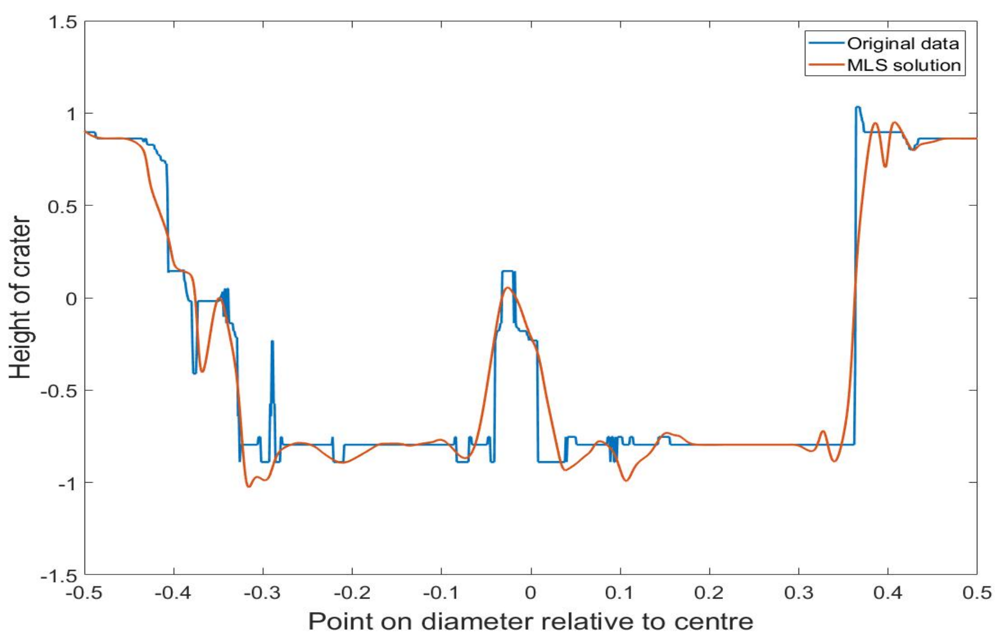

Figure 15: Trial 2 - Plot of MLS solution h against the original data h_i, in which $||h(x_i) - h_i||_2 = 4.6193$ to 4 d.p

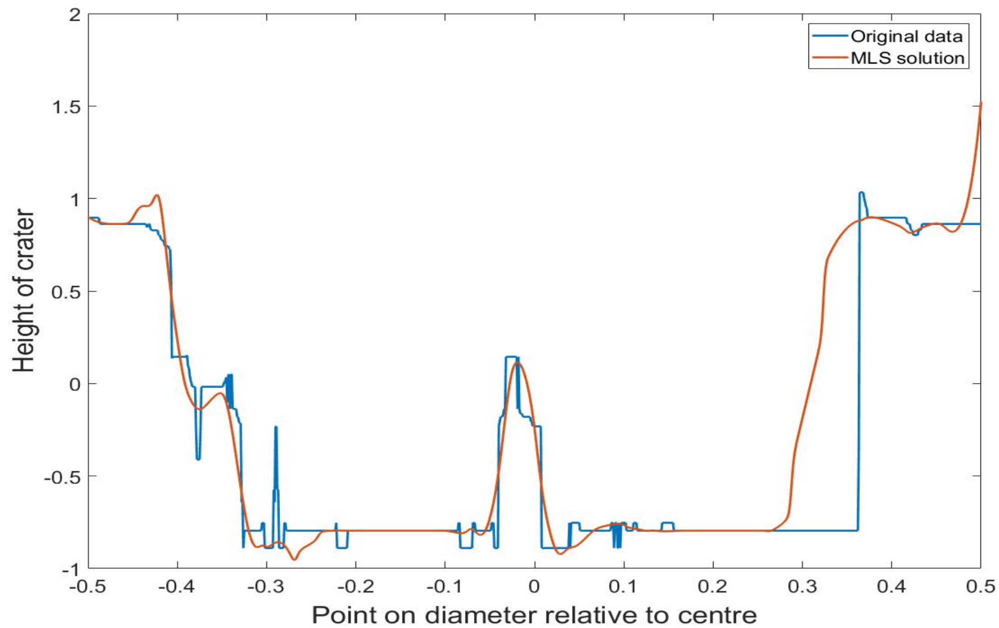

Figure 16: Trial 3 - Plot of MLS solution h against the original data h_i, in which $||h(x_i) - h_i||_2 = 1.1380 \times 10^1$ to 4 d.p

After analysing the MLS solutions for these three trials in which a randomly selected subset of 5% of the original data was used, it is evident that the it is not only the spacing of the points but more significantly the distribution of them that affects the quality of the fit. Just from analysing the plots alone, Figure 15 shows that the features of the original data are included in the MLS solution, such as the steep changes in elevation at the sides of the crater as well as the central peak. Figure 14 shows a slightly less successful MLS solution in this regard, since much of the data appears skewed at some points such as at $x = 0$ and $x = 0.45$, suggesting that there were a disproportionate number of data points from such regions in the random subset.

However, the solution in Figure 16 shows beyond any doubt the effect of a skewed distribution of data points, since the deviation of the MLS solution is great in the region about $x = 0.3$, which resulted in $||h(x_i) - h_i||_2 = 1.1380 \times 10^1$ for this trial. Comparison of $||h(x_i) - h_i||_2$ for each of the three trials is numerical evidence of the quality of the fits in each trial, since Trial 2 had the lowest value, followed by Trial 1 and then Trial 3, but this is expected given the quality (or perhaps lack of quality) of the plots in Figures 14, 15 and 16.

After having investigated the effects of changing n, η and the size and distribution of a subset of data points on the quality of the MLS solution, it would appear that for our given weight function for smaller values of η the distribution of data points selected has a greater impact on the fit than say larger values of η. By the nature of the weight function $\theta(r)$ (see Question 3), for smaller values of η we observe that $\theta(r)$ decreases exponentially quickly, meaning that the most significant contributions in minimising (3) are when $|x - x_i|$ is small i.e. when x_i is close to x. Since we are interested in fitting the solution over the entire domain, ideally we would want a fairly uniform distribution of data points that are not dense in particular intervals and sparse in others. For $\eta \ll 1$, the effect of not having such a distribution is substantial when taking a small random sample, as we previously saw when the subset included only 5% of the original data.

Naturally as well, we would also wish that for any value of η that we include as many data points in our subset as is possible without increasing the operational cost too greatly, in order to achieve a fairer representation of *all* of the data; this is especially true for smaller values of η, where it is imperative to represent the entire domain in our subset so that any sudden changes in depth are incorporated into our MLS solution. But in short, as η is decreased in order to take an appropriate random sample which still produces a good MLS solution we would require that as many data points as is feasibly possible are included in the subset, and that these points are roughly equally spread out across the domain.