3 Least squares method

For approximating the continuous function $y(x)$ defined on the interval $[a,b]$ with an algebraic polynomial $P_n(x;1)=\sum_{i=0}^{n} a_i x^i \in M_n^1$ using least squares method, we choose the constants $a_0, a_1, \ldots, a_n$, which minimize the least squares error $E^C(a_0, a_1, \ldots, a_n)$, where

$$E^C(a_0, a_1, \ldots, a_n) = \int_a^b (y(x) - P_n(x))^2 dx.$$

A necessary condition for real coefficient $a_0, a_1, \ldots, a_n$ that minimize the error $E^C(a_0, a_1, \ldots, a_n)$ is that

$$\frac{\partial E^C}{\partial a_i} = 0, \text{ for each } i = 0, 1, \ldots, n.$$

Thus, one can get the following linear system of equations

$$A_{LSM}^C X = b^C, \tag{3.1}$$

where $X = [a_0, a_1, \ldots, a_n]^T$, A_{LSM}^C is the $(n+1)\times(n+1)$ coefficient matrix and b^C is the $(n+1)\times 1$ column vector. The $(i,j)^{th}$ element of the matrix A_{LSM}^C and i^{th} element of the column vector b^C are given by

$$(A_{LSM}^C)_{i,j} = \int_a^b x^{i+j-2} dx, \quad \text{and}, \quad (b^C)_i = \int_a^b y(x) x^{i-1} dx,$$

respectively. Similarly, in discrete case, for approximating a data set $(x_k, y_k),\ k = 1, 2, \ldots, m$, with an algebraic polynomial $P_n(x;1) \in M_n^1$ using least squares method, we choose the constants $a_0, a_1, \ldots, a_n$ which minimize the least squares error $E^D(a_0, a_1, \ldots, a_n)$, where

$$\begin{aligned} E^D(a_0, a_1, \ldots, a_n) &= \sum_{k=1}^{m} (y_k - P_n(x_k))^2, \\ &= \sum_{k=1}^{m} y_k^2 - 2\sum_{i=0}^{n} a_i (\sum_{k=1}^{m} y_k x_k^i) + 2\sum_{i=0}^{n}\sum_{j=0}^{n} a_i a_j (\sum_{k=1}^{m} x_k^{i+j}). \end{aligned}$$

For E^D to be minimized it is necessary that $\frac{\partial E^D}{\partial a_i} = 0$, for each $i = 0, 1, \ldots, n$. Thus, one can get following linear system of equations

$$A_{LSM}^D X = b^D, \tag{3.2}$$

where A_{LSM}^D is the $(n+1)\times(n+1)$ coefficient matrix and b^D is the $(n+1)\times 1$ column vector. The $(i,j)^{th}$ element of the matrix A_{LSM}^D and i^{th} element of the column vector b^D are given by

$$(A_{LSM}^D)_{i,j} = \sum_{k=1}^{m} x_k^{i+j-2}, \quad \text{and}, \quad (b^D)_i = \sum_{k=1}^{m} y_k x_k^{i-1},$$

respectively.

4 Modified least squares method

In this Section, we proposed a modified least squares method based on a fractional polynomials that belongs to the space M_n^λ. Apply a similar technique which is discussed in Section 3 for any algebraic polynomial $P_n(x;\lambda) = \sum_{i=0}^{n} a_i x^{i\lambda} \in M_n^\lambda$ to approximating the continuous function $y(x)$ defined on the interval $[a,b]$, we choose the constants $a_0, a_1, \ldots, a_n$ which minimize the least squares error $E^C(a_0, a_1, \ldots, a_n; \lambda)$, where

$$E^C(a_0, a_1, \ldots, a_n; \lambda) = \int_a^b (y(x) - P_n(x;\lambda))^2 dx. \tag{4.1}$$

A necessary condition for real coefficient $a_0, a_1, \ldots, a_n$, that minimize the error $E^C(a_0, a_1, \ldots, a_n; \lambda)$ is that

$$\frac{\partial E^C}{\partial a_i} = 0, \text{ for each } i = 0, 1, \ldots, n.$$

Thus, one can get the following linear system of equations

$$A^C_{MLSM} X = d^C, \tag{4.2}$$

where A^C_{MLSM} is the $(n+1)\times(n+1)$ coefficient matrix and d^C is the $(n+1)\times 1$ column vector. The $(i,j)^{th}$ element of the matrix A^C_{MLSM} and i^{th} element of the column vector d^C are given by

$$(A^C_{MLSM})_{i,j} = \int_a^b x^{\lambda_{i-1}+\lambda_{j-1}} dx, \quad \text{and}, \quad (d^C)_i = \int_a^b y(x) x^{\lambda_{i-1}} dx,$$

respectively. Similarly, in discrete case, for approximating the data set $(x_k, y_k),\ k = 1, 2, \ldots, m$, with an algebraic polynomial $P_n(x;\lambda)$ using least squares method, we choose the constants $a_0, a_1, \ldots, a_n$, which minimize the least square error $E^D(a_0, a_1, \ldots, a_n; \lambda)$, where

$$\begin{aligned} E^D(a_0, a_1, \ldots, a_n; \lambda) &= \sum_{k=1}^{m} (y_k - P_n(x_k;\lambda))^2, \\ &= \sum_{k=1}^{m} y_k^2 - 2\sum_{k=1}^{m} P_n(x_k;\lambda) y_k + \sum_{k=1}^{m} (P_n(x_k;\lambda))^2, \\ &= \sum_{k=1}^{m} y_k^2 - 2\sum_{i=0}^{n} a_i \Big(\sum_{k=1}^{m} y_k x_k^{\lambda_i}\Big) + 2\sum_{i=0}^{n}\sum_{j=0}^{n} a_i a_j \Big(\sum_{k=1}^{m} x_k^{\lambda_i+\lambda_j}\Big). \end{aligned} \tag{4.3}$$

For E^D to be minimized it is necessary that $\frac{\partial E^D}{\partial a_i} = 0$, for each $i = 0, 1, \ldots, n$. Thus, one can get following linear system of equations

$$A^D_{MLSM} X = d^D, \tag{4.4}$$

where A^D_{MLSM} is the $(n+1)\times(n+1)$ coefficient matrix and d^D is the $(n+1)\times 1$ column vector. The $(i,j)^{th}$ element of the matrix A^D_{MLSM} and i^{th} element of the column vector d^D are given by

$$(A^D_{MLSM})_{i,j} = \sum_{k=1}^{m} x_k^{\lambda_{i-1}+\lambda_{j-1}}, \quad \text{and}, \quad (d^D)_i = \sum_{k=1}^{m} y_k x_k^{\lambda_{i-1}},$$

respectively. We know that the large value of n, the matrix A^C_{MLSM} and A^D_{MLSM} become ill-conditioned, which causes significant errors in estimating the parameters $a_i, i = 0, 1, \ldots, n$. This difficulty can be avoided if the functions belonging to the space M^λ_n, denoted by $L_i(x;\lambda, W),\ i = 0, 1, \ldots, n$, are so chosen that they are orthogonal with respect to the weight function $W(x)$ over the interval $[a,b]$. In this case the error function in the continuous and discrete case becomes

$$E^C(a_0, a_1, \ldots, a_n; \lambda) = \int_a^b W(x)(y(x) - \sum_{i=0}^{n} a_i L_i(x;\lambda, W))^2 dx, \tag{4.5}$$

and

$$E^D(a_0, a_1, \ldots, a_n; \lambda) = \sum_{k=1}^{m} W(x_k)(y_k - \sum_{i=0}^{n} a_i L_i(x_k;\lambda, W))^2, \tag{4.6}$$

respectively. The necessary condition for real coefficients $a_0, a_1, \ldots, a_n$ which minimize the error gives the normal equations. The normal equations in the continuous and discrete cases are

$$\int_a^b W(x)(y(x) - \sum_{i=0}^{n} a_i L_i(x;\lambda, W)) L_j(x;\lambda, W) dx = 0,\ \ j = 0, 1, \ldots, n, \tag{4.7}$$

and

$$\sum_{k=1}^{m} W(x_k)(y_k - \sum_{i=0}^{n} a_i L_i(x_k;\lambda, W)) L_j(x_k;\lambda, W) = 0,\ \ j = 0, 1, \ldots, n, \tag{4.8}$$

respectively. One can observe from the Equations (4.7) and (4.8) the parameter value can be determined directly. Thus the use of orthogonal functions not only avoids the problem of ill-conditioning but also determines the constants $a_i,\ i=0,1,\dots,n$, directly. As discussed in Section 2, the Müntz-Legendre polynomials $L_i(x;\lambda)\in M_n^\lambda,\ i=0,1,\dots,n$, are orthogonal for the weight function $W(x)=1$ on the interval $[0,1]$. If we consider the following error function in the continuous and discrete case

$$E^C(a_0,a_1,\dots,a_n;\lambda)=\int_0^1 W(x)(y(x)-\sum_{i=0}^n a_iL_i(x;\lambda))^2dx, \tag{4.9}$$

and

$$E^D(a_0,a_1,\dots,a_n;\lambda)=\sum_{k=1}^m W(x_k)(y_k-\sum_{i=0}^n a_iL_i(x_k;\lambda))^2, \tag{4.10}$$

respectively. One can easily observe that if we find the necessary condition for the parameters $a_0,a_1,\dots,a_n$, we get the systems of equations. For finding the value of $a_i,\ i=0,1,\dots,n$, we have to solve the equation system because Müntz-Legendre polynomials are not orthogonal with respect to the weight function $W(x)\neq 1$. Therefore, one may face the problem of ill-conditioning for a large value of n. This problem can be avoided if we generate the orthogonal polynomials with respect to the weight function $W(x)$.

Remark 1. The modified least squares method is based on the fractional polynomials, which depend on the fractional parameter λ. If we put $\lambda=1$ in the Equations (4.2) and (4.4) then we get (3.1) and (3.2) respectively. Therefore, in the modified least squares method, we have one additional degree of freedom compared to the classical least squares method in the form of fractional parameter λ. So, we tune the parameter λ to get the desired results.

Remark 2. Since Müntz-Legendre polynomials are orthogonal with respect to weight function $W(x)=1$. We know that the fractional derivative/integral of the polynomial belonging to space M_n^λ is again in space M_n^λ when λ and fractional order are the same. Hence, while solving the fractional differential equations using the least squares method and avoiding the difficulty due to ill-conditioning, we need polynomials belonging to the space M_n^λ, which are orthogonal to weight function $(b-x)^{-\lambda}$ or $(x-a)^{-\lambda}$. Moreover, usually, in applications, only a part of the given data needs more attention; for example, in some cases, the data may have more accuracy in some regions than in others. In such cases, the weight function indicates where data should be given more importance, and it should be chosen accordingly.

Now, we are going to discuss the Theorem, which helps us to generate the fractional orthogonal polynomials belonging to the space M_n^λ with respect to the weight function $W(x)$.

Theorem 4.1. *The set of fractional polynomial functions $\{L_0(x;\lambda,W),L_1(x;\lambda,W),\dots,L_n(x;\lambda,W)\}$ defined in the following way is orthogonal on $[a,b]$ with respect to the weight function $W(x)$.*

$$L_0(x;\lambda,W)=1,\quad L_1(x;\lambda,W)=x^\lambda-B_1,$$

where

$$B_1=\frac{\int_a^b W(x)x^\lambda(L_0(x;\lambda,W))^2dx}{\int_a^b W(x)(L_0(x;\lambda,W))^2dx},$$

and when $i\geq 2$,

$$L_i(x;\lambda,W)=(x^\lambda-B_i)L_{i-1}(x;\lambda,W)-C_iL_{i-2}(x;\lambda,W),$$

where

$$B_i=\frac{\int_a^b W(x)x^\lambda(L_{i-1}(x;\lambda,W))^2dx}{\int_a^b W(x)(L_{i-1}(x;\lambda,W))^2dx},\quad C_i=\frac{\int_a^b W(x)x^\lambda L_{i-1}(x;\lambda,W)L_{i-2}(x;\lambda,W)dx}{\int_a^b W(x)(L_{i-2}(x;\lambda,W))^2dx}.$$

Proof. We prove that the above result holds by mathematical induction; firstly, we will consider $L_0(x;\lambda,W)$ and $L_1(x;\lambda,W)$ and show that they are orthogonal.

$$\begin{aligned}\int_a^b W(x)L_0(x;\lambda,W)L_1(x;\lambda,W)dx &= \int_a^b W(x)(x^\lambda-B_1)dx\\ &= \int_a^b W(x)x^\lambda dx - B_1\int_a^b W(x)dx\\ &= 0.\end{aligned}$$