

SAYISAL ARAZİ MODELLERİNDE

EĞRİ YERLEŞTİRME YÖNTEMLERİ

YÜKSEK LİSANS TEZİ

18824

Müh. Ergin TARI

Tezin Enstitüye Verildiği Tarih: 18 HAZİRAN 1991

Tezin Savunulduğu Tarih : 2 TEMMUZ 1991

Tez Danışmanı

: Prof.Dr. Orhan
BAYKAL

Diğer Jüri Üyeleri

: Prof.Dr. Olcay
ÖZTAN

Doç.Dr. Oğuz
MÜFTÜOĞLU

Y. C.

TEMMUZ 1991

Yükseköğretim Kurulu
Dokümantasyon Merkezi



Çalışmamın gerçekleşmesinde, tez yöneticiliğimi üstlenen Sayın Prof.Dr. Orhan BAYKAL'a, değerli katkıları için Sayın Prof.Dr. Olcay ÖZTAN'a, tezin yazımındaki titiz çalışması için Sayın Zübeyde KÖSE'ye ve yardımları için değerli çalışma arkadaşlarıma teşekkürü bir borç bilirim.

Ergin TARI

İÇİNDEKİLER

ÖZET	v
SUMMARY	vi
BÖLÜM 1. GİRİŞ	1
BÖLÜM 2. SAYISAL ARAZİ MODELİ (SAM) KAVRAMI	3
2.1. Sayısal Arazi Modelinin Oluşturulması İçin Yöntemler	4
2.1.1. Raster (Ağ) Yöntemi	4
2.1.2. Üçgenleme Yöntemi	5
2.2. Sayısal Arazi Modelinin Oluşturulma Evreleri	7
2.2.1. SAM İçin Verilerin Toplanması	7
2.2.2. SAM Verilerinin İşlenmesi	8
BÖLÜM 3. (Veri) Sıralama	9
3.1. Sayısal Arazi Modelinde Sıralamanın Yeri	9
3.2. Genel Esaslar	10
3.3. Sıralama Algoritmalarının Ortak Özellikleri	11
3.4. Sıralama Algoritmaları	12
3.4.1. Damlacık (Bubble) Sıralaması	12
3.4.2. Seçme Sıralaması	14
3.4.3. Araya Sokma Sıralaması	15
3.4.4. Shell Sıralaması	16
3.4.5. Yığın (Heap) Sıralaması	17
3.4.6. Hızlı Sıralama (Quicksort)	27
3.4.7. Sıralama Algoritmalarının Karşılaştırılması	28
3.4.8. Bir Sıralama Programı ve Bazı Araştırmalar	29
3.4.8.1. Bir Parçalı Yöntem	31
3.4.8.2. İki Parçalı Yöntem	32
3.4.8.3. Uygulamalar	34
BÖLÜM 4. EĞRİ YERLEŞTİRME	39

4.1. En Küçük Kareler Yöntemi	40
4.1.1. Giriş	40
4.1.2. Dengeleyen Doğru	43
4.1.3. Dengeleyen Eğri	47
4.1.4. Veri Doğrusallaştırma Yöntemi	48
4.1.4.1. Veri Doğrusallaştırma İçin Dönüşümler	51
4.1.5. Dengeleyen Eğri Olarak Polinom	56
4.1.5.1. Polinom Salınımı	58
4.2. Spline Fonksiyonları İle Enterpolasyon	61
4.2.1. Parça Parça Doğrusal Enterpolasyon	61
4.2.2. Parça Parça Kübik Spline Fonksiyonları	63
4.2.3. Kübik Spline'in Varlığı	65
4.2.2.2. Kübik Splineların Oluşturulması	68
4.2.2.3. Kübik Splineların Tanımlanması	70
4.3. Fourier Serileri ve Trigonometrik Polinomlar	77
4.3.1. Fourier Serileri	77
4.3.2. Trigonometrik Polinomlar	83
BÖLÜM 5. SONUÇ VE ÖNERİLER	87
KAYNAKLAR	91
ÖZGEÇMİŞ	93

ÖZET

Sayısal Arazi Modeli, arazi üzerinde dağılmış koordinatları ve yükseklikleri bilinen noktalardan yararlanarak bilgisayar yardımı ile arazinin sayısal olarak gösterilmesidir. Sayısal arazi modeli, sayısal durum modeli ve sayısal yükseklik modelinden oluşur. Bir çok çalışmaya altlık oluşturan sayısal arazi modelinin etkin bir şekilde oluşturulması gerekir. Sayısal arazi modelinin oluşturulmasında kullanılan yazılım bu etkinliği sağlayan temel unsurdur. Yazılımın etkinliğini ise bellek ve hız parametrelerinin uygun kullanımı sağlar. Hız parametresini etkileyen en önemli faktör de bilgilere ulaşımıdır. Bilgilere ulaşımın en hızlı şekilde yapılabilmesi ise bu bilgilerin belli bir düzende sıralanmış olmasına bağlıdır. Sayısal arazi modeli gibi bilgileri tekrar tekrar sıralamayı gerektiren uygulamalarda ise sıralama işleminin bellek ve hız optimizasyonunu sağlaması önemlidir.

Sayısal yükseklik modelinde enterpolasyon sonucu elde edilen aynı kotlu noktaların birleştirilmesi işlemi bir eğri yerleştirme işlemidir. Eğri yerleştirme, en küçük kareler yöntemi ile 2. dereceden bir polinom, 3. ve daha yüksek dereceli bir polinom veya trigonometrik polinomlar kullanılarak gerçekleştirilebilir. Polinomlar ile eğri yerleştirmenin polinom salınımı adı verilen bir sakıncası Spline Fonksiyonlarının kullanılmasını gerektirir. Spline fonksiyonlarında da temelde polinom yaklaşımı kullanılır. Ardışık noktalar arasında oluşturulan grafik parçalarını birleştirmek temelinden yola çıkarak ortaya konan spline fonksiyonlarının en çok kullanılanı kübik spline fonksiyonlarıdır.

Önemi nedeni ile burada sıralama ve eğri yerleştirme yöntemleri üzerinde ayrıntılı bir biçimde durulmuştur. Literatürde verilen sıralama yöntemleri hız açısından karşılaştırılmış hangisinin Sayısal Arazi modeli için uygun olacağı araştırılmış ayrıca sıralama için bir yöntem verilmiştir. Sayısal yükseklik modelinde enterpolasyon sonucu elde edilen aynı kotlu noktalardan dengeleyen polinomların geçirilmesi sakıncası bir örnek ile gösterilmiştir.

SUMMARY

CURVE FITTING METHODS IN DIGITAL TERRAIN MODELS

Digital terrain model is a digital representation of the terrain, based on measurements on the reference points by means of a detailed computer program series named software.

Digital terrain model software which serves for a wide range engineering area must be planned for using optimum computer memory and providing maximum speed. Using computer memory is the subject of modern memory storing methods. Providing maximum speed depends on the form of arranging of the knowledge. In our study knowledge is generally in the form of numbers which are x,y planimetric coordinates or h coordinates.

The best way to get over the problem of loosing time is arranging the knowledge in a desired order. This arranging procedure is called sorting. If an element in an unsorted list of n elements is searched, the maximum number of comparisons is n. When a sorted list of n elements is searched for an element, the maximum number of comparisons in a searching method called "binary search" is, (k is the number of comparison)

$$k = \log_2 n$$

This number k is very attractive and always desired. Sorting must be managed properly in order to make searching so fast. In the studies, which require sorting many times, like digital terrain model the speed of sorting is very important. There are some algorithms for making faster sort. They can be written as follows,

1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Shell Sort
5. Heap Sort
6. Quick Sort

1- Bubble Sort

This algorithm is the best known and the most infamous sorting algorithm. Its popularity is derived from its catchy name and its simplicity. The bubble sort algorithm uses the exchange method of sorting. The general concept behind the bubble sort is the repeated comparisons and, if necessary, exchanges of adjacent elements.

2- Selection Sort

In this algorithm, the element with the lowest value is selected and exchanged with the first element. Then from the remaining elements, the element with the least key is found and exchanged with the second element, and so forth, up to the last two elements.

3- Insertion Sort

This algorithm initially sorts the first two members of the array. Next, the algorithm inserts the third member into its suitable position in relation to the first two members. Then, the fourth element is inserted into the list of three elements. And the process continues until all elements have been sorted.

4- Shell Sort

This algorithm, derived from insertion sort, is based on diminishing increments. In this algorithm, all elements that are $n/2$ positions (n is the number of elements of array) apart are sorted. Then all elements that are $n/4$ positions apart from the first step are sorted. The process continues and finally, all those adjacent to each other (1 positions apart) are sorted.

5- Heap Sort

First we have to define what a heap is. If an array a , given with n elements, satisfies the relation

$$a_{j/2} > a_j \quad \text{for } 1 < j/2 < j < N$$

then this array is said "heap". If rearranging the array into an order that forms a heap is managed, then sorting becomes very easy. The top of the heap, which is the largest element yet unsorted is pulled off. Then its largest underling is promoted to the top of the heap. Next its largest underling is promoted, and so on.

6- Quick Sort

This algorithm is the best sorting algorithm currently available. Algorithm's basic idea is partition.

The general procedure is to select a value and then to partition the array into two parts with all elements greater than or equal to the partition value on one side and those less than the partition value on the other. This process is then repeated for each remaining part until the array is sorted.

In digital terrain models, after the interpolation in digital elevation model, interpolation of curves is considered. In general, interpolation of curves is said curve fitting. The best known curve fitting method is the method of least-squares curve fitting. This method can be extended to many nonlinear cases. These cases can be written as exponential fitting and polynomial fitting.

The exponential fitting requires the data to fit an exponential function like $y=C.\exp (Ax)$. But this is not an efficient way for contouring which does not fit in any function. A product of polynomial fitting, which is called spline function, have the solution of the problem of polynomial wiggle.

Polynomial wiggle becomes an important constraint not to use a polynomial of degree 6 or above unless it is known that the true function, which is studied, is a polynomial. A polynomial of degree N can have $N-1$ relative maxima and minima and the graph can wiggle in order to pass through the points. Another method is to "piece together" the graphs of lower degree polynomials $S_k(x)$ and interpolate between the successive nodes (x_k, y_k) and (x_{k+1}, y_{k+1}) . (Figure 1) The reason of using lower degree polynomials is preventing the polynomial wiggle.

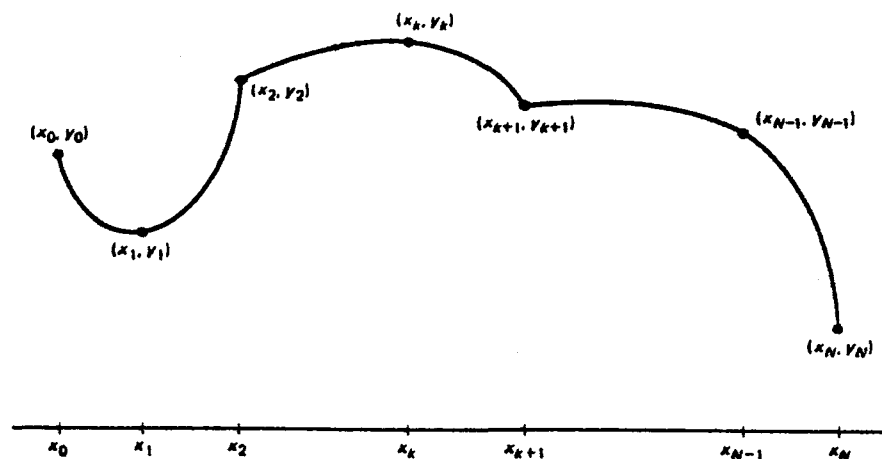


Figure 1. Piecewise Polynomial Interpolation

The two adjacent portions of the curve $y=S_k(x)$ and $y=S_{k+1}(x)$, which lie above $|x_k, x_{k+1}|$ and, $|x_{k+1}, x_{k+2}|$, respectively, pass through the common knot (x_{k+1}, y_{k+1}) . The two portions of the graph are "tied together" at the knot (x_{k+1}, y_{k+1}) and the set of functions $\{S_k(x)\}$ form a piecewise polynomial curve which is denoted by $S(x)$.

Mathematically, it is possible to construct a cubic functions $S_k(x)$ on each interval $|x_k, x_{k+1}|$ so that the resulting piecewise curve $y=S(x)$ and its first and second derivatives are all continuous on the larger interval $|x_0, x_N|$. The continuity of $S'(x)$ means that the graph $Y=S(x)$ will not have sharp corners. The continuity of $S''(x)$ means that the "radius of curvature" is defined at each point.

Let us consider the $N+1$ points $\{S(x_k, y_k)\}$ where the abscissas are ordered $x_0 < x_1 < \dots < x_N$. The function $S(x)$ is called a cubic spline if there exist N cubic polynomials $S_k(x)$ such that:

$$\begin{aligned} \text{I- } S(x) = S_k(x) = & S_{k,0} + S_{k,1}(x-x_k) + S_{k,2}(x-x_k)^2 + \\ & + S_{k,3}(x-x_k)^3 \quad \text{for } x \text{ in } |x_k, x_{k+1}| \text{ and} \\ & k=0, 1, \dots, N-1 \end{aligned}$$

$$\text{II- } S(x_k) = y_k \quad \text{for } k=0, 1, \dots, N$$

The spline passes through each data point.

$$\text{III- } S_k(x_{k+1}) = S'_{k+1}(x_{k+1}) \quad \text{for } k=0, 1, \dots, N-2.$$

The spline forms a continuous function

$$\text{IV- } S'_k(x_{k+1}) = S''_{k+1}(x_{k+1}) \quad \text{for } k=0, 1, \dots, N-2$$

The spline forms a smooth function.

$$\text{V- } S''_k(x_{k+1}) = S''_{k+1}(x_{k+1}) \quad \text{for } k=0, 1, \dots, N-2.$$

The second derivative is continuous.

Since $S(x)$ is piecewise cubic, its second derivative $S''(x_N)$ is piecewise linear on $|x_0, x_N|$. The linear lagrange interpolation formula gives the following representation for $S''(x) = S''_k(x)$:

$$S''_k(x) = S''(x_k) \frac{x - x_{k+1}}{x_k - x_{k+1}} + S''(x_{k+1}) \frac{x - x_k}{x_{k+1} - x_k}$$

Using $m_k = S''(x_k)$, $m_{k+1} = S''(x_{k+1})$ and $h_k = x_{k+1} - x_k$ in this equation yields,

$$S''_k(x) = \frac{m_k}{h_k} (x_{k+1} - x) + \frac{m_{k+1}}{h_k} (x - x_k)$$

for $x_k \leq x \leq x_{k+1}$ and $k=0,1,\dots,N-1$. Integrating last equation twice will give two constants of integration, and the result can be manipulated so that it has the form,

$$S_k(x) = \frac{m_k}{6h_k} (x_{k+1} - x)^3 + \frac{m_{k+1}}{6h_k} (x - x_k)^3 + p_k (x_{k+1} - x) + q_k (x - x_k)$$

Substituting x_k and x_{k+1} into equation and using the values $y_k = S_k(x_k)$ and $y_{k+1} = S_k(x_{k+1})$ yields the following equations that involve p_k and q_k , respectively:

$$y_k = \frac{m_k}{6} h_k^2 + p_k h_k \quad \text{and} \quad y_{k+1} = \frac{m_{k+1}}{6} h_k^2 + q_k h_k$$

Solving these two equations for p_k and q_k results in the following expression for the cubic function $S_k(x)$:

$$S_k(x) = \frac{m_k}{6h_k} (x_{k+1} - x)^3 + \frac{m_{k+1}}{6h_k} (x - x_k)^3 + \left(\frac{y_k}{h_k} - \frac{m_k h_k}{6} \right) (x_{k+1} - x) + \left(\frac{y_{k+1}}{h_k} - \frac{m_{k+1} h_k}{6} \right) (x - x_k)$$

Using the derivatives of these equation, an equation involving the unknown coefficients m_{k-1} , m_k , m_{k+1} can be written as follows:

$$h_{k-1} m_{k-1} + 2(h_{k-1} - h_k) m_k + h_k m_{k+1} = u_k$$

$$\text{where } U_k = 6(d_k - d_{k-1})$$

$$\text{for } k=1,2,\dots,N-1.$$

After the coefficients $\{m_k\}$ are determined by using above equation and endpoint constraints, the spline coefficients $\{S_{k,j}\}$ for $S_k(x)$ are computed using the formulas:

$$S_{k,0} = y_k, \quad S_{k,1} = d_k - \frac{h_k (2m_k + m_{k+1})}{6}$$

$$S_{k,2} = \frac{m_k}{2}, \quad S_{k,3} = \frac{m_{k+1} - m_k}{6h_k}$$

Each cubic polynomial $S_k(x)$ can be written in nested multiplication from:

$$S_k(x) = ((S_{k,3}w + S_{k,2})w - S_{k,1})w + y_k \quad \text{where } w = x - x_k$$

and $S_k(x)$ is used on the interval $x_k \leq x \leq x_{k+1}$.



BÖLÜM 1

1- GİRİŞ

Bilimsel ve teknik gelişmelerin hızla devam ettiği yüzyılımızın ikinci yarısında özellikle bilgisayarların devreye girmesi ile bir çok alanda olduğu gibi Jeodezi ve Fotogrametri biliminde de büyük bir gelişme gözlenmektedir. Bu gelişmeler; yapılan işlemlerde hızın artması, kullanılan hacim ve/veya hacimlerin azalması (ciltler halinde ölçme çizelgeleri yerine disket vb), estetik görünümün güzelleşmesi, işgücü tasarrufu sağlanması gibi bir çok değişik şekilde karşımıza çıkmaktadır. Jeodezi ve Fotogrametri alanında bu gelişmelerin çoğu aynı anda etkili olmaktadır. Bu gelişmeler; çizim, hesap ve ölçme işlerine ayrı ayrı hız kazandırmakta, kırtasiyeyi azaltmakta, çizim için gerekli zamanı kısalttığı için estetik görünümü artırmakta ve göreceli olarak büyük işgücü tasarrufu sağlamakta, maliyeti düşürmektedir. Bütün bu işlemler sırasında klasik yöntemlerden büyük değişiklikler ortaya çıkmaktadır. Bu değişiklikler ise bilgisayardan maksimum yararı sağlayacak şekilde planlanmakta ve uygulanmaktadır. Konunun en önemli noktası bilgisayardan maksimum yararı sağlayacak şekilde planlamayı en üst seviyeye çıkarması gereken yazılım (software) olmaktadır. Yazılım yukarıda sayılan yararları sağlayabilmesi için genellikle klasik hesaplama mantıklarından ayrılmakta ve bilgisayarın sahip olduğu hesaplama hız ve kapasite ile uyumlu olarak planlanmak zorundadır. Bu nedenle sürekli yeni hesaplama algoritmaları araştırılmakta ve uygulama sonuçları ortaya konmaktadır.

Bilgisayarın Jeodezi ve Fotogrametri biliminde kullanıldığı uygulama alanlarından birisi de Sayısal Arazi Modeli (S.A.M)'dir. Arazide var olan belirli sayıda dayanak noktalarına bağlı olarak kullanılan bir enterpolasyon yöntemi ile arazinin sayısal gösterimi şeklinde tanımlanan sayısal arazi modeli bu çalışmanın temel konusudur. Sayısal arazi modelinin oluşturulması sırasında ortaya çıkan işlemlerin büyüklüğü karşısında bilgisayarın sınırlı olan bellek ve hız kapasitesi sorununu aşmak için bilgisayarın optimum bir şekilde kullanılması, yani verilerin sıralanması, büyük lineer denklem sistemlerinin çözümünde modern bellek depolama yöntemlerinin kullanılması gerekir.

Sınırlı sayıda dayanak noktasından yararlanarak uygun bir enterpolasyon işlemi ile istenen diğer noktaların konumlarının belirtilmesi ve bunun uzantısı olarak eş yükselti eğrilerinin çizimi Sayısal Arazi Modeli'nin önemli bir problemini oluşturur.

Yukarıda belirtilen problemler için ayrı ayrı hesap yöntemleri ortaya konmuştur. Bu hesap yöntemleri, klasik hesaplama tekniklerine göre bilgisayar tarafından en iyi şekilde uygulanabilmesi için büyük farklılıklara uğramaktadır.

BÖLÜM 2

2- SAYISAL ARAZİ MODELİ KAVRAMI

İlk olarak 1955 yılında Massaschusetts Teknoloji Enstitüsü tarafından ortaya atılan SAM kavramı bugün bilgisayar teknolojisinin ulaştığı noktaya paralel olarak büyük gelişme göstermektedir [1]. Başlangıçta sadece karayolu projelerinin bazı evrelerini otomasyona sokmak için ortaya atılan düşünce, günümüzde çok geniş bir uygulayıcı kesimi tarafından kullanılmaktadır. Kullanım alanları olarak sanayii, tıp, mimarlık, inşaat, madencilik, ziraat, harita mühendisliği ve askeri hizmetler sayılabilir [2].

SAM, arazi üzerinde dağılmış koordinatları ve yükseklikleri bilinen noktalar yardımı ile bilgisayar işlemlerinden yararlanarak arazinin sayısal olarak gösterilmesidir [1]. Bu gösterim ise bir çok çalışmaya altlık oluşturur. Bu çalışmalar arasında topoğrafik haritaların yapımı, geçkilerin tasarımı, en ve boy kesitlerin çizimi, hacim belirlenmesi, baraj, havaalanı, kanal tasarımı, erozyonun önlenmesi, mikrodalga sistemleri, iletişim hatlarının oluşturulması vardır [2],[3],[4]. Böyle geniş bir uygulama alanı olan çalışmanın kullanıma getirilmesinde en önemli eleman bilgisayar programlarıdır. Kullanıcı için gerekli verinin doğru ve hızlı bir şekilde hazırlanması SAM'in amacıdır.

SAM geniş anlamda, ayrıntı noktalarına ait konum ve yükseklik bilgilerinin her ikisini, dar anlamda ise yalnızca yükseklik bilgilerini içerir. Diğer bir deyişle

SAM geniş anlamda, sayısal durum modeli (S.D.M) ve sayısal yükseklik modelinin (S.Y.M) her ikisini birden, dar anlamda ise sadece sayısal yükseklik modelini içerir [5], [6].

2.1- Sayısal Arazi Modelinin Oluşturulması için Yöntemler

Sayısal arazi modeli iki yöntemle oluşturulur.

1. Raster (ağ) Yöntemi
2. Üçgenleme Yöntemi

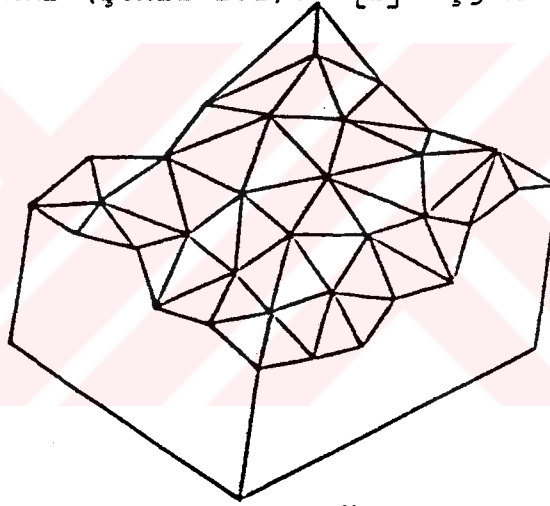
2.1.1- Raster (Ağ) Yöntemi

Bir arazi parçası için sayısal arazi modelinin elde edilebilmesi, bu arazi parçası üzerinde dağılmış koordinatları bilinen noktaların varolmasını gerektirir. Koordinatları bilinen bu dayanak noktaları rastlantısal olduğu gibi düzgün bir kareler veya dikdörtgenler ağıının kesişme noktaları olarak seçilebilir. Arazi parçası üzerinde dağılmış olan bu dayanak noktaları yersel ölçmelerle doğrudan doğruya belirlenebildikleri gibi topoğrafik harita ve planlardan veya fotogrametrik olarak hava fotoğraflarından dolaylı olarak da belirlenebilir. Bu durumda raster (ağ) yöntemi uygulanır. Yersel ölçmelerle veriler, arazide doğrudan elde edilmişse doğruluk yüksek olmasına karşın harcanan zaman fazladır. Rastlantısal olarak yapılan bu arazi ölçmelerinden, araziye kapsayacak şekilde oluşturulan kareler veya dikdörtgenler ağı düğüm noktalarının yükseklikleri uygun bir enterpolasyon yöntemi kullanılarak da bulunabilir [2]. Böylece SAM'da "Raster Yöntemi" kullanılabilir. Bu durumda presizyon kaybı olduğu gözönüne alınmalıdır. Ağ gözleri (gridler) ne kadar küçük olursa araziye temsil etmedeki doğruluk da o kadar büyük olur [2].

Düşüm noktalarının hesaplanmasında kullanılan dayanak noktalarının, yeterli yoğunluk ve doğruluğa sahip olması, yani yeterince araziye temsil etmeleri bütün yöntemlerde olduğu gibi bu yöntem için de gereklidir. Bu yöntem, ekonomik olması nedeni ile verilerin topoğrafik harita ve planlardan veya fotogrametrik olarak hava fotoğraflarından elde edilmesi durumunda önerilir.

2.1.2- Üçgenleme Yöntemi

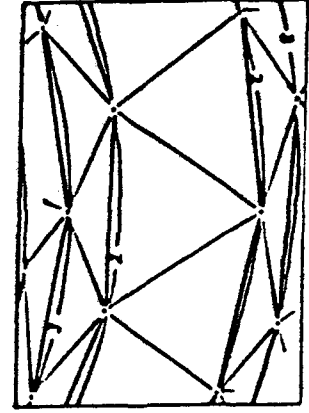
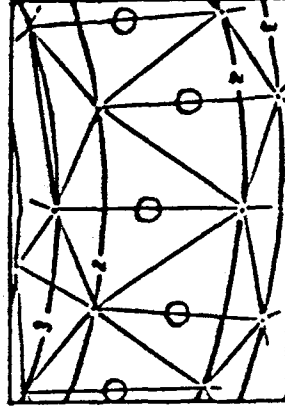
Bu yöntemde arazi, rastlantısal veya düzgün olarak dağılmış olan dayanak noktalarının birleştirilmesi ile düzlem üçgenlerden oluşan çok yüzlü (polihedron) bir yüzle kaplanır (Şekil 2.1). [2] Üçgenleme yönteminin



Şekil 2.1. Düzlem Üçgenlerle Arazi Yüzeyi (Polihedron)

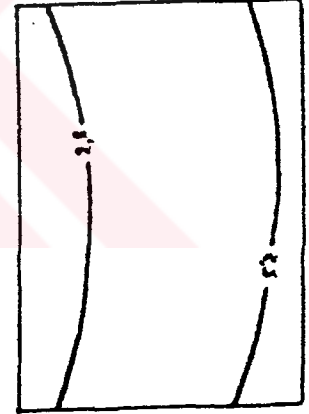
uygulandığı ilk çalışmalarda su toplama çizgisi, su dağıtma çizgisi, şev kenarı, yol gibi arazide bulunan yapay ve doğal çizgiler üzerinde bulunan dayanak noktalarının üçgenleme işleminde kullanılmasına bir öncelik verilmemiş tüm noktaların aynı öncelikte olduğu varsayımına dayanılmıştır. Böyle bir üçgenleme arazi yapısını gerçek anlamda gösteren çizgiler ile üçgen kenarlarının kesişmesi gibi arazinin doğru temsil edilmesini engelleyen bir sorun doğurur. Bunun için arazinin yapısını gösteren karakteristik çizgilerin (dere, su toplama-dağıtma çizgisi, yol vb) üçgenlerin bir kenarını oluşturması sağlanır [2].

Bu duruma bir örnek Şekil 2.2, 2.3, 2.4 ve 2.5 te görülmektedir.



Şekil 2.2. Yanlış Üçgenleme

Şekil 2.3. Doğru Üçgenleme



Şekil 2.4. (Şekil 2.2) den Enterpole Edilmiş 2,5 m Değerli Eşyükseklik Eğrileri.

Şekil 2.5. (Şekil 2.3) den Enterpole Edilmiş 2,5 m Değerli Eşyükseklik Eğrileri.

Üçgenleme yöntemi ile oluşturulan SAM aşağıdaki koşulları sağlamalıdır [2].

- 1- Her dayanak noktası en az bir üçgende yerelmalıdır.
- 2- Arazi yüzeyini örten üçgenlerin birbiri ile ara kesiti sıfır olmalıdır.

3- Ağı oluşturan kenarların uzunluklarının toplamı minimum olmalıdır.

SAM oluşturulması için yöntemlerin karşılaştırılması [6]'da ayrıntılı olarak incelenmektedir.

2.2- SAM'in Oluşturulma Evreleri

Sayısal arazi modelinin oluşumu başlıca iki evrede gerçekleşir. Bunlar

- 1- SAM için verilerin toplanması
- 2- SAM'de verilerin işlenmesi

2.2.1- SAM için Verilerin Toplanması

SAM verileri, yersel ölçmelerle araziden, kartografik sayısallaştırma ile topografik haritalardan, fotogrametrik ölçmeler ile fotoğraflardan elde edilir. SAM'in iskeletini oluşturan verilerin toplanması işlemine örnek-leme adı verilir. SAM için örnekleme evresi önemli bir evredir ve amaca uygun olmayan bir örneklemenin yaratacağı eksiklikleri hiçbir enterpolasyon yöntemi gideremeyeceği gibi örneklemenin yoğunluğu ve dağılım biçimi SAM'in doğruluğunu doğrudan etkiler [3].

SAM verilerinin toplanması sırasında kullanılan ileri teknoloji ürünleri, verilerin SAM tarafından kullanılacak şekle getirilmesinde büyük kolaylıklar sağlamaktadır. Bunlar arasında doğrudan X,Y,Z 'i çıkış olarak veren elektronik takeometreler, bunları kaydeden veri toplayıcılar (data recorder), büyük doğruluklara ulaşan uydu görüntüleri ve sayısallaştırmayı daha doğru, hızlı ve basit kılan elektronik sayısallaştırıcılar (digitizer) ilk akla gelenlerdir.

2.2.2- SAM Verilerinin İşlenmesi

Dayanak noktalarından yararlanarak ağın noktalarının yükseklikleri çeşitli enterpolasyon yöntemlerinden biri ile hesaplanır. Böylelikle arazi sayısal olarak belirlenmiş olur [1].

Sayısal yükseklik modelinde (S.Y.M) yüksekliklerin belirlenmesi işlemi, bilgisayar programı ile seçilen enterpolasyon yöntemi ya da yöntemleri kullanılarak yapılır.

SAM verilerinin işlenmesi uzun süre gerektiren bir evredir. SAM'ni oluşturmak için verilerin işlenmesi aşamasındaki bu süreyi kısaltmak için çeşitli yöntemler kullanılır. Bu yöntemlere örnek olarak, verilerin sıralanması, bu sıralama için en hızlı ve bilgisayar belleğinde en az yer işgal eden yöntemin seçilmesi, ortaya çıkacak büyük denklem takımlarının çözümünde, bilgisayar belleğinde en az yer tutmak ve en yüksek hıza ulaşmak için modern bellek depolama yöntemlerinin kullanılması sayılabilir. Ayrıca SAM için düzenlenecek programların hız ve bellek açısından optimize edilmesi ve gereksiz yere yüksek presizyonlu değişken kullanımından kaçınılmasına dikkat edilmesi gerekir.

SAM verilerinin SYM'de değerlendirilmesi sonucunda bu verilerin çeşitli matematiksel algoritmalar yardımı ile SDM'ne aktarılması gerekir. Bu aktarım ise ağırlıklı eş yükselti eğrilerinin geçirilmesi problemini kapsar. Bu eğrilerin geçirilmesi, çeşitli enterpolasyon yöntemleri ile işlenen SYM verilerinden yararlanarak gerçekleştirilir. Eğri geçirme ile ilgili geniş bilgi bölüm 4'te verilmektedir.

BÖLÜM 3

3- SIRALAMA

3.1- Sayısal Arazi Modelinde Sıralamanın Yeri

Sayısal arazi modelinin temel ögesi olan yazılımın optimum bir şekilde amacına ulaşması bellek ve hız parametrelerinin etkileşimli kullanılması ile gerçekleşir. Bellek kazanımı, lineer denklem sistemlerinin çözümünde modern bellek depolama yöntemlerinin kullanımı ile sağlanır. Hızın artırılması daha çok bilgiye ulaşılması ve bilginin işleme sokulması prosesleri ile ilgilidir. Bilgiye ulaşımın en hızlı şekilde yapılabilmesi ise bilginin dizilişi ile yakından ilgilidir. Herhangi bir sırasız dizi için bilgiye ulaşım dizinin eleman sayısı kadar karşılaştırma gerektirirken, sıralı bir dizi için, n dizinin eleman sayısı, k karşılaştırma sayısı olmak üzere, "ikili arama" olarak bilinen yöntemde,

$$n=2^k \quad (3.1)$$

$$k=\log_2 n \quad (3.2)$$

ile verilmiştir [7]. Burada k maksimum karşılaştırma sayısıdır. Görüldüğü gibi bilginin hızlı bir şekilde kullanıma alınabilmesi kesin olarak bilgilerin sıralanması işlemi ile ilgilidir. Bilgilerin SAM'da kullanılması sırasında bazı aşamalarda değişik elemanların farklı parametrelere göre tekrar tekrar sıralanması gerekir. Örnek olarak bir bölge içindeki noktaların hızla belirlenebilmesi için x ve y koordinatlarından oluşan dizilerin, ilgili bazı bilgilerin de beraberinde düzenlenmesi gözönüne

alınarak sıralanması gereği verilebilir. Ancak bellekte, sıralanan dizi elemanlarının sıralanmış şekli ile değil de sadece ve sadece dizinin indislerinin (program numaralarının) depolandığına ve sıralanmış diziye ilişkin bilgilerin de bu indisler yardımı ile bulunduğuna dikkat etmelidir. Böylece bellekten büyük yer kazancı sağlanmış ve programa büyük hız verilmiş olur.

Sıralamada kullanılan yöntemin hızı doğrudan doğruya yazılımın da hızını etkiler. Bu nedenle burada sıralama yöntemlerine ayrıntılı olarak girilecek, düzenlenen bir sıralama programı tanıtılacak, bir programın ne kadarlık bir hız kazanımı sağlayacağı üzerine yapılan araştırma ve sonuçları tablolar ve grafikler biçiminde verilecektir.

3.2- Genel Esaslar

Sıralama, kısaca benzer bilgilerden oluşan bir dizinin artan ya da azalan sırada düzenlenmesi işlemidir [7]. Yani n sayıdaki a_i elemanının,

$$a_1 \leq a_2 \leq \dots \leq a_n \quad (3.3)$$

veya

$$a_1 \geq a_2 \geq \dots \geq a_n \quad (3.4)$$

şekline getirilmesidir.

Sıralama algoritmaları genel olarak üç sınıfa ayrılırlar [7]. Bunlar

- Yer değiştirme
- Seçme
- Araya sokma

dır.

Bu yöntemleri anlayabilmek için bir masa üzerindeki kartlar dizisi gözönüne alınır. Yer değiştirme (diğer bir ismi değış-tokuş) yönteminde, masa üzerindeki kartların sıraya konulmasında bir kart belli bir düzen içinde diğeri ile yer değıştirilir ve böylece masa üzerindeki bütün kartlar sıralı duruma gelinceye kadar işleme devam edilir. Değış-tokuş yönteminin temeli dizinin elemanlarının yerlerini değıştirerek sıralamayı gerçekleştirmektir.

Seçme yönteminde ise en küçük kart sonra da bu karttan bir büyük kart ele alınır. Böylece işleme devam edilir ve kartlar sıralanmış olur. Bu yöntem en azından sıralama işlemi bitene kadar dizinin sıralanmış biçimi için bellekte ikinci bir vektörü gerektirir.

Araya sokma yöntemi ile sıralamada ise masanın üzerindeki her kartı birinciden başlayarak sırası ile ele alıp öncekilerine göre yerleştirmeyi gerektirir. Görüldüğü gibi araya sokma yöntemi de ek bir bellek gerektirmektedir.

3.3- Sıralama Algoritmalarının Ortak Özellikleri

Sözkonusu edilen her üç yöntem içinde birçok değışik algoritma geliştirilmiştir. Her algoritmanın kendine has iyi ve kötü yanları vardır. Bu yöntemler için aşağıdaki özellikler verilebilir.

- 1- Normal durumdaki bir diziyi sıralama hızı
- 2- En iyi ve en kötü durumdaki bir diziyi sıralama hızı
- 3- Sıralama prosesinin, dizi elemanlarının diziliş şeklinden bağımsız olması
- 4- Dizinin dizilişindeki özel durumdan yararlanma

Bu özellikler aşağıda kısaca açıklanmaktadır.

1- Sıralanacak dizinin hiçbir özel durum taşımaması halinde algoritmanın hızının ne olacağı önemli bir özelliktir ve algoritmanın iyi ve kötü nitelendirilmesine esas olan bir ölçüttür. Özel durumlar, dizinin bir kısmının veya tamamının sıralı veya tersinden sıralı olmasıdır.

2- En iyi ve en kötü durum: Sıralama için kullanılan bazı algoritmaların hızı, dizinin sıralanmadan önceki durumları ile çok ilgilidir. Normal bir durum için çok iyi bir performans gösteren algoritma, özel bir durumda çok kötü bir performans gösterebilir. Örnek olarak sıralı bir diziyi Quicksort algoritmasının bazı uyarlamalarının inanılmaz ölçüde uzun bir sürede çözmesini gösterebiliriz [8]. Hatta bu durumla karşılaşmamak için dizi önceden rastgele bir diziliş durumuna getirilir.

3- Algoritmanın işleyiş sırasında dizinin durumu ne olursa olsun algoritmanın aynı sonuca aynı yolla ulaşabilmesi durumunu ifade eder.

4- Algoritmanın, sıralama prosesi sırasında dizinin özel durumu varsa bu durumdan doğal olarak yararlanmayı ifade eder.

3.4- Sıralama Algoritmaları

Bu bölümde bazı sıralama algoritmaları, algoritma ölçütleri ile değerlendirilerek verilecek ve daha sonra da algoritmalar boyut (bellek) ve hız açısından karşılaştırılacaktır.

3.4.1- Damlacık (Bubble) Sıralaması

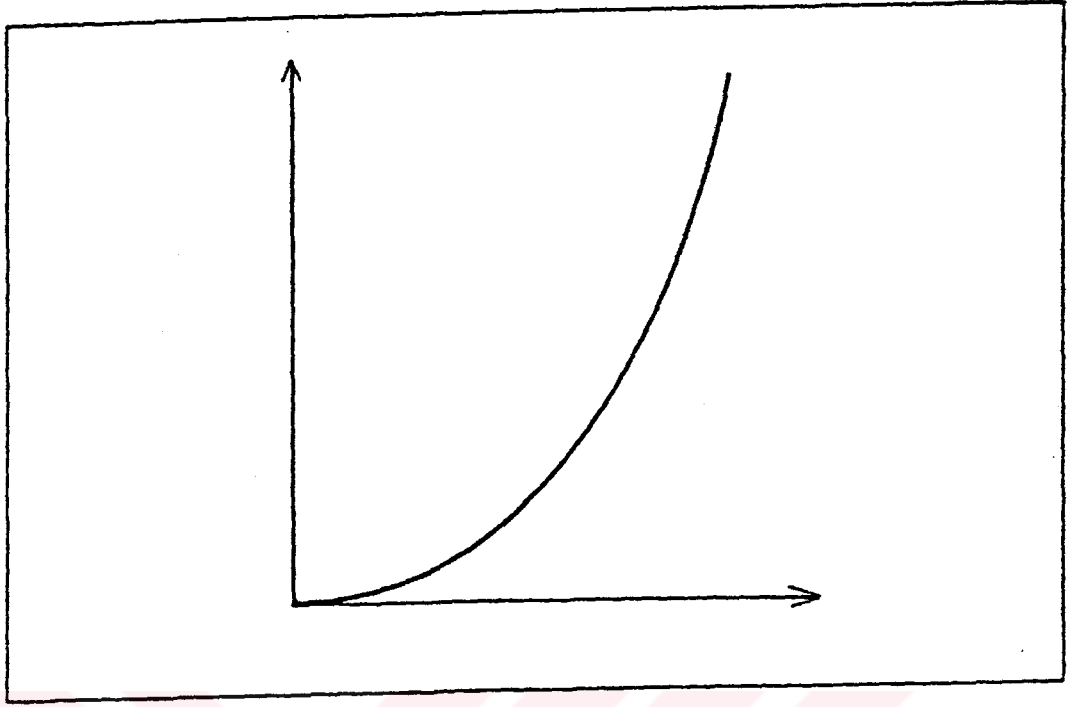
En iyi bilinen ve aynı zamanda da hiç kullanılmayan bir algoritmadır. Algoritma değiş-tokuş yöntemine dayanır.

Genel düşüncesi arda arda karşılaştırmalar ile gerekliyse komşu elemanların yerlerini değiştirerek sıralamayı gerçekleştirmektir. Yöntemin ismi bir kaba giren damlanın kendisine ait seviyeye yerleşmesinden doğmuştur [7].

Bu algoritma ile sıralama yaparken n dizinin eleman sayısı olmak üzere $1/2(n^2-n)$ sayıda karşılaştırma yapılır. Algoritmanın işleyişi basitçe şu şekildedir:

İlk durum :	4	3	1	2
1.adım :	1	4	3	2
2.adım :	1	2	4	3
3.adım :	1	2	3	4

Sıralı bir dizinin sıralanması durumunda, yani en iyi durumda sıfır sayıda yer değiştirme yapılır. Ortalama bir durumda $3/4(n^2-n)$ sayıda yer değiştirme ve en kötü durumda ise $3/2(n^2-n)$ sayıda yer değiştirme yapılır. Damlacık algoritmasının hesap zamanı dizi eleman sayısının karesi ile orantılı olması nedeni ile n^2 algoritması olarak da adlandırılır. Yani, küçük diziler için önemsizken, büyük diziler için çok büyük hesaplama zamanları gerektirir. Sadece karşılaştırmalar gözönüne alındığında bile bu kolaylıkla görülebilir. Her karşılaştırmamanın 0.001 saniye aldığı kabul edilirse, 10 elemanlı bir dizi 0.05 saniyede, 100 elemanlı bir dizi 5 saniyede ve 10.000 elemanlı bir dizi 500 saniyede sıralanırken 100.000 elemanlı bir dizi ise ancak 1400 saatte sıralanabilecektir. (Bu rakamlar sadece karşılaştırma amacı ile keyfi olarak verilmiştir). Şekil 3.1'de hesaplama zamanının eleman sayısı ile ilişkisi görülmektedir.



Şekil 3.1. n^2 Eğrisi

3.4.2- Seçme Sıralaması

Seçme sıralaması, en küçük elemanı bulup ilk elemanla yer değiştirme ve sonra da kalan $n-1$ eleman için aynı işlemi tekrarlamak şeklinde gerçekleşir. Bu algoritma da $1/2(n^2-n)$ karşılaştırma gerektirir. En iyi durumda yer değiştirme sayısı $3(n-1)$ ve en kötü durumda yer değiştirme sayısı $n^2/4+3(n-1)$ 'dir [7]. En iyi durumda, yani dizinin sıralı olması durumunda $n-1$ sayıda eleman hareket edecek ve her hareket 3 yer değiştirme gerektirecektir. Algoritmanın işleyişi basit olarak şu şekildedir.

İlk durum :	2	4	1	3
1.adım :	1	4	2	3
2.adım :	1	2	4	3
3.adım :	1	2	3	4

3.4.3- Araya Sokma Sıralaması

Basit sıralama algoritmalarının sonuncusu olan araya sokma ile sıralamada ilk önce dizinin ilk iki elemanı sıralanır, sonra üçüncü eleman bu sıralı iki elemana göre sıralamadaki yerine yerleştirilir, daha sonra dördüncü eleman bu sıralı üç elemana göre sıralamadaki yerine konulur. Prosese bu şekilde devam edilir. Algoritmanın işleyişi basit olarak şu şekildedir:

```

İlk durum: 4  3  1  2
1.adım    : 3  4  1  2
2.adım    : 1  3  4  2
3.adım    : 1  2  3  4

```

Damlacık ve seçme sıralamasından farklı olarak bu sıralama algoritmasında karşılaştırmaların sayısı dizinin ilk durumuna büyük ölçüde bağlıdır. Dizi sıralı ise karşılaştırmaların sayısı $n-1$, dizi tersten sıralı ile karşılaştırmaların sayısı $1/2(n^2+n)-1$ ve bu iki durum arasında bir durum ise $1/4(n^2+n-2)$ 'dir. Yer değiştirmelerin sayısı ise en iyi durum için $2(n-1)$, ortalama durum için $1/4(n^2+9n-10)$ ve en kötü durum için $1/2(n^2+3n-4)$ olmaktadır [7]. Görüldüğü gibi araya koyma ile sıralamada en kötü durum için rakamlar, damlacık ve seçme sıralamasındaki kadar kötüdür, fakat ortalama durum için çok az bir farkla daha iyidir. Araya sokma algoritmasında en kötü ve en iyi durum arasındaki oran, aşağıda açıklandığı gibi doğrusaldır.

	Karşılaştırma sayısı	Yer değiştirme sayısı
En kötü durum	$1/2(n-1)(n+2)$	$1/2(n-1)(n+4)$
En iyi durum	$n-1$	$2(n-1)$
$\frac{\text{En kötü durum}}{\text{En iyi durum}}$	$1/2(n+2)$	$1/4(n+4)$

3.4.4- Shell Sıralaması

Shell sıralaması, D.L. Shell tarafından bulunmuş olması nedeniyle bu adı almış olmasına rağmen işleyişinin birbirleri üzerine kümelenmiş ıstırdye kabuklarını andırması dolayısıyla bu ad yakıştıırılmış gibi gözükmeğtedir [7].

Algoritma aslında araya sokma algoritmasının geliş-tirilmiş biçimidir. Algoritmanın ana fikri şöyledir: Her sayı, kendisinden dizi boyutu ile orantılı olarak (bu orantı n dizi boyutu olmak üzere $n/2$) belli bir uzaklıktaki elemanla eşleşir ve bu ikili grup kendi arasında araya sokma algoritması ile sıralanır. Sonra bu ikili grup diğer bir ikili grupla birleşerek dörtlü gruplar oluşturur ve bu dörtlü grubun sıralaması da araya sokma algoritması ile yapılır. Sonra her grup bir diğeri ile birleşerek $n/2$ elemanlı son 2 grup da sıralanarak sıralama gerçekleştirilir.

16 elemanlı bir dizinin bu algoritma ile sıralanması aşağıda gösterilmektedir. Örnekte büyük indis büyük sayıyı göstermektedir. Bu yüzden aşağıdaki dizi büyükten küçüğe doğru sıralıdır. Bu dizinin bu yöntemle küçükten büyüğe doğru dizilmesi şu şekildedir.

İlk durum:

$a_{16} \ a_{15} \ a_{14} \ a_{13} \ a_{12} \ a_{11} \ a_{10} \ a_9 \ a_8 \ a_7 \ a_6 \ a_5 \ a_4 \ a_3 \ a_2 \ a_1$

1. adım:

$(a_8 \ a_{16}) \ (a_7 \ a_{15}) \ (a_6 \ a_{14}) \ (a_5 \ a_{13}) \ (a_4 \ a_{12}) \ (a_3 \ a_{11}) \ (a_2 \ a_{10}) \ (a_1 \ a_9)$

2. adım:

$(a_4 \ a_8 \ a_{12} \ a_{16}) \ (a_3 \ a_7 \ a_{11} \ a_{15}) \ (a_2 \ a_6 \ a_{10} \ a_{14}) \ (a_1 \ a_5 \ a_9 \ a_{13})$

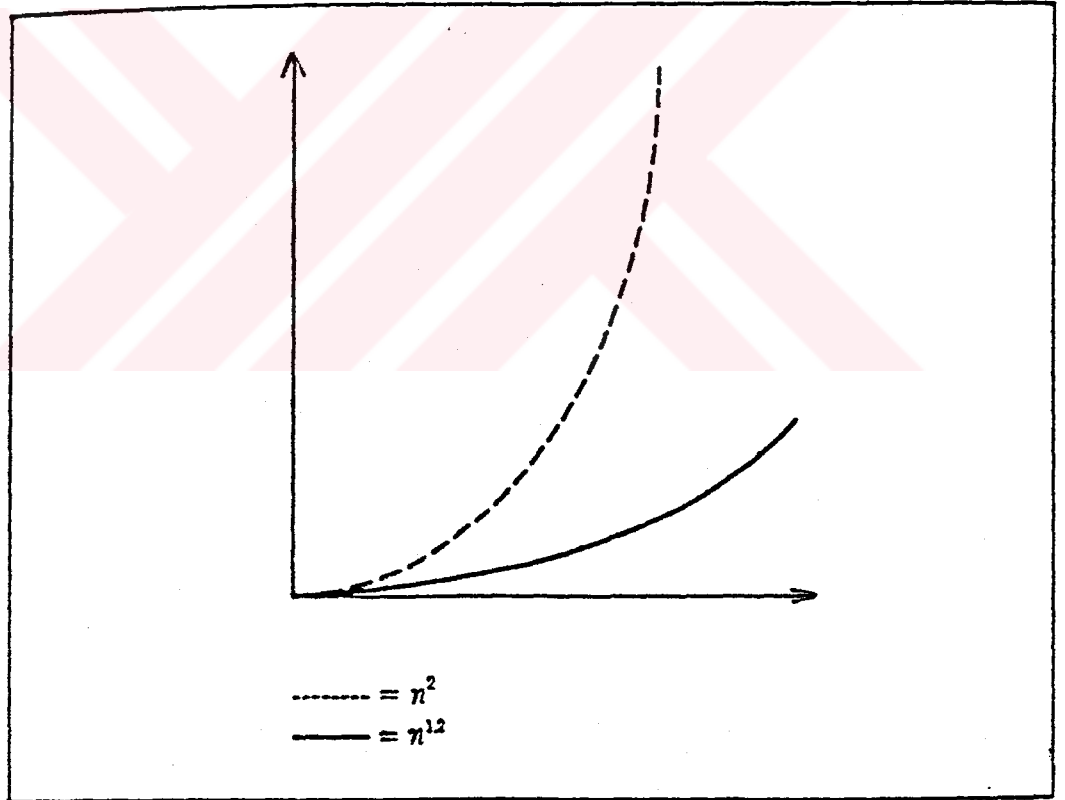
3. adım:

$$(a_2 a_4 a_6 a_8 a_{10} a_{12} a_{14} a_{16}) \quad (a_1 a_3 a_5 a_7 a_9 a_{11} a_{13} a_{15})$$

4. adım:

$$(a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15} a_{16})$$

Orjinal verilerin sıralanmasında, en kötü olasılıkla bile gerekli olan işlem sayısının $n^{3/2}$ olduğu görülebilir. Düzensiz sıralı veriler için, en azından $N > 6000$ olmak üzere işlem sayısı yaklaşık $n^{1.27}$ olur. Bu ise n^2 algoritmalarına göre azımsanamayacak bir ilerlemeyi gösterir. Bu karşılaştırma Şekil 3.2'de görülmektedir.



Şekil 3.2 n^2 ve $n^{1.2}$ Eğrileri

3.4.5- Yığın (Heap) Sıralaması

Heapsort tercih edilen bir sıralama yöntemidir. Yardımcı depolama yerine gerek göstermez. İşlem sayısı

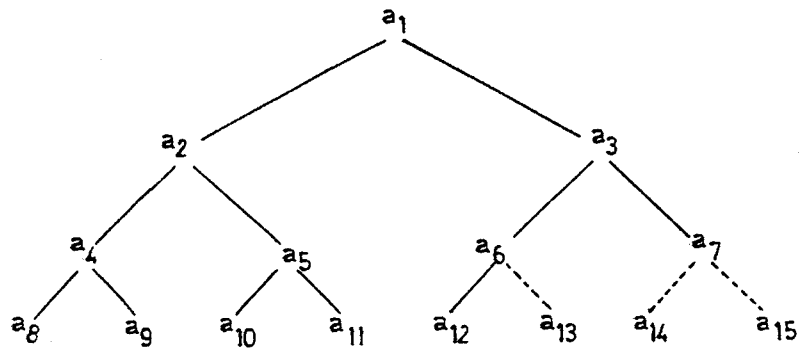
$N \log_2 N$ ile ifade edilebilir. En kötü koşullarda bulunan bir dizinin sıralanması bile ortalama çalışma süresinden %20 fazladır [8]. N elemanlı $(a_i, i=1,2,\dots,N)$ bir dizi verilmiş olsun. Bu kümeden,

$$2 < j < n \text{ ve } i = j/2 \text{ (Fortran'da } i = j/2, \text{Basic'de } i = \text{FlX}(j/2)) \quad (3.5)$$

olmak üzere

$$a_i > a_j \quad (3.6)$$

şeklinde oluşturulan diziye "yığın" (heap) adı verilir [9]. Algoritmanın adı bu tanımdan gelmektedir. Eğer Şekil 3.3'deki gibi düzenlenmiş sayıları gözönüne alırsak (3.5) ve (3.6) tanımlaması, en üstte a_1 , sonraki ayırım için a_2, a_3 ve daha sonra bunların herbirinden olan ayırımlar için de (a_4, a_5) , (a_6, a_7) elemanlarının simgelediği düğüm noktalarından oluşan bir ağaç yapısını ortaya çıkarır. Bu şekilde oluşan ağaç yapısının içindeki herhangi bir düğüm noktasına ait eleman bu düğüm noktasından çıkan her iki dalın ucundaki elemandan büyük veya en azından eşit değere sahiptir. Verilen dizi bu şekilde bir yığın oluşturacak biçimde düzenlenirse, yani heap kümesi biçimine getirilirse sıralama çok kolay olur.



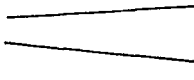
Şekil 3.3. Heap Dizilişi

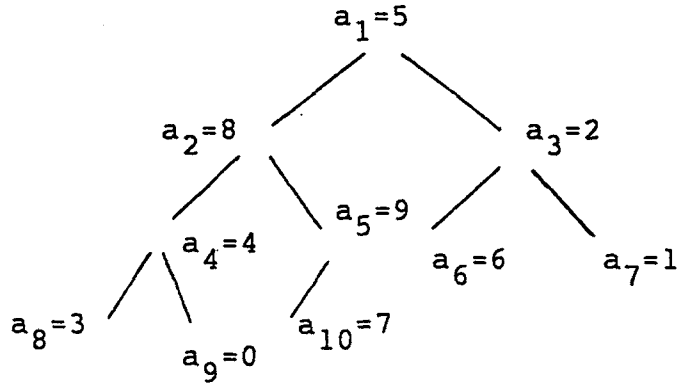
Aşağıda ağaç yapısının, heap'in ve heap'den sıralı dizinin elde edilişi bir örnekle şematik olarak gösterilecektir.

5 8 2 4 9 6 1 3 0 7

ile verilen on elemanlı dizinin sıralanması sözkonusu olsun. Bu durumda önce ağaç yapısının kurulması gerekir. (3.5) ifadesinden $j=2,3,\dots,10$ ile alt düğüm noktalarının, $i=j/2$ ile de üst düğüm noktalarının olduğu Tablo 3.1'den, bunun da ağaç sistemine karşılık geldiği Şekil 3.4'ten görülmektedir. Bu ağaç sistemi verilen dizinin ilk şeklini (Şekil 3.5) temsil etmektedir.

Tablo 3.1. (3.5) İfadesinden Elde Edilen Yapı

Üst Düğüm Noktası	$i=j/2$	$j=2,3,\dots,10$	Alt Düğüm Noktası
$5=a_1$ 	1	2	$a_2=8$
	1	3	$a_3=2$
$8=a_2$ 	2	4	$a_4=4$
	2	5	$a_5=9$
$2=a_3$ 	3	6	$a_6=6$
	3	7	$a_7=1$
$4=a_4$ 	4	8	$a_8=3$
	4	9	$a_9=0$
$9=a_5$ 	5	10	$a_{10}=7$

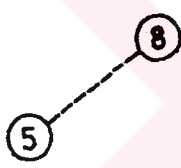


Şekil 3.4. Dizinin İlk Ağaç Yapısındaki Gösterimi

5
8
2
4
9
6
1
3
0
7

Şekil 3.5. Verilen Dizi

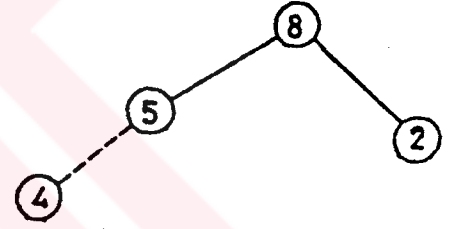
Heap ağaç yapısının elde edilmesi işlemi şematik olarak Şekil 3.6'da görülmektedir.



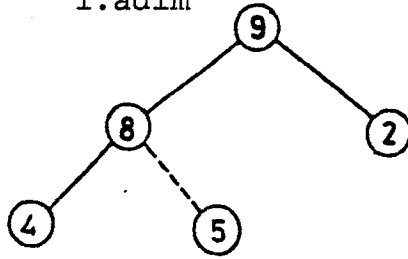
Şekil 3.6.a. 1. adım



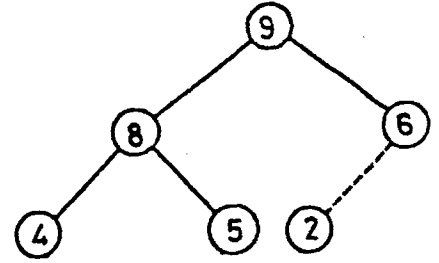
Şekil 3.6.b. 2. adım



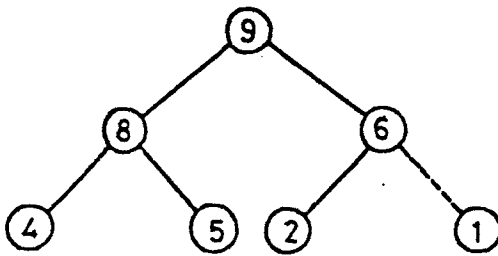
Şekil 3.6.c. 3. adım



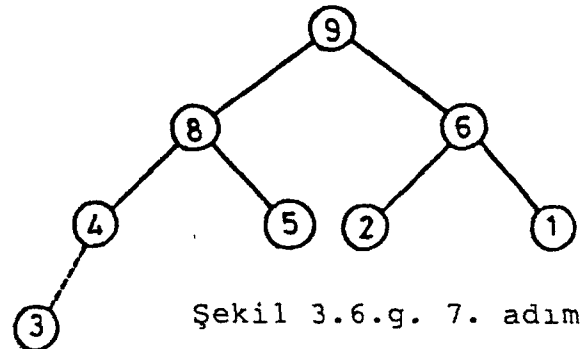
Şekil 3.6.d. 4. adım



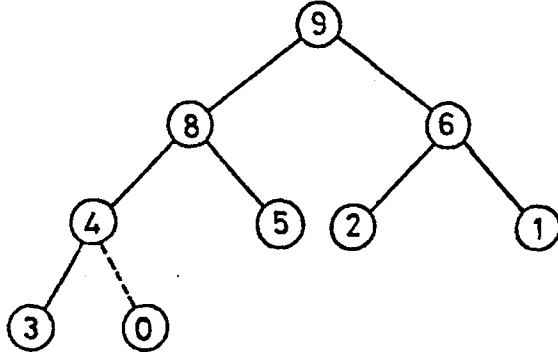
Şekil 3.6.e. 5. adım



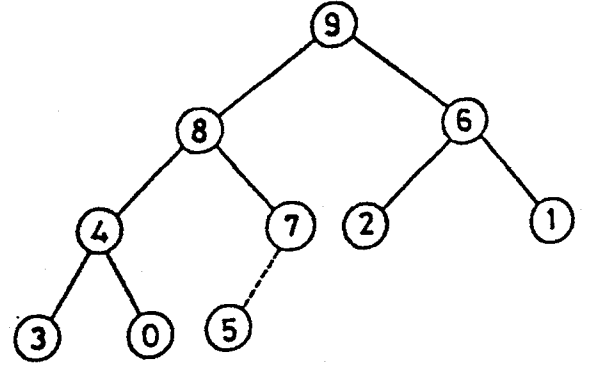
Şekil 3.6.f. 6. adım



Şekil 3.6.g. 7. adım

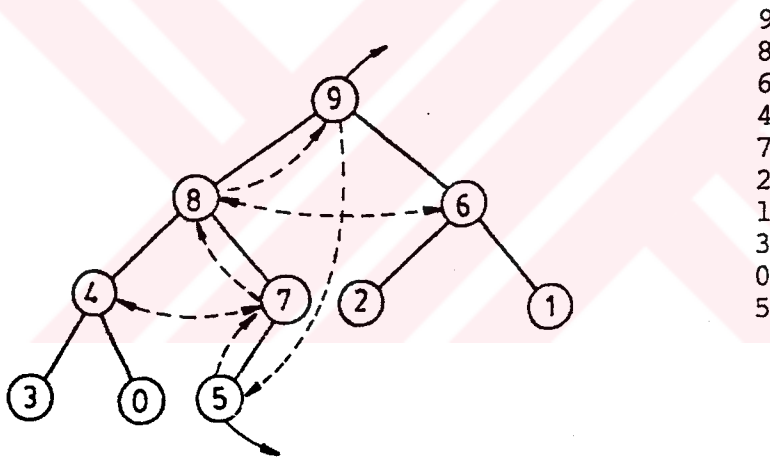


Şekil 3.6.h. 8.adım

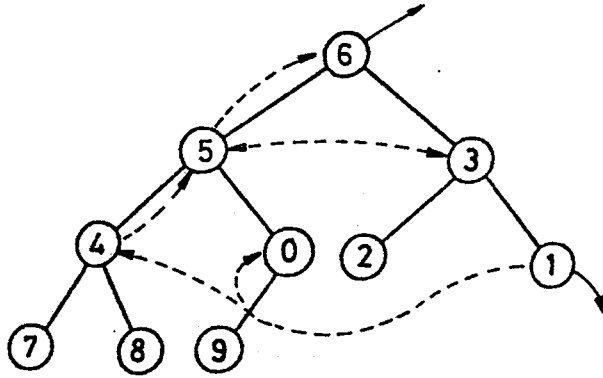


Şekil 3.6.i. 9.adım

Şekil 3.6'dan oluşan bu heap ağaç yapısı içindeki sıralama prosesi şematik olarak Şekil 3.7'de görülmektedir.

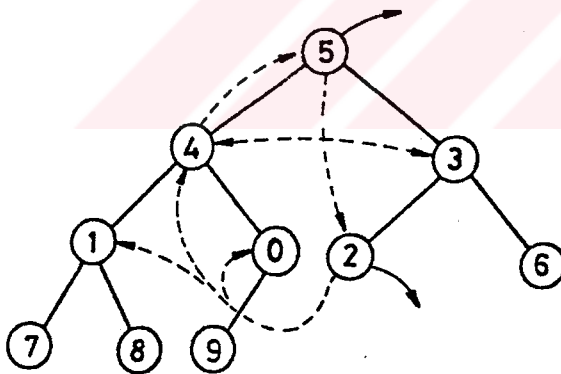


Şekil 3.7.a İlk Heap



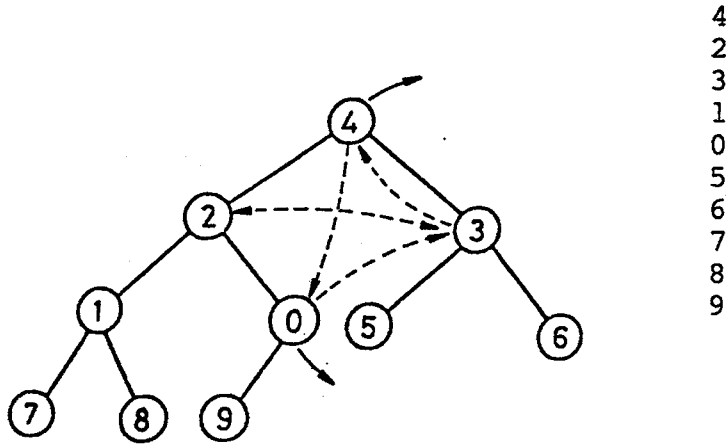
6
5
3
4
0
2
1
7
8
9

Şekil 3.7.d 3. Adım

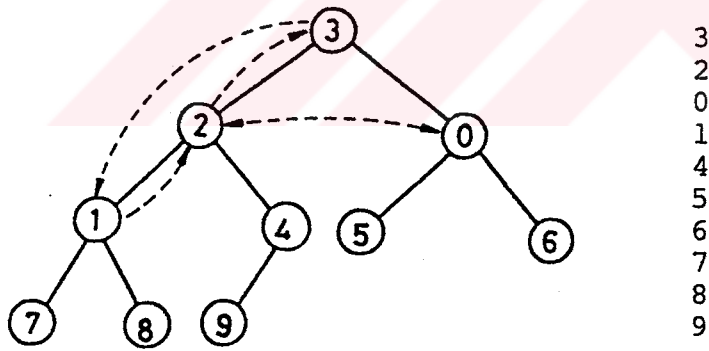


5
4
3
1
0
2
6
7
8
9

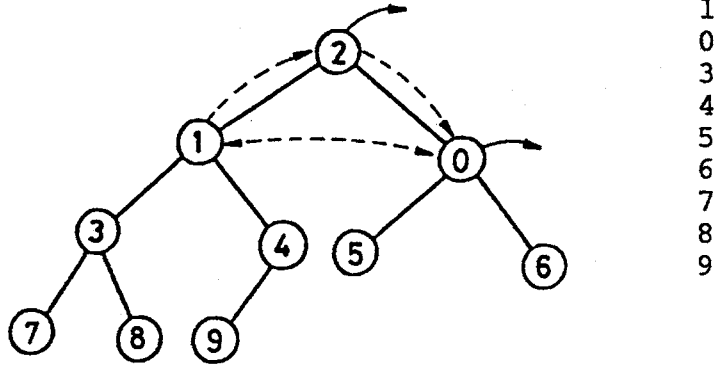
Şekil 3.7.e 4. Adım



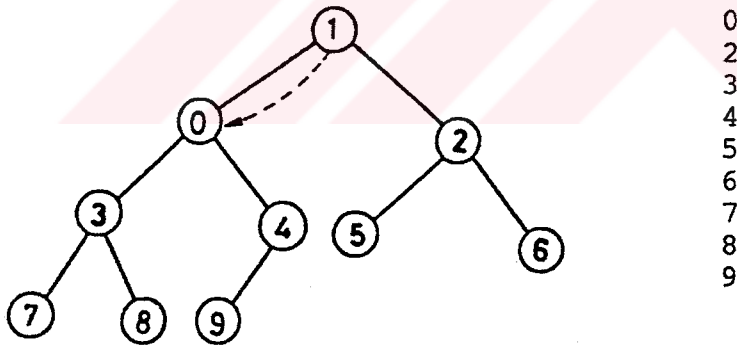
Şekil 3.7.f 5. Adım



Şekil 3.7.g 6. Adım



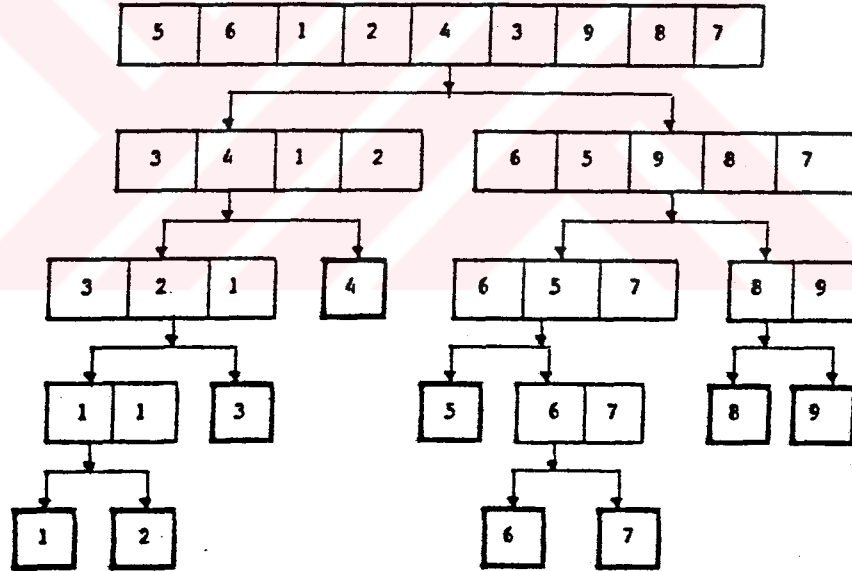
Şekil 3.7.h 7. Adım



Şekil 3.7.i 8. Adım

3.4.6- Hızlı Sıralama (Quicksort)

Bilinen ve geniş bir şekilde kullanılan en iyi sıralama algoritmasıdır. Yer değiştirme yönteminin esasına dayalı olup diğer yöntemlerle kıyaslandığında olağanüstü hızlıdır. Algoritma dizinin bölünmesi düşüncesi üzerine kuruludur. Genel prosedür, dizi elemanlarının seçilen bir elemandan (bu değer karşılaştırmacı olarak adlandırılır) büyük ve eşit olanlarını bir bölüme, küçük olanlarını diğer bir bölüme yerleştirilmesi şeklindedir. Ayrılan her bölümün aynı şekilde alt bölümlerine ayrılması işlemi dizi sıralanıncaya kadar devam eder. Algoritmanın en belirgin özelliği tekrarlı olarak aynı işlemi yapmasıdır. Algoritmanın işleyişi Şekil 3.8'de görülmektedir.



Şekil 3.8. Quicksort İşleyişi

Algoritmanın işleyişi içerisinde karşılaştırmacı değerlerin seçimi rastgele veya küçük bir ortalama alma işlemi ile de gerçekleştirilebilirse de bunun önemi büyük değildir. Ancak optimal bir sıralama için değerlerin ortasındaki bir elemanın seçilmesi uygundur. Birçok dizi

için bu değeri bulmak kolay değildir, ancak algoritma hemen hemen seçimin her durumunda iyi sonuç verir. En kötü durum olan karşılaştırmacı değer dizinin en büyük ve en küçük elemanı olması durumunda bile performans düşmez ve sıralamanın özellikleri üzerinde olumsuz etki yapmaz. Ancak işleyiş sırasında her bölüm için karşılaştırmacı değer olarak en büyük elemanın seçilmesi durumunda algoritma bir n^2 sıralamasına dönüşür. Her bölümde en büyük elemanın seçilmesi çok küçük bir olasılıkla olabileceği için bu önemli bir sorun olarak görülmemektedir. Sıralı dizilerin olması durumunda ise bu diziler sıralama işlemi öncesi rasgele sıralanarak sorun elemine edilmektedir.

Algoritma normal durumda $n \cdot \log n$ sayıda karşılaştırma ve $n \cdot \log n / 6$ sayıda yer değiştirme yapar [7]. Bunlar ise daha önce gördüğümüz algoritmalarla karşılaştırıldığında çok iyi değerlerdir.

3.4.7- Sıralama Algoritmalarının Karşılaştırılması

Yukarıda sözü edilen altı değişik sıralama algoritması için, C programlama dilinde 80286 mikroişlemcili 16 Mhz hızında bir PC-AT bilgisayarında çalıştırılan programlar ile elde edilen sonuçlar tablo 3.3.'de görülmektedir.

Tablo 3.3. Algoritmaların Karşılaştırılması

N Algoritma	1500	7500	15.000	20.000	25.000	30.000
Damlacık	11 sn	282 sn	1126 sn	1999 sn	3123 sn	4495 sn
Seçme	6 sn	142 sn	567 sn	1008 sn	1574 sn	2267 sn
Araya Sokma	4 sn	108 sn	429 sn	760 sn	1187 sn	1707 sn
Shell	0.99sn	5.05sn	10.55sn	14.34sn	18.24sn	21.76sn
Heap	0.22sn	1.37sn	2.97sn	4.18sn	4.89sn	5.22sn
Quick	0.16sn	1.04sn	1.98sn	2.80sn	3.52sn	4.23sn

Tablo 3.3.'deki değerlere dikkat edilirse $4495/3123=30000^2/25000^2$, $282/11=7500^2/1500^2$ vb. değerleri damlacık sıralaması için n^2 algoritması yakıştırmasının gerçeğe ne kadar uygun olduğu görülmektedir. Ayrıca, shell sıralaması için de $21.76/18.24=30.000^{1.27}/25.000^{1.27}$, $18.24/14.34=20.000^{1.27}/15.000^{1.27}$ vb. değerleri bu algoritmanın $n^{1.27}$ ile ($n>6000$ için) orantılı olduğunu göstermektedir. Tablo 3.3.'deki değerler yardımı ile çizilen grafikler Şekil 3.9.ve 3.10.'da görülmektedir.

3.4.8- Bir Sıralama Programı ve Bazı Araştırmalar

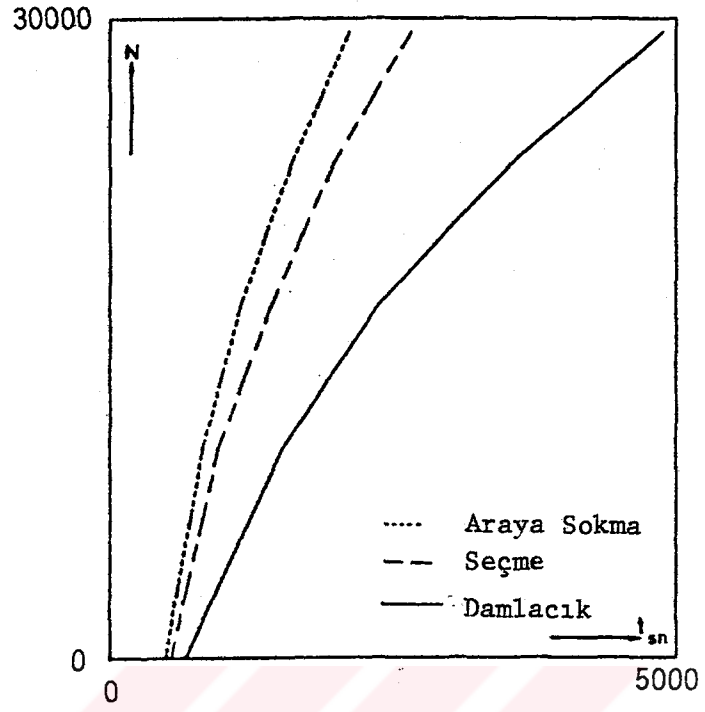
Geliştirilecek olan sıralama yönteminde, bir karşılaştırma ve bu karşılaştırma sonucu gerekiyorsa bir yer-değiştirme işlemi, bunun bilgisayar uygulamasındaki karşılığı olan en içteki çevrimin bir dönümü, bir işlem birimi olarak tanımlanmıştır.

Buna göre n elemanlı bir dizinin sıralanması için,

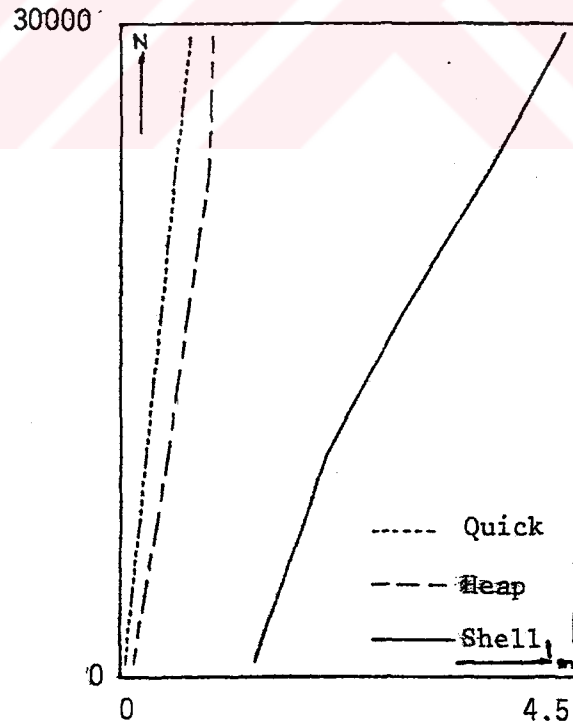
$$q = \frac{n(n-1)}{2} \quad (3.8)$$

tane işlem ortaya çıkar (bak.(3.4.1)). Bunun içinde zaman denklemlerinin önceden saptanmasına gerek vardır. Zaman denklemleri her bilgisayar için ayrıdır. Zaman denklemi bir bilgisayar için bir defa belirlendikten sonra herhangi bir sayıda elemana sahip bir dizinin hangi sürede sıralanabileceği hesaplanabilir. Hatta zaman denklemi programın başına konmak suretiyle programın büyük bir yaklaşıklıkla ne kadar çalışacağı baştan bilinebilir ve buna göre davranılır. Özellikle hızı düşük bilgisayarlar için zaman denkleminin belirlenmesi çok önemlidir. Herhangi bir bilgisayarda bu sayıdaki işlemin yapılabilmesi için gereken süre,

$$t = K. \frac{n(n-1)}{2} \quad (3.9)$$



Şekil 3.9. Yavaş Algoritmalar



Şekil 3.10. Hızlı Algoritmalar

olur. Burada K; bir işlem biriminin icra edilebilmesi için geçen süredir. Gözönüne alınan bir bilgisayarda, n_0 sayıdaki elemanın sıralanması için geçen süre t_0 olarak saptanmış ise, herhangi bir n sayıdaki elemanın sıralanması için geçen süre,

$$t = \frac{t_0}{n_0(n_0-1)} \cdot n \cdot (n-1) \quad (3.10)$$

ifadesi ile bulunabilir [10]. Burada daha sonra genelleştirilecek bir yöntemin ilk iki aşaması verilecektir. Bunlar bir parçalı yöntem ve iki parçalı yöntem olarak adlandırılacaktır.

3.4.8.1- Bir Parçalı Yöntem

n elemanlı bir dizi m parçaya ayrılırsa, m parçanın her birinin ayrı ayrı sıraya dizilmesi için,

$$q_1 = \frac{\frac{n}{m}(\frac{n}{m}-1)}{2} \cdot m \quad (3.11)$$

sayıda işlem gerekir. m tane sıralanmış dizinin tek bir dizi biçiminde sıralanması için, düzenlenen programda,

$$q_2 = 2mn + n \quad (3.12)$$

tane işlem ortaya çıkar (tanımlanan birimde). Yapılan ilk işlem sıralama, ikinci işlem ise birleştirme işlemi olarak adlandırılmıştır. Bu durumda işlem sayısı,

$$q = q_1 + q_2 = \frac{1}{2} n \left(\frac{n}{m} - 1 \right) + 2mn + m \quad (3.13)$$

olur. q fonksiyonu sürekli bir fonksiyon olarak gözönüne alınabilir. İşlem sayısının hangi m değeri için minimum olacağı,

$$\frac{dq}{dm} = 0$$

dan,

$$m = \frac{n}{\sqrt{2(2n+1)}} \quad (3.14)$$

olarak bulunur. Minimum işlem sayısı, (3.14) deki m değeri (3.13)de yerine konursa,

$$q_{\min} = n \left(\sqrt{2(2n+1)} - \frac{1}{2} \right) \quad (3.15)$$

olarak bulunur. Herhangi bir bilgisayarda bu sayıda işlemin yapılabilmesi için gereken süre,

$$t_q = K_q \cdot n \left(\sqrt{2(2n+1)} - \frac{1}{2} \right) \quad (3.16)$$

olur. Burada K_q ; bir işlem biriminin icra edilebilmesi için geçen süredir. Gözönüne alınan bir bilgisayarda n_o^q sayıdaki elemanın sıralanması için geçen süre t_o^q olarak saptanmış ise, herhangi n sayıdaki elemanın sıralanması için geçen süre,

$$t_q = \frac{t_o^q}{n_o^q \left(\sqrt{2(2n_o^q+1)} + \frac{1}{2} \right)} \cdot n \cdot \left(\sqrt{2(2n+1)} - \frac{1}{2} \right) \quad (3.17)$$

ifadesi ile bulunabilir.

3.4.8.2- İki parçalı yöntem

Bir parçalı yönteme göre bir dizinin sıralanmasında dizi $m=n/\sqrt{2(2n+1)}$ sayıda parçaya ayrılırsa işlem miktarı minimum olmaktadır. Mademki dizinin parçalanması ile işlem sayısı azalıyor, o halde bir parçalı yöntem ile elde edilen parça dizilerin her birinin de yeniden parçalanması düşünülebilir. O halde bir parçalı yöntem ile bir dizinin sıralanması için gereken işlem sayısına göre daha az işlem sayısına sahip bir yöntem ortaya konabilir. Bu bölümde, bir parçalı yöntemin daha gelişmiş olan, iki parçalı yöntem olarak adlandırılan böyle bir yöntem ortaya konacaktır.

n elemanlı bir dizi m_1 parçaya bölündükten sonra ortaya çıkan her bir parçanın elemanları bu defada m_2 parçaya bölünürse,

$$P = \frac{\frac{n}{m_1 m_2} \left(\frac{n}{m_1 m_2} - 1 \right)}{2} \cdot m_1 m_2 + (m_2 + 2 \frac{n}{m_1} m_2) m_1 + m_1 + 2n m_1 \quad (3.18)$$

sayıda işlem ortaya çıkar. İşlem sayısının minimumu,

$$\frac{\partial P}{\partial m_1} = -\frac{1}{2} \frac{n^2}{m_2 m_1^2} + m_2 + 1 + 2n = 0 \quad (3.19)$$

$$\frac{\partial P}{\partial m_2} = -\frac{1}{2} \frac{n^2}{m_1 m_2^2} + m_1 + 2n = 0 \quad (3.20)$$

denklemlerinden bulunur. (3.19) ve (3.20) den

$$m_2 = \frac{1+2n}{2n} m_1 \quad (3.21)$$

elde edilir. (3.20) ve (3.21) den

$$n^2 - \frac{1}{2} \left(\frac{1+2n}{n} \right)^2 m_1^4 - n \left(\frac{1+2n}{n} \right)^2 m_1^3 = 0 \quad (3.22)$$

yazılabilir. m_1 , n'nin yanında yeteri kadar küçük kalıyor ise büyük n'ler için (3.21) ve (3.22) den,

$$m = m_1 = m_2 = \sqrt[3]{\frac{n}{4}} \quad (3.23)$$

yaklaşık değeri bulunabilir. (3.23) değeri (3.18) de yerine konursa minimum işlem sayısı,

$$P_{\min} = 3.77976315 n^{\frac{4}{3}} - 0.5n + 0.39685026 n^{\frac{2}{3}} + 0.62996053 n^{\frac{1}{3}} \quad (3.24)$$

bulunur. İşlem sayısı $n^{1.33}$ ile orantılıdır. Bölüm (3.4.4) deki sonuç ile karşılaştırılabilir. Bu ise n^2 algoritmasına göre azınsanmayacak bir durumdur. Herhangi bir bilgisayar- da bu sayıda işlemin yapılabilmesi için gereken süre,

$$t_p = K_p (3.77976315 n^{\frac{4}{3}} - 0.5n + 0.39685026 n^{\frac{2}{3}} + 0.62996053 n^{\frac{1}{3}}) \quad (3.25)$$

olur. Gözönüne alınan bilgisayarda n_o^p sayıdaki elemanın sıralanması için gereken süre t_o^p olarak saptanmışsa,

$$K_p = \frac{t_o^p}{3.77976315 (n_o^p)^{4/3} - 0.5n_o^p + 0.39685026 (n_o^p)^{2/3} + 0.62996053 (n_o^p)^{1/3}} \quad (3.26)$$

ifadesinden bulunabilir.

3.4.8.3- Uygulamalar

Uygulamalar Amstrad CPC 464 mini bilgisayarında yapılmıştır. Tablo 3.4 de, değişik sayıda elemanı olan dizilerin, damlacık yöntemine göre BASIC dilinde düzenlenmiş program ile sıralanması için gereken süreler ve ayrıca hesapla bulunan süreler verilmiştir. $n_o = 2916$ için bilgisayarda $t_o = 6^{\text{saat}} 41^{\text{dak}} 40^{\text{san}} = 24100$ saniye bulunmuştur. Buna göre zaman ifadesi (3.10)dan,

$$t = 0.0028352474 \cdot n \cdot (n-1) \quad (3.27)$$

dir.

Tablo 3.4. Damlacık Yöntemi ile Sıralama İçin Gerekli Süre

Sıralamaya Giren Eleman Sayısı	Bilgisayarda Saptanan Süre	Hesapla Bulunan Süre
1	2	3
168	1 ^{dak} 23 ^{sn}	1 ^{dak} 20 ^{sn}
528	13 ^{dak} 17 ^{sn}	13 ^{dak} 11 ^{sn}
1088	56 ^{dak} 07 ^{sn}	55 ^{dak} 57 ^{sn}
1680	2 ^{sa} 13 ^{dak} 48 ^{sn}	2 ^{sa} 13 ^{dak} 22 ^{sn}
2280	4 ^{sa} 05 ^{dak} 26 ^{sn}	4 ^{sa} 05 ^{dak} 35 ^{sn}
2916	6 ^{sa} 41 ^{dak} 40 ^{sn}	6 ^{sa} 41 ^{dak} 40 ^{sn}

Tablo 3.5. bir parçalı yöntem ile sıralamada (3.14) formülünün, Tablo 3.6. ise iki parçalı yöntemde (3.23) formülünün doğruluğunu sayısal örnek olarak doğrulamaktadır. (3.14) formülü $n=2916$ için $m=27$, (3.23) formülü $n=2916$ için $m_1=m_2=m=9$ parçalama sayısını vermektedir (Tablo 3.5. ve 3.6. da (*) işaretli satırlara bakınız). Bir parçalı ve iki parçalı yöntemin karşılaştırılması için Tablo 3.5. ve Tablo 3.6. incelenmelidir. Verilen bir ve iki parçalı yöntemlerin süre açısından hızlılığının tartışılabilmesi için Tablo 3.5. ve Tablo 3.6., Tablo 3.4. ile karşılaştırılmalıdır.

Tablo 3.5. Bir Parçalı Yöntemde Parçalama Sayısı ile Sürenin Değişimi

Sıralamaya Giren Eleman Sayısı	Parçalama Sayısı m	Saptanan Süre		
		sa	dak	sn
2916	2	3	23	34
2916	6	1	11	45
2416	12		40	59
2916	18		32	32
2916*	27*		29	12*
2916	36		29	35
2916	54		34	05
2916	81		43	50
2916	108		55	03
2916	162	1	18	28

Tablo 3.6. İki Parçalı Yöntemde Parçalama Sayısı ile Sürenin Değişimi

Sıralamaya Giren Eleman Sayısı	Parçalama Sayısı m	Saptanan Süre		
		sa	dak	sn
2916	2-2	1	45	30
2916	4-4		32	00
2916	6-6		19	55
2880*	8-8*		16	36*
2916	9-9		16	29
2900	10-10		16	22
3000	10-10		17	04
2880	12-12		16	51
2940	14-14		18	22
2816	16-16		18	50
2916	18-18		21	06
2916 (6. ve 7. satırdan)	10-10		16	29

Bilgisayarda elemanların bir parçalı yöntem ile sıralanmasında

$$n_q^o = 3420 \quad \text{değeri için } t_q^o = 36^{\text{dak}} 49^{\text{sn}}$$

bulunmuştur. Buna göre bir parçalı yöntemde zaman denklemi (3.17) den,

$$t_q = 0.00554568.n.(\sqrt{2(2n+1)-0.5}) \quad (3.28)$$

olarak elde edilir. Bilgisayarda elemanların iki parçalı yöntem ile sıralanmasında

$$n_p^o = 4000 \quad \text{için } t_p^o = 24^{\text{dak}} 36^{\text{sn}}$$

bulunmuştur. Buna göre iki parçalı yöntemde zaman denklemi (3.25) ve (3.26) dan,

$$K_p = 0.00619881$$

$$t_p = 0.02343 n^{4/3} - 0.0030994n + 0.0024600.n^{2/3} + 0.00390500^{1/3} \quad (3.29)$$

olarak elde edilir. Tablo 3.7. de, bir parçalı yöntem ile değişik elemanlı dizilerin bilgisayarda sıralanması için gereken ve (3.28) zaman denkleminde hesapla bulunan süreleri verilmiştir. Parçalama için (3.14) formülü kullanılmıştır.

Tablo 3.7. Bir Parçalı Sıralama Yönteminde Değişik Elemanlı Dizilerin Sıralanması İçin Bilgisayarda Gereken ve Hesapla Bulunan Süreler

Sıralamaya Giren Eleman Sayısı	(3.14)den Bulunan Parça Sayısı	Bilgisayarda Saptanan Süre		(3.28) zaman Denkleminde Hesaplanan Süre	
		dak	sn	dak	sn
168	7		30		24
528	12	2	27	2	13
1088	17	6	55	6	35
1680	21	13	02	12	39
2280	24	20	22	20	01
2916	27	29	12	28	59
3420	30	36	49	36	49

Tablo 3.8 de, iki parçalı yöntem ile değişik elemanlı dizilerin bilgisayarda sıralanması için gereken ve de (3.29) zaman denkleminde hesapla bulunan süreleri verilmiştir. Parçalama için (3.23) formülü kullanılmıştır.

Tablo 3.8. İki Parçalı Sıralama Yönteminde Değişik Elemanlı Dizilerin Sıralanması İçin Bilgisayarda Gereken ve Hesapla Bulunan Süreler

Sıralamaya Giren Eleman Sayısı (n)	(3.23)den Bulunan Parça Sayısı $m=m_1=m_2$	Bilgisayarda Saptanan Süre		(3.29) Zaman Denkleminde Hesaplanan Süre	
		dak	sn	dak	sn
108	3		17		11
256	4		48		37
500	5	1	50	1	31
864	6	3	36	3	10
1372	7	6	24	5	53
2048	8	10	33	10	03
2916	9	16	29	16	08
4000	10	24	36	24	36

Tablo 3.9 da Damlacık yöntemi, bir parçalı yöntem ve iki parçalı yöntemin rahat karşılaştırılabilmesi için aynı sayıda elemana sahip dizilerin sıralanması için gereken süreler saniye cinsinden verilmiştir.

Tablo 3.9. Damlacık, Bir Parçalı ve İki Parçalı Yöntemlerin Karşılaştırılması

Sıralamaya Giren Eleman Sayısı (n)	Damlacık Yönteminde Süre sn	Bir Parçalı Yöntemde Süre sn	İki Parçalı Yöntemde Süre Zaman Denkleminden sn
168	83	30	21.29
528	797	147	98.54
1088	3367	415	259.12
1680	8028	782	463.12
2280	14726	1222	696.51
2916	24100	1752	967.61
3420	33149	2209	1197.29

Uygulamada otomasyona girecek olan nüfus, vergi, gümrük, emniyet, devlet istatistik enstitüsü gibi kamu kuruluşlarında, stok kontrolü ve benzeri işlerle uğraşan firmalarda elemanların sıralanması işlemleri ile karşılaşılır. Bu sıralama işlemlerinin özellikle masa bilgisayarlarında yapılmasında zaman sorunu ortaya çıkar. Bir dizi elemanlarının ne kadar süre içinde sıralanabileceğinin baştan bilinmesinde yarar olur.

BÖLÜM 4

4- EĞRİ YERLEŞTİRME

Deney sonuçları üzerinde işlemler yapılırken, iki ya da daha fazla değişken arasındaki bağıntıları göstermek üzere doğrular, eğriler, ya da yüzeyler yerleştirmek söz konusu olur. Bazı problemler için bu işlem kolaydır ve elle yapılabilir, ancak bazı problemler için hesaplar çok uzun olur ve yüksek hızlı modern hesap makineleri olmaksızın gerçekleştirmek olanaksızdır [11].

Eğri yerleştirme probleminin ortaya çıkışı Newton, Leibniz ve Gauss'a kadar uzanır [12]. Eğri yerleştirme işlemine gerek duymamızın başlıca nedenleri şunlardır [11].

1. Değişkenler arasındaki bağıntıyı gösteren uygun bir araç aranması,
2. Sonuçların bulunması için bu bağıntının gerekmesi,
3. Ölçme (veya deney) noktalarının geçirilen eğri çevresindeki dağılımlarına bakarak, önceden yaptığımız bazı kabullerin ne derece güvenilebilir olduğunun bilinmesine gerek duyulması,
4. Denenmek istenen bir teorinin bulunması,
5. Kurulmuş olan bir teorinin geçirilecek olan eğrinin biçimini andırması.

Bu gerekliliklere örnek olarak Psikolog Strong'un hatırdı kalma oranının Log T'ye lineer bir fonksiyonla bağılı olduğunu varsayarak bu hipotezini yaptığı deneyler sonucunda görmesi ve $S=1/2 g t^2$ formülü ile bilinen serbest atış hareketinin bir eğri geçirilmesi için uygun olması verilebilir [8].

4.1- En Küçük Kareler Yöntemi

4.1.1- Giriş

Bir çok alanda, $\{x_k\}$ absisleri farklı olmak üzere $(x,y) \dots (x_N, y_N)$ noktalarından oluşan dataların bir kümesini üreten deneylerle oldukça sık karşılaşılır. Sayısal yöntemlerin bir amacı, bu değişkenleri ilişkilendiren $y=f(x)$ fonksiyonunu belirlemektir. Genellikle, önce $f(x)$ fonksiyonunun belirlenmesi için kullanılan modellerden biri seçilir, sonra bu modelin katsayıları hesaplanır. Kullanılacak fonksiyon modeli için çok sayıda olanak vardır. Genel olarak, fonksiyonun şeklini saptayan fiziksel duruma uygun temel bir matematiksel model vardır.

Matematik modelin yanlış ya da uygun olmayan bir fonksiyon şeklinde seçilmesi ölçmeler sırasındaki düzensiz hatalara bir de böyle bir modelin seçilmesinden dolayı düzenli hatalar yükleyecektir. Bu düzenli hataya "teori hatası" ya da "matematiksel model hatası" adı verilir. Daha sonra yapılacak enterpolasyon (ya da ekstrapolasyon) hesaplarının başarısı, matematiksel model hatası yaratmayan bir fonksiyonun seçimine bağlıdır [13].

Burada

$$y=f(x)=Ax+B \quad (4.1)$$

şeklindeki lineer fonksiyonlar modelini ele alacağız.

Genellikle, ölçmelerde deneysel hatalar bulunur ve $\{x_k\}$, $\{y_k\}$ ölçme değerleri ile (4.1)'den elde edilen kesin $f(x_k)$ değerleri arasında e_k ölçme hatası gibi bir hata ortaya çıkar (matematiksel model ile ölçme arasındaki fark). Ölçme hatası olarak adlandırılan e_k değeri, $f(x_k)$ değeri ve y_k ölçme değeri arasında

$$f(x_k) = y_k + e_k \quad (4.2)$$

eşitliği yazılabilir.

Noktaların yakınından geçen (4.1) denkleminin en iyi doğrusal yaklaşımı nasıl bulunur? Bu soruya cevap vermek için hataların (sapma veya artık denilir) incelenmesi gerekir [14].

$$e_k = f(x_k) - y_k \quad 1 \leq k \leq N \quad (4.3)$$

Eğrinin ölçme noktalarından ne kadar uzak olduğunu değerlendirmek için (4.3)'teki düzeltmelerle birlikte birkaç kriter vardır.

$$\text{Maksimum Hata: } E_\infty(f) = \max_{1 \leq k \leq N} \{|f(x_k) - y_k|\} \quad (4.4)$$

$$\text{Ortalama Hata: } E_1(f) = \frac{1}{N} \sum_{k=1}^N |f(x_k) - y_k| \quad (4.5)$$

$$\text{Karesel Ortalama Hata: } E_2(f) = \left[\frac{1}{N} \sum_{k=1}^N |f(x_k) - y_k|^2 \right]^{1/2} \quad (4.6)$$

(koh)

Tablo 4.1. de x_k y_k değerleri ve $f(x_k)$ fonksiyonu verilmiştir.

Tablo 4.1. Hataların Elde Edilişi

x_k	y_k	$f(x_k)=8.6-1.6 x_k$	$ e_k $	e_k^2
-1	10.0	10.2	0.2	0.04
0	9.0	8.6	0.4	0.16
1	7.0	7.0	0.0	0.00
2	5.0	5.4	0.4	0.16
3	4.0	3.8	0.2	0.04
4	3.0	2.2	0.8	0.64
5	0.0	0.6	0.6	0.36
6	-1.0	-1.0	0.0	0.00
			2.6	1.40

Tablo 4.1.'de verilen değerlerden

$$E(f)=0.8$$

$$E_1(f)=0.325$$

$$E_2(f)=0.41833$$

kriterleri bulunur. Maksimum hatanın en büyük olduğunu görüyoruz ve eğer bir nokta çok hatalı ise onun değeri maksimum hatayı belirler. $E_1(f)$ ortalama hatası çeşitli noktalardaki hataların mutlak değerlerinin ortalamasıdır. Bu kriter, hesabı kolay olduğu için sıkça kullanılır.

Veri noktaları kümesine bir doğru ile en iyi yaklaşım (4.4), (4.5) veya (4.6) eşitliklerinin verdiği değerlerden birisi minimum yapılarak sağlanır. $E_2(f)$ kriterini minimum yapmak hesap açısından çok daha kolay olduğu için $E_2(f)$ 'in seçimi gelenek haline gelmiştir.

4.1.2- Dengeleyen Doğru

$(x_1, y_1) \dots (x_N, y_N)$ gibi N noktadan oluşan ve $\{x_k\}$ absisleri farklı bir kümede $E_2(f)$ 'yi minimum yapan $y=f(x)=Ax+B$ doğrusuna dengeleyen doğru denir. Dengeleyen doğru $(x_1, y_1) \dots (x_N, y_N)$ örnekleme noktalarından yararlanılarak $E_2(f)$ değerini minimum yapan $y=f(x)=Ax+B$ doğrusuna ait A ve B parametrelerinin bulunması ile elde edilir. (4.6)'da verilen bu değer minimumu sadece ve sadece düzeltmelerin karelerinin toplamının minimum yapılması ile bulunabilir. Bunun için (4.1) ve (4.6)'dan yazılan

$$E(A, B) = \sum_{k=1}^N (Ax_k + B - y_k)^2 = N[E_2(f)]^2 \quad (4.7)$$

fonksiyonunun hangi (A, B) noktasında minimum olduğu araştırılır. Bu düşünceye göre, (4.7) fonksiyonunda $\{x_k\}$ ve $\{y_k\}$ değerlerinin sabit, A ve B değerlerinin de değişken olduklarına dikkat edilmelidir.

$E(A, B)$ fonksiyonunu minimum yapan bir noktada $\partial E / \partial A$ ve $\partial E / \partial B$ kısmi türevlerinin her ikisi de sıfır olur. B'yi sabit tutup A'ya göre türev alırsak

$$\frac{\partial E}{\partial A} = \sum_{k=1}^N 2(Ax_k + B - y_k)^1 (x_k + 0 - 0) = 2 \sum_{k=1}^N (Ax_k^2 + Bx_k - x_k y_k)$$

(4.8)

ve A'yı sabit tutup B'ye göre türev alırsak

$$\frac{\partial E}{\partial B} = \sum_{k=1}^N 2(Ax_k + B - y_k)^1 (0 + 1 - 0) = 2 \sum_{k=1}^N (Ax_k + B - y_k) \quad (4.9)$$

elde edilir.

(4.8) ve (4.9)'da elde edilen kısmi türevleri sıfıra eşitleyip, toplamının dağılma özelliğini kullanırsak,

$$0 = \sum_{k=1}^N (Ax_k^2 + Bx_k - x_k y_k) = A \sum_{k=1}^N x_k^2 + B \sum_{k=1}^N x_k - \sum_{k=1}^N x_k y_k \quad (4.10)$$

$$0 = \sum_{k=1}^N (Ax_k + B - y_k) = A \sum_{k=1}^N x_k + N \cdot B - \sum_{k=1}^N y_k \quad (4.11)$$

ortaya çıkar.

(4.10) ve (4.11) eşitlikleri

$$\left(\sum_{k=1}^N x_k^2 \right) A + \left(\sum_{k=1}^N x_k \right) B = \sum_{k=1}^N x_k \cdot y_k \quad (4.12)$$

$$\left(\sum_{k=1}^N x_k \right) A + N \cdot B = \sum_{k=1}^N y_k$$

olarak 2'ye 2'lik bir normal denklem sistemi şeklinde yazılabilir. (4.12) lineer sisteminin çözümü Cramer kuralı kullanılarak elde edilebilir. Tablo 4.1'deki veri noktaları kullanılarak, A ve B'yi içeren (4.12) denklem sistemi Tablo 4.2.'deki şekilde hesaplanarak,

Tablo 4.2. Normal Denklem Katsayılarının Elde Edilmesi

x_k	y_k	x_k^2	$x_k y_k$
-1	10	1	-10
0	9	0	0
1	7	1	7
2	5	4	10
3	4	9	12
4	3	16	12
5	0	25	0
6	-1	36	-6
20	37	92	25

$$92 \quad A+20 \quad B = 25$$

$$20 \quad A+8 \quad B = 37$$

olarak elde edilir. Bu denklem sistemi çözülürse

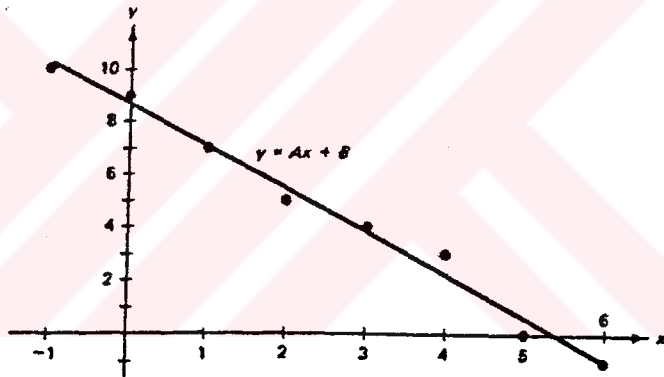
$$A = -1.6071429$$

$$B = 8.6428571$$

bulunur. Böylece dengeleyen doğru

$$y = -1.6071429x + 8.6428571$$

olur. (Şekil 4.1)



Şekil 4.1. Dengeleyen Doğru

$$y = -1.6071429x + 8.6428571$$

M bilinen bir sabit olmak üzere $f(x) = Ax^M$ fonksiyonu bazı durumlarda veri noktalarına daha iyi uygunluk gösterir. Bu durumlarda yalnızca A parametresi bulunur. Bunun için en küçük kareler yöntemini kullanarak

$$E(A) = \sum_{k=1}^N (Ax_k^M - y_k)^2 \quad (4.13)$$

fonksiyonunun minimumu aranır [15]. Bu durumda $E'(A) = 0$ olmalıdır. (4.13)'den türev alınırsa

$$E'(A) = 2 \sum_{k=1}^N (Ax_k^M - y_k) (x_k^M) = 2 \sum_{k=1}^N (Ax_k^{2M} - x_k^M y_k) \quad (4.14)$$

ve buradan $y = Ax^M$ matematiksel modeli için A katsayısı

$$A = \frac{\sum_{k=1}^N x_k^M y_k}{\sum_{k=1}^N x_k^{2M}} \quad (4.15)$$

olarak bulunur.

Gravitasyon sabitinin bulunması için yapılan deney sonuçları Tablo 4.2 de verilmiştir. Burada d metre cinsinden uzunluk, t ise saniye cinsinden zamandır. Burada matematik model $1/2 gt^2$ biçiminde belli olduğu için en uygun modelin araştırılmasına gerek yoktur. Bu modelde $M=2$ 'dir Tablo 4.3.'deki değerler (4.15) formülünde kullanılırsa

$$A = 4.9073 \quad \text{ve} \quad g = 2A = 9.8146 \text{ m/sn}^2$$

bulunur.

Table 4.3. Ölçme Değerleri

Zaman t_k (sn)	Uzunluk, d_k (m)	$d_k t_k^2$	t_k^4
0.200	0.1960	0.00784	0.0016
0.400	0.7850	0.12560	0.0256
0.600	1.7665	0.63594	0.1296
0.800	3.1405	2.00992	0.4096
1.000	4.9075	<u>4.90750</u>	<u>1.0000</u>
		7.68680	1.5664

4.1.3- Dengeleyen Eğri

En küçük kareler yöntemine göre eğri uydurulması yöntemi bir çok doğrusal olmayan duruma genelleştirilebilir. Örneğin, N tane $\{(x_k, y_k)\}$ veri noktası olan bir deney için matematik model

$$y = C \exp(Ax) \quad (4.16)$$

şeklinde üssel bir fonksiyon olarak seçilsin. Seçilen fonksiyonun veri noktalarına en iyi yaklaşım durumuna ilişkin parametrelerin elde edilmesi en küçük kareler yöntemine göre

$$E(A, C) = \sum_{k=1}^N [C \cdot \exp(Ax_k) - y_k]^2 \quad (4.17)$$

nin minimumunun bulunmasını gerekli kılar.

A ve C 'ye göre $E(A, C)$ 'nin kısmi türevleri

$$\frac{\partial E}{\partial A} = 2 \sum_{k=1}^N [C \cdot \exp(Ax_k) - y_k] [Cx_k \cdot \exp(Ax_k)] \quad (4.18)$$

$$\frac{\partial E}{\partial C} = 2 \sum_{k=1}^N [C \cdot \exp(Ax_k) - y_k] [\exp(Ax_k)] \quad (4.19)$$

olur ve (4.18) ve (4.19)'daki türevler sıfıra eşitlendiğinde üssel fonksiyon için normal denklemler

$$C \cdot \sum_{k=1}^N x_k \exp(2Ax_k) - \sum_{k=1}^N x_k y_k \exp(Ax_k) = 0 \quad (4.20)$$

$$C \cdot \sum_{k=1}^N \exp(2Ax_k) - \sum_{k=1}^N y_k \exp(Ax_k) = 0$$

bulunur. (4.20)'deki sistem A ve C bilinmeyenlerine göre doğrusal değildir ve denklem sistemi Newton enterpolasyon yöntemi ile çözülebilir.

(4.20) denklem sisteminin diğer bir çözüm yöntemi Nelder-Mead simpleks algoritmasının kullanılmasıdır [14]. $E(A,C)$ 'nin minimumunu doğrudan doğruya veren bu yöntem de iteratif bir yöntemdir ve her iki yöntemin de çözümü yaklaşımı iyi olan bir başlangıç değerini gerektirir.

4.1.4- Veri Doğrusallaştırma Yöntemi

Üssel fonksiyon için diğer bir yaygın yaklaşım (4.16)'nın her iki tarafının logaritmasının alınmasıdır. (4.16) modelinin,

$$\ln(y) = A.x + \ln(c) \quad (4.21)$$

şeklindeki logaritma fonksiyonunda

$$Y = \ln(y) \quad X = x \quad B = \ln(c) \quad (4.22)$$

biçimindeki değişken ve sabit dönüşümleri yapılırsa,

$$Y = A.x + B \quad (4.23)$$

doğru denklemi elde edilir [15]. Bölüm (4.1)'deki en küçük kareler modeline göre dengeleyen doğrunun bulunması prosesine dönüştürülmüş olur.

En küçük kareler yöntemi, $\{(x_k, y_k)\}$ veri kümesinden dönüştürülmüş olan $\{(x_k, \ln(y_k))\}$ veri kümesine uygulanır. A ve B katsayıları

$$\begin{aligned} A \left(\sum_{k=1}^N x_k^2 \right) + B \left(\sum_{k=1}^N x_k \right) &= \sum_{k=1}^N x_k \ln(y_k) \\ A \left(\sum_{k=1}^N x_k \right) + BN &= \sum_{k=1}^N \ln(y_k) \end{aligned} \quad (4.24)$$

lineer denklem sistemi çözülerek bulunur. (4.16)'daki gerekli C katsayısı

$$C=\exp(B) \quad (4.25)$$

formülü ile hesaplanır.

(4.16) modeline uyan deneylerde, veri doğrusallaştırma denen bu teknik en küçük kareler yönteminde ortaya çıkan nonlinear denklem takımının çözümünde iteratif prosedürü kullanma çabasına girmek istenilmediğinde kullanılan iyi bilinen bir yöntemdir [14].

Tablo 4.4'te $y=C.\exp(Ax)$ dengeleyen eğrisinin parametrelerinin bulunması için 5 veri noktası ve bunların (4.22) ifadesinden elde edilen değişken ve sabit dönüşümleri verilmiştir.

Tablo 4.4. Veri Doğrusallaştırma Yöntemi ile Üssel Model için Katsayıların Elde Edilmesi

x_k	y_k	$\ln(y_k)$	x_k^2	$x_k \ln(y_k)$
0	1.5	0.40547	0	0.00000
1	2.5	0.91629	1	0.91629
2	3.5	1.25276	4	2.50553
3	5.0	1.60944	9	4.82831
4	7.5	2.01490	16	8.05961
10		6.19886	30	16.30974

A ve C'yi bulmak için (4.24) lineer sistemi, Tablo 4.4'ten yararlanılarak

$$30A+10B=16.30974$$

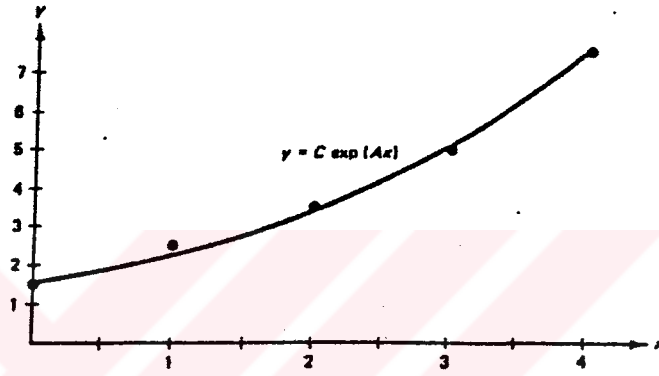
$$10A+5B = 6.19886$$

olarak elde edilir. Bu denklem çözülürse $A=0.391202$ ve

$B=0.457367$ bulunur. Buradan, $C=\exp(0.457367)=1.5799$ ve dengeleyen eğri

$$y=1.5799 \exp (0.391202x) \quad (4.26)$$

olur. (Şekil 4.2)



Şekil 4.2. Veri Doğrusallaştırması ile Elde Edilen $y=1.5799 \exp (0.391202x)$ Şeklindeki Dengeleyen Eğri

Tablo 4.4.'deki doğrusallaştırılmamış veri noktaları kullanılarak $E(A,C)$ 'yi minimum yapan (4.20) denklem sistemi oluşturulur ve bu denklem sistemi iteratif bir yöntemle çözülmürse (4.16) denkleminin A ve C katsayıları bulunur ve buradan

$$y=1.6109 \exp(0.38357x) \quad (4.27)$$

dengeleyen eğrisi elde edilir. Veri doğrusallaştırılmasıyla elde edilen (4.26) ile $E(A,C)$ 'nin minimum yapılmasıyla elde edilmesiyle (4.27)'yi karşılaştırmak veri doğrusallaştırma yöntemi hakkında genel bir karar verebilmek için gereklidir. (4.26) ve (4.27)'nin karşılaştırılması Tablo 4.5'te görülmektedir.

Tablo 4.5. Her İki Yöntemle Üssel Model İçin Bulunan Dengeleyen Eğrilerin Karşılaştırılması

x_k	y_k	$1.5799e^{0.39120x}$	$1.6109e^{0.38357x}$
0	1.5	1.5799	1.6104
1	2.5	2.3363	2.3640
2	3.5	3.4548	3.4692
3	5.0	5.1088	5.0911
4	7.5	7.5548	7.4713

Katsayılar arasında çok küçük bir fark vardır ve [0.4] aralığında fonksiyon değerlerinin %2'den çok fark etmediği görülmektedir. Enterpolasyon amaçları için eğriler arasında gözle görülür büyüklükte bir fark olmaz. Bununla birlikte, verideki hataların dağılımı normal dağılımda ise (4.27) genellikle ekstrapolasyon için daha iyidir. Veri noktalarının dışında $x=10$ için en küçük kareler yaklaşımı (4.26), veri doğrusallaştırma (4.27) ile elde edilen 78.993 değerinden yaklaşık %6 fark eden 74.626 değerindedir.

4.1.4.1- Veri Doğrusallaştırma İçin Dönüşümler

Veri doğrusallaştırma tekniği, bilim adamları tarafından $y=C \cdot \exp(Ax)$, $y=C x^4$, $y=A \cdot \ln(x)+B$ ve $y=A/x+B$ gibi modeller için kullanılmaktadır. Model seçildikten sonra değişkenlerin ve sabitlerin transformasyonu, doğrusal bir bağıntı elde edilecek şekilde yapılmalıdır. Örneğin, $y=D/(x+c)$ modeli $X=xy$, $Y=y$, $C=-1/A$ ve $D=-B/A$ değişken ve sabit dönüşümleri ile $Y=AX+B$ doğrusal modeline dönüştürülebilir. Diğer kullanışlı transformasyonlar Tablo 4.6'da verilmektedir [14].

Tablo 4.6. Veri Doğrusallaştırması İçin Değişken Dönüşümleri

$y = f(x)$	Doğrusal Şekil $Y = AX + B$	Değişken ve Sabit Dönüşümleri
$y = \frac{A}{x} + B$	$y = A \frac{1}{x} + B$	$X = \frac{1}{x}, Y = y$
$y = \frac{D}{x+C}$	$y = \frac{-1}{C} (xy) + \frac{D}{C}$	$X = xy, Y = y$ $C = \frac{-1}{A}, D = \frac{-B}{A}$
$y = \frac{1}{Ax+B}$	$\frac{1}{y} = Ax+B$	$X = x, Y = \frac{1}{y}$
$y = \frac{x}{A+Bx}$	$\frac{1}{y} = A \frac{1}{x} + B$	$X = \frac{1}{x}, Y = \frac{1}{y}$
$y = A \ln(x) + B$	$y = A \ln(x) + B$	$X = \ln(x), Y = y$
$y = C \exp(Ax)$	$\ln(y) = Ax + \ln(C)$	$X = x, Y = \ln(y)$ $C = \exp(B)$
$y = Cx^4$	$\ln(y) = A \ln(x) + \ln(C)$	$X = \ln(x), Y = \ln(y)$ $C = \exp(B)$
$y = (Ax+B)^{-2}$	$y^{-1/2} = Ax+B$	$X = x, Y = y^{-1/2}$
$y = Cx \exp(-Dx)$	$\ln\left(\frac{y}{x}\right) = -Dx + \ln(C)$	$X = x, Y = \ln\left(\frac{y}{x}\right)$ $C = \exp(B), D = -A$
$y = \frac{L}{1+C \exp(Ax)}$	$\ln\left(\frac{L}{y} - 1\right) = Ax + \ln(C)$	$X = x, Y = \ln\left(\frac{L}{y} - 1\right)$

Doğrusal model en genel biçimde

$$f(x) = \sum_{j=1}^M c_j f_j(x) \quad (4.28)$$

şeklinde ifade edilir. Bunun için $\{(x_k, y_k)\}$ veri noktaları kümesinin ve M tane doğrusal bağımsız fonksiyonun bir kümesi olan $f_j(x)$ verilmelidir. M adet doğrusal fonksiyonun kombinasyonu ile verilen $f(x)$ modelinde;

$$E(c_1, c_2, \dots, c_M) = \sum_{k=1}^N [f(x_k) - y_k]^2 = \sum_{k=1}^N \sum_{j=1}^M [c_j f_j(x_k) - y_k]^2 \quad (4.29)$$

biçiminde ifade edilen hataların kareleri toplamının minimum olma koşulundan M adet $\{c_j\}$ katsayısı hesaplanır.

E'nin minimum olması için kısmi türevlerin sıfır olması gerekir ($\partial E / \partial c_i = 0$, $i=1, 2, \dots, M$). Bu gereklilikten,

$$\sum_{k=1}^N \sum_{j=1}^M [c_j f_j(x_k) - y_k] [f_i(x_k)] = 0 \quad i=1, 2, \dots, M \quad (4.30)$$

sistemi elde edilir [14]. (4.30) denklem sistemi bilinmeyen $\{c_j\}$ katsayılarına göre düzenlenirse

$$\sum_{j=1}^M \left[\sum_{k=1}^N f_i(x_k) f_j(x_k) \right] c_j = \sum_{k=1}^N f_i(x_k) y_k \quad i=1, 2, \dots, M$$

(4.31)

şeklinde $M \times M$ 'lik lineer denklem sistemi oluşur. Bunlar normal denklemler adını alır.

(4.31) denklem sistemi, işlemlerin nasıl yapıldığının anlaşılması ve bir sisteme bağlanmış olması nedeni ile matris formunda gösterilir.

F ve F^T matrisleri,

$$F = \begin{bmatrix} f_1(x_1) & f_2(x_1) & f_3(x_1) & \dots & f_M(x_1) \\ f_1(x_2) & f_2(x_2) & f_3(x_2) & \dots & f_M(x_2) \\ f_1(x_3) & f_2(x_3) & f_3(x_3) & \dots & f_M(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_1(x_N) & f_2(x_N) & f_3(x_N) & \dots & f_M(x_N) \end{bmatrix} \quad (4.32)$$

$$F^T = \begin{bmatrix} f_1(x_1) & f_1(x_2) & f_1(x_3) & \dots & f_1(x_N) \\ f_2(x_1) & f_2(x_2) & f_2(x_3) & \dots & f_2(x_N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_M(x_1) & f_M(x_2) & f_M(x_3) & \dots & f_M(x_N) \end{bmatrix} \quad (4.33)$$

olarak gösterilsin. F^T ve Y kolon vektörünün çarpımı,

$$F^T Y = \begin{bmatrix} f_1(x_1) & f_1(x_2) & f_1(x_3) & \dots & f_1(x_N) \\ f_2(x_1) & f_2(x_2) & f_2(x_3) & \dots & f_2(x_N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_M(x_1) & f_M(x_2) & f_M(x_3) & \dots & f_M(x_N) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad (4.34)$$

gözönüne alınsın. (4.34)'teki $F^T Y$ çarpımının i . satır elemanı

$$\sum_{k=1}^N f_i(x_k) y_k \quad (4.35)$$

dır. Dikkat edilirse bu (4.31) eşitliğindeki kolon vektörünün i . elemanı ile aynıdır. Şimdi de $M \times M$ 'lik bir matris F^T ve F matrislerinin çarpımı

$$F^T F = \begin{bmatrix} f_1(x_1) & f_1(x_2) & f_1(x_3) & \dots & f_1(x_N) \\ f_2(x_1) & f_2(x_2) & f_2(x_3) & \dots & f_2(x_N) \\ f_3(x_1) & f_3(x_2) & f_3(x_3) & \dots & f_3(x_N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_M(x_1) & f_M(x_2) & f_M(x_3) & \dots & f_M(x_N) \end{bmatrix} \begin{bmatrix} f_1(x_1) & f_2(x_1) & f_3(x_1) & \dots & f_M(x_1) \\ f_1(x_2) & f_2(x_2) & f_3(x_2) & \dots & f_M(x_2) \\ f_1(x_3) & f_2(x_3) & f_3(x_3) & \dots & f_M(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_1(x_N) & f_2(x_N) & f_3(x_N) & \dots & f_M(x_N) \end{bmatrix} \quad (4.36)$$

gözönüne alınsın $F^T F$ çarpımının i . satır ve j . kolonun elemanı,

$$\sum_{k=1}^N f_i(x_k) f_j(x_k) = f_i(x_1) f_j(x_1) + f_i(x_2) f_j(x_2) + \dots + f_i(x_N) f_j(x_N) \quad (4.37)$$

dir. Bu ise (4.34) eşitliğindeki i . satırdaki c_j katsayısı ile aynıdır. O halde (4.34),

$$F^T F C = F^T Y \quad (4.38)$$

matris şeklinde yazılabilir. (4.28)'de verilen modele ilişkin C katsayılar vektörünün bulunması için, F ve F^T matris elemanlarının hesaplanıp depolanması, sonra $F^T F$ ve $F^T Y$ çarpımlarının bulunması daha sonra da ortaya çıkan denklem sisteminin çözülmesi gerekir.

4.1.5- Dengeleyen Eğri Olarak Polinom

Yukarıda sözü edilen $f(x)$ fonksiyonu $j=0$ 'dan $j=M$ 'ye kadar olmak üzere $f_j(x)=x^j$ fonksiyonlar kümesinden oluşturulursa,

$$f(x)=c_1+c_2x+c_3x^2+\dots+c_{m+1}x^M \quad (4.39)$$

şeklindeki M . dereceden bir polinom elde edilir.

Burada en küçük kareler yöntemi uygulanarak dengeleyen eğri olarak parabolün nasıl bulunacağı ve bunun M . dereceden bir polinoma genelleştirilmesi gösterilecektir. $(x_1, y_1) \dots (x_N, y_N)$ gibi N nokta ve model olarak

$$y=f(x)=Ax^2+Bx+C \quad (4.40)$$

parabolü verilsin A, B ve C katsayılarının

$$E(A, B, C) = \sum_{k=1}^N (Ax_k^2 + Bx_k + C - y_k)^2 \quad (4.41)$$

değerini minimum yapan değerleri araştırılır. A, B ve C 'ye göre E 'nin kısmi türevleri

$$\frac{\partial E}{\partial A} = 2 \sum_{k=1}^N (Ax_k^2 + Bx_k + C - y_k) (x_k)^2$$

$$\frac{\partial E}{\partial B} = 2 \sum_{k=1}^N (Ax_k^2 + Bx_k + C - y_k) (x_k) \quad (4.42)$$

$$\frac{\partial E}{\partial C} = 2 \sum_{k=1}^N (Ax_k^2 + Bx_k + C - y_k)$$

dir. Toplamanın dağılma özelliği kullanılarak, A, B, C değerleri normal denklemleri oluşturmak üzere toplamın dışına taşınabilir. O zaman,

$$\begin{aligned}
& \left(\sum_{k=1}^N x_k^4 \right) A + \left(\sum_{k=1}^N x_k^3 \right) B + \left(\sum_{k=1}^N x_k^2 \right) C = \sum_{k=1}^N x_k^2 y_k \\
& \left(\sum_{k=1}^N x_k^3 \right) A + \left(\sum_{k=1}^N x_k^2 \right) B + \left(\sum_{k=1}^N x_k \right) C = \sum_{k=1}^N y_k x_k \\
& \left(\sum_{k=1}^N x_k^2 \right) A + \left(\sum_{k=1}^N x_k \right) B + N \cdot C = \sum_{k=1}^N y_k
\end{aligned} \tag{4.43}$$

normal denklemleri elde edilir.

Tablo 4.7. En Küçük Kareler Yöntemine Göre Dengeleyen Eğri Olarak Parabolün Katsayılarının Elde Edilişi

x_k	y_k	x_k^2	x_k^3	x_k^4	$x_k y_k$	$x_k^2 y_k$
-3	3	9	-27	81	-9	27
0	1	0	0	0	0	0
2	1	4	8	16	2	4
4	3	16	64	256	12	48
3	8	29	45	353	5	79

Tablo 4.4.'te verilen veri noktalarının arasından geçen en uygun parabol için A, B ve C katsayılarını bulmak için (4.43)'den

$$353 A + 45 B + 29 C = 79$$

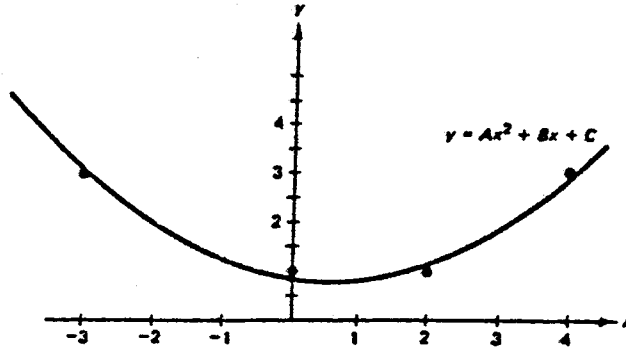
$$45 A + 29 B + 3 C = 5$$

$$29 A + 3 B + 4 C = 8$$

lineer denklemleri elde edilir. Lineer sistemin çözümü sonunda $A=585/3.278$, $B=-631/3.278$ ve $C=1.394/1.639$ bulunur ve istenen parabol

$$y = \frac{585}{3.278} x^2 - \frac{631}{3.278} x + \frac{1.394}{1.638} = 0.178462 x^2 - 0.192495 x + 0.850519$$

olur. (Şekil 4.3)



Şekil 4.3. Dengeleyen Eğri Olarak Parabol

4.1.5.1- Polinom Salınımı

Nonlineer veri üretmek için dengeleyen polinomun kullanılması baştan yanıltıcıdır. Dikkat edilirse N . dereceden bir polinom $N-1$. lokal ekstremuma sahiptir. Eğer veri noktaları gerçekten bir polinom eğrisi üzerinde bulunmuyorsa, o zaman dengeleyen polinomların büyük salınımlar yaptığına dikkat etmelidir. Bu olay, polinom salınımı adını alır ve yüksek dereceli polinomlarda daha belirgin olur. Bu yüzden, seçilen modelin bir polinom olduğu bilinmedikçe 6. veya daha yüksek dereceli polinom nadiren kullanılır [14].

Örneğin, $f(x)=x^{-2}+(|x-1|)^{1/2}$ fonksiyonu $x=\{0.2,1,1.5,2,3,5\}$ verisi için kullanılmış ve Tablo 4.8'deki polinom katsayıları elde edilmiştir. Bu polinomların değerlendirilmesi sonucunda bulunan değerler Tablo 4.9'da verilmektedir. Ayrıca Tablo 4.9'da polinomların $[3,5]$ aralığında aldığı değerler ve Şekil 4.4'de de bu değerlerden yararlanarak çizilen grafik görülmektedir. Dikkat edilirse polinomlar $[3,5]$ aralığında veriden büyük farklılık ve salınım göstermektedir. Buradaki $P_5(x)$ polinomu $(n+1)$ noktadan geçen n . dereceden Lagrange enterpolasyon polinomuna karşılık gelmektedir [16].

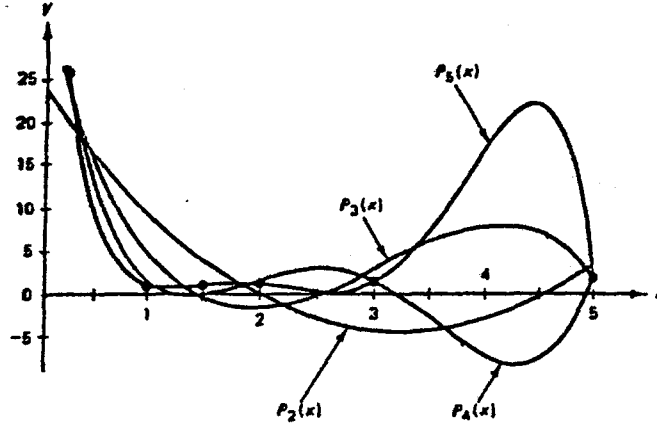
Tablo 4.8. Polinom Katsayıları

	$p_2(x)$	$p_3(x)$	$p_4(x)$	$p_5(x)$
C_1	23.59	32.57	39.40	45.73
C_2	-17.07	-42.04	-77.97	-121.62
C_3	2.62	16.08	53.29	123.32
C_4	-	-1.78	-14.51	-58.07
C_5	-	-	1.33	12.66
C_6	-	-	-	-1.01

Tablo 4.9. Polinom Değerleri

$\begin{matrix} p_i \\ x_i \end{matrix}$	$f(x)$	$p_2(x)$	$p_3(x)$	$p_4(x)$	$p_5(x)$
0.2	25.89	20.28	24.79	25.82	25.89
1	1	9.14	4.83	1.55	1.0
1.5	1.15	3.88	-0.32	0.13	1.15
2	1.25	-0.08	-1.43	1.89	1.25
3	1.53	-4.06	3.12	1.42	1.53
5	2.04	3.70	1.88	2.04	2.04
3.5	1.66	-4.08	6.10	-2.56	7.44
4	1.79	-2.80	7.78	-6.86	17.16
4.5	1.92	-0.20	6.82	-7.38	22.05

Tablo 4.9 ve Şekil 4.4 incelenirse, $p_5(x)$ polinomunun, 6 veri noktasının tümünden geçmesine rağmen en kötü yaklaşım olduğu anlaşılır. Eğer bir polinom kullanma zorunluluğu varsa $p_2(x)$ veya $p_3(x)$ seçilmelidir.



Şekil 4.4. Polinom Salınımı

$(x_1, y_1) \dots (x_N, y_N)$ gibi N veri noktasına uyan M . dereceden

$$P_M(x) = c_1 + c_2 x + c_3 x^2 + c_4 x^3 + \dots + c_M x^{M-1} + c_{M+1} x^M \quad (4.44)$$

dengeleyen polinomunun katsayıları,

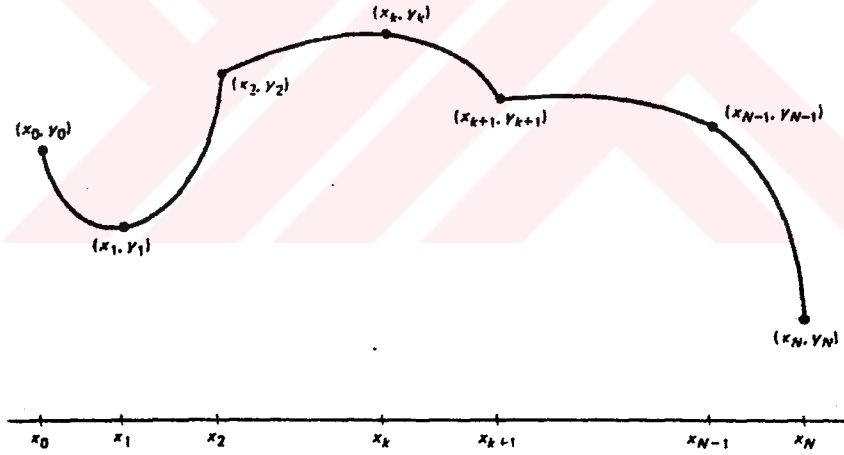
$$\begin{aligned} & \left(\sum_{k=1}^N x_k^{2M} \right) c_{M+1} + \left(\sum_{k=1}^N x_k^{2M-1} \right) c_M + \dots + \left(\sum_{k=1}^N x_k^M \right) c_1 = \sum_{k=1}^N x_k^M y_k \\ & \left(\sum_{k=1}^N x_k^{2M-1} \right) c_{M+1} + \left(\sum_{k=1}^N x_k^{2M-2} \right) c_M + \dots + \left(\sum_{k=1}^N x_k^{M-1} \right) c_1 = \sum_{k=1}^N x_k^{M-1} y_k \\ & \vdots \\ & \left(\sum_{k=1}^N x_k^M \right) c_{M+1} + \left(\sum_{k=1}^N x_k^{M-1} \right) c_M + \dots + N c_1 = \sum_{k=1}^N y_k \end{aligned} \quad (4.45)$$

ile bulunur.

Aşağıda 4.2. bölümünde doğrusal olmayan eğri yerleştirme için bir başka yöntem verilmiştir.

4.2- Spline Fonksiyonları ile Enterpolasyon

$(N+1)$ noktadan oluşan $\{(x_k, y_k)\}$ kümesi için polinom enterpolasyonu genellikle tatmin edici değildir. Bölüm 4.1.5.1.'de (Polinom Salınımı) tartışıldığı gibi, N . dereceden bir polinom $N-1$ tane relatif maksimum ve minimuma sahiptir ve grafik noktalardan salınarak geçer. Diğer bir yöntem, düşük dereceli $S_k(x)$ polinomuna ait grafik parçalarını birleştirmek ve ardışık (x_k, y_k) ve (x_{k+1}, y_{k+1}) noktaları arasında enterpole etmektir. (Şekil 4.5). $[x_k, x_{k+1}]$ ve $[x_{k+1}, x_{k+2}]$ 'nin üzerine bulunan $y=S_{k+1}(x)$ ve $y=S_k(x)$ eğrileri (x_{k+1}, y_{k+1}) ortak noktasından geçer. Grafiğin iki parçası (x_{k+1}, y_{k+1}) noktasında "birlikte düşümlenir" ve $\{S_k(x)\}$ fonksiyonlar kümesi $S(x)$ ile belirtilen parça parça polinom eğrisini oluşturur.



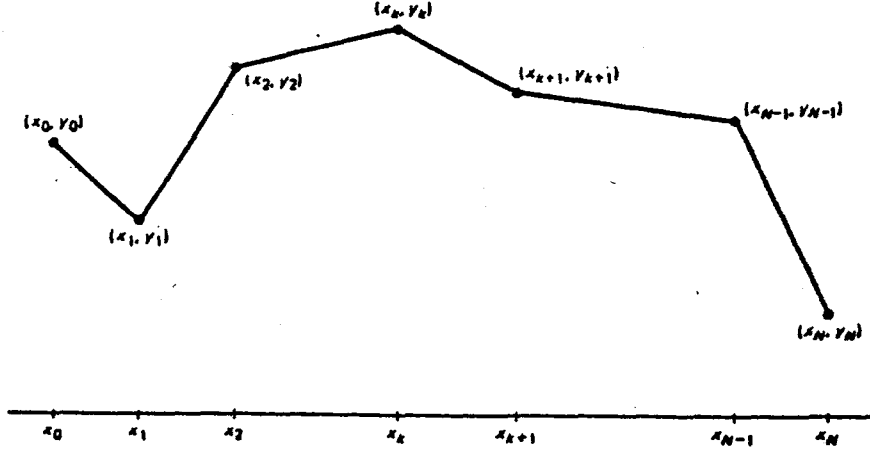
Şekil 4.5. Parça Parça Polinom Enterpolasyonu

4.2.1- Parça Parça Doğrusal Enterpolasyon

Kullanılan en basit polinom, 1. dereceden bir polinomdur ve noktalardan geçen doğru parçalarını içeren bir poligon oluşturur.

$$S_k(x) = y_k \frac{x - x_{k+1}}{x_k - x_{k+1}} + y_{k+1} \frac{x - x_k}{x_{k+1} - x_k} \quad x_k \leq x \leq x_{k+1} \quad (4.46)$$

$k=0,1,2,\dots,N-1$ için (4.46)'da verilen eğri Şekil 4.6'da görülen poligonu verir.



Şekil 4.6. Parça Parça Doğrusal Enterpolasyon
(Doğrusal Bir Spline)

Eşdeğer bir açıklama,

$$d_k = (y_{k+1} - y_k) / (x_{k+1} - x_k) \quad (4.47)$$

eğimi göstermek üzere, bir noktadan geçen eğimi verilen

$$y = S_k(x) = y_k + d_k(x - x_k) \quad (4.48)$$

doğru denklemi kullanılarak verilebilir.

Buradan $k=0,1,\dots,N-1$ için doğrusal spline fonksiyonu

$$S(x) = \begin{cases} y_0 + d_0(x - x_0) & x_0 \leq x \leq x_1 \\ y_1 + d_1(x - x_1) & x_1 \leq x \leq x_2 \\ \vdots & \vdots \\ y_k + d_k(x - x_k) & x_k \leq x \leq x_{k+1} \\ \vdots & \vdots \\ y_{N-1} + d_{N-1}(x - x_{N-1}) & x_{N-1} \leq x \leq x_N \end{cases} \quad (4.49)$$

$S(x)$ 'in kesin hesabı için (4.49) denklemi, (4.46) denkleminde görünüm açısından daha iyidir. Absislerin $x_0 < x_1 < \dots < x_{N-1} < x_N$ şeklinde sıralandığı varsayılınsın. Herhangi bir x değerinin içinde bulunacağı $[x_k, x_{k+1}]$ kapalı aralığı, $x-x_1, x-x_2, \dots, x-x_{N-1}, x-x_N$ ardışık farklarının, $x-x_{k+1} < 0$ olmak üzere en küçük tamsayı değeri elde edilmeye kadar hesaplanmasıyla bulunur. Böylece k 'yi $x_k < x < x_{k+1}$ olacak şekilde elde edilir ve $S(x)$ spline fonksiyonunun değeri,

$$S(x) = s_k(x) = y_k + d_k(x - x_k) \quad x_k < x < x_{k+1} \quad (4.50)$$

olur.

Bu yol daha yüksek dereceli polinomlara da genişletilebilir. Örneğin, eğer $x_0, x_1, \dots, x_{2M}$ verilmişse o zaman, $[x_{2k}, x_{2k+2}]$ kapalı alt aralığında parça parça ikinci dereceden bir polinom oluşturulabilir. Böylece oluşturulan ikinci dereceden spline fonksiyonlarının x_{2k} gibi indisi çift sayı olan düğüm noktalarında eğriliğin birdenbire değişmesi bu fonksiyonların bir sakıncasıdır ve bu durum bunların grafiğinde istenmeyen kıvrımlara veya distorsiyonlara neden olur. İkinci dereceden spline fonksiyonlarının ikinci türevi çift sayılı düğüm noktalarında süreksizdir. Eğer parça parça kübik polinomlar kullanılırsa o zaman birinci ve ikinci türevler sürekli duruma getirilebilir.

4.2.2- Parça Parça Kübik Spline Fonksiyonları

Bir veri noktaları kümesine bir polinom eğrisini uydurmanın, tasarlama ve bilgisayar grafikleri alanında uygulamaları vardır. Tasarımcı, hata söz konusu olmaksızın veri noktaları arasından geçen "pürüzsüz bir eğri" çizmek ister. Bir eğri cetvelinin kullanılması ve gözle bakıldığında, pürüzsüz görünen bir eğri çizilmesi yaygındır. Matematiksel olarak, her bir $[x_k, x_{k+1}]$ aralığında $s_k(x)$

kübik fonksiyonlarını, parça parça bu fonksiyonlardan oluşan $y=S(x)$ fonksiyonunun kendisi, birinci ve ikinci türevleri $[x_0, x_N]$ geniş aralığının tamamında sürekli olacak şekilde kurma olanağı vardır. $S'(x)$ 'in sürekliliği, $Y=S(x)$ grafiğinin keskin köşelere sahip olmayacağı, $S''(x)$ 'in sürekliliği ise "eğrilik yarıçapının" her noktada tanımlı olduğu anlamına gelir.

Absisleri $x_0 < x_1 < \dots < x_N$ şeklinde sıralanmış olan N tane (x_k, y_k) noktalarını gözönüne alalım. Eğer aşağıda verilen koşulları sağlayan N tane kübik polinom varsa, yine aşağıda tanımlanmış olan $S(x)$ fonksiyonuna kübik spline adı verilir [14].

$$I. S(x) = S_k(x) = S_{k,0} + S_{k,1}(x-x_k) + S_{k,2}(x-x_k)^2 + S_{k,3}(x-x_k)^3$$

$$x_k < x < x_{k+1} \quad k=0,1,\dots,N-1$$

$$II. S(x_k) = y_k, \quad k=0,1,\dots,N$$

dir. Yani, spline bütün veri noktalarından geçer.

$$III. S_k(x_{k+1}) = S_{k+1}(x_{k+1}), \quad k=0,1,\dots,N-2$$

dir. Yani spline sürekli bir fonksiyon şeklindedir.

$$IV. S'_k(x_{k+1}) = S'_{k+1}(x_{k+1}), \quad k=0,1,\dots,N-2$$

dir. Yani spline'in birinci türevi sürekli dir.

$$V. S''_k(x_{k+1}) = S''_{k+1}(x_{k+1}), \quad k=0,1,\dots,N-2$$

dir. Yani spline'in ikinci türevi sürekli dir.

4.2.2.1- Kübik Spline'in Varlığı

Eğer I-V özelliklerini sağlayan bir kübik spline var ise araştırmak gerekir. Bu nedenle öncelikle kübik spline'in varlığı gösterilmelidir. Her $S_k(x)$ kübik polinomunun dört tane bilinmeyen sabiti vardır. Bu nedenle toplam $4N$ tane saptanacak katsayı var demektir. Diğer bir şekilde, $4N$. dereceden serbestliğin veya koşulun kesin olarak gerekli olduğu söylenebilir. Kübik spline fonksiyonunun II, III, IV ve V'de verilen özellikleri sağlaması istenir. Bu nedenle veri noktalarının II özelliğini sağlaması isteğinden $N+1$ koşul, III, IV ve V özelliklerini sağlaması isteklerinden de her biri için $N-1$ koşul yazılabilir. Böylece $N+1+3(N-1) = 4N-2$ koşul belirlenir. Bu, ikinci dereceden serbestlik gösterir ve bu koşullara sınır (uç nokta) koşulları denir. Bunlar daha sonra tartışılacak olan x_0 ve x_N 'deki $S'(x)$ veya $S''(x)$ 'i de kapsar. Şimdi denklemlerin çıkarılması ile ilgilenilebilir.

$S(x)$, her bir parçada kübik olduğu için $S(x)$ 'in ikinci türevi $S''(x)$, $[x_0, x_N]$ aralığının her bir alt aralığında doğrusaldır. Doğrusal lagrange enterpolasyon formülü $S''(x) = S''_k(x)$ için şu gösterimlerle verilebilir:

$$S''_k(x) = S''(x_k) \frac{x - x_{k+1}}{x_k - x_{k+1}} + S''(x_{k+1}) \frac{x - x_k}{x_{k+1} - x_k} \quad (4.51)$$

(4.51)'de $m_k = S''(x_k)$, $m_{k+1} = S''(x_{k+1})$ ve $h_k = x_{k+1} - x_k$ koyulursa

$$S''_k(x) = \frac{m_k}{h_k}(x_{k+1} - x) + \frac{m_{k+1}}{h_k}(x - x_k) \quad k=0, 1, \dots, N-1 \quad (4.52)$$

elde edilir. (4.52)'nin iki kere integrali alınır

$$x_{k+1} - x = u, \quad x - x_k = v \quad \text{değişken dönüşümleri ile}$$

$$-dx = du, \quad dx = dv$$

$$\frac{d S'_k(x)}{dx} = \frac{m_k}{h_k} \cdot u + \frac{m_{k+1}}{h_k} v$$

$$\frac{d S_k(x)}{dx} = -\frac{1}{2} \frac{m_k}{h_k} u^2 + \frac{1}{2} \frac{m_{k+1}}{h_k} v^2 + c_1$$

$$S_k(x) = \frac{1}{6} \frac{m_k}{h_k} u^3 + \frac{1}{6} \frac{m_{k+1}}{h_k} v^3 + c_1 x + c_2$$

$$S_k(x) = \frac{1}{6} \frac{m_k}{h_k} (x_{k+1} - x)^3 + \frac{m_{k+1}}{h_k} (x - x_k)^3 + c_1 x + c_2 \quad (4.53)$$

bulunur. İntegral sabitleri için,

$$-1 \cdot p_k + 1 \cdot q_k = c_1$$

$$x_{k+1} \cdot p_k - x_k q_k = c_2$$

sabit değişimleri yapılırsa

$$S_k(x) = \frac{1}{6} (x_{k+1} - x)^3 + \frac{1}{6} \frac{m_{k+1}}{h_k} (x - x_k)^3 + (q_k - p_k) x + x_{k+1} p_k - x_k q_k$$

ve sonuç

$$S_k(x) = \frac{1}{6} (x_{k+1} - x)^3 + \frac{1}{6} \frac{m_{k+1}}{h_k} (x - x_k)^3 + p_k (x_{k+1} - x) + q_k (x - x_k) \quad (4.54)$$

şeklinde bulunur.

(4.54) eşitliğinde $y_k = S_k(x_k)$ ve $y_{k+1} = S_k(x_{k+1})$ yazılırsa p_k ve q_k sabitleri için

$$y_k = \frac{m_k}{6} h_k^2 + p_k h_k \quad (4.55a)$$

ve

$$x_k = \frac{m_{k+1}}{6} h_k^2 + q_k h_k \quad (4.55b)$$

eşitlikleri elde edilir. Bu iki eşitlikten p_k ve q_k çözülür ve (4.54) eşitliğinde bu değerler yerlerine konursa $S_k(x)$ kübik fonksiyonu,

$$S_k(x) = \frac{m_k}{6h_k} (x_{k+1}-x)^3 + \frac{m_{k+1}}{6h_k} (x-x_k)^3 + \left(\frac{y_k}{h_k} - \frac{m_k h_k}{6}\right) (x_{k+1}-x) + \left(\frac{y_{k+1}}{h_k} - \frac{m_{k+1} h_k}{6}\right) (x-x_k) \quad (4.56)$$

olarak bulunur.

(4.56) gösteriminin yalnızca $\{m_k\}$ bilinmeyen katsayılarını içeren bir şekle indirgendiğine dikkat edilmelidir. Bu katsayıları bulmak için (4.56)'nın türevi alınırsa,

$$S'_k(x) = -\frac{m_k}{2h_k} (x_k-x)^2 + \frac{m_{k+1}}{2h_k} (x-x_k)^2 - \left(\frac{y_k}{h_k} - \frac{m_k h_k}{6}\right) + \frac{y_{k+1}}{h_k} - \frac{m_{k+1} h_k}{6} \quad (4.57)$$

bulunur.

(4.57) türevinin x_k 'daki değeri, $d_k = (y_{k+1} - y_k)/h_k$ olmak üzere, gerekli kısaltmalardan sonra,

$$S'_k(x) = -\frac{m_k}{3} h_k - \frac{m_{k+1}}{6} h_k + d_k \quad (4.58)$$

olur.

Benzer olarak, $S'_{k-1}(x)$ bağıntısının elde edilmesi ve bunun x_k 'daki değeri için $k-1$ ile k yer değiştirerek,

$$S'_{k-1}(x_k) = \frac{m_k}{3} h_{k-1} + \frac{m_{k-1}}{6} h_{k-1} + d_{k-1} \quad (4.59)$$

elde edilir.

Şimdi (4.58) ve (4.59) eşitlikleri IV özelliğinde yerine konursa, m_{k-1} , m_k ve m_{k+1} arasında önemli bir bağıntı olan,

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = u_k \quad k=1, 2, \dots, N-1 \quad (4.60)$$

bağıntısı elde edilir. Burada $u_k = 6(d_k - d_{k-1})$ 'dir. O halde ikinci dereceden serbestisi olan bir çözüm vardır.

4.2.2.2- Kübik Spline'ların Oluşturulması

(4.60) denkleminde bilinmeyenlerin $\{m_k\}$ değerleri olduğu, diğer terimlerin ve çarpanların $\{(x_k, y_k)\}$ verilerinden basit aritmetik işlemler ile elde edilebileceği görülür. (4.60) sistemi, $N+1$ bilinmeyen içeren $N-1$ sayıda bir lineer denklem sistemidir. Bu yüzden iki ek eşitlik gereklidir. Bunlar (4.55) sistemindeki $(N-1)$ eşitlikten m_N ve birinci eşitlikten m_0 elemine edilerek kullanılır. Uç nokta koşulları (sınır koşulları) için standart proses tablo 4.10'da özetlenmiştir [14].

Tablo 4.10. Kübik Bir Spline İçin Uç Nokta Koşulları (Sınır Koşulları)

Koşul	m_0 ve m_N eşitlikleri
I- Kenetlenmiş Kübik Spline	$m_0 = \frac{3}{h_0} [d_0 - S'(x_0)] - \frac{m_1}{2}$ $m_N = \frac{3}{h_{N-1}} [S'(x_N) - d_{N-1}] - \frac{m_{N-1}}{2}$
II- Doğal Spline	$m_0 = 0, m_N = 0$
III- $S''(x)$ 'in uç noktalarda ekstrapolasyonu	$m_0 = m_1 - \frac{h_0(m_2 - m_1)}{h_1}$ $m_N = m_{N-1} + \frac{h_{N-1}(m_{N-1} - m_{N-2})}{h_{N-2}}$
IV- $S''(x)$ uç nokta yakınındaki değerler	$m_0 = m_1 \quad m_N = m_{N-1}$
V- $S''(x)$ her uç noktada belirlenir	$m_0 = S''(x_0) \quad m_N = S''(x_N)$

Tablo 4.10'daki V nolu proses gözönüne alınsın. Eğer m_0 verilirse o zaman $h_0 m_0$ hesaplanabilir ve (4.60)'in ilk eşitliği ($k=1$ olduğunda)

$$2[h_0 + h_1]m_1 + h_1 m_2 = u_1 - h_0 m_0 \quad (4.61)$$

olur. Benzer olarak m_N verilirse $h_{N-1} m_N$ hesaplanabilir ve (4.55)'in son eşitliği ($k=N-1$ olduğunda)

$$h_{N-2} m_{N-2} + 2(h_{N-2} + h_{N-1})m_{N-1} = u_{N-1} - h_{N-1} m_N \quad (4.62)$$

olur. (4.61) ve (4.62) eşitlikleri $k=2, 3, \dots, N-2$ için (4.60) denkleminde elde edilen eşitliklerle birlikte $m_1, m_2, \dots, m_{N-1}$ katsayılarını içeren $N-1$ doğrusal eşitliği oluşturur.

Tablo 4.10'dan seçilen, özel sınır koşulları ne olursa olsun, (4.60) eşitliğinden 1. ve $(N-1)$. eşitlikler yine yazılabilir ve $m_1, m_2, \dots, m_{N-1}$ 'i içeren $H.M=V$ şeklindeki tridiagonal lineer denklem sistemi elde edilebilir.

$$\begin{bmatrix} b_1 & c_1 & & & \\ a_1 & b_2 & c_2 & & \\ & & & & \\ & & a_{N-3} & b_{N-2} & c_{N-2} \\ & & & a_{N-2} & b_{N-1} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{N-2} \\ m_{N-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N-2} \\ v_{N-1} \end{bmatrix} \quad (4.63)$$

(4.63)'teki lineer denklem sisteminin köşegen olması bir avantajdır ve tek bir çözüme sahiptir. h_k değeri $h_k = x_{k+1} - x_k$ olarak bölüm 4.2.3'de tanımlanmıştı. Buradan çekilen $x_{k+1} = x_k + h_k$ değeri (4.56) eşitliğinde yerine konur ve $d_k = (y_{k+1} - y_k) / h_k$ olduğu da gözönüne alınırsa (4.56) denklemi

$$S_k(x) = \frac{m_{k+1} - m_k}{6h_k} (x - x_k)^3 + \frac{m_k}{2} (x - x_k)^2 + \left[d_k - \frac{h_k (2m_k + m_{k+1})}{6} \right] (x - x_k) + y_k \quad (4.64)$$

şekline gelir. $S_k(x)$ kübik spline fonksiyonunun katsayıları, (4.64) denklemi ve λ özelliği karşılaştırılırsa,

$$S_{k,0} = y_k \quad S_{k,1} = d_k - \frac{h_k (2m_k + m_{k+1})}{6} \quad (4.65a)$$

$$S_{k,2} = \frac{m_k}{2} \quad S_{k,3} = \frac{m_{k+1} - m_k}{6h_k} \quad (4.65b)$$

şeklinde elde edilir. $\{m_k\}$ katsayıları saptandıktan sonra $S_k(x)$ kübik spline fonksiyonunun katsayıları (4.65) formüllerinden hesap edilir.

Her $S_k(x)$ kübik polinomu, hesaplama işleminin verimli yapılabilmesi için, diğer bir deyişle bilgisayarda seri açılımlarından kaçınabilmek için,

$$S_k(x) = [(S_{k,3} \omega + S_{k,2}) \omega + S_{k,1}] \omega + y_k, \quad \omega = x - x_k \quad (4.66)$$

şeklinde yazılabilir. Burada her $S_k(x)$, için $x_k \leq x \leq x_{k+1}$ aralığında kullanılır.

4.2.2.3- Kübik Spline'ların Tanımlanması

$(x_0, y_0) \dots (x_N, y_N)$ kübik bir spline'i oluşturan $N+1$ nokta olsun. Tablo 4.10'dan seçilen sınır koşulları ile birlikte 4.60 eşitliği 4.63 şeklinde bir lineer denklem sistemi elde etmek için kullanılabilir. Tridiagonal sistem $m_1, \dots, m_{N-1}$ katsayıları için çözülür. m_0 ve m_N katsayıları ise Tablo 4.10'dan seçilen sınır koşullarından bulunur. (4.65) eşitlikleri spline katsayılarını bulmak için kullanılır. Aşağıdaki lineer denklem sistemleri Tablo 4.10'daki durumlar için kullanılır.

1.Durum: Kenetlenmiş kübik spline; $S'(x_0)$ ve $S'(x_N)$ biliniyor ise 4.58'den $k=0$ için

$$S'(x_0) = -\frac{m_0}{3} h_0 - \frac{m_1}{6} h_0 + d_0$$

$$m_0 h_0 = -3 S'(x_0) - \frac{1}{2} h_0 m_1 + 3 d_0$$

4.61'de $h_0 m_0$ yerine konursa,

$$2[h_0 + h_1]m_1 + h_1 m_2 = u_1 + 3S'(x_0) + \frac{1}{2} h_0 m_1 - 3d_0$$

$$[2h_0 - \frac{1}{2} h_0 + 2h_1]m_1 + h_1 m_2 = u_1 - 3[d_0 - S'(x_0)]$$

$$(\frac{3}{2} h_0 + 2h_1)m_1 + h_1 m_2 = u_1 - 3[d_0 - S'(x_0)]$$

bulunur. Benzer olarak,

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = u_k \quad k=2, 3, \dots, N-2$$

ve

$$h_{N-2}m_{N-2} + 2(h_{N-2} + \frac{3}{2} h_{N-1})m_{N-1} = u_{N-1} - 3[S'(x_N) - d_{N-1}]$$

bulunur.

2.Durum: Doğal spline; eğrinin kapalı bir eğri olmaması durumunda sınır koşulları olarak, iki uç noktasındaki ikinci türevlerin sıfır olduğu, yani eğriliğin sıfır olduğu düşünülerek [17], $S''(x_0)=0$ ve $S''(x_N)=0$ alınırsa (4.61)'den

$$2(h_0 + h_1)m_1 + h_1 m_2 = u_1$$

bulunur. Benzer olarak

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = u_k \quad k=2, 3, \dots, N-2$$

ve

$$S''(x_N) = 0, m_N = 0 \quad \text{ile 4.62'den}$$

$$h_{N-2}m_{N-2} + 2(h_{N-2} + h_{N-1})m_{N-1} = U_{N-1}$$

bulunur.

3.Durum: Uç noktalarındaki $S''(x_0)$ ve $S''(x_N)$ ekstrapolasyon ile elde edilecek değere eşit alınır ve Tablo 4.10'dan alınan m_0 değeri 4.61'de yerine konursa,

$$2[h_0 + h_1]m_1 + h_1m_2 = U_1 - h_0 \left[m_1 - \frac{h_0(m_2 - m_1)}{h_1} \right]$$

$$2[h_0 + h_1]m_1 + h_0m_1 + \frac{h_0^2}{h_1}m_1 + h_1m_2 - \frac{h_0^2}{h_1}m_2 = U_1$$

$$(3h_0 + 2h_1 + \frac{h_0^2}{h_1})m_1 + (h_1 - \frac{h_0^2}{h_1})m_2 = U_1$$

bulunur. Benzer olarak,

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = U_k \quad k=2, 3, \dots, N-2$$

ve m_N değeri 4.62'de yerine konursa,

$$(h_{N-2} - \frac{h_{N-1}^2}{h_{N-2}})m_{N-2} + (2h_{N-2} + 3h_{N-1} + \frac{h_{N-1}^2}{h_{N-2}})m_{N-1} = U_{N-1}$$

bulunur.

4.Durum: Uç noktalarındaki $S''(x_0)$ ve $S''(x_N)$ komşu sabitler olarak alınır, 4.61'den $m_0 = m_1$ ile,

$$(3h_0 + 2h_1)m_1 + h_1m_2 = U_1$$

bulunur. Benzer olarak,

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = U_k \quad k=2,3,\dots,N-2$$

ve 4.62'den $m_N = m_{N-1}$ ile,

$$h_{N-2}m_{N-2} + (2h_{N-2} + 3h_{N-1})m_{N-1} = U_{N-1}$$

bulunur.

5.Durum: $S''(x_0)$ ve $S''(x_N)$ verilmiş ise 4.61'den,

$$2(h_0 + h_1)m_1 + h_1m_2 = U_1 - h_0S''(x_0)$$

bulunur. Benzer olarak,

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = U_k \quad k=2,3,\dots,N-2$$

ve 4.62'den ,

$$h_{N-2}m_{N-2} + 2(h_{N-2} + h_{N-1})m_{N-1} = U_{N-1} - h_{N-1}S''(x_N)$$

bulunur [14].

$(0,0)$, $(1,0.5)$, $(2,2)$ ve $(3,1.5)$ noktalarından geçen splinelerin 5 farklı sınır koşuluna göre bulunması aşağıda verilmiştir.

1.durum kenetli spline: $h_0=1$, $h_1=1$, $h_2=1$, $d_0=0.5$, $d_1=1.5$, $d_2=-0.5$, $U_1=6(d_1-d_0)$ ve $U_2=6(d_2-d_1)$ 'i kullanarak iki eşitlik elde edilir.

$$\left(\frac{3}{2}+2\right)m_1 + m_2 = 6(1.5-0.5) - 3(0.5-0.2)$$

$$m_1 + \left(2 + \frac{3}{2}\right)m_2 = 6(-0.5-1.5) - 3.(-1.0+0.5)$$

Bu eşitliklerden,

$$\begin{bmatrix} 3.5 & 1.0 \\ 1.0 & 3.5 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 5.1 \\ -10.5 \end{bmatrix}$$

olur. Bu lineer denklem sistemi çözülürse $m_1=2.52$ ve $m_2=-3.72$ olarak elde edilir. Tablo 4.10'daki 1. eşitlik m_0 ve m_3 'ü hesaplamak için kullanıldığında,

$$m_0 = 3(0.5 - 0.2) - \frac{2.52}{2} = -0.36$$

$$m_3 = 3(-1.0 + 0.5) + \frac{3.72}{2} = 0.36$$

bulunur. Sonra $m_0=-0.36$, $m_1=2.52$, $m_2=-3.72$ ve $m_3=0.36$ değerleri 4.59 eşitliğinde kullanıldığında spline katsayıları,

$$0 < x < 1 \text{ için } S_0(x) = 0.48x^3 - 0.18x^2 + 0.2x$$

$$1 < x < 2 \text{ için } S_1(x) = -1.04(x-1)^3 + 1.26(x-1)^2 + 1.28(x-1) + 0.5$$

$$2 < x < 3 \text{ için } S_2(x) = 0.68(x-2)^3 + 1.86(x-2)^2 + 0.68(x-2) + 2$$

bulunur.

2. durum doğal spline: h, d ve u için 1. durumdaki değerleri kullanarak,

$$2(1+1)m_1 + m_2 = 6(1.5 - 0.5)$$

$$m_1 + 2(1+1)m_2 = 6(-0.5 - 1.5)$$

elde edilir. Bu eşitliklerden,

$$\begin{bmatrix} 4.0 & 1.0 \\ 1.0 & 4.0 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 6.0 \\ -12.0 \end{bmatrix}$$

lineer denklem sistemi elde edilir ve çözüm yapılırsa $m_1=2.4$ ve $m_2=-3.6$ bulunur. $m_0=S''(x_0)=0$ ve $m_3=S''(x_3)=0$ olduğu için 4.65 eşitlikleri kullanıldığında spline katsayıları,

$$0 \leq x \leq 1 \quad \text{için} \quad S_0(x) = 0.4x^3 + 0.1x$$

$$1 \leq x \leq 2 \quad \text{için} \quad S_1(x) = -(x-1)^3 + 1.2(x-1)^2 + 1.3(x-1) + 0.5$$

$$2 \leq x \leq 3 \quad \text{için} \quad S_2(x) = 0.6(x-2)^3 - 1.8(x-2)^2 + 0.7(x-2) + 2.0$$

şeklinde olur.

3. durum: Lineer sistem,

$$\begin{bmatrix} 6.0 & 0.0 \\ 0.0 & 6.0 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 6.0 \\ -12.0 \end{bmatrix}$$

şeklinde oluşur ve $m_1=1.0$ ve $m_2=-2.0$ bulunur. Tablo 4.10'dan yararlanarak m_0 ve m_3 değerleri,

$$m_0 = 1.0 - (-2.0 - 1.0) = 4.0$$

$$m_3 = -2.0 + (-2.0 - 1.0) = -5.0$$

olarak hesaplanır ve spline katsayıları,

$$0 \leq x \leq 1 \quad \text{için} \quad S_0(x) = -0.5x^3 + 2.0x^2 - x$$

$$1 \leq x \leq 2 \quad \text{için} \quad S_1(x) = -0.5(x-1)^3 + 0.5(x-1)^2 + 1.5(x-1) + 0.5$$

$$2 \leq x \leq 3 \quad \text{için} \quad S_2(x) = -0.5(x-2)^3 - (x-2)^2 + (x-2) + 2.0$$

olarak bulunur.

4. durum: $S''(x)$, uç noktalarda komşu sabitler olarak alınır. Lineer sistem,

$$\begin{bmatrix} 5.0 & 1.0 \\ 1.0 & 5.0 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 6.0 \\ -12.0 \end{bmatrix}$$

şeklinde oluşur ve $m_1=1.75$ ve $m_2=-2.75$ bulunur. $S''(x)$, uç noktaların komşularındaki sabitler olduğu için $m_0=1.75$ ve $m_3=2.75$ olarak seçilir ve spline katsayıları,

$$\begin{aligned} 0 \leq x \leq 1 \quad \text{için} \quad S_0(x) &= 0.875x^2 - 0.375x \\ 1 \leq x \leq 2 \quad \text{için} \quad S_1(x) &= -0.75(x-1)^3 + 0.875(x-1)^2 + \\ &\quad + 1.375(x-1) + 0.5 \\ 2 \leq x \leq 3 \quad \text{için} \quad S_2(x) &= -1.375(x-2)^2 + 0.875(x-2) + 2.0 \end{aligned}$$

şeklindedir.

5. durum: Lineer sistem,

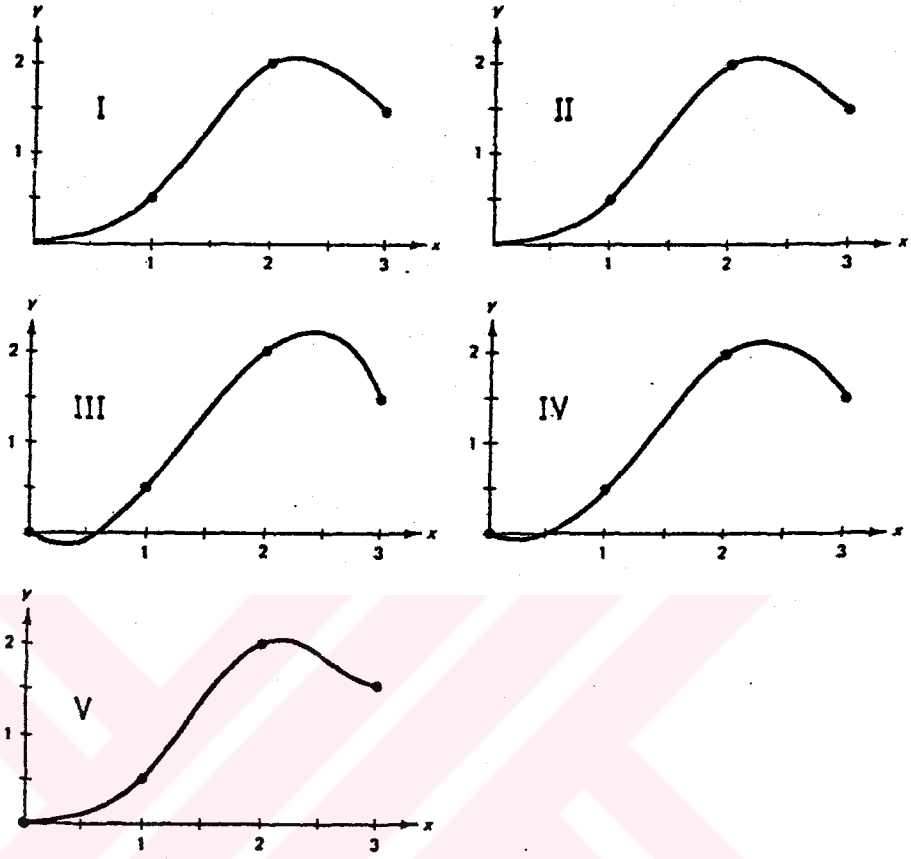
$$\begin{bmatrix} 4.0 & 1.0 \\ 1.0 & 1.0 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 6.3 \\ -15.3 \end{bmatrix}$$

şeklindedir ve çözüm $m_1=2.7$ ve $m_2=-4.5$ bulunur. $m_0=S''(x_0)$ ve $m_3=S''(x_3)$ olarak alındığından $m_0=-0.3$ ve $m_3=3.3$ olur. Bu değerler için spline katsayıları,

$$\begin{aligned} 0 \leq x \leq 1 \quad \text{için} \quad S_0(x) &= 0.5x^3 - 0.15x^2 + 0.15x \\ 1 \leq x \leq 2 \quad \text{için} \quad S_1(x) &= -1.2(x-1)^3 + 1.35(x-1)^2 + 1.35(x-1) + 0.5 \\ 2 \leq x \leq 3 \quad \text{için} \quad S_2(x) &= 1.3(x-2)^3 - 2.25(x-2)^2 + 0.45(x-2) + 2 \end{aligned}$$

şeklinde olur.

Yukarıda her bir durum için verilen çözümlerin kübik splineleri Şekil 4.7'de görülmektedir.



Şekil 4.7. Kübik Splineler

4.3- Fourier Serileri ve Trigonometrik Polinomlar

4.3.1- Fourier Serileri

Bilim adamları ve mühendisler, sık sık ses ve ışık gibi periyodik karaktere sahip fiziksel olayları incelerler. Bunlar periyodik $f(x)$ fonksiyonları ile tanımlanırlar ve bütün x 'ler için,

$$g(x+p) = g(x) \quad (4.67)$$

şeklinde gösterilirler. p sayısına fonksiyonun periyodu denir.

Bu konu için 2π periyotlu fonksiyonları incelemek yeterli olur. Eğer $g(X)$, P periyotlu bir fonksiyon ise $X=px/2\pi$ değişken dönüşümü ile $g(X)$ fonksiyonu 2π periyotlu bir fonksiyon haline dönüşür. (x değişkeninin periyoda T ise $P=P.T/2\pi$ 'den $T=2\pi$ olur) Böylece P periyotlu $g(X)$ fonksiyonu,

$$g(X+P)=g\left(\frac{Px}{2\pi}+T\right)=g\left(\frac{Px}{2\pi}+2\pi\right)=g\left(\frac{Px}{2\pi}\right)=g\left(\frac{Px}{2\pi}\right)=f(x)$$

(4.68)

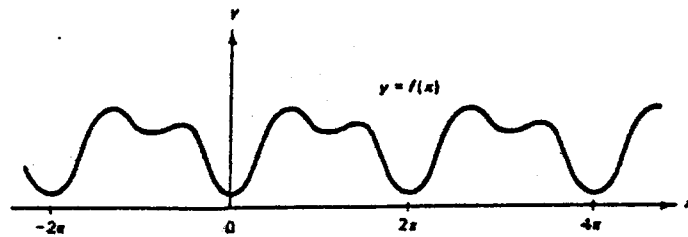
şeklinde 2π periyotlu $f(x)$ fonksiyonu durumuna gelir. Bundan sonra bu bölümde, $f(x)$ 'in 2π periyotlu periyodik bir fonksiyon olduğu varsayılacaktır. Yani bütün x 'ler için,

$$f(x+2\pi)=f(x)$$

(4.69)

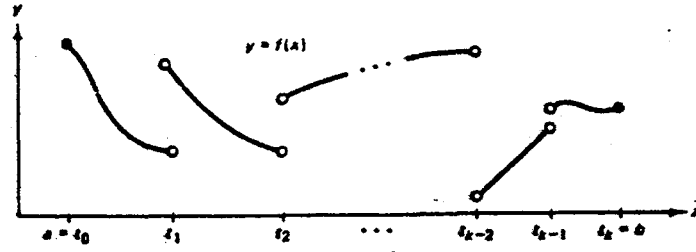
olur. $y=f(x)$ 'in grafiği, Şekil 4.8'de gösterildiği gibi, 2π uzunluğundaki her aralıkta grafiğin parçasını tekrarlayarak elde edilir.

$\sin(jx)$ ve $\cos(jx)$ fonksiyonları 2π periyotlu fonksiyonlara örnek olarak gösterilebilir. (j pozitif tam sayıdır). Burada, herhangi bir 2π periyotlu fonksiyon, $a_j \cos(jx)$ ve $b_j \sin(jx)$ terimlerinin toplamı ile gösterilebilir mi sorusu ile karşılaşılır. İleride yanıtın evet olduğu görülecektir.



Şekil 4.8. 2π Periyotlu Bir $f(x)$ Fonksiyonu

t_i noktaları, $a=t_0 < t_1 < t_2 \dots t_{k-1} < t_k=b$ olmak üzere, $f(x)$ fonksiyonu her $t_{i-1} < x < t_i$ açık aralığında sürekli ve t_i noktalarında sağdan ve soldan limitlere limitlere sahipse $f(x)$ fonksiyonu $[a,b]$ kapalı aralığında parça parça sürekli dir denir. (Şekil 4.9)



Şekil 4.9. $[a,b]$ 'de Parça Parça Sürekli Bir Fonksiyon

Eğer $f(x)$ 2π periyotlu periyodik bir fonksiyon, $f(x)$ ve $f'(x)$ parça parça sürekli ise $f(x)$ fonksiyonunun Fourier serisi,

$$\frac{a_0}{2} + \sum_{j=1}^{\infty} [a_j \cos(jx) + b_j \sin(jx)] \quad (4.70)$$

dir. Buradaki $\{a_j\}$ ve $\{b_j\}$ katsayıları Euler tarafından,

$$a_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(jx) dx \quad j=0,1,2,\dots \quad (4.71)$$

$$b_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(jx) dx \quad j=1,2,3,\dots \quad (4.72)$$

şeklinde verilmiştir [15].

(4.70) Fourier serisindeki $a_0/2$ sabit teriminin $1/2$ çarpanı, (4.71) genel formülünden $j=0$ ile a_0 katsayısının elde edilebilmesi için konulmuştur. (4.70) Fourier serisi x 'in bütün değerleri için $f(x)$ fonksiyonuna

yakınsar. (4.70) fourier serisi, $f(x)$ 'in süreksiz olduğu noktalarda ise fonksiyonun bu süreksiz olduğu noktadaki soldan ve sağdan limitlerinin ortalamasına yakınsar.

$([f(x^-)+f(x^+)])/2$). Fourier serisi gösterimi,

$$f(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} [a_j \cos(jx) + b_j \sin(jx)] \quad (4.73)$$

şeklinde elde edilir.

Örnek olarak $f(x)=x/2$ fonksiyonunun $-\pi < x < \pi$ aralığında

$$f(x) = \sum_{j=1}^{\infty} \frac{(-1)^{j+1}}{j} \sin(jx) = \sin(x) - \frac{\sin(2x)}{2} + \frac{\sin(3x)}{3} - \dots$$

fourier serisi ile gösterilebileceği aşağıdaki gibi kanıtlanabilir. (4.71) ve (4.72) Euler integralleri kullanılarak,

$$a_j = \frac{1}{\pi} \int_{-\pi}^{\pi} \frac{x}{2} \cos(jx) dx = \frac{x \sin(jx)}{2\pi j} + \frac{\cos(jx)}{2\pi j^2} \Big|_{-\pi}^{\pi} = 0$$

$$j=1,2,3,\dots$$

ve

$$\begin{aligned} b_j &= \frac{1}{\pi} \int_{-\pi}^{\pi} \frac{x}{2} \sin(jx) dx = \frac{-x \cos(jx)}{2\pi j} + \frac{\sin(jx)}{2\pi j^2} \Big|_{-\pi}^{\pi} = \\ &= \frac{(-1)^{j+1}}{j} \quad j=1,2,3,\dots \end{aligned}$$

elde edilir. a_0 katsayısı ayrı bir hesaplama ile

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} \frac{x}{2} dx = \frac{x^2}{4\pi} \Big|_{-\pi}^{\pi} = 0$$

olarak bulunur. Yukarıdaki hesaplamalar kosinüs fonksiyonunun tüm katsayılarının sıfır olduğunu gösterir.

$f(x)$ 'in kısmi toplamı,

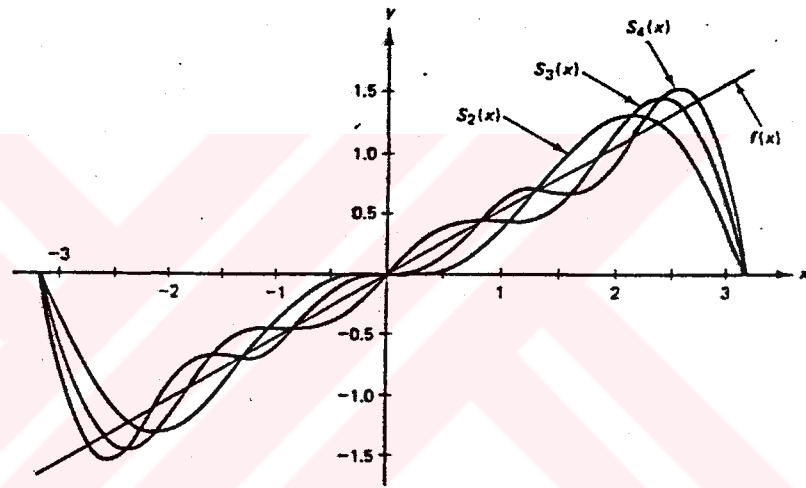
$$S_2(x) = \sin(x) - \frac{\sin(2x)}{2}$$

$$S_3(x) = \sin(x) - \frac{\sin(2x)}{2} + \frac{\sin(3x)}{3}$$

ve

$$S_4(x) = \sin(x) - \frac{\sin(2x)}{2} + \frac{\sin(3x)}{3} - \frac{\sin(4x)}{4}$$

olup grafikleri Şekil 4.10.'da gösterilmiştir [14].



Şekil 4.10. $[-\pi, \pi]$ Kapalı Aralığında $f(x)=x/2$ Fonksiyonu ve $S_2(x)$, $S_3(x)$ ve $S_4(x)$ Fourier Serisi ile Yaklaşımları.

Fourier serilerinin bazı genel özellikleri şu şekildedir:

1- Fourier Serilerinin Eklenmesi: Eğer $f(x)$ ve $g(x)$ 'in Fourier gösterimi varsa, o zaman $h(x)=f(x)+g(x)$ 'in de Fourier gösterimi vardır ve $f(x)$ ve $g(x)$ 'in katsayılarının eklenmesi ile elde edilebilir.

2- Kosinüs Serisi: $f(x)$, her x için $f(-x)=f(x)$ şeklinde bir çift fonksiyon olduğunda eğer $f(x)$, 2π periyotlu ve $f(x)$ ve $f'(x)$ parça parça sürekli ise o zaman $f(x)$ için Fourier serisi yalnızca kosinüslü terimleri içerir.

$$f(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} a_j \cos(jx) \quad (4.74)$$

Burada,

$$a_j = \frac{2}{\pi} \int_0^{\pi} f(x) \cos(jx) dx \quad j=0,1,2,\dots \quad (4.75)$$

dir.

3- Sinüs Serisi: $f(x)$, her x için $f(-x) = -f(x)$ şeklinde bir tek fonksiyon olduğunda eğer $f(x)$, 2π periyotlu ve $f(x)$ ve $f'(x)$ parça parça sürekli ise o zaman $f(x)$ için Fourier serisi yalnızca sinüslü terimleri içerir.

$$f(x) = \sum_{j=1}^{\infty} b_j \sin(jx) \quad (4.76)$$

Burada

$$b_j = \frac{2}{\pi} \int_0^{\pi} f(x) \sin(jx) dx \quad j=1,2,3,\dots \quad (4.77)$$

dir.

Örneğin $f(x) = |x|$ fonksiyonunun $-\pi < x < \pi$ aralığında,

$$\begin{aligned} f(x) &= \frac{\pi}{2} - \frac{4}{\pi} \sum_{j=1}^{\infty} \frac{\cos((2j-1)x)}{(2j-1)^2} \\ &= \frac{\pi}{2} - \frac{4}{\pi} \left[\cos(x) + \frac{\cos(3x)}{3^2} + \frac{\cos(5x)}{5^2} + \dots \right] \end{aligned} \quad (4.78)$$

şeklinde Fourier'in kosinüs serisi ile gösterilebileceği aşağıdaki gibi kanıtlanabilir.

$f(x)$ fonksiyonu bir çift fonksiyondur. 2. özellik nedeni ile yalnızca $\{a_j\}$ katsayılarının hesaplanması yeterlidir. 4.75'ten,

$$a_j = \frac{2}{\pi} \int_0^{\pi} x \cos(jx) dx = \frac{2x \sin(jx)}{\pi j} + \frac{2 \cos(jx)}{\pi j^2} \Big|_0^{\pi}$$

$$= \frac{2 \cos(j\pi) - 2}{\pi j^2} = \frac{2[(-1)^j - 1]}{\pi j^2} \quad j=1,2,3,\dots$$

bulunur. j çift olduğunda $[(-1)^j - 1] = 0$ olduğundan, kosinüs serisi yalnızca tek indisli terimleri içerir. Tek indisli katsayılar,

$$a_1 = \frac{-4}{\pi}, \quad a_3 = \frac{-4}{\pi 3^2}, \quad a_5 = \frac{-4}{\pi 5^2} \dots$$

olarak elde edilir. a_0 katsayısı ayrı bir hesaplama ile,

$$a_0 = \frac{2}{\pi} \int_0^{\pi} x dx = \frac{x^2}{\pi} \Big|_0^{\pi} = \pi$$

şeklinde bulunur. Böylece 4.78'deki istenen katsayılar bulunmuş olur.

4.3.2- Periyodik Fonksiyon Uydurma

Dengeleyen eğri olarak uydurulan Fourier serisine burada "Trigonometrik Polinom" adı verilecektir.

$[-\pi, \pi]$ kapalı aralığı,

$$-\pi = x_0 < x_1 < \dots < x_N = \pi \quad x_k = -\pi + 2\pi k/N \quad (4.79)$$

olmak üzere, x 'in eşit dağılmış değerleri tarafından eşit parçalara bölünmüş olsun. $y_0 = y_N$ olmak üzere, $\{(x_k, y_k) : k=0,1,\dots,N\}$ veri noktalarına en iyi uyan periyodik bir fonksiyon geçirilecektir. Eğer $2M+1 \leq N$ ise, o zaman,

$$P(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} [a_j \cos(jx) + b_j \sin(jx)] \quad (4.80)$$

şeklindeki bir trigonometrik fonksiyonun uydurulabilmesi için yeterli sayıda nokta vardır. En küçük kareler yöntemi $\{a_j\}$ ve $\{b_j\}$ katsayılarının bulunması için

kullanılabilir. Bunun için,

$$E(P(x)) = \sum_{k=1}^N [P(x_k) - y_k]^2 \quad (4.81)$$

şeklindeki minimum fonksiyonu kurulur ve bu fonksiyonu minimum yapan $\{a_i\}$, $\{b_i\}$ kümeleri Fourier serisinin katsayıları olur. (4.79)'daki gibi noktalar eşit dağıldığında (4.80)'deki katsayıların hesaplanması aşağıda görülmektedir.

Trigonometrik polinom uydurma; $y_0 = y_N$ olmak üzere, $N-1$ tane $\{(x_k, y_k)\}$ veri noktasının (4.79)'daki gibi absislerinin eşit dağılmış olduğu kabul edildiğinde, eğer $2M+1 \leq N$ ise (4.81)'deki ifadeyi minimum yapan (4.80) Fourier serisinin katsayıları,

$$a_j = \frac{2}{N} \sum_{k=1}^N f(x_k) \cos(jx_k) \quad j=0,1,2,\dots,M \quad (4.82)$$

ve

$$b_j = \frac{2}{N} \sum_{k=1}^N f(x_k) \sin(jx_k) \quad j=1,2,3,\dots,M \quad (4.83)$$

olurlar.

(4.82) ve (4.83) formülleri en küçük kareler prosesi ile tanımlanmasına rağmen bunlar (4.71) ve (4.72)'deki Euler formüllerindeki integrallere sayısal yaklaşımlar olarak da görülebilir. Euler formülleri sürekli bir fonksiyonun Fourier serisi için katsayıları, oysa (4.82) ve (4.83) formülleri veri noktalarına eğri uydurma için trigonometrik polinomun katsayılarını verir.

Örnek olarak $f(x)=x/2$ fonksiyonu ile üretilen veri noktaları kullanılmıştır. Daha fazla nokta kullanıldığında trigonometrik polinom katsayıları, Fourier serisi katsayılarına daha yakın olurlar.

5. dereceden bir trigonometrik polinom için
(M=5),

$$x_k = -\pi + k \frac{\pi}{6}, \quad k=1,2,\dots,12$$

ifadesinden elde edilen eşit aralıklı absis ve $f(x)=x/2$ 'den elde edilen fonksiyon değerleriyle üretilen 12 veri noktası arasından geçen trigonometrik polinom aşağıda bulunmuştur. Bu trigonometrik polinomun katsayıları, aynı şekilde üretilen 60 ve 360 veri noktası kullanılarak elde edilen trigonometrik polinomun katsayıları ve sayfa 80'deki örnekteki aynı fonksiyonun Fourier serisi açılımının ilk beş terimi karşılaştırılmıştır.

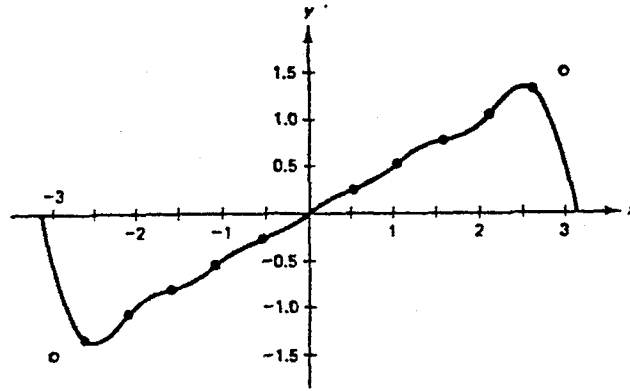
Trigonometrik polinomun periyodik olarak devam edeceği varsayıldığı için, bir süreksizlik noktasında, örneğin π noktasında $f(\pi)$ fonksiyon değeri,

$$f(\pi) = \frac{f(\pi^-) - f(\pi^+)}{2} = \frac{\pi/2 - \pi/2}{2} = 0 \quad (4.84)$$

formülü ile hesaplanır [14]. $f(x)$ fonksiyonu bir çift fonksiyondur, bu nedenle sinüslü terimlerin bütün katsayıları sıfırdır. (Yani j 'nin her değeri için $a_j=0$) Böylece 5. dereceden trigonometrik polinom 4.82 ve 4.84 formüllerinden,

$$\begin{aligned} P(x) = & 0.9770486 \cos(x) - 0.4534498 \cos(2x) + 0.26179938 \\ & \cos(3x) - 0.1511499 \cos(4x) + 0.0701489 \cos(5x) \end{aligned} \quad (4.85)$$

olarak elde edilir. $P(x)$ 'in grafiği Şekil 4.11'de görülmektedir.



Şekil 4.11. $y=x/2$ Çizgisi Üzerinde Belirlenmiş 12 Veri Noktasından Elde Edilen 5. dereceden $P_5(x)$ Trigonometrik Polinomu

5. dereceden trigonometrik polinomun katsayıları, enterpolasyon noktalarının sayısı 60'a ve 360'a çıktığında yavaşça değişir. Noktaların sayısı arttıkça katsayılar $f(x)$ 'in Fourier serisi açılımı ile elde edilen katsayılarla yaklaşır. Sonuçlar Tablo 4.11'de verilmiştir. Tablo 4.11'den, çeşitli sayıdaki veri noktalarından elde edilen trigonometrik polinom katsayıları ile Fourier serisi katsayıları ve artan veri noktaları sayısına göre trigonometrik polinom katsayılarındaki değişim kolaylıkla karşılaştırılabilir.

Tablo 4.11. $[-\pi, \pi]$ 'de $f(x)=x/2$ 'ye Yaklaşımlar İçin Trigonometrik Polinom Katsayılarının Karşılaştırılması

	Trigonometrik Polinom Katsayıları			Fourier Serisi Katsayıları
	12 Nokta	60 Nokta	360 Nokta	
b_1	0.97704862	0.99908598	0.99997462	1.0
b_2	-0.45344984	-0.49817096	-0.49994923	-0.5
b_3	0.26179939	0.33058726	0.33325718	0.33333333
b_4	-0.15114995	-0.24633386	-0.24989845	-0.25
b_5	0.07014893	0.19540972	0.19987306	0.2

BÖLÜM 5

5- SONUÇ VE ÖNERİLER

Sayısal arazi modelinin oluşturulması sürecinde eş yükselti eğrilerinin geçtiği noktaların belirlenmesi bu noktaların uygun bir şekilde birleştirilmesi ve bu işlemler sırasında elde edilen bilgilerin mühendislik amaçları için kullanılmasının optimum bir şekilde gerçekleştirilmesi ancak bilgilerin sıralı olması durumunda mümkündür. (3.2) eşitliğinden oluşturulan Tablo 5.1'deki değerler incelenirse bilgilerin sıralanmasının kesin gerekliliği ortaya çıkar. Bu tabloda N dizinin eleman sayısı olup birinci kolondaki k, sırasız dizilerde 2. kolondaki k ise sıralı dizilerde bilgiye ulaşmak için gerekli maksimum karşılaştırma sayısını göstermektedir.

Tablo 5.1. Sıralı ve Sırasız Dizilerde Karşılaştırma Sayısı

k=N	$k=\log_2 N$
1	2
100	7
200	8
500	9
1000	10
2000	11
5000	13
10000	14
20000	15
50000	16
100000	17
1000000	20

Tablo 5.1'den bilgilerin sıralı olmasının gerekliliği tartışma götürmez bir şekilde açıktır. Sayısal

arazi modeli gibi bilgilere tekrar tekrar ulaşılmasını gerektiren uygulamalarda sıralama işleminin önemi büyüktür.

Sayısal arazi modeli, diğer mühendislik dallarına da hizmet üretmekte kullanılır. Sayısal arazi modeli konusunda hız ve bellek faktörünü gözardı etmek bu hizmetlerin de yeterli ve etkin bir şekilde gerçekleştirilmesini engeller. Bu nedenlerle sayısal arazi modeli yazılımlarının hız açısından verimliliğini olurunca artırmak gerekir. Sıralama işlemi, hız artırımını sağlayan tek yoldur. Hız artırımını olurunca büyütmek de sıralama yöntemlerini karşılaştırmak ve bunların özelliklerini iyi bilmek ve en hızlı yöntemi seçmekle mümkündür. Bunun için bölüm 3'te sıralama yöntemleri ayrıntılı bir şekilde incelenmiştir. Bu bölümde yapılan araştırmalara dayanılarak,

1. Sayısal Arazi Modelinin oluşturulması için geliştirilecek programlarda sıralama yöntemlerinin kullanılması,

2. Sıralama programları içinde Quicksort algoritmasının kullanılması, önerilir.

Sayısal arazi modeli konusu içinde en çok karşılaşılan problem eş yükselti eğrili planların çizilmesidir. Sayısal yükseklik modelinde yapılan enterpolasyon işlemlerinden sonra aynı kotlu noktaların birleştirilmesi gerekir. Birleştirme işlemi, haritacılığın gerektirdiği kurallara ve doğruluğa uygun şekilde gerçekleştirilmelidir. Bunun için, bu noktalardan geçen uygun bir fonksiyonun (matematiksel modelin) seçilmesi gerekir. Burada dengeleyen eğri aynı kotlu noktalardan geçmediği için dengeleyen eğri modellerinin seçilmesi ve bunların parametrelerinin bulunması söz konusu değildir. Aynı kotlu n noktadan geçecek şekilde n . dereceden bir polinomun seçilmesi ve parametrelerin belirlenmesi, büyük salınımlara neden

olduđu, haritacılık kurallarına ve doğruluđuna uygun olmadığı için önerilmemektedir. Bölüm 4'de bu durum bir örnekle gösterilmiştir. Trigonometrik seriler ve polinomlar da aynı gerekçe ile ve bu fonksiyonların periyodik olaylarda uygun olması, eş yükselti eğrilerinin ise periyodik olmaması nedeni ile salık verilmemektedir. Ayrıca eş yükselti eğrilerinin tek değerlikli fonksiyonlarla ifadesi olanaksız olduđu için de yüksek dereceli polinomların kullanılması mümkün değildir.

Aynı kotlu noktalardan geçen eş yükselti eğrilerinin özelliđini en iyi bir biçimde sağlayan matematiksel model olarak spline fonksiyonları görölmektedir. Bu nedenle aynı kotlu noktalardan geçen eğrilerin çizimi için spline fonksiyonları önerilir.

KAYNAKLAR

- [1] GÜLER,A., Sayısal Arazi Modellerinde Enterpolasyon Yöntemleri, Harita Dergisi, Ocak 1978, Sayı 85.
- [2] ÖZER,H., Sayısal Arazi Modeli Oluşturma Yöntemleri, Harita Dergisi, Ocak 1989, Sayı 102.
- [3] GÜLER,A., Sayısal Arazi Modellerinde İki Enterpolasyon Yöntemi ile Denemeler, Harita ve Kadastro Müh. Dergisi, 1985, Sayı 52-53
- [4] MILLER,C.L., LAFLAMME,R.A., The Digital Terrain Model Theory and Application, Photogrammetric Engineering and Remote Sensing, 24,3,1958
- [5] KOYUNCU,D., Sayısal Arazi Modelleri, Harita Dergisi, Temmuz 1981, Sayı 87.
- [6] YANALAK,M., Sayısal Arazi Modelleri ve Kullanılan Enterpolasyon Yöntemleri, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Ens., Ocak 1991.
- [7] SHILDT,H., Advanced Turbo C, Borland-Osborne/McGraw-Hill, Programming Series, 1987
- [8] PRESS,W.H., FLANNERY, B.P, TEUKOLSKY,S.A., WATTERING, W.T., Numerical Recipes in C, Cambridge University, Press, 1988
- [9] TREMBLAY,J.P., BUNT,R., An Introduction to Computer Science: An Algorithmic Approach, McGraw-Hill Computer Science Series, International Student Edition, 1979.
- [10] ÖZTAN,O., Sıralama Algoritmaları Üzerine Bazı Öneriler, Yayınlanmamış Makale
- [11] MOSTELLER,F., ROURKE,E.K.R., THOMAS,Jr.G.B., Probability With Statistical Applicatinos, Addison Wesley Publishing Company, Inc, 1961
- [12] ŞENMAN,R., Doğrusal Olmayan Eğri Uydurma, Yüksek Lisans Tezi, İTÜ Fen Bilimleri Enst., 1984.

- [13] BAYKAL, O., ÖZTAN, O., GÜRDOĞAN, İ.H., Dengeleyen Eğri Olarak Polinom ve Fourier Serisi, İTÜ Dergisi, Cilt 41, No 1-2, İstanbul, 1983.
- [14] MATTEWS, J.H., Numerical Methods, Prentice-Hall, Inc, Englewood Cliffs, New Jersey 07632, 1987
- [15] SOKOLNIKOF, I.S., REDHEFFER, R.M., Mathematics of Physics and Modern Engineering, McGraw-Hill Book Company, Kogakusha Company, Ltd. International Student Edition, 1966.
- [16] AKTAŞ, Z., ÖNCÜL, H., URAL, S., Sayısal Çözümleme, Cilt 1, ODTÜ Bilgisayar Mühendisliği Bölümü, 1981.
- [17] GÜLER, A., Ardışık Noktaların Parametrik Kübik Splaynlarla Birleştirilmesi, Harita ve Kadastro Müh. Dergisi, Sayı 56-57, 1986.

ÖZGEÇMİŞ

1967 yılında Bulancak'ta doğmuştur. Lise öğrenimini Trabzon Lisesinde tamamladıktan sonra 1984 yılında İ.T.Ü. İnşaat Fakültesi, Jeodezi ve Fotogrametri Mühendisliği Bölümünde lisans öğrenimine başlamıştır. 1988 yılında lisans öğrenimini tamamlayıp aynı yıl İ.T.Ü. Fen Bilimleri Enstitüsü İnşaat Anabilim Dalı, Jeodezi ve Fotogrametri Mühendisliği programında Yüksek Lisans öğrenimine başlamıştır. 1989 yılında İ.T.Ü. İnşaat Fakültesi Jeodezi ve Fotogrametri Mühendisliği Bölümünde Araştırma Görevlisi olarak göreve başlamış ve halen bu görevi sürdürmektedir.

W. G.
Yükseköğretim Kurulu
Doküman Yönetim Merkezi