

Kırgızistan-Türkiye Manas Üniversitesi Yayınları: 224
Ders Kitapları Dizisi: 64

MATLAB ile
KİMYA MÜHENDİSLİĞİNDE
MATEMATİKSEL MODELLEME

Prof. Dr. Selahattin GÜLTEKİN

Bişkek-2017

УДК 519.8
ББК 22.18
Г 98

Г 98 Гүлтекин С. **Matlab İle Kimya Mühendisliğinde Matematiksel Modelleme** –Б.: КТМУ. 2017 – 148 б.

ISBN 978-9967-9038-2-1

Математикалык Моделдөө, инженердик тармакта дагы, табигый-так илимдер тармагында дагы өтө кеңири колдонулууда. Моделдөөнүн максаты, ар кандай кубулуштардын физикалык жана/же химиялык абалыны математикалык теңдемелер аркылуу туюнтуу. Көпчүлүк учурларда практикалык иш жасоо мүмкүн эмес же болбосо өтө көп каражатты талап кылат. Бул себептен боло турган кубулуштарды теңдемелер аркылуу туюнтуп, алына турган жыйынтыктарды алдын ала айтууга болот. Моделдөөдө теңдемелер өтө жөнөкөй да өтө татаал да болушу мүмкүн. Ар кандай кабыл алуулардын (таануулардын) негизинде теңдеменин жөнөкөй же татаал экендиги белгиленет. Компьютерлердин татаал математикалык теңдемелер системаларын чыгарууда жана симуляциясында колдонулуусу, азыркы учурда математикалык моделдөөнүн маанилүүлүгүн арттырууда. Мындай математикалык моделдөө үчүн иштелип чыккан ар кандай техникалык эсептөө пакет программаларын түрлөрү дүйнөдө өтө көп. Бул китепте болсо, дүйнөдө өтө кеңири тараган MATLAB (англисче «Matrix Laboratory» сөзүнүн кыскартылганы) Пакет Программасын колдонуу аркылуу моделдөө теңдемелерин чыгарылыштарын алдык.

Matematiksel Modelleme, hem mühendislik ve hem de fen alanında yaygın olarak kullanılmaktadır. Modellemeden kasıt, olayın fiziksel ve/veya kimyasal durumunu matematiksel denklemlerle ifade etmektir. Çoğu zaman deney yapmak ya imkânsızdır veya çok pahalıdır. Bu nedenle, masa başında olayı denklemlerle ifade edip, neticeleri kestirmek mümkündür. Modelleme denklemleri çok basit olabileceği gibi çok karmaşık da olabilir. Kabullenmelere göre, denklemlerin basitliği veya karmaşıklığından bahsedilir. Karmaşıklık matematiksel denklem sistemlerinin çözümlerinde ve simülasyonunda bilgisayarların daha fazla kullanılması ile matematiksel modelleme son zamanlarda gittikçe önem kazanmaktadır. Piyasada çok çeşitli hazır teknik hesaplama paket programları vardır. Biz burada, çok yaygın olan MATLAB Paket Programını kullanarak model denklemleri çözmeye çalıştık.

M 1602110000-17

ISBN 978-9967-9038-2-1

УДК 519.8
ББК 22.18
© КТМУ, 2017

ÖNSÖZ

Elinizdeki kitabı, gerek yurtdışında (Suudi Arabistan, Malezya, Kırgızistan), gerekse Türkiye’de çeşitli üniversitelerde verdiğim **Matematiksel Modelleme** ders notlarını bir araya getirerek hazırladım.

İstanbul’daki **Üsküdar Üniversitesi**nden 2016-2017 akademik yılında bir yıllık *sabbatical* izni alarak Kırgızistan’daki *Kırgızistan-Türkiye Manas Üniversitesi*ne (KTMÜ) geldim. Burada kimya mühendisliği bölümünde verdiğim derslerin yanında da bu kitap ortaya çıktı.

Özellikle, İngilizceleri yeterli olmayan öğrencilere yönelik böyle bir kitabın hazırlanması bu coğrafyada önem taşımaktadır.

Kitapta **MATLAB** programı, hemen hemen bütün modelleme çözümlerinde sıkça kullanıldığı için *Bölüm 2’de* yeteri derecede MATLAB ile ilgili bilgiler, örneklerle beraber, verilmiştir.

Bölüm 1’de matematiksel modelleme ile ilgili genel bilgiler sunulmuştur.

Bölüm 3, deneysel verilerin işlenmesi ile ilgili olup, benim çok önem verdiğim bir konudur.

Bölümler 4 ve 5, sırasıyla, lineer olmayan ve lineer denklemlerin çözümü ile ilgilidir.

Bölüm 6, nümerik türev ve nümerik integrasyonu kapsamaktadır.

Bölüm 7, diferansiyel denklemleri işlemektedir.

Bölüm 8’de ise kimya mühendisliğinde önemli olan bazı proses ve birimlerinin modellemesine yer verilmiştir.

Bölüm 9 ise **Mikroskobik** ve **Makroskobik** denkliliklere ayrılmıştır.

Bölüm 10 ve 11 ise, sırasıyla, çözümlü ve çözülecek problemleri kapsamaktadır.

Her bölüm sonunda yeteri kadar çözülmesi gereken problemler verilmiştir. Daha sonraki bölümlere geçmeden önce bunların çözülmesi şiddetle önerilir.

Kırgızistan-Türkiye Manas Üniversitesi huzur dolu bir ortama sahiptir. Trafik problemi, *çoğu öğretim üyeleri KTMÜ kampusunda yaşadıklarından dolayı*, yoktur. Burası kitap yazmak için, bence, dünyanın en uygun yerlerinden birisidir.

Üsküdar Üniversitesine, KTMÜ’ya gelmeme izin verdiği için, şükranlarımı sunarım.

KTMÜ Kimya Mühendisliği Bölümünden araştırma görevlisi **Alimzhan Kolbekov’a**, Bilgisayar Mühendisliği Bölümünden öğretim görevlisi **M. Selim Elmalı’ya** yardımlarından dolayı teşekkür ederim.

Nihayet bu kitabın yazılmasında, direkt veya indirekt olarak, destek olan öğrenci ve öğretim üyesi arkadaşlarıma teşekkürü bir borç bilirim.

Kitabın öğrencilere faydalı olmasını dilerim.

Bütün kritiklere açığız. Herhangi bir öneriniz için (gultekinsss@gmail.com) adresine yazabilirsiniz.

Prof. Dr. Selahattin Gültekin
Bişkek-Kırgızistan
5 Haziran, 2017

İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
BÖLÜM 1- MATEMATİKSEL MODELLEME PRENSİPLERİ.....	1
Amaç.....	1
1.1 Genel Bakış.....	1
1.2 Modellemede Kullanılan Terimler.....	2
Özet.....	6
BÖLÜM 2 - MATLAB© Paket Programı	7
Amaç.....	7
Giriş	7
2.1 MATLAB’ın Genel Kullanımı	8
2.2 Diziler: Vektörler ve Matrisler	10
2.2.1 Vektörler.....	10
Vektör Oluşturma	10
2.3 Vektör ve Matris İşlemleri	12
2.3.1 Matrislerde Toplama ve Çıkarma İşlemleri:	12
2.3.2 Matrislerin Çarpımı:	12
2.3.3 Dizilerin Eleman- Elemana İşlemleri (<i>Element-by-Element Operations</i>).....	12
2.3.4 Matrisin Transpozu.....	13
2.3.5 Bir Matrisin Determinantı (det Komutu).....	13
2.3.6 Matrisin Tersi (inv Komutu)	13
2.3.7 Bir Matrisin Rankı (rank komutu).....	13
2.4 Lineer (Doğrusal) Denklem Sistemleri.....	13
2.5 Polinomlar.....	14
2.5.1 Polinomların Değerini Bulma (polyval Komutu).....	14
2.5.2 Polinomların Köklerini Bulma (roots Komutu)	15
2.5.3 Polinomların Çarpımı (<i>conv</i> Komutu).....	15
2.5.4 Polinomların Birbirleriyle Bölümü (<i>deconv</i> Komutu)	15
2.5.5 İnterpolasyon, Ekstrapolasyon ve Regresyon	16
2.5.6 Verilere Polinom Uydurmak (polyfit Komutu).....	16
2.6 Grafikler.....	17
2.6.1 İki Boyutlu (2D) Grafikler	17
plot Komutu	17
ezplot komutu.....	18
2.7 Programlama.....	19
2.7.1 M-Dosyaları (M-Files)	19
2.7.1.1 Düz-yazı Dosyaları (script files)	19
2.7.1.2 Fonksiyon Dosyaları (<i>function files</i>).....	20
2.7.2 Koşul İfadeleri (Conditional Statements).....	22
2.7.3 Döngüler (Loops)	26
for döngüleri	26
2.8 Sembolik İşlemler	28
2.8.1 MATLAB İle Yapılabilecek Sembolik İşlemler	28
2.8.1.1 İntegrasyon.....	28
Belirsiz İntegraller (int komutu).....	28
Belirli İntegraller	29
2.8.1.2 Sembolik Limit Alma.....	30
2.8.1.3 Cebirsel Denklemlerin Sembolik Çözümü (solve komutu)	30
2.8.1.4 Diferansiyel Denklemlerin Sembolik Çözümü (<i>dsolve</i> komutu).....	31
2.9 Nümerik (Sayısal) İşlemler	34

2.9.1 Matematiksel İşlemler	34
2.9.1.1 Kök Bulma	34
2.9.1.2 Nümerik İntegrasyon.....	35
2.9.1.3 Diferansiyel Denklemlerin Nümerik Çözümü	35
Özet.....	37
PROBLEMLER	37
BÖLÜM 3 - DENEYSEL VERİLERİN FORMÜLE EDİLMESİ.....	41
Amaç.....	41
3.1 Proseslerde Verilerin Direkt ve İndirekt Ölçülmesi	41
3.2 Üstel ve Kuvvet Fonksiyonları	47
3.2.1 Logaritmik Koordinatlar.....	48
3.3 Dağınık Verilere Doğru Uydurma	50
3.3.1 En Küçük Kareler Yöntemi (The Least Squares Method)	50
3.3.2 Regresyon Katsayısı	57
Özet.....	58
PROBLEMLER	58
BÖLÜM 4 - LİNEER OLMAYAN CEBİRSEL DENKLEMLERİN KÖKLERİNİN BULUNMASI.....	60
Amaç.....	60
4.1 Grafik Yöntem	60
4.2 Nümerik Yöntemler	61
4.2.1 Basit İterasyon Yöntemi.....	61
4.2.2 Newton-Raphson (N-R) Yöntemi	62
4.2.3 Yarıya Bölme Yöntemi (Bisection Method).....	64
4.2.4 Regula-Falsi, R-F, (Yer Değiştirme) Yöntemi.....	64
4.2.5 Secant Yöntemi	66
Özet.....	66
PROBLEMLER	66
BÖLÜM 5 - LİNEER DENKLEM SİSTEMLERİ	68
Amaç.....	68
Giriş	68
5.1 Lineer Denklem Sistemlerinin Çözümü	68
Gauss Eleme Yöntemi.....	68
Özet.....	71
PROBLEMLER	71
BÖLÜM 6 - NÜMERİK TÜREV VE NÜMERİK İNTEGRASYON.....	74
Amaç.....	74
6.1 Nümerik Türev.....	74
6.2 Nümerik İntegrasyon	75
6.2.1 Yamuklar Yöntemi (Trapezoids Rule).....	76
6.2.2 Simpson Parabolik Yöntemi.....	78
Özet.....	81
PROBLEMLER	81
BÖLÜM 7 - DİFERANSİYEL DENKLEMLERİN ÇÖZÜMLERİ.....	82
Amaç.....	82
7.1 Nümerik Çözümler	82
7.1.1 Euler Tek Adım Yöntemi	82
7.1.2 Runge-Kutta (R-K) Yöntemi.....	84
7.2 Analitik Çözümler.....	86
Adi Diferansiyel Denklemlerin (ODE) Analitik (Sembolik) olarak çözümü	86
Diferansiyel Denklemlerin Bilgisayar Çözümleri.....	86

dsolve Komutu	87
7.2.1 Çoklu Diferansiyel Denklemler.....	89
Özet.....	93
BÖLÜM 8 - KİMYA MÜHENDİSLİĞİNDE ÖNEMLİ BAZI PROSES VE BİRİMLERİNİN MODELLENMESİ	94
Amaç.....	94
8.1 Kanatçıklardan Isı Transferi Modeli	94
8.2 Buharlaştırıcı Modeli	96
8.3 Reaktörlerin Modellenmesi.....	99
8.3.1 Kazan Tipi Kesikli Reaktör (KR) (<i>Batch Reactor</i>).....	100
8.3.2 Tam Karıştırılmalı Sürekli Reaktör (TKSR) (<i>Continuous-Stirred-Tank-Reactor, CSTR</i>).....	101
8.3.3 Piston Akışlı Reaktör (PAR) (<i>Plug-Flow-Reactor, PFR</i>).....	102
8.4 Denge Prosesleri	105
8.4.1 Tek Kademeli Denge Prosesleri	105
8.4.2 Gaz-Sıvı Sistemler.....	106
8.4.3 Ters Akımlı Çok Kademeli Prosesler.....	107
Özet.....	110
BÖLÜM 9 - MİKROSKOBİK VE MAKROSKOBİK DENKLİKLER	112
Amaç.....	112
9.1 Mikroskobik Denklikler.....	112
9.1.1 Süreklilik Denklemi	112
Vektör Notasyonları	114
9.2 Makroskobik Denklikler.....	115
9.2.1 Toplam Kütle Denkliği (Süreklilik Denklemi)	115
9.2.2 Komponent (Bileşen) Denklikleri	117
9.2.3 Enerji Denkliği	119
Özet.....	120
PROBLEMLER	120
BÖLÜM 10 - KİMYA MÜHENDİSLİĞİNDEN SEÇİLMİŞ ÇÖZÜMLÜ ÖRNEKLER	122
Amaç.....	122
BÖLÜM 11 - MATEMATİKSEL MODELLEME İLE İLGİLİ ÇÖZÜLECEK BAZI PROBLEMLER	136
Amaç.....	136
KAYNAKLAR.....	138
DİZİN	140

BÖLÜM 1- MATEMATİKSEL MODELLEME PRENSİPLERİ

Amaç

Bu bölümün bitiminde,

- *Matematiksel Modellemenin ne olduğu*
- *Neden matematiksel modellemeye gerek olduğu*
- *Matematiksel Modelleme yapılırken hangi adımların takip edildiği*
- *Matematiksel modellemenin gerçek-dünya problemlerine nasıl uygulandığı*
- *Nümerik hesaplamaların da analitik hesaplamalar kadar güvenilir olduğu*

gerçeklerini anlamış olacağız

1.1 Genel Bakış

Matematiksel modellemenin anlamı, herhangi bir **fiziksel veya kimyasal olayı laboratuvara götürmeden kâğıt üzerinde matematiksel denklemlerle ifade ederek ortaya koymaktır**. Yani modelleme, fiziksel dünyayı *matematik eşitliklerle* tarif etmeyi amaçlar.

Böylece, eğer proses laboratuvarında gerçekleşecek olsaydı, ne netice alacağımızı laboratuvara gitmeden kâğıt üzerinde tahmin edebiliyoruz. Bir prosesin modellenenebilmesi için modelleyicinin fiziksel ve kimyasal yasaların eksiksiz olarak uygulaması ve prosesi en ince noktalarına kadar çok iyi bilmesi lazımdır. Daha doğrusu, modelleyici sağlam bir eğitim temeline (*background*) sahip olmalıdır.

Matematiksel modellemede temel dört korunum yasasından yararlanılır. Bunlar:

- Kütle**
- Enerji**
- Momentum ve**
- Elektriksel yük**

yasalarıdır. Ayrıca şüphesiz ki *transfer olayları, termodinamik, kinetik, matematik ve bilgisayar* gibi dallardan da büyük yardım görülmektedir. Bu söylediğimiz yasaların hepsini ilgili konularda uygulayacağız.

Neden modellemeye gerek vardır? Bazı hallerde deneysel yollara başvurmak olanaksız olabilir.

Matematiksel olarak formülendirilmiş modeli bir sistemin değişik koşullardaki davranışını öngörmede bir araç olarak kullanılmaktadır. Sistemin simülasyonu (benzetimi) da matematiksel modellemeye dayanmaktadır.

Matematiksel modelleme, mühendislik ve fen alanında, tıp ve ekonomi alanında sıkça kullanılmaktadır. Tıp alanında, örneğin, **kanserli bir hücrenin büyüme hızı**, modelleme ile ortaya konulmaktadır.

Karmaşık matematiksel denklem sistemlerinin çözümlerinde bilgisayarların daha fazla kullanılması ile matematiksel modelleme son zamanlarda gittikçe önem kazanmaktadır.

Endüstriyel problemlerin çoğu lineer olmayan denklemlerle ifade edilmektedir. Lineer denklem(ler) nispeten kolayca çözülür. Ancak, lineer olmayanların bazen *analitik* (tam=exact) çözümü olmayabilir. O zaman da *nümerik* (yaklaşık) çözümlere başvurulur.

Gerek analitik, gerekse nümerik çözümleri manuel yapabiliriz, ancak biz daha çok **MATLAB Teknik Hesaplama Programını** kullanarak çözümleri elde edeceğiz. Şunu özellikle belirtelim ki **matematiksel modellemede esas, olayın doğru bir şekilde denklemlerle formüle edilmesidir. Denklemlerin çözümleri ise, önem bakımından, ikinci sırada gelir.**

Nümerik yöntemleri tek tek ele alıp genel denklemlerini elde ettikten sonra bunları MATLAB’de programlayıp çözeceğiz.

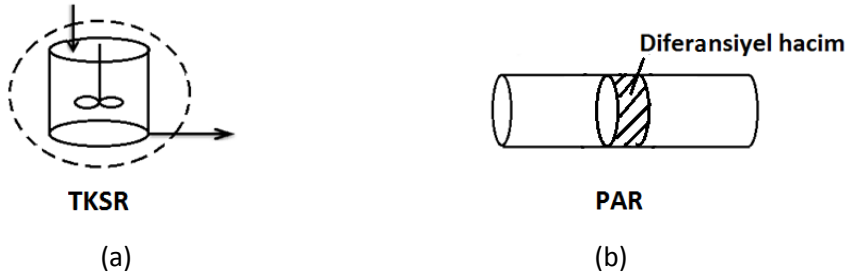
Mühendislik ve fen bilimlerinin her alanında modelleme yapma gereği vardır. Örneğin, kimya mühendisliğinde aşağıdaki alanlarda özellikle modelleme yapılmaktadır:

1. *Taşınum Olayları (Akışkanlar Mekaniği, Isı ve Kütle transferi)*
2. *Kimyasal Reaksiyon Mühendisliği (Reaktör Tasarımı)*
3. *Proses Dinamik ve Kontrol*
4. *Fabrika Tasarımı*
5. *Ekonomi*

Özellikle, *proses dinamik ve kontrol* tamamen matematiksel modelleme üzerine kurulmaktadır.

Modellemede doğru “**kontrol sistemi**” (*kontrol veya diferansiyel hacim* olarak da isimlendirilir) seçimi çok önemlidir. Çünkü bu seçilen kısım modellenen tüm sistemi temsil etmek üzere kullanılacaktır. Kontrol hacmi, sisteme göre veya modelin duyarlılık derecesine ve modelden hangi spesifik sonuçlar istendiğine bağlı olarak değişebilir. Sistemler, **yığın özellikli** (*lumped-parameter-system*) veya **dağınık özellikli** (*distributed-parameter-system*) diye ikiye ayrılır. Kimyasal reaksiyon mühendisliğindeki **Tam Karıştırmalı Sürekli Reaktör (TKSR)** yığın sisteme, **Piston Akışlı Reaktör (PAR)** de dağınık sisteme örnek verilebilir (Şekil 1.1’e bakınız). Dağınık özellikli sistemlerin modellenmesi daha karmaşık olup çoğu kez kısmi diferansiyel denklemlere kadar gidebilir.

Matematiksel modellemenin matematiksel karmaşıklığını azaltmak üzere geçerli kabullerin nasıl seçileceğini biraz sonra ele alacağız.



Şekil 1.1 Yığın ve Dağınık Sistemler a) Sistem: tüm reaktör b) Sistem: diferansiyel hacim

1.2 Modellemede Kullanılan Terimler

Bağımsız değişkenler: Uzunluk, yarıçap veya sistemin durumunu ifade eden zaman gibi büyüklüklerdir. İstenildiği gibi değiştirilebilir.

Bağımlı değişkenler: Bağımsız değişkenlerin fonksiyonu olan ve model denklemleri ile çıkarılan büyüklüklerdir. Bir PAR’da zaman ve uzunluğun fonksiyonu olan konsantrasyon gibi $[C=f(t,z)]$

Diferansiyel denklemler: Bağımlı değişkenlerin bağımsız değişkenlere göre değişim hızlarını, yani *türevlerini* içeren denklemlerdir. Örneğin, dv/dt , v ’nin t ye göre değişim hızını belirler ki o da ivmedir.

Sınır şartları (koşulları): Bağımlı değişkenlerin ve/veya türevlerinin bağımsız değişkenlerin belli değerlerindeki değerleridir. Diferansiyel denklemlerin *tam çözümü* için sınır değerleri gereklidir. $y(2)=4$, $y(x_0)=y_0$ gibi

Başlangıç şartları: Bağımlı değişkenlerin ve/veya türevlerinin sınır değerleri, bağımsız değişkenin başlangıç değerleri için belirlenirse, *başlangıç şartları* olarak adlandırılır. $y(0)=1$, $y'(0)=2$ gibi

Yığın parametreleri: Sistemde konuma bağlı olarak değişmeyen büyüklükler. TKSR’deki sıcaklık ve konsantrasyon gibi.

Dağılık parametreler: Sistemde konuma bağlı olarak değişen büyüklükler. PAR'daki konsantrasyon, dönüşüm ve reaksiyon hızı gibi.

Giriş değişkenleri: Modelde daha sonraki çıktıları belirleyen büyüklüklerdir. Başlangıçtaki sıcaklık, debi ve lineer hız gibi

Çıkış değişkenleri: Verilen giriş değişkenlerinin ve parametre değerlerinin sonucuna bağlı olan ve keyfi değerler verilemeyen büyüklüklerdir.

Şunu unutmamak lazım, modellemede amaç, *çıkış değişkenlerinin kestirimidir*.

Mühendisin, modellemede oynadığı, belki de, en önemli rol geçerli hangi varsayımları yapabileceği hakkında fikir yürütebilmesidir. Burada çoğunlukla, kapsamlı bir tanım ile yeteri kadar iyi bir cevap elde etme arasında bir mühendislik uzlaşması gerekmektedir. Asıl amaçtan sapmadan akılcı, olabildiğince çok kolaylaştırıcı varsayımlar yapmak gerekir

Proseste meydana gelen esas olayı kapsayan bir modelin geliştirilmesi çok sayıda beceri, yaratıcılık ve deneyim gerektirir. Modelleme alanı mühendisin yaratıcılık ve yenilikçiliğinin prosesin başarısında anahtar rol oynadığı bir alandır. Modelleme yapanların iyi bir fizik, kimya ve şüphesiz ki matematik bilgisine sahip olmaları gerekir.

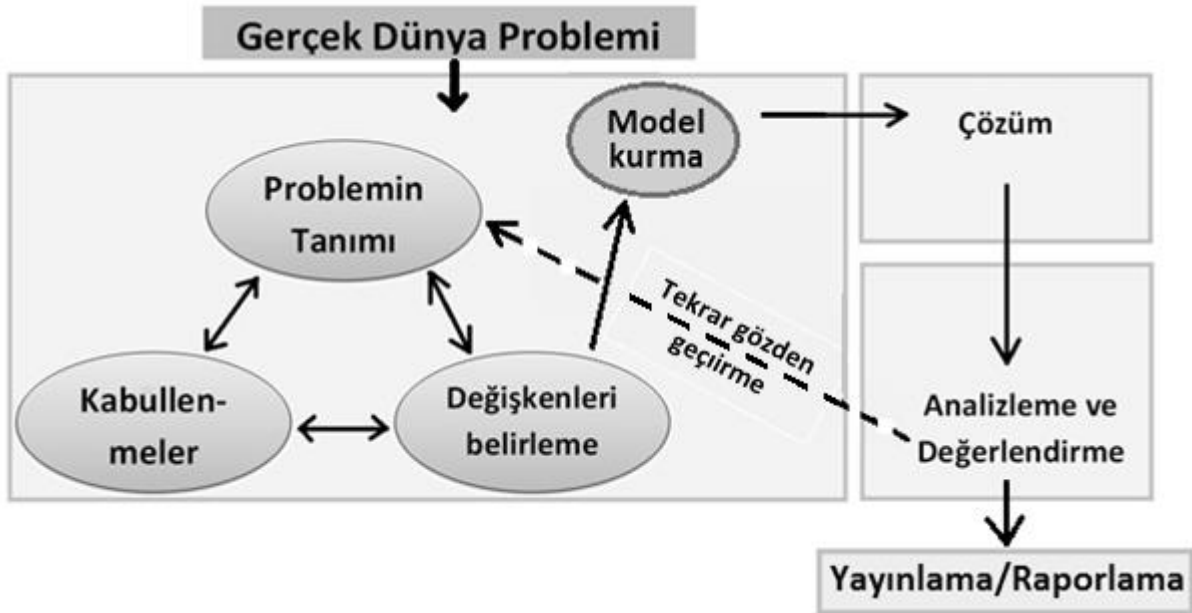
Bir model,

a) Kesin olarak gerekenden daha fazla ayrıntı bulundurmamalı ve

b) Olabildiğince az sayıda parametre içermelidir.

$y = (x_1, x_2, \dots, x_n, a, b, c, \dots, f)$ gibi bir model denklemini göz önüne alalım. Burada, $x_1, x_2, \dots, x_n$ bağımsız değişkenleri; $a, b, c, \dots$ parametreleri, f ise zorlayıcı fonksiyonu temsil etmektedir.

Modellemede takip edilecek yol, aşağıdaki diyagramda basit bir şekilde verilmiştir.



Modellemedeki 7 adım:

- 1) Problemi tam olarak tanımlama,
- 2) Akılcı kabullenmeler yapma,
Burada kabullenmenin fiziksel anlamda bir dayanağının olması gerekir. Makul olmayan kabullenmelerden uzak durulmalı. Kabullenmelerle modelleme basite indirgenebilir. Ancak, çok da basit olup gerçek problemden uzaklaşmasını önlemek lazım (*Make the things simple but not too simple, Einstein*). Çok detaylı modellemenin çözümünün de ona göre zor olacağı açıktır.
- 3) Sistemde önemli olan değişkenleri belirleme,

- 4) Matematiksel modeli kurma,
- 5) Modeli, *analitik* veya *numerik* olarak çözme. Burada, özellikle MATLAB'dan yararlanacağız.
- 6) Çıkan çözümü irdeleme,
Eğer çözüm fiziksel olarak bir anlam taşımıyorsa bizim yukarıdaki (1-5) adımları tekrar gözden geçirmemiz lazımdır.
- 7) Her şey yolunda ise sonuçları yayımlama veya raporlama

Matematiksel Modellemede **bağımlı değişken** (*çıktı*), aşağıdaki gibi bir ilişki ile gösterilebilir, demiştik:

Bağımlı değişken = f (bağımsız değişkenler, parametreler, zorlayıcı fonksiyon)

Bağımlı değişken sistemin **halini**, *bağımsız değişkenler* sistemdeki değişkenleri gösterir (zaman, uzunluk, vs.)

Parametreler: Sistemin özelliklerini veya bileşimini gösterir.

Zorlayıcı fonksiyon ise dışarıdan sistem üzerine etki eden fonksiyondur (*yer çekimi kuvveti* gibi).

Şimdi aşağıda **Chapra**'nın [1] kitabından *adapte* edilen son derece öğretici örneği, ilk matematiksel modelleme örneği olarak ele alalım. Bu örnek, aynı zamanda nümerik hesaplamaların ne kadar kuvvetli (geçerli) olduğunu da göstermektedir.

Örnek Problem

Bir paraşütçünün zamana göre düşüş hızını, fiziksel yasalar çerçevesinde modelleyelim.

Serbest düşüş: $\mathbf{F} = m \mathbf{a}$

$\mathbf{F}$: Paraşütçü üzerine etki eden *net* kuvvet, a = ivme (m/sn^2), m = kütle (kg)

$\mathbf{F} = \mathbf{F}_U + \mathbf{F}_D$, $\mathbf{F}_D$ = aşağıya doğru çeken kuvvet = mg (ağırlık)

$\mathbf{F}_U$ = sürtünmeden dolayı yukarıya çeken kuvvet = $-c\vartheta$,

burada c = sürtünme katsayısı, ϑ = düşüş hızı.

$$\frac{d\vartheta}{dt} = a = \frac{F}{m} = \frac{F_D + F_U}{m} = \frac{mg - c\vartheta}{m} = g - \frac{c}{m}\vartheta, \quad \text{yani}$$

$$\frac{d\vartheta}{dt} = g - \frac{c}{m}\vartheta$$

diferansiyel denklemini elde ettik ve bunu $t = 0$, $\vartheta = 0$ *başlangıç şartını* kullanarak ϑ 'nin t ile nasıl değişeceğini bulmak için çözmeliyiz.

Bu denklem, lineer ve birinci mertebeden bir diferansiyel denklem olup *integral faktörünü* kullanarak analitik (sembolik) olarak çözebiliriz [2]

Çözüm: $\vartheta(t) = \frac{gm}{c} (1 - e^{-(\frac{c}{m})t})$ olur.

Bu denklemde t bağımsız değişken, $\vartheta(t)$ bağımlı değişken, m ve c parametreler, g ise zorlayıcı fonksiyondur.

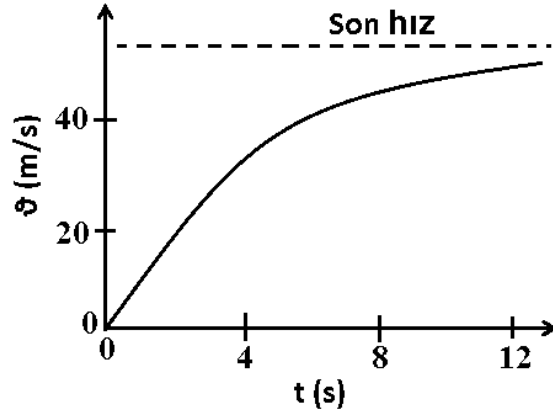
Eğer

$g = 9.8 \text{ m/sn}^2$, $c = 12.5 \text{ kg/sn}$, $m = 68.1 \text{ kg}$ alınırsa analitik çözüm, adım adım, aşağıda Tablo 1.1'deki gibi olur.



Tablo 1.1

t (sn)	θ (m/sn)
0	0
2	16.4
4	27.77
8	41.1
10	44.87
12	47.49
∞	53.39



Şekil 1.2 Modelin grafiksel çözümü

Eğer $\theta(t)$ **analitik** olarak bulunamıyorsa, o zaman bunu çözmek için **nümerik** yöntemi kullanmalıyız.

Bu problemin bir de numerik olarak çözümünü yapalım. Böylece nümerik yöntemlerin ne kadar gerçeğe yakın olduğunu da görmüş oluruz.

$$\frac{d\theta}{dt} = \frac{\Delta\theta}{\Delta t} = \frac{\theta(t_{i+1}) - \theta(t_i)}{t_{i+1} - t_i}; \frac{d\theta}{dt} = \lim_{\Delta t \rightarrow \infty} \frac{\Delta\theta}{\Delta t}$$

$$\frac{\theta(t_{i+1}) - \theta(t_i)}{t_{i+1} - t_i} = g - \frac{c}{m}\theta(t_i) \quad \text{olduğunu hatırlayalım}$$

$$\Rightarrow \theta(t_{i+1}) = \theta(t_i) + \left[g - \frac{c}{m}\theta(t_i) \right] (t_{i+1} - t_i); \quad (t_{i+1} - t_i) = \Delta t;$$

$\Delta t = 2$ sn ise

$\Delta t = 2$ sn için çözüm Tablo 1.2’de, diğer Δt ’ler için ise sonuçlar Tablo 1.3’de verilmiştir

Tablo 1.2

t (sn)	θ (m/sn)
0	0
2	19.6
4	32
8	44.82
10	47.97
12	49.96
∞	53.39

Δt küçüldükçe numerik çözüm, analitik çözüme yaklaşmaktadır

Aşağıdaki Tablo 1.3 bu gerçeği çok güzel bir şekilde özetlemektedir.

Tablo 1.3

Analitik çözüm			Karşılaştırma	Numerik çözüm					
m=68.1kg; c=12.5kg/sn; g=9.8m/sn				Analitik-Nümerik					
			Δt = 2 sn		Δt = 0.5 sn		Δt = 0.01 sn		
t (sn)	g (m/sn)		t (sn)	g (m/s)	t (sn)	g (m/sn)	t (sn)	g (m/sn)	
0	0		0	0	0	0	0	0	
2	16.4		2	16.4	2	17.06	2	16.41	
4	27.77		4	27.77	4	28.67	4	27.83	
8	41.1		8	41.1	8	41.95	8	41.13	
10	44.87		10	44.87	10	45.6	10	44.9	
12	47.49		12	47.49	12	48.09	12	47.51	
∞	53.39		∞	53.39	∞	53.39	∞	53.39	
$\vartheta(t) = \frac{gm}{c} (1 - e^{-(\frac{c}{m})t})$			$\vartheta(t_{i+1}) = \vartheta(t_i) + \left[g - \frac{c}{m} \vartheta(t_i) \right] \Delta t$						

Değişkenlerin karşılıklı ilişkilerini ifade eden matematiksel denklemler iki gruba ayrılabilir: Cebirsel denklemler ve diferansiyel denklemler.

Daha sonraki bölümlerde denklem çeşitlerini ve bunların çözümlerini ele alacağız.

Şimdi burada matematiksel modellemeye bir ara verip, *Bölüm 2'de MATLAB İleri Hesaplama Programını* kısaca ele alalım. Buradaki MATLAB bilgileri, kitapta geçen problemleri çözebilme yeterliliğinde olmasına dikkat edilmiştir. Daha detaylı **MATLAB** kitapları için verilen kaynaklara başvurunuz [3-5].

Özet

Bu bölümde matematiksel modellemenin tarifi verildikten sonra onun gerekliliği vurgulanmış ve takip edilecek 7 adım tek tek verilmiştir. Matematiksel modelleme gerçek bir problem olan serbest düşüş hızına uygulanmıştır. Nümerik hesaplamalar ile analitik hesaplamaların nasıl birbirine yaklaşabileceği seçeceğimiz adım büyüklüğü ile ilgili olduğu ortaya konulmuştur.

BÖLÜM 2 - MATLAB© Paket Programı

Amaç

Bu bölümün bitiminde,

- *MATLAB Paket Programı hakkında genel bilgiler*
- *Vektör ve Matrisler*
- *MATLAB’de teknik hesaplamaların yapılışı*
- *Polinomlar ve ilgili hesaplamalar*
- *Verilen verilere eğri uydurma*
- *Lineer denklem sistemleri*
- *Grafik çizimi*
- *MATLAB’de Programlama*
- *Sembolik ve Nümerik hesaplamalar*
- *Cebirsel ve Diferansiyel denklemlerin çözümü*

konularını anlamış olacağız.

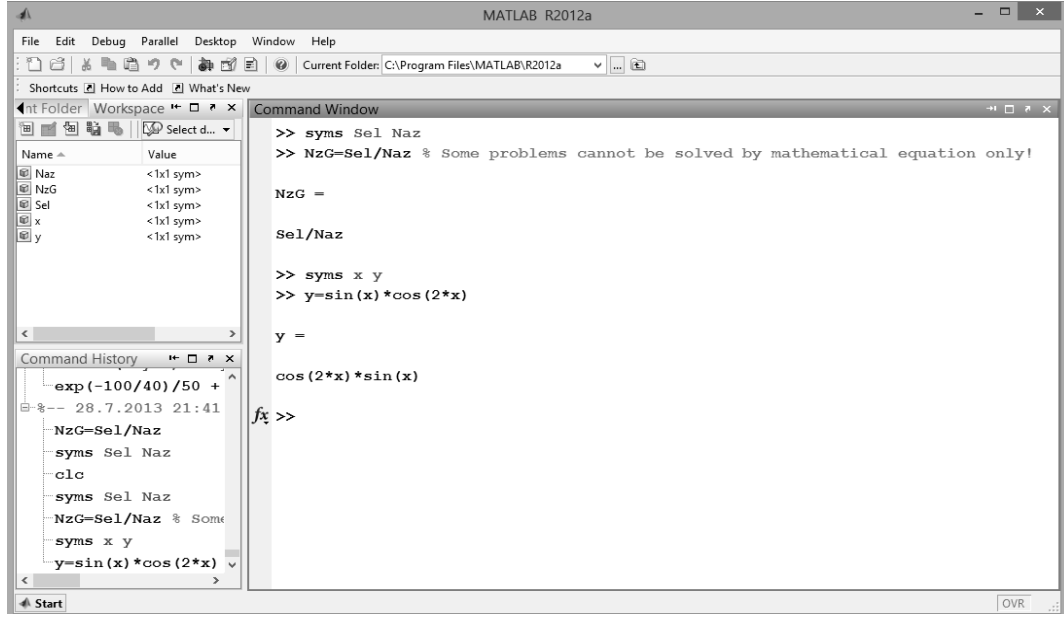
Giriş

MATLAB, **Mathworks** firması tarafından matematiksel problemlerin çözümü için İngilizce olarak geliştirilmiş, matris tabanlı, etkileşimli bir yazılım paketidir. MATLAB adı İngilizce olarak “**MAT**rix (Matris)” ve “**LAB**oratory (Laboratuvar)” kelimelerinin ilk heceleri birbirine eklenerek oluşturulmuştur. MATLAB ile yapılabilecek bazı işlemler şunlardır [6-10]:

1. *Matematiksel işlemler*
2. *Nümerik (Sayısal) İşlemler*
3. *Sembolik İşlemler*
4. *Grafikler: Grafik alet kutusu kullanılarak 2 veya 3 boyutlu olarak kolayca gerçekleştirilebilir.*
5. *Veri Analizi*
6. *Simülasyon (Benzetim) ve Model Oluşturma*
7. *Optimizasyon*

MATLAB çalıştırıldıktan sonra ekranda bir pencere açılır. Bu pencere, Şekil 2.1’de de görüleceği gibi, genelde 3 alt-pencereden oluşur. Bu alt-pencereler, kullanıcının isteğine göre değiştirilebilir. Pencereler şunlardır:

1. *Aktif Klasör/Çalışma Alanı (Current Directory/Workspace)*
2. *Komut Geçmişi (Command History)*
3. *Komut Penceresi (Command Window)*



Şekil 2.1 MATLAB pencereleri

2.1 MATLAB'in Genel Kullanımı

MATLAB, daha önce değinilen alanlardaki problemleri çok kısa zamanda çözebilir. Bu çözümler için, MATLAB'in kütüphanelerinde *tanımlı* bulunan fonksiyonlar, kullanıcılar tarafından oluşturulan *M-Dosyaları* (M-Files) ve *fonksiyon dosyaları* (function files) veya yine MATLAB'in kendi içinde yüklü olan *alet kutuları* (toolbox) kullanılabilir.

MATLAB'de İşlemciler (Operatörler)

Aritmetik İşlemciler

<u>İşlem</u>	<u>Sembol</u>	<u>Örnek</u>
Toplama: $a + b$	+	$3+2$
Çıkarma: $a - b$	-	$5-3$
Çarpma: $a * b$	*	$3*4$
Bölme: a/b	/	$3 / 4$
Ters bölme: $b \backslash a$	\	$5 \backslash 3$
Üs alma: a^b	^	5^2

İşlem önceliği: *parantez* $\rightarrow$ *üs* $\rightarrow$ *bölme / çarpma* $\rightarrow$ *toplama / çıkarma* şeklindedir.

İlişkisel işlemciler

<u>İşlemci</u>	<u>Anlamı</u>	<u>Örnek</u>
<	... den küçük	$5 < 8$
>	... den büyük	$9 > 3$

$< =$	... den küçük veya eşit	$a \leq b$
$> =$	... den büyük veya eşit	$a \geq b$
$= =$	Eşit	$a = b$
$\sim =$	Eşit değil	$a \neq b$

Not: “=” işareti atama için kullanılır

Mantıksal İşlemciler

<u>İşlemci</u>	<u>Sembol</u>	<u>Örnek</u>
&	AND (VE)	A & B
	OR (VEYA)	A B
~	NOT (DEĞİL)	~ A

İfade doğru ise cevap **1**, yanlış ise **0** verir.

MATLAB’de çok sayıda **yerleşik** (built-in) matematiksel fonksiyonlar vardır. Bazıları:

<u>Fonksiyonlar</u>	<u>Sembol</u>
sin (x)	sin
cos (x)	cos
tan (x)	tan
arcsin (x)	asin
arccos (x)	acos
arctan (x)	atan
exponansiyel, e^x	exp
tabii logaritma	log (e tabanına)
10 tabanlı log.	log 10
Karekök, $\sqrt{x}$	sqrt
Mutlak, $ x $	abs
İki tabanına göre hiperbolic fonksiyonlar.	log 2
$\sinh(x) = \frac{e^x - e^{-x}}{2}$	sinh
$\cosh(x) = \frac{e^x + e^{-x}}{2}$	cosh

Değişkenler

- Atama ile değişik değerler alan büyüklüklere değişken denir (atama “=” işareti ile)

değişken = ifade (sayılar, değişkenler, işlemciler, fonksiyonlar)

- *Değişken* isimleri bir harfle başlar ve alfanümerik karakter ile altçizgi “_” den oluşabilir ve en fazla 67 karaktere sahiptir.
- $a = 4, b = 8$ % gibi virgülle ayırarak birden fazla ifade aynı satıra yazılabilir.
- Eğer ifade tek satıra sığmaz ise satır sonuna “...” işareti konularak alttaki satırda devam edilir.

Eğer atama yapılmazsa sonuç “**ans**” değişkenine atanır.

- Değişkenlerin aktif listesi ‘**who**’, boyutları da ‘**whos**’ ile listelenir.
- **clear a b** % **a** ve **b** değişkenlerini siler.
- **clear all** % bütün değişkenleri siler.
- **clc** (*clean comand window*) % komut penceresini temizler, fakat değişkenler olduğu gibi bellekte kalır. (% işareti ile başlayan bilgi açıklayıcı bilgi olup MATLAB tarafından göz önüne alınmaz)

2.2 Diziler: Vektörler ve Matrisler

Dizi, en genel matematiksel tanımı ile nümerik ve metinsel değerler topluluğudur. MATLAB’de hemen hemen her şey bir dizi olarak işleme konur [11-14]. MATLAB’de üç çeşit dizi ifadesi kullanılmaktadır:

Bir nümerik dizi, *skalar*, *vektör* veya *matris* olabilir:

- 1×1 dizisi, bir skaları (*scalar*) göstermektedir.
- **$m \times 1$** veya **$1 \times n$** dizisi, bir vektörü (*vector*) göstermektedir (tek boyutlu dizi, **1D**)
- **$m \times n$** veya **$n \times m$** dizisi, bir matrisi (*matrix*) göstermektedir (iki boyutlu dizi, **2D**)

1×1 dizisi sabit veya tek elemanlı matris, **$n \times 1$** dizisi *sütun vektörü* ve **$1 \times n$** dizisi ise *satır vektörü* olarak tanımlanabilmektedir. Bir dizinin eleman sayısı, satır ile sütun sayısının çarpımıdır.

2.2.1 Vektörler

Vektörler, $m \times 1$ veya $1 \times n$ boyutlu diziler olarak tanımlanmaktadır. $m \times 1$ boyutlu diziyeye “sütun (kolon) vektörü” denir ve dizinin eleman sayısı **m** tanedir; $1 \times n$ boyutlu diziyeye “*satır vektörü*” demiştik ve dizinin eleman sayısı **n** tanedir.

Vektör Oluşturma

MATLAB’de vektör oluşturma üç yolla gerçekleşir:

1. *Direkt olarak (köşeli parantez [...] kullanmakla)*

Örnek .2.1 Köşeli parantez kullanarak vektör oluşturma

```
>> v = [1 7 -9 10 12 25]
```

```
v =
```

```
1      7     -9     10     12     25
```

Bu işlem, **v** adlı bir satır vektörü oluşturur. Elemanlar arasına virgül konur ya da boşluk bırakılır. Örnekte görüldüğü gibi “**enter**” tuşuna basıldığında vektörü çıktı olarak görürüz. Eğer çıktısını görmek istemiyorsak vektörü yazdıktan sonra **sonuna** noktalı virgül “;” koyarız,

```
v = [2 5 7 11]; gibi.
```

Dikkat edildiği üzere, yukarıda verilen örneklerde vektörler satır vektörlerdir. Bir satır vektörünü sütun vektörüne çevirmek istiyorsak *transpozunu (devriğini)* almamız gerekir, bu da tek kesme işareti “'” ile olur.

Örnek .2.2 Bir vektörün transpozunu (devriğini) bulmak

```
v = [2 5 -8 -11]’ (Satır vektörünün transpozu)
```

v =
2
5
-8
-11

2. Eşit aralıklı elemanlar kullanarak; [“:” iki nokta üst üste işareti ile veya **linspace**, **logspace** komutlarıyla]

a- Vektör elemanları birbirlerini, sabit miktarda artan veya azalan bir değerle (step size) takip ederler ise “:” işleci (colon operator) bu tür bir işlem için en temel yöntemdir.

Örnek 2.3 Kolon operatörü (:) ile vektör oluşturma

v = 1:8

v = 1 2 3 4 5 6 7 8

v değişkeni 1, 2, 3, 4 ... ve 8 tamsayılarını üreterek bir satır vektörünü gösterir.

Daha genel olarak,

$$v = x_{\text{ilk}} : \Delta x_{\text{artış}} : x_{\text{son}} \text{ şeklindedir}$$

Artış miktarı, $\Delta x_{\text{artış}}$, belirtilmezse MATLAB bunu **1 birim** olarak kabul eder. Artış miktarı pozitif veya negatif, herhangi bir sayı olabilir.

Örnek.2.4 Kolon operatörü ile vektör oluşturma

p = 0.1:0.2:1

p = 0.1000 0.3000 0.5000 0.7000 0.9000

b- **linspace** ve **logspace** komutlarını kullanmak.

Bu durumda başlangıç ve bitiş noktaları aralığında kaç nokta olacağı belirtilir. **linspace**, lineer aralıklı bir vektör üretmekte kullanılır ve özellikle grafik çizimlerinde çok yararlıdır.

Genel gösterim: **linspace** (x_{ilk} , x_{son} , **n**) şeklinde olup **n**=vektördeki eleman sayısıdır.

Örnek 2.5 **linspace** komutu ile vektör oluşturma

linspace(3, 17, 2) neyi ifade eder?

Çözüm:

Bu örnekte 3, *ilk değer*, 17 *son değer* ve 2 de ilk değer ile son değer aralığındaki (ilk ve son değer dâhil) nokta sayısıdır.

Eğer nokta sayısı belirtilmezse iki nokta arasında MATLAB lineer olarak **100** eşit elemanlı satır vektörü verir.

c- Özel (utility) fonksiyonlar kullanarak (**rand**, **randn**, **ones**, **zeros** komutlarıyla) vektörler hazırlanabilir.

Ayrıca, MATLAB’de özel işlemler yapan hazır fonksiyonlar da vardır. Bunlar, örneğin,

min, **max**, **mean**, **sum**, **prod**, **sort**, ve **length**

komutlarıdır.

Örnek 2.6 Özel işlemler yapan hazır fonksiyonların vektörlerle kullanımı

$x=[4\ 7\ 9\ 11\ 3\ 5\ 8]$ vektörüne

$\min(x)$, $\max(x)$, $\text{mean}(x)$, $\text{sum}(x)$, $\text{length}(x)$, $\text{prod}(x)$, $\text{sort}(x)$

fonksiyonlarını uygulayıp sonucu bulunuz.

Çözüm:

Cevaplar sırasıyla 3, 11, 6.7143, 47, 7, 332640 ve [3 4 5 7 8 9 11] dir.

2.3 Vektör ve Matris İşlemleri**2.3.1 Matrislerde Toplama ve Çıkarma İşlemleri:**

MATLAB’de toplama ve çıkarma işlemleri için “+” ve “-” sembolleri kullanılır. Boyutları aynı olan işlemler için tanımlıdır.

2.3.2 Matrislerin Çarpımı:

Matrislerde çarpma işlemi için “*” sembolü kullanılır. Çarpma işlemlerinde matrislerin iç boyutları aynı olmalıdır. $A*B$ çarpımında A ’nın sütun sayısı ile B ’nin satır sayısı eşit olmalıdır.

MATLAB’de bölme işlemi için iki farklı (/ ve \) bölme işlemcisi vardır:

$x=b/A$ matris denklemi, $Ax=b$ ‘nin çözümünü gösterir. Bu aynı zamanda $x=A^{-1}*b$ şeklinde de gösterilebilir.

$x=b\backslash A$ matris denklemi ise $xA=b$ ‘nin çözümünü gösterir.

Örnek 2.7 Bir matrisin bir vektöre bölümü

$A = [1\ 2\ 3; 4\ 5\ 6]$; $b = [9\ 11\ 21]$; ise x vektörünü bulunuz.

Çözüm:

$$x=A^{(-1)} *b' = [1\ 2\ 3; 4\ 5\ 6; 6\ 3\ 19]^{(-1)} * [9\ 11\ 21]' = \text{inv}(A)*b'$$

$$x = -5.4561$$

$$3.9123$$

$$2.2105$$

bulunur (b vektörünün transpozunun alındığını ve $A^{(-1)}$ yerine de $\text{inv}(A)$ kullanıldığına dikkat ediniz. inv komutu hakkında ileriki bölümlerde daha kapsamlı bilgi verilecektir).

2.3.3 Dizilerin Eleman- Elemana İşlemleri (Element-by-Element Operations)

Eleman-eleman çarpım

$a = [3\ 7\ -3\ 4]$; $b = [1\ 4\ -2\ -2]$ olsun

Eleman-eleman çarpımı $a.*b$ şeklinde gösterilir.

“.” işaretini *eleman-eleman* işlemini göstermektedir.

Çarpım: $a.*b$, a dizisinin her bir elemanı, sırasıyla b dizisinin karşılıklı her bir elemanı ile çarpılır

Sağa Bölme: $a./b$, “a” dizisinin her bir elemanı, sırasıyla “b” dizisinin her bir elemanına bölünür.

Sola Bölme: $a.\backslash b$, “b” dizisinin her bir elemanı, sırasıyla “a” dizisinin her bir elemanına bölünür.

“.^” üs alma: $a.^b$, “a” dizisinin her bir elemanı sırasıyla “b” dizisindeki karşılıklı elemanların üssüne yükseltilir.

Örnek 2.8 Eleman elemana işlemler

Aşağıdaki **a** ve **b** vektörleri için:

a= [3 7 -3 4]; **b**= [1 4 -2 -2] ise

Çözüm:

a.*b= 3 28 6 -8

a./b= 3.0000 1.7500 1.5000 -2.0000

a.\b= 0.3333 0.5714 0.6667 -0.5000

a.^b= 1.0e+003 * (0.0030 2.4010 0.0001 0.0001) bulunur.

Vektörler için verdiğimiz bu örnekler aynı zamanda matrisler için de geçerlidir.

2.3.4 Matrisin Transpozu

Bir **A** matrisinin transpozu “**A'**” ile bulunur. “**'**” transpoz operatörü ile matrisin satır ve sütunları yer değiştirilmiş olur.

2.3.5 Bir Matrisin Determinantı (det Komutu)

“**det(A)**” fonksiyonu ile bulunur. Matris, kare matris olmalıdır.

2.3.6 Matrisin Tersi (inv Komutu)

Bir **A** matrisinin tersi “**inv(A)**” fonksiyonu ile bulunur. Matris, tekil özelliğine sahip olmaması (*non-singular*) ve kare (*square*) matris olmalıdır.

Örnek 2.9 Bir matrisin tersini (inversion) bulma

A = [2 5; -2 4]; **inv(A)** =

0.2222 -0.2778

0.1111 0.1111

2.3.7 Bir Matrisin Rankı (rank komutu)

Bir matrisin rankı, lineer denklem sistemindeki bağımsız denklem sayısını gösterir.

2.4 Lineer (Doğrusal) Denklem Sistemleri

Sistem, bilinmeyen sayısına göre derecelendirilir. **n. dereceden** bir denklem sisteminin genel yazılımı aşağıdaki gibidir:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

.....

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

Burada **a_{ij}** katsayılarıdır, **b_i** bilinenlerdir ve **x_i** ile ifade edilenler ise *bilinmeyenler*dir. Yukarıdaki denklemleri matris-vektör şeklinde yazarsak:

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{\mathbf{b}}$$

Burada A matrisine **katsayılar matrisi** denir. Yukarıdaki matris işleminin harfsel gösterimi aşağıdaki gibidir:

$$Ax=b$$

Denklemin her iki tarafı A matrisinin (katsayılar matrisi) tersi ile çarpılırsa bilinmeyenler vektörü elde edilir.

$$x = A^{-1} * b ; \quad \text{MATLAB de } A^{-1} \text{ matrisi, } \mathbf{inv}(A) \text{ fonksiyonu ile bulunabilir.}$$

Örnek 2.10 Kare matrisli bir denklem sisteminin çözümü

Aşağıdaki denklem takımını MATLAB'in **inv** komutu ile çözünüz.

$$3x_1 + 4x_2 + 5x_3 + 6x_4 = 7$$

$$x_1 - 2x_2 + 3x_3 + x_4 = 0$$

$$-x_1 + 3x_2 + 5x_3 - 3x_4 = -12$$

$$7x_1 - x_2 - x_3 + 2x_4 = 11$$

Çözüm:

$$A = \begin{Bmatrix} 3 & 4 & 5 & 6 \\ 1 & -2 & 3 & 1 \\ -1 & 3 & 5 & -3 \\ 7 & -1 & -1 & 2 \end{Bmatrix}, \quad b = \begin{bmatrix} 7 \\ 0 \\ -12 \\ 11 \end{bmatrix} = [7 \quad 0 \quad -12 \quad 11]$$

$$x = \mathbf{inv}(A) * b = \begin{bmatrix} 0.8924 \\ -0.2363 \\ -1.0353 \\ 1.7407 \end{bmatrix}$$

2.5 Polinomlar

Polinom, sabit katsayı ve pozitif tamsayı üsse sahip **tek değişkenli** bir fonksiyondur. Polinomların genel olarak matematiksel ifadesi aşağıdaki gibidir [15-21].

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

MATLAB'de ise polinomlar, katsayılarının *azalan sırada* sıralandığı birer **satır vektörü** ile gösterilmektedir. Bu nedenle, bir polinomu MATLAB'e tanıtabilmek için sadece katsayılardan oluşan satır vektörünün MATLAB'e girilmesi yeterlidir.

Örnek 2.11 Satır vektörü ile bir polinomu ifade etmek

$p = [8 \ 0 \ 4 \ -2 \ 0 \ 1 \ 0 \ 10]$ satır vektörü girildiğinde, MATLAB tarafından $p(x) = 8x^7 + 4x^5 - 2x^4 + x^2 + 10$ polinomu olarak algılanmaktadır

2.5.1 Polinomların Değerini Bulma (polyval Komutu)

MATLAB ortamında bir polinomun değerini bulmak için **"polyval(p,x)"** komutu kullanılmaktadır. Burada **p** polinomun azalan sırada sıralanmış katsayı matrisini, **x** ise polinomun değişkenini göstermektedir. **x** önceden tanımlanmış bir sayı olabildiği gibi vektör veya matris de olabilir. **x** sayı olarak tanımlanmışsa, polinom gösteriminde skalar çarpım, **x** vektör veya matris olarak tanımlanmışsa vektörel çarpım kullanılmalıdır

$$p(x) = 8 * x^3 + 4 * x^2 - 2 * x - 3; \quad \% \text{ skalar değişkenli polinomun tanımlanması}$$

$$p(x) = 8 * x^3 + 4 * x.^2 - 2 * x - 3; \quad \% \text{ vektörel değişkenli polinomun tanımlanması}$$

polyval(p,x) **p** polinom katsayısına sahip polinomun **x** deki değerini hesaplar.

Örnek 2.12 *polyval komutu ile bir polinomun değerini hesaplama*

$p(x) = 8x^3 + 4x^2 - 2x - 3$ polinomunun $x = 2$ değeri için aldığı sonucu, **polyval** komutu kullanarak bulunuz.

Çözüm:

```
p=[8 4 -2 -3];           % katsayı vektörünün tanıtılması
x=2;                     % değişken değerinin tanıtılması
B=polyval(p,x)=73        % polinom değerinin B değişkenine atanması
```

2.5.2 Polinomların Köklerini Bulma (roots Komutu)

P(x) = 0 şeklindeki denklemin kökleri **roots (P)** şeklindeki bir ifade ile bulunur. Burada **P** polinom katsayı vektörüdür.

Örnek 2-13 *Bir polinomun köklerini roots komutu ile bulma*

$P(x) = x^3 + 4x^2 - 3x - 7 = 0$ denkleminin köklerini bulunuz.

Çözüm:

```
P = [1 4 -3 -7]; r = roots (P)
r = -4.3194
    1.4437
   -1.1233
```

Üçüncü dereceden bir polinomun 3 tane kökü olmalıdır. Bunlar gerçek, kompleks veya karışımları olabilir.

2.5.3 Polinomların Çarpımı (conv Komutu)

Polinomların çarpımında “**conv**” komutu kullanılır. Polinomların çarpımında ya bir skalar değerle polinomun çarpımı, ya da iki polinomun birbirleriyle çarpımı söz konusu olur.

Örnek 2.14 *İki polinomun conv komutu ile çarpımını bulma*

Vektörel katsayıları aşağıda verilen polinomların çarpımını bulunuz.

$P_1 = [3 \ 7 \ 2 \ 4]$; $P_2 = [1 \ 1 \ 1]$

Çözüm:

```
PP = conv (P1, P2) = [3 10 12 13 6 4] veya
PP(x) = 3x5 + 10x4 + 12 x3 + 13 x2 + 6x +4 olur.
```

2.5.4 Polinomların Birbirleriyle Bölümü (deconv Komutu)

İki polinomu birbirine bölmek için “**deconv**” komutu kullanılır.

f_1 polinomu, f_2 polinomuna bölündüğünde bölüm “**d**” de, kalan “**r**” de saklanır. Fonksiyon şöyledir:

$$[d, r] = \text{deconv} (f_1, f_2)$$

Örnek 2.15 *İki polinomun bölümünde bölüm ve kalanı bulma*

Diyelim $f_1 = 5x^3 + 3x^2 + 2x + 1$ ve $f_2 = x^2 + 2x + 1$ olsun. f_1 polinomunu f_2 polinomuna bölünüz.

Çözüm:

$$P_1 = [5 \ 3 \ 2 \ 1]; P_2 = [1 \ 2 \ 1]$$

$$[d, r] = \text{deconv}(P_1, P_2)$$

$$d = [5 \ -7], r = [11 \ 8] . \quad \text{Diğer bir deyimle,}$$

$$\frac{5x^3 + 3x^2 + 2x + 1}{x^2 + 2x + 1} = 5x - 7 + \frac{11x + 8}{x^2 + 2x + 1}$$

2.5.5 İnterpolasyon, Ekstrapolasyon ve Regresyon

Tablo halinde verilen sayısal verilerin *ara* değerini bulmaya **interpolasyon**, *dış* değerlerini bulmaya da **ekstrapolasyon** denir. Gerek interpolasyon gerekse ekstrapolasyon yapmada en uygun yol, o değerlerden geçen bir polinomun denklemini bulmak ve onu kullanmaktır. Tablo halinde verilen değerlerden geçen en uygun polinomu bulma işlemine ise **regresyon** denir [22-26].

İki bilinen değer arasında MATLAB ile interpolasyon yapmak için “**tek boyutlu: interp1**”, üç bilinen değer arasında interpolasyon yapmak için de “**iki boyutlu: interp2**” veya **griddata** komutları kullanılır.

x ve **y** iki vektör olsun (*zaman – basınç gibi*)

$y_i = \text{interp1}(x, y, x_i)$ ifadesi, verilen **x** ve **y** vektörlerindeki x_i ara değerine karşılık gelen y_i değerini verir.

$y_i = \text{interp1}(x, y, x_i)$ veya $y_i = \text{interp1}(x, y, x_i, \text{'metot'})$ ifadeleri kullanılabilir.

Burada **metot**: *linear, spline* veya *cubic* şeklinde olabilir.

interp1 komutu aynı zamanda **ekstrapolasyon** için de kullanılabilir.

Örnek 2.16 ‘spline’ kullanarak ekstrapolasyon yapma

$x = [3 \ 5 \ 8 \ 12]; y = [45 \ 67 \ 92 \ 150];$ ise $x=13$ değerinde **y**’nin değeri nedir? Metot olarak “spline” kullanınız.

Çözüm:

$$x = [3 \ 5 \ 8 \ 12]; y = [45 \ 67 \ 92 \ 150];$$

$$y_{13} = \text{interp1}(x, y, 13, \text{'spline'})$$

$$y_{13} = 175.1905 \text{ bulunur.}$$

Regresyonda **en küçük kareler metodu** (*The least square method*) [27-31] kullanılır. Bununla ilgili analitik türetimler ileride verilecektir.

2.5.6 Verilere Polinom Uydurmak (polyfit Komutu)

MATLAB’de verilen değerlere bir polinom uydurmak **polyfit(x, y, n)** şeklindeki komutla mümkündür. **x** ve **y** vektörleri, ‘**n**’ ise polinomun derecesini gösterir.

Örnek 2.17 Verilere doğru uydurmak ve veriler ile uydurulan doğrunun aynı grafikte gösterimi

$x = 1: 6;$ ve $y = [0.5 \ 0.8 \ 0.95 \ 1.2 \ 1.5 \ 1.7]$ verilen iki vektör olsun.

Bu verileri Kartezyen koordinat sisteminde (*x-y grafiği*) gösterelim. Sonra da bu verilerden geçen doğrunun denklemini **en küçük kareler yöntemi** ile bulup çizelim, bulunan doğruyu ve verileri aynı grafikte gösterelim (Şekil 2.2)

$$P = \text{polyfit}(x, y, 1)$$

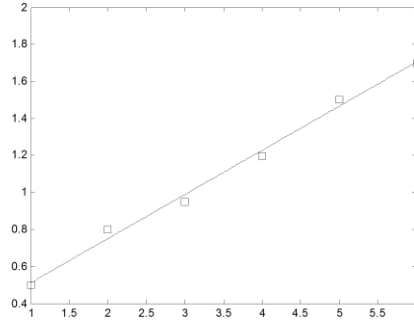
$$P = [0.2386 \ 0.2733] \text{ bulunur. Yani } y = 0.2386x + 0.2733 \text{ şeklinde bir denkleme uymaktadır.}$$

$xx = 1: 0.01: 6$ (daha sağlıklı bir grafik elde etmek için x eksenindeki artımlar 1 (bir) kadar değil, 0.01 kadar artırıyoruz)

$yy = \text{polyval}(P, xx);$

$\text{plot}(x, y, 's', xx, yy, 'r-')$

Elde edilen grafik Şekil 2.2 de gösterilmiştir.



Sekil 2.2 En Küçük Kareler yöntemiyle bir doğrunun çizilmesi

2.6 Grafikler

MATLAB’ de, verilerin ya da fonksiyonların grafiksel olarak gösterilmesi için pek çok seçenek vardır. .

MATLAB’de 2 ve 3-boyutlu grafikler elde etmek mümkündür [32-36].

2.6.1 İki Boyutlu (2D) Grafikler

MATLAB ile pek çok grafik türü oluşturulabilir: Çizgisel, alan, çubuk (bar), dilim, kontur, dağılım ve kutupsal grafikler bunlar arasındadır.

İki boyutlu grafik çiziminde en çok kullanılan **plot**, **ezplot**, **semilogx**, **semilogy**, **loglog** ve **plotyy** komutlarından **plot** ve **ezplot** komutlarını göreceğiz.

plot Komutu

plot(x,y) şeklinde kullanılır

Her nokta için **(x,y)** değerleri verilmiştir. İlk nokta için, **x** ‘in ilk değeri ve **y**’nin ilk değeri alınır. Bu nedenle **x** ve **y** vektörlerinin eleman sayıları aynı olmalıdır, yoksa MATLAB hata verir. Daha genel olarak **plot** komutu aşağıdaki şekilde verilebilir.

plot(x,y,'çizgi türü')

Çizdirmek istediğimiz çizgisel grafiğin özellikleri girilmiştir (Kod yazılımına bkz).‘çizgi türü’ ve **plot** komutunda kullanılabilecek bazı özellikler aşağıda açıklanmıştır.

Tablo 2.1 Çizgi Tipi

Gösterim	Anlamı	Örnek
-	düz çizgi	_____
--	kesik çizgi	- - - - -
:	noktalı çizgi (varsayılan)	- - - - -
-.	kesik ve noktalı çizgi sıralı	- - - - -

MATLAB’de *çizgi türü* içinde **çizginin rengi, tipi ve nokta sembolleri** gösterimi tanımlanabilir.

Bunlar herhangi bir sıra içinde, aşağıda verilen kısa kodları kullanılarak tanımlanabilir. Çizgi tipi Tablo 2.1’de, renkler Tablo 2.2’de ve nokta sembolleri de Tablo 2.3’de verilmiştir.

Tablo 2.2 Renkler

Gösterim	Renk
y	Sarı
m	Magenta
c	Çıyan
r	Kırmızı
g	Yeşil
b	Mavi
w	Beyaz
k	Siyah

Tablo 2.3 Nokta Sembolleri

Gösterim	İşaret	Gösterim	İşaret
+	Artı	d	Elmas
o	Çember	^	Yukarı üçgen
*	Yıldız	v	Aşağı üçgen
.	Nokta	<	Sola üçgen
x	Çarpı	>	Sağa üçgen
s	Kare	p	Beşgen

ezplot komutu

Aşağıda **ezplot** komutunun genel kullanımı ve açıklaması verilmiştir:

ezplot(fonksiyon)

Verilen fonksiyonu varsayılan aralık olan $-2\pi < x < 2\pi$ için çizdirir.

Örnek 2.18 ezplot komutunun kullanımı

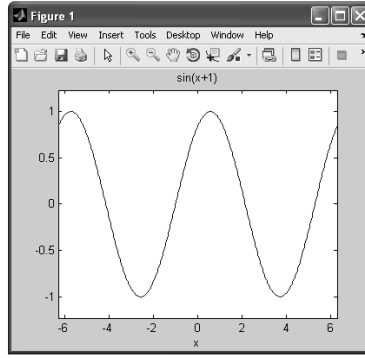
$y = \sin(x+1)$ grafiğini Kod 2.1 deki gibi ezplot komutuyla çiziniz.

Çözüm:

ezplot komutunun en sade halidir ve varsayılan aralık olan $-2\pi < x < 2\pi$ aralığı için grafik çizdirilmiştir (Şekil 2.3)

```
>> ezplot('y-sin(x+1)')
```

Kod 2.1

Şekil 2.3 **ezplot** kullanımı

2.7 Programlama

En genel ifadesiyle programlama, belli bir problemin çözümü için bir programlama dili aracılığıyla yazılan programın **tasarımı**, **kodlaması**, **test edilmesi** ve **güncellenmesi** işlemlerinin bütünüdür.

Programın algoritması hazırlandıktan sonra bir programlama dili (FORTRAN, C, C++, Pascal, Basic gibi) kullanılarak kodlama, yani program yazma işlemine geçilir. MATLAB **komut penceresi** (*command window*) ve özellikle MATLAB **editörü** (*editor*) program yazma işlemi için kullanılabilecek alanlardır.

Basit programlar için güncelleme sürecine gerek yoktur. Burada sadece çok basit programlar ve kodlar verileceğinden, bu işlem hakkında bilgi vermeye gerek duyulmamıştır.

Burada ilk olarak kodların yazılacağı **M-dosyaları** (**script** ve **function**) daha sonra **inline** komutu ve son olarak da bir algoritma oluşturmak için gerekli olan **koşul ifadeleri** (*conditional statements*) ile **döngüler** (*loops*) anlatılacaktır.

2.7.1 M-Dosyaları (M-Files)

MATLAB’ de kodlar genellikle komut penceresi (*command window*) kullanılarak girilir. Geçmiş komutlar tekrar kullanılmak istendiğinde ise komut geçmişi (*command history*) penceresi kullanılarak istenilen komuta ulaşılabilir. Birkaç satırda oluşan basit programlar düşünüldüğünde bu yaklaşımda bir problem yoktur. Çünkü hata yapıldığında önceki satırlar kolaylıkla yeniden yazılabilir ve kod tekrar kullanılmak istendiğinde komut geçmişinden veya **yukarı ok** ile komutlar kolayca tekrar çağrılabilir. Fakat satır sayısı fazla, karmaşık programlarda komut penceresi ve komut geçmişi yeterli olmaz. Böyle durumlarda komutlar **m-uzantılı dosyalara** yazdırılır ve tekrar ve tekrar çağrılıp kullanılabilir.

M-uzantılı bir dosya oluşturmak için öncelikle bir yazı editör programı (*text editor*: MATLAB editör, notepad, wordpad gibi) açılır ve yazılacak olan kodlar buraya yazılır. Kodlar yazıldıktan sonra bu dosya, örneğin, “**dosya_adı.m**” şeklinde kaydedilir. Daha sonra MATLAB komut penceresinde sadece dosya adı (**.m yazılmadan**) yazılarak dosya içindeki komutlar çalıştırılır.

Oluşturulan **m-dosyasının** çalıştırılabilmesi için dosyanın aktif MATLAB klasörü (*current directory*) içinde olması gerekir. MATLAB komut editörü, dosyanın kaydedileceği klasör olarak otomatik olarak aktif klasörü seçer. Aktif klasörün yeri MATLAB penceresinin üst kısmında kısa yol tuşlarının sağındadır. Aktif klasörün içinde bulunan dosyalar MATLAB penceresinin solunda aktif klasör (*current directory*) penceresinde gösterilir. Eğer kaydedilen dosya burada görülüyor ise dosya MATLAB tarafından kullanılabilir. Dosyanın içindeki komutları çalıştırmak için yapılması gereken tek şey dosya adını komut penceresinde yazıp **enter**’e basmaktır. Böylece dosyanın içindeki tüm komutlar çalıştırılır.

2.7.1.1 Düz-yazı Dosyaları (script files)

Komut penceresine yazılabilecek komutlar direkt olarak **m-dosyasının** içine yazılarak oluşturulurlar. Böylece kullanılan komutlar saklanabilir ve tekrar tekrar kullanılabilir.

2.7.1.2 Fonksiyon Dosyaları (function files)

Fonksiyon dosyaları, MATLAB' in özelliklerini genişletmek için kullanılan dosyalardır. **Düz-yazı dosyalarından farklı olarak, fonksiyon dosyaları girdi değerleri alıp çıktı değerler üretebilirler.** Aslında MATLAB' de kullanılan komutların çoğu fonksiyon dosyaları olarak önceden hazırlanmışlardır. MATLAB açıldığında bu dosyaları tarar ve komutları kullanıma hazır hale getirir.

Fonksiyon dosyaları oluşturulurken komutlar yazılmadan önce m-dosyasının bir fonksiyon dosyası olduğunu MATLAB' e göstermek için aşağıda verilenlerden biri dosyanın başında kullanılmalıdır.

function *fonksiyon_adı*

Bu ifade ile *fonksiyon_adı* isimli bir fonksiyon oluşturulur. Hiçbir girdi değeri almaz ve çıktı değeri üretmez. Düz-yazı dosyası ile aynı işlevi görür.

function *fonksiyon_adı*(x) veya **function** *fonksiyon_adı*(x,y,z)

İlk kullanımda fonksiyon bir x değeri alır. İkinci kullanımda ise birden fazla değer (x,y,z) aldığından her girdi değeri arasına virgül konulmuştur. Hiçbir çıktı değeri ise üretilmez.

function x= *fonksiyon_adı* (a,b,c) **function** [x,y,z]=*fonksiyon_adı*(a,b,c)

İlk kullanımda girdi değerleri olarak (a, b, c) değerlerini almıştır ve çıktı değeri olarak ise sadece x değerini üretmiştir. İkinci kullanımda ise yine aynı girdi değerlerini almasına rağmen birden fazla çıktı değeri üretilmiştir. Bu nedenle bu değerler bir satır vektörü şeklinde gösterilmiştir.

Yukarıda verilen fonksiyon tanımlarından biri, yazılacak olan fonksiyonun türüne göre seçildikten sonra, programın komutları yazılarak dosya tamamlanır. Daha sonra dosya kaydedilir. **Burada önemli olan dosya adı ile *fonksiyon_adı* ile gösterilen fonksiyon adının aynı olmasıdır.**

Fonksiyon türüne göre fonksiyonun komut penceresinde kullanımı da değişir. İlk tür için sadece fonksiyon adını yazmak yeterlidir:

```
>> fonksiyon_adı
```

İkinci kullanımda ise, fonksiyon girdi değerleri aldığından, bu değerlerin parantez içinde fonksiyon adından sonra verilmesi gerekir:

```
>> fonksiyon_adı(x_değeri,y_değeri, z_değeri)
```

Son kullanımda ise fonksiyon bir çıktı değeri ürettiğinden fonksiyon çıktısı bir değişkene atanmalıdır:

```
>> x=fonksiyon_adı(a_değeri,b_değeri,c_değeri)
```

Örnek 2.19 m-fonksiyon dosyasının kullanımı

Bir dönemde verilen iki ara sınav ve final sınavlarına notlarına göre öğrencilerin son dönem notunu aşağıda verilen denkleme göre hesaplayan *fonksiyon dosyasını* yazınız!

$$Sonnot=0.3(Sınav\ 1)+0.3(Sınav\ 2)+0.4(Final)$$

Daha sonra aşağıda notları verilen öğrencilerinin yarı dönem son notlarını hesaplatınız. Öğrencilerin dönem sonu ortalama notunu hesaplayınız.

Öğrenci	Sınav 1	Sınav 2	Final
Öğrenci 1	87	90	82
Öğrenci 2	56	70	78
Öğrenci 3	76	68	92
Öğrenci 4	91	92	88

Çözüm:

Öncelikle kullanılacak fonksiyon türü seçilmelidir. Öğrenci notlarını argüman olarak alacağı için girdi bölümü olmalıdır. Son notların ortalamasının bulunması istendiği için çıktı notlarının bir değişkene atanması gerekir. Bu nedenle üçüncü fonksiyon türü kullanılacaktır.

Fonksiyon dosyası (Kod 2.2a) ve açıklamaları aşağıda verilmiştir:

```
function n=fonk_sonnot(s1,s2,f)
```

```
n=0.3*s1+0.3*s2+0.4*f;
```

Kod 2.2a

1. Üçüncü fonksiyon türüne göre fonksiyon tanımlanması yapılmıştır. Burada dönem ara sınav notları **s1,s2** ve final notu **f** olarak alınmıştır. Fonksiyon sonucu ise **n** değişkenine atanmıştır.

2. Verilen notlara göre öğrencilerin dönem sonu not değeri **n** olarak hesaplanmıştır.

Aşağıda **fonk_sonnot** fonksiyonu kullanılarak öğrencilerin son notları ve ortalama sonnot hesaplanmıştır (Kod 2.2b)

1. İlk öğrencinin son notunu hesaplamak için 87, 90 ve 82 notları fonksiyona argüman olarak verilmiştir.

2. Fonksiyon sonucu **sonnot1** değişkenine atanmıştır.

3. Aynı işlem ikinci, üçüncü ve dördüncü öğrenci içinde yapılmıştır ve her öğrenci için son not hesaplanmıştır (**sonnot2, sonnot3, sonnot4**).

4. Bulunan son notlar toplanarak öğrenci sayısına (4) bölünmüştür. Elde edilen ortalama, **ort** değişkenine atanmıştır.

```

sonnot1=fonk_sonnot(87,90,82)
sonnot1 =
85.900000000000006
sonnot2=fonk_sonnot(56,70,78)
sonnot2 =
69
sonnot3=fonk_sonnot(76,68,92)
sonnot3 =
80

sonnot4=fonk_sonnot(91,92,88)

sonnot4 =

90.099999999999994

>> ort=(sonnot1+sonnot2+sonnot3+sonnot4)/4

ort =

81.250000000000000

```

Kod 2.2b

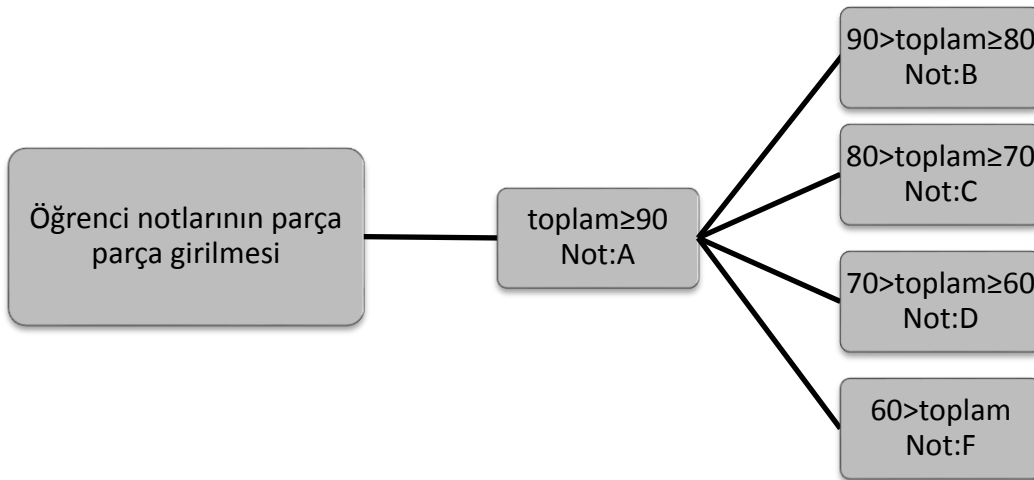
2.7.2 Koşul İfadeleri (Conditional Statements)

Bir program içerisinde kod akışı her zaman yukarıdan aşağıya doğru sırasıyla olmayabilir. Elde edilen veya girilen verilere göre program farklı şekilde dallanıp ilerleyebilir [37,38]. Bir örnek ile açıklanacak olursa:

Bir dersten öğrencilerin alacakları harf notlarının hesaplanması isteniyor. Puanlama aşağıda verilen şekilde yapılacaktır:

100-90	→	A
80-90	→	B
70-80	→	C
60-70	→	D
0-59	→	F

Öğrencinin toplam genel son notuna göre alacağı harf notu değişebilir. Yani program akışı şematik olarak gösterilecek olunursa:



Görüldüğü gibi, genel toplam **sonnot** hesabından sonra program akışı toplam nota göre 5 farklı yol izleyebilir. Burada, hangi yola gidileceğini belirlemek için yapılması gereken, toplam not hesabından sonra koşul ifadeleri eklenerek programın yönlendirilmesidir.

MATLAB’ de koşul ifadesi için kullanılabilecek 4 farklı komut vardır. Bunlar **if**, **elseif**, **else** ve **switch** komutlarıdır.

if komutu

Pek çok programlama dilinde ortak olan bu komutun genelde kullanımı aşağıda verilmiştir:

```

if (koşul(lar))
    komutlar
end
  
```

if komutu şu anlama gelmektedir:

Eğer **if** komutundan sonra parantez içinde verilen koşul veya koşullar sağlanıyorsa, komutlar ifadesi ile gösterilen işlemleri yap ta ki **end** ifadesine gelene kadar. Eğer koşul veya koşullar sağlanmıyorsa direkt olarak **end** ifadesinden sonraki satıra git. **end** ifadesi ile MATLAB’ e **if** komutunun bittiği bildirilir.

Koşul ifadeleri tanımlanırken bazen **bir tek koşul yeterli olmayabilir**. İki veya daha fazla koşulu bir bütün halinde tanımlamak gerekebilir. MATLAB’de bunun için kullanılan operatörler ve bunların kullanımları aşağıda verilmiştir:

İşlec	Adı	Açıklama
	veya (or)	Eğer verilen koşullardan biri doğru ise sonuç doğrudur, tamamı yanlış ise sonuç yanlıştır
&	ve (and)	Eğer verilen ifadelerin tamamı doğru ise sonuç doğrudur, bir tanesi yanlış ise tamamı yanlıştır
~	değil (not)	Verilen koşulun tersini alır. Yani doğru ise yanlış, yanlış ise doğru yapar.

Örnek 2.20

Bir öğrencinin matematik dersinden aldığı notlar aşağıda verilmiştir.

Arasınan 1	Arasınan 2	Final
69	87	83

Bu derste toplam puan

$$Puan=0.3*(Arasınan\ 1 + Arasınan\ 2) + 0.4*(Final)$$

şeklinde hesaplanıyorsa, bu öğrencinin dersten hangi harf notu alacağını hesaplayınız.

A	B	C	D	F
100-90	80-89	70-79	60-69	<60

Çözüm

Problemin MATLAB çözümü (Kod 2.3) ve açıklamaları aşağıda verilmiştir:

1. Öğrencinin notları girilmiştir ve puan verilen formüle göre hesaplanmıştır.
2. Sırasıyla not tanımlamalarına geçilmiştir.

Eğer puan $\geq$ 90 ise, not=A

Eğer puan $\geq$ 80 ve puan $<$ 90 ise, not=B

Eğer puan $\geq$ 70 ve puan $<$ 80 ise, not=C

Eğer puan $\geq$ 60 ve puan $<$ 70 ise, not=D

Eğer puan $<$ 60 ise, not=F

3. Not yazdırıldığında, öğrencinin B aldığı görülür.

4. Puan yazdırıldığında da, öğrencinin 80 topladığı görülür. Yani bulunan B notu doğrudur.

Bu işlemi yaptırmanın en kolay yolu tabi ki bu değildir. Toplanan puana bakılarak öğrencinin notu hemen söylenebilirdi. Fakat 1 değil de 100 öğrencinin olduğu bir sınıf düşünüldüğünde bu kodlama anlam kazanır.

elseif komutu

Ardarda yazılan **if** komutlarında, MATLAB her **if** komutunu *ayrı ayrı* inceler. Yukarıda verilen Örnek 2.20 'nin çözüm kodu incelendiğinde, MATLAB, kullanılan 5 **if** komutunu da inceler. Hâlbuki *not* değerine ikinci **if** komutunda ulaşılmıştır. Yani sonradan incelenen 3 **if** komutu gereksizdir ve kodun çalışma süresini uzatır. Bu nedenle, MATLAB 'de **elseif** isimli başka bir komut tanımlanmıştır. Bu komut tek başına kullanılmaz, bir **if** komutunun devamında kullanılabilir. **elseif** komutunun genel kullanımı aşağıda verilmiştir.

Görüldüğü gibi bir **if** komutundan sonra istenildiği kadar **elseif** komutu tanımlanabilir. Gösterilen kullanım şu anlama gelmektedir:

1. Eğer **if** komutu içinde verilen koşul ya da koşullar doğru ise, **if** komutu ile birinci **elseif** komutu arasındaki '**komutlar 1**' ile gösterilen işlemleri yap ve **end**'e git. Eğer doğru değil ise birinci **elseif** komutuna git.

% koşul ifadeleri

not1=69;not2=87;final=83;

puan=0.3*(not1+not2)+0.4*final;

if (puan $\geq$ 90)

not='A'

end

if (puan $\geq$ 80&puan $<$ 90)

not='B'

end

if (puan $\geq$ 70&puan $<$ 80)

not='C'

end

if (puan $\geq$ 60&puan $<$ 70)

not='D'

end

if (puan $<$ 60)

not='F'

end

not =

B

Kod 2.3

```

if (koşul(lar))
    komutlar 1
elseif (koşul(lar))
    komutlar 2
.
.
elseif (koşul(lar))
    komutlar n
end

```

2. Eğer birinci **elseif** komutu içinde verilen koşul ya da koşullar doğru ise, birinci **elseif** komutu ile ikinci **elseif** komutu arasındaki '**komutlar 2**' ile gösterilen işlemleri yap ve **end**'e git. Eğer doğru değil ise ikinci **elseif** komutuna git.

3. Eğer ikinci **elseif** komutu içinde verilen koşul ya da koşullar doğru ise, ikinci **elseif** komutu ile **end** komutu arasındaki '**komutlar 3**' ile gösterilen işlemleri yap ve **end**'e git. Eğer doğru değil ise **end** de git. Durum bu şekilde (**n-1**) tane **elseif** koşuluna kadar devam eder.

Yani, herhangi bir **if** veya **elseif** içinde aranan koşul sağlanırsa komutlar uygulanır ve kalan **elseif**'lere bakılmadan direkt olarak **end**'e gidilir. Böylece bakılmasına gerek kalmayan kısımlar atlanmış olur ve kısa sürede sonuca varılır. Diğer bir deyişle, birbiriyle bağlantılı koşul ifadeleri yazılırken, ilk koşulda **if**, diğerlerine **elseif** kullanılır.

Yukarıda verilen Örnek 2.20'yi **elseif** komutu kullanılarak tekrar çözelim.

Kod 2.4 incelendiğinde, ikinci **elseif** komutundan sonraki koşullar kullanılmamıştır. Çünkü bir **elseif** ifadesine gelinebilmesi için, önceki koşulların yanlış olması gerekir. Örneğin puan=65 olduğunu düşünülürse:

1. Öncelikle **if(puan>=90)** ifadesine gelir. Bu şart sağlanmadığı için bir sonrakine gider.

2. Sonra **elseif(puan>=80)** ifadesine gelir. Burada ayrıca Örnek 2.20' deki gibi **puan<90** koşuluna gerek yoktur. Çünkü **elseif(puan>=80)** ifadesine gelebilmesi için zaten **if(puan>=90)** koşulunun yanlış olması gerektir. Yani **puan<90** koşulu önceden sağlanmıştır. **puan=65<80** olduğundan bir sonraki **elseif** 'e geçilir

3. **elseif(puan>=70)** koşulu yanlış olduğundan sonraki koşula geçilir.

4. **elseif(puan>=60)** koşulu doğru olduğundan:

```
not='D'
```

komutu işlem görür.

else komutu

elseif komutu gibi tek başına kullanılamayan bu komut, **if** ya da **elseif** komutu ile birlikte kullanılabilir. **else** komutunun genel kullanım biçimleri aşağıda verilmiştir:

```

>> not1=69; not2=87; final=83;
>> puan=0.3*(not1+not2)+0.4*final;
>> if(puan>=90)
not='A';
elseif(puan>=80)
not='B';
elseif(puan>=70)
not='C';
elseif(puan>=60)
not='D';
elseif(puan<60)
not='F';
end
>> not
not =
B

```

Kod 2.4

İlk kullanımda, eğer **if** komutu içinde verilen koşul ya da koşullar doğru ise, **if** ile **else** arasında kalan '**komutlar 1**' diye isimlendirilen komutlar takımı işlem görür. Eğer yanlış ise kodun **else** kısmına gelir ve **else** ile **end** kısımları arasında kalan ve '**komutlar_{else}**' diye gösterilen bölümdeki satırlar işlem görür.

İkinci kullanımda ise, öncelikle **if** komutu içine bakılır. Eğer koşul ya da koşullar doğru ise '**komutlar 1**' işlem görür. Eğer yanlış ise **elseif** komutunun koşuluna bakılır. Doğru ise '**komutlar 2**' işlem görür. Burada **else** komutuna geçmeden istenildiği kadar **elseif** komutu kullanılabilir. Yukarıdaki kullanım için, **elseif** koşul ya da koşulları yanlış ise **else**'e geçilir ve '**komutlar_{else}**' işlem görür.

```
if(koşullar)

komutlar 1
elseif(koşul(lar))
komutlar 2
.
.
.
else
komutlarelse
end
```

2.7.3 Döngüler (Loops)

Aynı komutların tekrar edilmesi işlemine döngü denir. Döngüler iki şekilde olabilir:

1. Daha önce belirlenen sabit bir sayı kadar (**for** döngüleri) tekrar edilen döngü
2. Belli bir koşul sağlandığı sürece (**while** döngüleri) devam edilen döngü

for döngüleri

```
for x=başlangıç:bitiş
komutlar
end
```

Komutlar ile gösterilen işlemleri, **x** vektörünün her elemanı için gerçekleştirir. Örneğin **x=1:10** için **komutlar** ile gösterilen işlemler **x=1, 2, ... ,10** olmak üzere 10 kez tekrar ettirilir.

```
for x= başlangıç:artış:bitiş
komutlar
end
```

Komutlar ile gösterilen işlemleri, **x** vektörünün her elemanı için tekrar eder. Örneğin, **x=1:0.5:10** için komutlar **x=1, 1.5, 2, ... ,9.5, 10** olmak üzere 19 kez tekrar ettirilir.

```
for x= [x1,x2,x3, ... , xn];
komutlar
end
```

Yine **x** vektörünün her elemanı için komutlar ile gösterilen işlemler tekrar ettirilir.

Yukarıda belirtildiği gibi, **for** döngüsü ile döngü içinde verilen komutlar, daha önce verilen **for** ifadesindeki vektörün her elemanı için tekrarlanır. Bu komutun genel kullanımı aşağıda verilmiştir

Komutlar yerine MATLAB’ de yapılabilecek her şey yazılabilir.

Örnek 2.21 **for** döngüsü kullanımı

Bir sınıftaki öğrencilerin matematik dersinden aldığı puanlar aşağıda verilmiştir

Buna göre öğrencilerin alacağı harf notlarını gösteren bir kod yazınız.

(Harf Notları Örnek 2.20 deki gibi verilecektir.)

Öğrenci	Not	Öğrenci	Not
Ogr. 1	87	Ogr. 4	75
Ogr. 2	68	Ogr. 5	82
Ogr. 3	98	Ogr. 6	58

Çözüm:

Verilen problemin MATLAB çözümü (Kod 2.5) ve açıklamaları aşağıda verilmiştir:

1. Öncelikle öğrenci puanları (Puan) ve isimleri (Isim) girilmiştir.
2. Altı öğrenci olduğu için **for** döngüsü ile **x=1:6**, yani **x=1,2, ... ,6** için döngü içindeki komutlar toplamda altı defa tekrar edilmiştir.
3. Her tekrar için önce öğrencinin aldığı not bulunur. Örneğin **x=1** için:

Puan(1)>=90 yanlış, sonraki

Puan(1)>=80 doğru, Not='B',

end 'e git

4. Not bulunduktan sonra öğrenci adı ve aldığı not **disp** komutu ile yazdırılır. *Birden fazla ifade aynı anda yazdırılacağından, disp komutu köşeli parantez ile kullanılmıştır.* Yine **x=1** için:

Isim(1,:) → 1. öğrencinin adı

Not(1) → 1. öğrencinin notu

disp([Isim(1,:) ' --> ' Not(x)]) ile

Ogr. 1 --> B gösterilmiştir.

5. Yukarıdaki üçüncü ve dördüncü maddeler **x=1, 2, 3, 4, 5, 6** için tekrar edilmiştir ve her öğrencinin adının karşısına aldığı not yazdırılmıştır.

```

Puan=[87 68 98 75 82 58];
Isim=['Ogr. 1' ;'Ogr. 2'; 'Ogr. 3'
;'Ogr. 4'; 'Ogr. 5' ;'Ogr. 6'];
>> for x=1:6
    if(Puan(x)>=90)
        Not(x)='A';
    elseif(Puan(x)>=80)
        Not(x)='B';
    elseif(Puan(x)>=70)
        Not(x)='C';
    elseif(Puan(x)>=60)
        Not(x)='D';
    else
        Not(x)='F';
    end

```

Kod 2.5

2.8 Sembolik İşlemler

MATLAB ortamında, sayısal değerler yerine **harfli ifadelerin** kullanıldığı matematiksel işlemlere sembolik işlemler denir. Kesin sonuç istendiğinde veya sayısal olarak hesaplanması zor olan problemler genellikle sembolik işlemlerin kullanıldığı durumlardır. MATLAB, sembolik çözümler için çok geniş bir içeriğe sahip olan **Matematik Sembolik Araç kutusunu (Symbolic Math Toolbox)** kullanır. Sembolik işlemlerin tamamlanma süresi, genelde sayısal işlemlere göre daha fazladır.

Sembolik ifadeler **syms** komutu ile tanımlanır

syms komutu **birden fazla** sembolik ifadeyi MATLAB ortamında tanımlamak için kullanılır.

Aşağıda bu komutun genel gösterimi verilmiştir:

syms *degisken1 degisken2... deęişkenN*

2.8.1 MATLAB İle Yapılabilecek Sembolik İşlemler

2.8.1.1 İntegrasyon

Belirsiz İntegraller (int komutu)

Bir fonksiyonun ters-türevine ya da başka bir ifade ile sınırları belirtilmemiş integral işlemlerine **belirsiz integral** denir.

Sembolik ifadelerin integralleri **int** komutu kullanılarak bulunur. Aşağıda **int** komutunun *belirsiz integraller* için kullanım şekilleri verilmiştir.

int(f) f fonksiyonun sembolik değişkene göre integralini alır. Birden fazla sembolik değişken varsa, MATLAB kendi varsayılan sıralamasına göre birini seçer. Bu sıra $t, x, y, z, \dots$ şeklinde devam eder. Yani ilk olarak t için bakar, yoksa x ve böyle devam eder.

int(f,x) f fonksiyonunun x sembolik değişkenine göre integralini alır.

Aşağıda gösterilen Kod 2.6 de, verilen bir fonksiyon için, **int** komutunun nasıl kullanılacağı gösterilmiştir. Önce sembolik değişkenler ve integrali istenen $y=\sin(x)$ fonksiyonu tanımlanmıştır. Daha sonra **int** komutu kullanılarak fonksiyonun sembolik olarak integrali alınmıştır.

```
>> syms x y
>> y=sin(x);
>> int(y,x)
ans =
-cos(x)
```

Kod 2.6

Örnek 2.22 Belirsiz sembolik integralin kullanımı

Aşağıda verilen kuvvet ifadesinin genel olarak integralinin nasıl alınacağını MATLAB kullanarak bulunuz.

$$I = \int x^n dx$$

Çözüm:

Problemin MATLAB çözümü Kod 2.7’de verilmiştir. Bu Kod incelendiğinde:

1. Öncelikle kullanılacak sembolik değişkenler (x,y,z,n) ve integrali alınacak ifade tanımlanmıştır.
2. **int** komutu kullanılarak integral hesaplanmıştır.

Dikkat edilecek olursa, MATLAB belirsiz integral işlemlerinin sonucunda oluşan ve sabit bir sayıyı ifade eden ‘ c ’ değişkenini otomatik olarak ifadeye *eklememiştir*. **int** komutu ile bulunacak olan sonuç, başka bir işlemde kullanılmadan önce elle bir ‘ c ’ sabiti eklenmelidir.

```
syms x y z n
y=x^n;
z=int(y,x)
z =
x^(n+1)/(n+1)

$$z = \frac{x^{n+1}}{n+1}$$

```

Kod 2.7

Belirli İntegraller

Belli bir aralık için yapılan integral alma işlemine **belirli integral** alma denir. MATLAB’ de belirli integrallerin sembolik çözümleri için yine **int** komutu kullanılır.

Aşağıda **int** komutunun belirli integraller için kullanım şekilleri verilmiştir.

int(f,a,b) f fonksiyonun sembolik değişkene göre $[a,b]$ aralığında integralini alır. Birden fazla sembolik değişken varsa, MATLAB kendi varsayılan sıralamasına göre birini seçer.

int(f,x,a,b) f fonksiyonunun x sembolik değişkenine göre $[a,b]$ aralığında integralini alır.

Örnek 2.23 Belirli integralin hesaplanması

Aşağıda verilen işlemin sonucunu MATLAB kullanarak bulunuz.

$$\int_a^b (x^4 + 2x - 1)dx$$

Çözüm:

Problemin MATLAB çözümü aşağıda verilmiştir,

Sonuç: $\frac{b^5}{5} - \frac{a^5}{5} + b^2 - a^2 + a - b$

Belirli integral işlemlerinde, sabit c değişkenleri birbirini götürdüğünden, işlem sonucuna eklemeye gerek yoktur.

2.8.1.2 Sembolik Limit Alma

MATLAB'in *sembolik aletkutusunu* kullanarak herhangi bir ifadenin limitini alabiliriz.

Genel olarak:

$$\text{limit}(ifade,a) \quad \text{şeklindedir.}$$

Burada a değerinde *ifadenin* limit değeri hesaplanacak demektir. a 'nın değeri belirtilmezse o zaman $a=1$ alınır.

2.8.1.3 Cebirsel Denklemlerin Sembolik Çözümü (solve komutu)

Bir veya birden fazla bilinmeyenlerle oluşturulan bir sistemi, verilen denklemleri kullanarak bilinmeyenleri bulma işlemine **denklem çözme** denir. Denklemler, bilinmeyen değişken sayısına göre isimlendirilirler. Tek bilinmeyenden oluşanlara, **bir bilinmeyenli denklemler**, iki bilinmeyenlilere, **iki bilinmeyenli denklemler** denir ve böyle devam eder. Verilen denklemleri sağlayan değerlere **denklemin kökleri** ve oluşturduğu kümeye de **çözüm kümesi** denir.

Bir denklem sisteminin çözülebilmesi için, bilinmeyenlerin sayısı kadar bağımsız denkleme ihtiyaç vardır. Eğer yeterli sayıda denklem yoksa bazı bilinmeyenlerin tam sayısal değeri bulunamaz, ancak diğer bilinmeyenler cinsinden sembolik değeri bulunabilir.

MATLAB de cebirsel denklem sistemlerini çözmek için **solve** komutu kullanılır.

Tek Değişkenli Denklemler

solve komutunun bir tek bilinmeyenli denklemler için kullanımı aşağıdaki gibidir.

solve('f=g') $'f=g'$ şeklinde ifade edilen denklemi sembolik değişken için çözer (Kod 2.8a)

solve('f') **solve** komutu içinde denklem bir ifadeye eşitlenmemişse MATLAB bu ifadeyi otomatik olarak $'f=0'$ şeklinde alır ve bilinmeyen sembolik değişken için çözer (Kod 2.8b)

Çözülmesi gereken denklemler iki şekilde tanımlanabilir:

1. Tırnak içine alarak direkt olarak **solve** komutu içine yazılabilir.

`solve('x^2=4')` gibi

2. **f** fonksiyonu önce tanımlanıp sonra **solve** komutunda kullanılabilir.

`syms x`

`f=x^2-4` (=0 olacak şekilde fonksiyon yazılmalıdır !)

`solve(f)`

veya

`f='x^2-4'` (Tırnak içinde yazdığımızdan x değişkenini sembolik olarak

önceden yazmaya gerek yok)

`solve(f)` gibi

Örnek 2.23 **solve** komutunun kullanımı

$f=x^2-5=0$ denklemini **solve** komutu ile çözünüz.

Çözümler aşağıda verilmiştir.

```
>> solve('x^2=5')
ans =
    5^(1/2)
   -5^(1/2)
```

Kod 2.8a

```
>> solve('x^2-5')
ans =
    5^(1/2)
   -5^(1/2)
>> f='y^2-5';
>> solve(f)
ans =
    5^(1/2)
   -5^(1/2)
```

Kod 2.8b

2.8.1.4 Diferansiyel Denklemlerin Sembolik Çözümü (**dsolve** komutu)

Bir denklemde, bir fonksiyonun hem kendisi hem de türevleri bulunuyorsa bunlara **diferansiyel denklemler** denir. Sadece tek bağımsız değişkenin oluşturduğu diferansiyel denklemlere **adi diferansiyel denklemler** (**Ordinary Differential Equations, ODE**) denir. Burada, sadece bu tek değişkenli diferansiyel denklemlerin çözümü anlatılacaktır.

Tek Değişkenli Diferansiyel Denklemler

Bu bölümde sadece tek denklemden oluşan diferansiyel denklemler incelenecektir. MATLAB, diferansiyel denklemleri (*başlangıç- değer ve sınır-değer problemleri*) çözmek için **dsolve** komutu kullanır. Bu komut içinde kullanılacak diferansiyel denklemler aşağıdaki gibi tanımlanır:

Fonksiyon türevleri **D** öneki ve eğer birinci mertebeden türev değilse, kaçınıcı mertebeden türev olduğunu gösteren bir sayı ile birlikte tanımlanır.

$$\frac{dy}{dx} \longrightarrow Dy, \quad \frac{d^2y}{dx^2} \longrightarrow D2y$$

$$\frac{d^5 y}{dx^5} \rightarrow D^5 y . \quad \text{Daha genel olarak} \quad \frac{d^n y}{dx^n} \rightarrow D^n y$$

Fonksiyonun kalan kısımları tıpkı **solve** komutundaki gibidir.

Bu komutun kullanım biçimleri ve açıklamaları aşağıda verilmiştir.

dsolve('denklem')

Verilen diferansiyel denklemi sembolik değişken için çözer. Eğer birden fazla sembolik değişken varsa, MATLAB kendi varsayılana göre birini seçer.

(Bakınız Kod 2.9a , Kod 2.9b)

dsolve('denklem','C1', ... , 'Cn')

Verilen diferansiyel denklemi, **C1,C2,...,Cn** olarak verilen başlangıç ve/veya sınır değerlere göre sembolik değişken için çözer. Eğer birden fazla sembolik değişken varsa, MATLAB kendi varsayılana göre birini seçer (Bakınız Kod 2.9c)

dsolve('denklem','C1', , 'Cn','x')

Verilen diferansiyel denklemi, **C1,C2,...,Cn** olarak verilen başlangıç ve/veya sınır değerlere göre '**x**' sembolik değişkeni için çözer (Bakınız Kod 2.9d).

Yukarıdaki kullanımlarda **C** ile gösterilen başlangıç/sınır değerler aşağıdaki örneklerdeki gibi verilir:

1. Denklem: $Dy=y+x'$	$C1: y(0)=0$	BD
2. Denklem: $D^2y+Dy+y=x^2-4$	$C1: y(0)=4,$ $C2: Dy(0)=1$	BD
3. Denklem: $D^2y+Dy+y=x^2-4$	$C1: y(0)=4$ $C2: y(1)=1$	SD

BD: Başlangıç-Değer

SD: Sınır-Değer

Fonksiyonun tam sonucuna ulaşmak için, eğer diferansiyel denklem bir başlangıç-değer problemi ise, en fazla **n. mertebeden türev** içeren bir denklem için, (n-1). türevden başlayarak geriye doğru her türev için bir başlangıç değeri verilmelidir. Örneğin, yukarıdaki ikinci örnekte, en yüksek türev mertebesi 2 olduğundan, 1. türev için bir başlangıç değeri ve fonksiyon için de bir başlangıç değeri verilmiştir. Eğer diferansiyel denklem bir sınır değer problemi ise, en fazla n. dereceden türev içeren bir denklem için n tane sınır değeri yazmak gerekir. 3. örnekte, 2. mertebeden bir diferansiyel denklem için iki sınır-değer tanımlanmıştır.

Kod 2.9a da önce sembolik değişkenler tanımlanmış ve **dsolve** komutu ile diferansiyel denklem çözülmüştür.

```
>> syms x y
>> y=dsolve('Dy+y=x^2-2*x-1')
y =
x^2-2*x-1+exp(-t)*C1
```

Kod 2.9a

Komut içinde bir başlangıç değeri tanımlanmadığından, sonuçta **C1** diye bir bilinmeyen sabit vardır.

Kod 2.9b’de denklemde z isimli ikinci bir değişken vardır. MATLAB, x değişkenini seçerek, denklemi x için çözmüştür

```
>> dsolve('Dy+y=x^2-2*z-1')
ans =
x^2-2*z-1+exp(-t)*C1
```

Kod 2.9b

Kod 2.9c’de önce, denklemdeki en büyük türev ikinci mertebeden olduğu için 2 adet başlangıç değeri verilmiştir (başlangıç-değer problemi).

Eğer bir tane verilseydi, sonuç yine C bilinmeyenleri içerirdi. İkinci bölümde ise fonksiyonun sınır değerleri verilmiştir (sınır değer problemi). Denklemdeki en büyük türev ikinci mertebeden olduğu için 2 adet sınır değeri verilmiştir.

Kod 2.9d’de ise, verilen denklem hiçbir sembolik değişken içermemektedir. Eğer en sonuna ‘ x ’ için denklemi çözmeseydik, MATLAB kendi varsayımlara göre çözecekti. MATLAB 2010 için bu değişken ‘ t ’ dir.

```
>> dsolve('D2y=cos(2*x)-y','y(0)=1','Dy(0)=0')
ans =
cos(t)*(1-cos(2*x))+cos(2*x)
>> dsolve('D2y=cos(2*x)-y','y(0)=0','y(1)=1')
ans =
sin(t)*(1+cos(2*x)*cos(1)-cos(2*x))/sin(1)-
cos(t)*cos(2*x)+cos(2*x)
Kod 2.9c
```

```
>> dsolve('D3u=u','u(0)=1','Du(0)=-1','D2u(0)=0','x')
ans =
-1/3*3^(1/2)*exp(-1/2*x)*sin(1/2*3^(1/2)*x)+
exp(-1/2*x)*cos(1/2*3^(1/2)*x)
Kod 2.9d
```

Örnek 2.24 *dsolve* komutunun Newton soğuma yasasına uygulanması

Bir cismin sıcaklık değişimi aşağıda verilen denklemle modellenebilir:

$$\frac{dT}{dt} = -0.2(T - T_o) \quad (\text{Newton 'nun soğuma yasası})$$

T = Sıcaklık ($^{\circ}\text{C}$), T_o = Ortamın (çevrenin) sıcaklığı, ($^{\circ}\text{C}$) , t =zaman (dakika)

Buna göre, sıcaklığı 30°C olan bir metal küre, 90°C olan bir suyun içine konulursa, kaç dakika sonra metalin sıcaklığı 70°C olur? (Suyun sıcaklığının hep 90°C ’ de kaldığı kabul edilmektedir).

Çözüm:

Sıcaklık değişimi için diferansiyel denklem çözülmürse $T=T(t)$ gibi zamana bağlı bir sıcaklık denklemi elde edilir. Burada, bu fonksiyonun değeri için:

$T(t)=70$

alınarak, sıcaklığın 70 olabilmesi için ne kadar zaman geçmesi gerektiği bulunabilir. Bunlar, Kod 2.10' de verilmiştir.

1. Verilen diferansiyel denklem **dsolve** komutu kullanılarak sembolik olarak çözülmüştür.

2. Elde edilen denklem 70'e eşitlenerek **solve** komutu ile çözülmüştür. Elde edilen sonuç:

$t=5\ln 3$ 'dür.

double komutu kullanılarak bu değer bir ondalık sayıya dönüştürülmüştür. Böylece kürenin sıcaklığı, $t=5.4931$ dakika sonra 70 °C olur.

```
>> dsolve('DT=-0.2*(T-90)','T(0)=30','t')
ans =
90-60*exp(-1/5*t)
>> solve('90-60*exp(-1/5*t)=70','t')
ans =
5*log(3)
>> double(5*log(3))
ans =
5.4931
```

Kod 2.10

2.9 Nümerik (Sayısal) İşlemler

Burada MATLAB'in nümerik işlemlerde nasıl kullanılacağı kısaca anlatılacaktır. Sayısal integral işlemlerinden diferansiyel denklemlere, türev almadan kök bulma işlemlerine kadar pek çok alanda kullanılan MATLAB, günümüzün en yaygın matematiksel hazır paket programıdır. MATLAB' de hazır olarak bulunan pek çok matematiksel komutun yanı sıra, kullanıcılar tarafından geliştirilen ve özel problemlerin çözümünde kullanılabilecek pek çok fonksiyon da bulunabilir. Fakat, bazı işlemlerin çözümünde (örneğin, integral hesaplarında) yaklaşık yöntemler kullanılacağından, çözüm analitik çözüm ile tam olarak aynı olmayabilir. Tam sonuç istenen problemler için, mümkünse, sembolik matematiksel işlemler kullanılmalıdır.

2.9.1 Matematiksel İşlemler

2.9.1.1 Kök Bulma

Fen ve mühendislik hesaplamalarında çoğu kez verilen bir ifadeyi sıfır yapan değeri, yani kökünü bulmaya gerek vardır. Eğer fonksiyon 1. veya 2. **dereceden** ise analitik olarak köklerini bulabiliriz. Fakat 2. dereceden fazla veya transandantal bir ifade ise analitik olarak kökünü bulmak hemen hemen imkânsızdır. Bu tip ifadelerin kökleri nümerik yöntemle yaklaşık olarak bulunabilir. İleride nümerik olarak kök bulma yöntemleri ele alınacaktır. Şimdilik, MATLAB ile kökleri bulma yollarını kısaca inceleyeceğiz.

fzero komutu

Tek denklem çözümü için kullanılır. Çözüm (kök), verilen bir başlangıç değeri etrafında aranır.

fzero için $f = 0$ şeklinde olmalıdır

Genel formül

$$fzero('f', x_0)$$

şeklindedir. Burada f verilen denklemi, x_0 ise kökün bulunacağı civardır. f lineer veya nonlineer olabilir.

Örnek 2.25 $fzero$ komutuyla kök bulma

$5x^3 - 2x^2 - x + 5 = 0$ denkleminin $x_0 = 2$ civarındaki kökü nedir?

Çözüm:

$$f = '5 * x^3 - 2 * x^2 - x + 5'$$

$$fzero(f, 2) = -0.9412$$

2.9.1.2 Nümerik İntegrasyon

Alan ve hacim hesapları gibi problemleri sonsuz küçük parçalara ayırıp daha sonra bu parçaları bir araya getirip toplama işlemine **integral** alma denir. Belli bir aralık için yapılan integral alma işlemine de **belirli integral** alma denir. Burada sadece belirli integraller için yaklaşık hesap yöntemleri anlatılacaktır.

Tırnak içinde verilen f fonksiyonunun $[a,b]$ aralığındaki integralini Simpson Yöntemi ile hesaplar. İntegral sonucu **inf** ifadesi gösterirse, eğride tekillik olduğu düşünülür.

$$quad('f', a, b)$$

[**Tekillik**: Çok küçük değer değişimleri için bile fonksiyon sonucunun çok fazla değiştiği durumlar] (Kod 2.11).

Örnek 2.26 $quad$ komutlarının kullanımı

Aşağıda verilen fonksiyonu, yukarıda verilen kullanımlar ile, $[1,10]$ aralığındaki integralini hesaplayalım.

$$f = \frac{x^2}{e^x}$$

Çözüm:

Kod 2.11'de verilmiştir.

```
quad('x.^2./exp(x)',1,10)
ans =
    1.8339
```

Kod 2.11

2.9.1.3 Diferansiyel Denklemlerin Nümerik Çözümü

Fonksiyonun kendisinin ve türevlerinin aynı anda bulunduğu denklemlere diferansiyel **denklemler denildiğini** daha önceki bölümde belirtmiştik. Tek bilinmeyenden oluşan diferansiyel denklemlere **adi diferansiyel denklemler(ODE)** denir. İki veya daha fazla bilinmeyenlerin oluşturduğu diferansiyel denklemlere, **kısmi diferansiyel denklemler (Partial Differential Equations, PDE)** denir. Bu bölümde sadece adi diferansiyel denklemler için **ode45** ile çözümler gösterilecektir.

Adi Diferansiyel Denklemler – Başlangıç-Değerli Sistemler (DE with initial conditions)*Tek Değişkenli Adi Diferansiyel Denklemlerin MATLAB Komutları ile Çözümü*

Bunu bir örnek ile açıklayalım.

Örnek 2.27 Bir diferansiyel denklemin **ode45** ile çözümü

Verilen

$$\frac{dy}{dx} = y + x^2 - 1 \quad y(0) = 0$$

diferansiyel denklemini MATLAB komutları ile çözünüz.

Çözüm:

Problemin MATLAB çözümü (Kod 2.12) ve açıklamaları aşağıda verilmiştir.

1. Öncelikle **ode** komutunda kullanılacak olan diferansiyel denklem fonksiyonu tanımlanmıştır.
2. Daha sonra **ode45** komutu kullanılarak diferansiyel denklem sayısal olarak çözülmüştür. Burada fonksiyon adının önüne **@** simgesi (kulp) eklenmiştir. Bunun yerine fonksiyon adı tırnak içinde de yazılabilirdi.
3. Sonuçlar **s** değişkenine atanmıştır. Her ne kadar denklem fonksiyonunda **t** ve **z** değişkenleri kullanılmış olsa da, **ode45** fonksiyonunun sonucu **x** ve **y** değişkenlerine **x**'in aralık değerlerine ulaşmak için '**s.x**' ve dizinin son elemanına ulaşmak için ise '**(end)**' eklenmiştir. Sonuç değerlerine ulaşmak için ise aynı şekilde '**s.y(end)**' kullanılmıştır.

fonksiyon_1.m dosyası

function s=fn1(t,z)

s=z+t^2-1;

end

MATLAB kodları

>> s=ode 45(@fn1,[0,10],[0]);

>> s.x(end)

ans =

10

>> s.y(end)

ans =

2.1799e+004

Kod 2.12

Bulunan sonuç,

$$y=21799$$

Bu değer, gerçek sonuç olan $y=21905$ sonucuna oldukça yakındır.

Özet

Bu kısımda, MATLAB ile yapılabilecek matematiksel hesaplamalar (lineer denklem sistemleri, polinomlar, grafik çizimleri, programlama, sembolik ve nümerik hesaplamalar, diferansiyel denklemler, integrasyon, limit gibi) örneklerle ele alınmış ve çok sayıda çözülecek problemler de bu bölümün sonunda verilmiştir.

PROBLEMLER

1) Verilen lineer denklem sistemlerini MATLAB komutları ile çözünüz.

a) $8x_1 + 2x_2 - 2x_3 = -2$

$$10x_1 + 2x_2 + 4x_3 = 4$$

$$12x_1 + 2x_2 + 2x_3 = 6$$

b) $2x_1 - 6x_2 - x_3 = -38$

$$-3x_1 - x_2 + 7x_3 = -34$$

$$-8x_1 + x_2 - 2x_3 = -20$$

2) Verilen bir elektrik devresi için aşağıda verilen denklemler yazılmıştır.

$$55I_1 - 25I_2 = -200$$

$$-37I_3 - 4I_2 = -250$$

$$-25I_1 - 4I_3 + 29I_2 = 100$$

Buna göre I_1, I_2, I_3 akım değerlerini bulunuz.

3) Dümdüz yere atılan bir topun, $s(t) = \frac{1}{2}gt^2 + v_0t + h_0$ denklemini kullanarak, verilen hız ve yükseklik değerinde yere ne zaman çarpacağını bulunuz.

s = nesnenin zamana bağlı konumu (0 zemin seviyesini göstermektedir. Pozitif yüksekliğin nesnenin yüksek bir platformdan atıldığını gösterdiğini unutmayınız.)

g = yerçekimi ivmesi ($9.8 \text{ m}^2/\text{sn}$ alınız)

v_0 = başlangıç hızı (m/sn)

h_0 =başlangıç yüksekliği (m)

4) $P(x)=5x^3+21x^2-x+10$ polinomunun $x=[-3 \ 2 \ 4 \ 7]$ için polyval komutu kullanarak değerlerini hesaplayınız..

5) $P(x)=\frac{4x^5+3x^3+3x^2+20x+9}{3x^4-6x^3-13x^2+5x-16}$ polinomlarının bölümünü bulunuz.

6) $P(x)=x-2-\frac{1}{x+4}+\frac{1}{x-1}-\frac{1}{x+3}$ polinomunu iki polinomun bölümü olarak gösteriniz.

7). $P(x)=8x^9+4x^7-5x^3+18x+67$ ise bu polinomun köklerini bulunuz.

8) $P(x) = x^3 + 13x^2 + 5x + 7$ ve $P(x) = x^3 + 13x^2 + 5x + 7$ polinomlarının çarpımını **conv** komutu kullanarak bulunuz.

9) $a=[0 \ 5 \ 9 \ 10 \ 17 \ 3]$ ve $b=[0 \ 4 \ 10 \ 22 \ 9 \ 7]$ ile tanımlanan 5. Dereceden iki polinomun katsayısı verilmiştir. Her iki polinomun $x=5$ değerini **polyval** ile hesaplayınız.

10) $P(x) = 4x^4 + 13x^3 + 2x^2 + 4x + 2$, $G(x) = 4x^4 + 13x^3 + 2x^2 + 4x + 2$

a) Verilen $P(x)$ ve $G(x)$ polinomlarının köklerini bulunuz.

b) $G(x)$ polinomunun $P(x)$ polinomuna bölümünü bulunuz.

c) Polinomların çarpımını bulunuz.

11) Tabloda belirlenen x değerlerine karşılık gelen y değerleri verilmiştir. x bağımsız değişken, y bağımlı değişkendir. Verilere göre 3.dereceden bu polinomun en küçük kareler metodu ile (**LSM**) regresyon eğrisini bulunuz ve $x=3$ değeri için polinom değerini bulunuz.

x	0	1	2	4	5
y	1	2	4	16	25

12) (1,3), (2,4), (5,2), (7,8), (9,2), (10,8) noktalarından geçen polinomu lineer interpolasyon yardımı ile bulunuz ve $x=1.5$ noktasındaki değeri hesaplayınız.

13) (1,3), (2,4), (5,2), (7,8), (9,2), (10,8) noktalarından geçen *cubic* polinomu interpolasyon yardımı ile bulunuz ve $x=2.5$ noktasındaki değerini hesaplayınız.

14) (1,3), (2,4), (5,2), (7,-8), (9,2), (10,8) noktalarına uyan en iyi polinomu belirleyiniz ve $x=4.5$ noktası için y değerini hesaplayınız.

15) (0,0), $(\pi/3,1)$, $(\pi,0)$ noktalarından geçen ikinci derece denklemi *polyfit* komutu kullanarak bulunuz.

16) $-3x^2+4x-21$ eğrisinin integralini **int** fonksiyonu ile bulunuz.

17) x' in $0 < x < 10$ aralığındaki değerleri için $P(x)=x^2$ eğrisinin altında kalan alanı bulunuz.

18) Aşağıda verilen integrallerin sonuçlarını hesaplayınız.

a) $\int_0^4 (1 - e^{-2x}) dx$

b) $\int_{-2}^4 (1 - x - 4x^3 + 2x^5) dx$

c) $\int_{-3}^5 (4x - 3)^3 dx$

d) $\int_0^{\pi/2} (6 + 3\cos x) dx$

e) $\int_{-2}^4 (1 - x - 4x^3 + 2x^5) dx$

f) $\int_1^2 \left(x + \frac{2}{x}\right)^2 dx$

g) $\int_0^3 x^2 e^x dx$

- 19) Düşen bir cisim için yukarı doğru oluşan tepki kuvvetinin cismin hızının karekökü ile doğru orantılı olduğunu farz edin. Bu durumda hız şu şekilde hesaplanabilir:

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{n}} t\right)$$

Burada, c_d = ikinci dereceden sürüklenme katsayısı. Eğer $v(0)=0$, $g=9.8 \text{ m/s}^2$, $m=68.1 \text{ kg}$ ve $c_d = 0.25 \text{ kg/m}$ olduğu kabul edilirse cismin 10 saniyede ne kadar düşeceğini analitik olarak hesaplayınız.

- 20) $W = \int p dV$ integralini göz önüne alalım. Burada W iş, p basınç ve V hacmi göstermektedir. MATLAB kullanarak aşağıda verilen veriler için yapılan işi hesaplayınız.

p(kPa)	336,0	294,4	266,4	260,8	260,5	249,6	193,6	165,6
V(m ³)	0,5	2,0	3,0	4,0	6,0	8,0	10,0	11,0

- 21) Aşağıda verilen tablodaki verileri kullanarak alınan yolu

t(dk)	1	2	3,25	4,5	6	7	8	9	9,5	10
v (m/sn)	5	6	5,5	7	8,5	8	6	7	7	5

hesaplayınız.

- 22) $ax^4 - e^x + x \sin x - e^x \cos x$ ifadesinin sembolik olarak (analitik) x 'e göre integrasyonunu bulunuz.

- 23) $I = \int_1^7 x e^x dx$ belirli integral ifadesinin nümerik olarak değerini bulunuz.

- 24) $x^2 \sin x - e^x \cos x$ ifadesinin sembolik olarak köklerini bulunuz.

- 25) Aşağıdakilerin köklerini **solve** fonksiyonu ile bulunuz.

- a) $x^5 + 2x^3 - \frac{1}{2}x^2 + 5x = 4$
b) $x^3 - 3x^2 - 4x = 2$
c) $x^5 - x^4 + x^3 = 17$

26) Aşağıdaki denklem sistemlerini MATLAB fonksiyonlarını kullanarak çözünüz

- | | | | |
|----|---|----|--|
| a) | $x_1 + 4x_2 - x_3 + 4x_4 = 4$
$x_1 - 3x_2 + x_3 - x_4 = 5$
$3x_1 + 2x_2 + 4x_3 - x_4 = 7$
$2x_1 - 2x_2 - x_3 + 4x_4 = 3$ | b) | $x_1 - 0.5x_2 + 0.7x_3 = 1$
$0.7x_1 + x_2 + 0.3x_3 = 0.5$
$-x_1 + 1.4x_2 + 1.5x_3 = 2$ |
| c) | $7x_1 + 3x_2 + x_3 = 21$
$6x_2 + 3x_3 + 7x_4 = -3$
$x_1 + 5x_2 + 4x_4 = 11$
$3x_1 + 2x_2 + 4x_4 = 3$ | d) | $2.4x_1 + 0.5x_2 + 3.9x_3 + 0.7x_4 = 5.7$
$x_1 + 1.2x_2 + 3x_3 + 4x_4 = -9.2$
$2x_1 - 7.1x_2 + 4.1x_3 = 2$
$4.2x_1 - 7.2x_2 + 1.2x_3 - 8.1 = 0$ |

27) Aşağıdaki denklem sistemlerini MATLAB ile çözünüz.

- | | | | |
|----|--|----|---|
| a) | $x_1 + x_2 + 2x_3 + 5x_4 = 2$
$3x_1 + 4x_2 + 2x_3 + 4x_4 = -5$
$2x_1 + 9x_2 + 3x_3 + 2x_4 = 2$
$2x_1 + x_2 + 2x_3 - 3x_4 = -4$ | b) | $6x_1 + 2x_2 + 2x_3 = -2$
$2x_2 + \frac{2}{3}x_3 + \frac{1}{3}x_4 = -3.8$
$x_1 + 2x_2 - 2x_3 = 0$ |
| c) | $x_1 + 4x_2 - x_3 + 4x_4 = 4$
$1.5x_1 - x_2 - 4.5x_3 = 9.5$
$-2.5x_1 - 1.5x_2 + x_3 = -7$ | d) | $2.5x_1 - 4x_2 - 2x_3 - 3x_4 = -20.29$
$3x_1 - 3x_2 - 4.3x_3 + 3x_4 = 8.89$
$2x_1 - 4x_2 - 3.25x_3 - x_4 = 4.84$
$x_1 - 5x_2 + 3.3x_3 - 20x_4 = -14.01$
$-12x_2 - 8x_3 + 3x_4 = 49$ |
| e) | $x_1 + 2x_2 - 12x_3 + 8x_4 = 27$
$5x_1 + x_2 + 7x_3 - 2x_4 = 4$
$-3x_1 + 7x_2 + 9x_3 + 5x_4 = 11$ | f) | $1.02x_1 - 0.25x_2 - 0.3x_3 = 0.515$
$-0.25x_1 + 0.14x_2 + 1.21x_3 = 2.78$
$-0.41x_1 - 1.13x_2 - 0.15x_3 = 1.555$ |
| g) | $4x_1 + 0.24x_2 - 0.08x_3 - 0.16x_4 = 8$
$0.09x_1 + 3x_2 - 0.15x_3 - 12x_4 = 7$
$0.04x_1 - 0.08x_2 + 4x_3 + 0.06x_4 = 20$
$0.02x_1 + 0.06x_2 + 0.04x_3 - 10x_4 = 1$ | h) | $-1.05x_1 + 3.96x_2 + 1.15x_3 = -2.71$
$3.96x_1 - 1.01x_2 = 9.5$
$-3.09x_1 - 1.15x_2 + 3.96x_3 = 5.98$ |
| i) | $x_1 + x_2 - 2x_3 = 1$
$3x_1 + 2x_2 + 4x_3 = 2$
$3x_1 + x_2 + x_3 = 4$
$2x_1 - 3x_2 + x_3 = -1$ | | |

BÖLÜM 3 - DENEYSEL VERİLERİN FORMÜLE EDİLMESİ

Amaç

Bu bölümün bitiminde,

- Proseslerdeki verilerin direkt ve indirekt ölçülmesi
- Lineer olmayan denklemlerin lineer olarak nasıl çizileceği
- Üssel ve kuvvet fonksiyonları ve onların semi-log ve log-log ölçeklerindeki çizimi
- En küçük kareler metodunun temelleri ve verilen verilere denklem uydurma

konularında yeterli bilgiye sahip olacağız

Detaylı olarak matematiksel modellemeye geçmeden önce, öğrencilere son derece yardımcı olacak bazı kavramları gözden geçirelim. Böylece öğrencinin, laboratuvarında elde ettiği ölçümleri nasıl değerlendirebileceğini uygulamalarla görelim.

3.1 Proseslerde Verilerin Direkt ve İndirekt Ölçülmesi

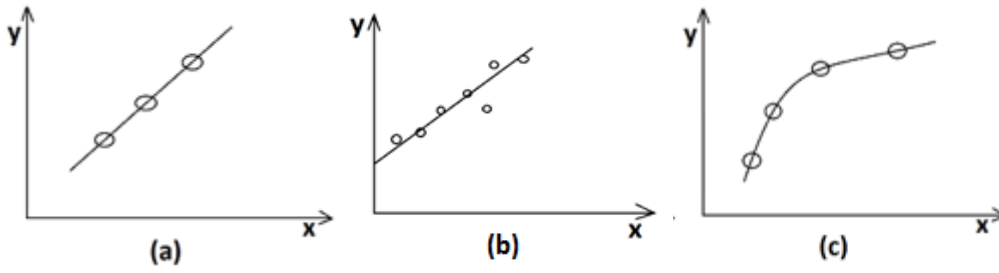
Herhangi bir, diyelim, kimyasal prosesin ilerleyişini takip edebilmek için **sıcaklık (T), basınç (P), konsantrasyon (=derişim, C), yoğunluk (ρ), absorbands (A)** vb. gibi değişkenlerden bir veya birkaçı ölçülür. Bunların bazıları **direkt** olarak (T, P gibi), bazıları da **endirekt** olarak ölçülür. Örneğin, bir prosesteki konsantrasyonunu direkt olarak ölçmek çoğu zaman mümkün değildir. Onun yerine diğer bir değişkenden, örneğin, absorbands, konsantrasyonun bir *fonksiyonu* olarak ölçülebilir. Sonra da *konsantrasyon-absorbans* arasındaki *kalibrasyon eğrisinden* veya aralarındaki *matematiksel bağıntıdan* her absorbands için gerekli konsantrasyon bulunabilir.

Diyelim ki **x-y** arasında aşağıdaki verileri ölçtük (Tablo 3.1.). Burada **y** proses değişkeni, **x** ise direkt ölçülen miktar olsun. Acaba **x** ile **y** arasında nasıl bir ilişki (bağıntı) vardır?

Tablo 3.1

x	0.83	1.2	1.8	2.4	1.9	3.5
y	14.8	15.5	18.5	23.4	25.1	31.1

Burada ilk akla gelen şey **x-y** arasındaki ilişkiyi bir *Kartezyen koordinat* sisteminde çizmektir. Eğer ilişki bir doğru veriyorsa, **x-y** arasındaki matematiksel bağıntıyı hemen bulabiliriz (Şekil 3.1a). Eğer verilerin hepsi aynı doğru üzerinde olmayıp sağa-sola sapmalar gösteriyorlarsa buna da yine bir doğru uydurmak mümkündür (Şekil 3.1b).. (Daha sonra **en küçük kareler metodu** kısmında böyle bir doğrunun nasıl çizildiğini göreceğiz). Eğer Şekil 3.1c’de olduğu gibi bir eğri çıkmışsa o zaman bu eğriye ya bir *polinom* uydurmak zorundayız yahut *interpolasyon* veya *ekstrapolasyonla*, sırasıyla, ara ve dış değerleri bulmak mümkündür. Yukarıdaki Tablo 3.1’de **x-y** arasında verilen bilgiler **MATLAB** paket program ile çizilmiş ve Şekil 3.2’de gösterilmiştir Her ne olursa olsun, bir mühendis daima, veriler eğri de olsa, şu veya bu şekilde koordinatlarda bazı değişiklikler yaparak çizimi **lineer** bir şekle koymaya çalışır. Çünkü lineer bir doğrudaki *sadece iki nokta bilmemiz o olayın formülü hakkında gerekli tam bilgiyi verir.*



Şekil 3.1 x-y arasındaki değişik ilişkiler

Eğer eğri hala lineer duruma getirilemiyorsa o zaman *iki noktalı lineer interpolasyon* kullanmak gerekir. x_1 ve x_2 birbirinden çok uzak değilse, (x_1, y_1) ve (x_2, y_2) noktalarından geçen doğrunun denklemi

$$y = y_1 + \frac{x - x_1}{x_2 - x_1}(y_2 - y_1) \quad \text{ile verilir.}$$

Bu denklem yardımıyla lineer interpolasyonu gerçekleştirebiliriz. *Ekstrapolasyonlarda* ise bu denklemin kullanılışında çok **dikkatli** olmak gerekir çünkü hiç beklenmedik yanlış sonuçlar elde edebiliriz!

Tablo 3.1'deki değerler alınarak aşağıdaki MATLAB programıyla Şekil 3.2. çizilmiştir

Kod 3.1

```
x=[0.83 1.2 1.8 2.4 2.9 3.5];
```

```
y=[14.8 15.5 18.5 23.4 25 31];
```

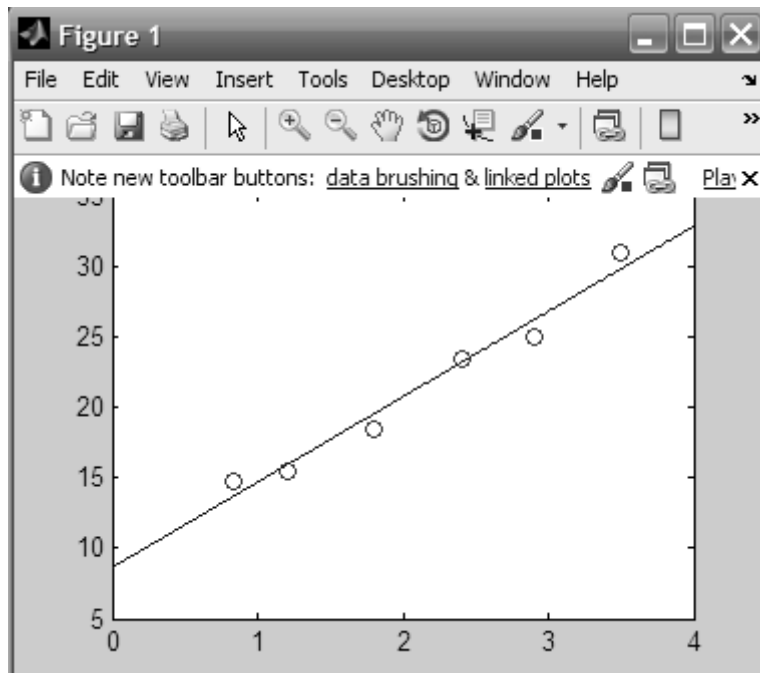
```
Y=polyfit(x,y,1)
```

```
Y = 6.0512 8.6288
```

```
xx=0:0.01:4;
```

```
yy=6.05*xx+8.63;
```

```
plot(x,y,'or',xx,yy,'-')
```

Şekil 3.2 Tablo 3.1'deki verilerin MATLAB ile **en küçük kareler yöntemi**ni kullanarak çizilmesi

Örnek 3.1 *Lineer interpolasyon kullanımı*

Değişik t zamanlarında " c " değerleri aşağıdaki şekilde ölçülmüştür.

c	3.4	7.1	10.2
t	2.5	5.2	7.7

$c(t=3.4)$ ve $c(t=8)$ 'i lineer interpolasyonla bulunuz.

Çözüm:

$$c = c_1 + \frac{t - t_1}{t_2 - t_1} (c_2 - c_1) = 3.4 + \frac{t - 2.5}{5.2 - 2.5} (7.1 - 3.4)$$

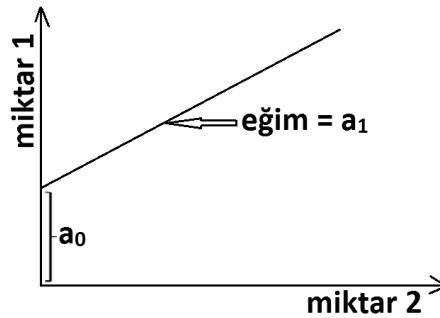
$c(3.4) = 4.8$ bulunur.

$c(8)$ 'i de siz bulunuz!

$y = ax + b$ ifadesi bir doğru denklemini verir ve bu doğrunun eğimi " a ", kesim noktası da " b " dir. Daha genel olarak şunu söyleyebiliriz: İki büyüklük arasındaki ilişki

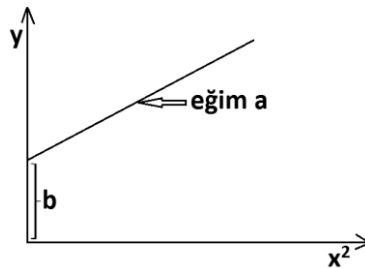
$$(\text{miktar 1}) = a_1 (\text{miktar 2}) + a_0$$

şeklinde ise o zaman (miktar 1) 'in (miktar 2) 'ye karşı çiziminden bir doğru elde ederiz. Bu durum Şekil 3.3'te gösterilmiştir.



Şekil 3.3 (Miktar 1)'in (miktar 2)'ye karşı çizimi

Örneğin $y = ax^2 + b$ şeklinde bir ifadeyi $y-x^2$ arasında çizersek Şekil 3.4'de görüldüğü gibi bir doğru elde ederiz.



Şekil 3.4 Lineer olmayan bir ifadeyi lineer halde çizmek

Örnek 3.2 *Lineer olmayan ifadelerin lineer halde çizilmesi*

Lineer olmayan bazı ifadelerin, düz doğru elde etmek için nasıl çizilmesi gerektiği aşağıda verilmiştir.

1. $y^2 = a x^3 + b$ ise $y^2 - x^3$ çizimi bir doğru verir

2. $y = \frac{a}{x^2} + b$ ise $y - \frac{1}{x^2}$ çizimi bir doğru verir

3. $\frac{1}{y} = a(x+6) + b$ ise $\frac{1}{y} - (x+6)$ çizimi bir doğru verir

4. $\cos y = a(x^3 + 5)$ ise $\cos y - (x^3 + 5)$ çizimi bir doğru verir ve orijinden geçer.

$r = \frac{k_1}{k_2 + k_3 c} \Rightarrow \frac{1}{r} = \frac{k_3}{k_1} c + \frac{k_2}{k_1}$; Böylece $\frac{1}{r} - c$ arasındaki ilişki eğim $= \frac{k_3}{k_1}$,

kesim noktası $= \frac{k_2}{k_1}$ olur

6. $y = 1 + (m x^2 + n)^{1/2}$ ise $(y-1)^2 - x^2$ arasındaki ilişki m =eğim, n =kesim noktası olur.

Durumu **özetleyecek** olursak:

$$f(y) = a_1 g(x) + a_0$$

formunda verilen bir ifadede $f(y)$, $g(x)$ ' e karşı çizilerek lineerlik elde edilebilir. Burada eğim " a_1 " ve kesim noktası " a_0 "dır.

$a_1 = \frac{f_2 - f_1}{g_2 - g_1}$, $a_0 = f_1 - a_1 g_1$ veya $a_0 = f_2 - a_1 g_2$ ifadesiyle verilir.

Bu gibi denklemleri, alacalığınız derslerin laboratuvar çalışmalarındaki deneylerde kullanma imkânını bulacaksınız.

Örnek 3.3 Verilere bir doğru uydurmak

Debi ile debimetre okuması arasındaki kalibrasyonda aşağıdaki ölçümler alınmıştır [30].

Debi, V (L/dk)	Debimetre okuması (R)
34.8	10
64.8	20
154.8	50
214.8	70
274.8	90

a) V-R kalibrasyon eğrisini çiziniz.

b) Debimetre okuması (R) 58 ise debi nedir?

Çözüm:

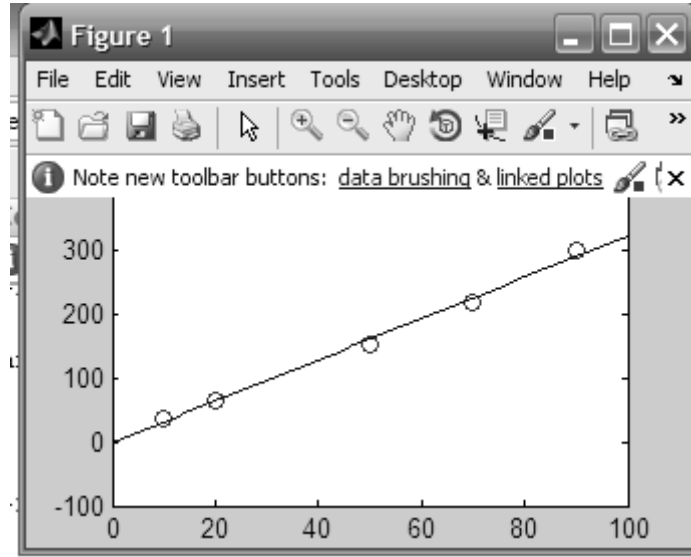
Önce **V** ile **R** arasındaki ilişki V-R düzleminde çizilirse Şekil 3.5'de görüldüğü gibi bir durumla karşılaşılır (*Sadece noktalara dikkat ediniz*). Doğrusal grafik ise MATLAB ile en küçük kareler metodu kullanılarak çizilmiştir

- a) Görüldüğü gibi V-R arasında hemen hemen lineer bir ilişki vardır. Buna göre en uygun doğru MATLAB ile çizildikten sonra, denklem

$$V=3.24R-0.68$$

olarak bulunur.

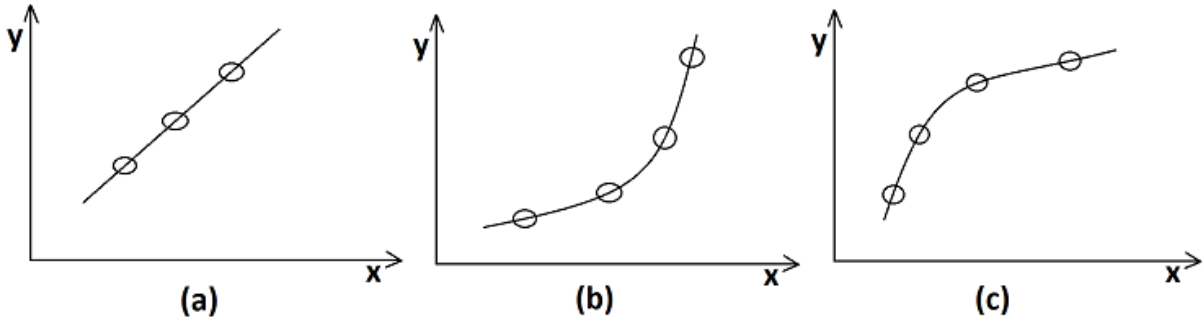
$$V(58)=3.24(58)-0.68=187.2 \text{ L/dk olarak bulunur.}$$



Şekil 3.5 Debi ile debimetre arasındaki bağıntı

Aşağıdaki soruyu cevaplayınız.

Şekil 3.6 (a), (b) ve (c) de x-y ilişkisi gösterilmiştir. *Lineer interpolasyon* kullanıldığında hangisinde *doğru* değer, daha *düşük* değer ve daha *yüksek* değer bulunur? Açıklayınız.



Şekil 3.6 Lineer interpolasyonun kullanılmasında ne gibi hatalara düşüleceğini gösteren grafikler

Daha önce de belirttiğimiz gibi bir mühendis daima, şu veya bu şekilde, değişkenler arasında lineer bir bağıntı kurmaya çalışır. Lineer değilse, o zaman onu çeşitli koordinatlarda lineer bir halde çizmeye çalışmalıdır.

Aşağıdaki denklemlere bir göz atalım:

$$y = 5x + 8 \quad \text{lineer}$$

$$y = a(x^2 - 2) + b \quad \text{non-lineer}$$

$$y = \sin(\pi x) / \sqrt{x} \quad \text{non-linear}$$

Gerek lineer, gerekse non-linear bir eşitlik verildiği zaman bağımsız değişken " x " in her değeri için bağımlı değişken " y " yi hesap edebiliriz. Ama bunun tersi her zaman o kadar kolay olmayabilir (y verilince x i hesaplamak gibi!).

Diyelim ki Şekil 3.6.a' da görüldüğü gibi her x için y lineer bir doğru üzerinde olsun. O zaman

$$y = a_1 x + a_0 \quad \text{eşitliğinden} \quad a_1 = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{ve} \quad a_0 = y_1 - a_1 x_1 \quad \text{veya} \quad y_2 - a_1 x_2 \quad \text{den de kesim noktasını}$$

bulabiliriz. " a_1 " ve " a_0 " bulunduktan sonra artık matematiksel eşitliğimiz elimizde olup istediğimiz şekilde kullanabiliriz.

Örnek 3.4 *Lineer olmayan bir bağıntıyı lineer halde çizmek*

Bir orifizdeki hacimsel debi V (mL/sn) ile sıcaklık T (°C) arasında aşağıdaki ölçümler yapılmıştır [30].

T	15	25	45	75
V	14.8	20.1	31.2	60.4

hacimsel debi " V " nin, " T " nin karekökü ile değiştiğine inanılmaktadır. Yani

$$V = a T^{1/2} + b$$

şeklindedir. Buna göre V ile T arasındaki tam bağıntıyı bulunuz.

Bunun için aşağıdaki tabloyu hazırlamamız gerekir:

$T(^{\circ}\text{C})$	15	25	45	75
$T^{1/2}$	3.87	5.00	6.71	8.66
$V(\text{mL/sn})$	14.76	20.14	27.73	38.47

" V " değeri, $T^{1/2}$ ye karşı çizilir.

Bu çizim, MATLAB yardımıyla aşağıdaki program kullanılarak çizilmiş ve denklem

$V = 4.913T^{(1/2)} - 0.450$ olarak bulunmuştur.

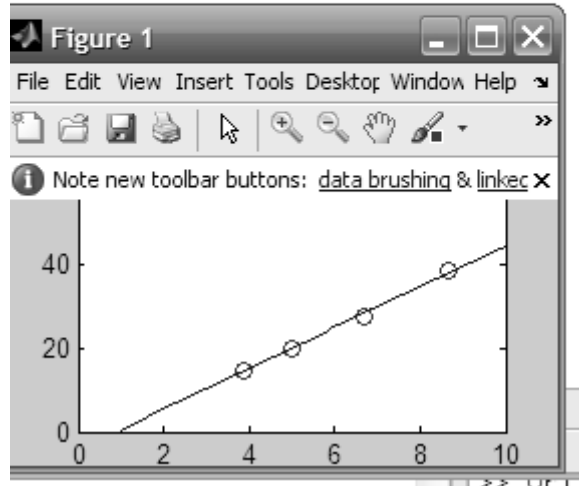
Lineerliğe uygunluğu da grafikte açıkça görülmektedir.

Kod 3.2 Örnek 3.4 için

```

T1=[3.87 5.00 6.71 8.66];
V=[ 14.76 20.14 27.73 38.47];
P=polyfit(T1,V,1)
% V=4.913*T^(1/2)-0.450
TT1=1:.1:10;
VV=polyval(P,TT1);
plot(T1,V,'or',TT1,VV,'-')

```



Şekil 3.7 -Örnek 3.4'e ait grafik

Burada görüldüğü gibi V ile $T^{1/2}$ arasında yaklaşık olarak bir lineerlik vardır.

3.2 Üstel ve Kuvvet Fonksiyonları

Mühendislikte sık sık karşımıza çıkıp da lineer olmayan iki tip fonksiyon vardır. Bunlar üstel (**eksponansiyel**=*exponential*) ve **kuvvet** (*power-law*) fonksiyonlarıdır. Burada logaritmanın, üssel fonksiyonun tersi olduğunu unutmamak gerekir. Bir hatırlatma için, örneğin,

$$P = \exp(Q) \rightarrow \ln P = Q$$

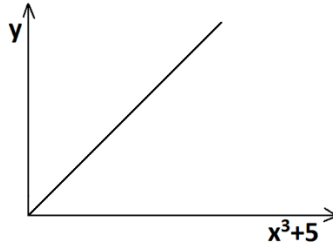
$$\ln(a*b) = \ln a + \ln b, \text{ ve } \ln(a/b) = \ln a - \ln b \text{ dir.}$$

Adi logaritma (10 tabanına göre) ile tabii logaritma arasında

$$\ln x = 2.3 \log_{10} x$$

bağıntısı geçerlidir.

Eğer $y = ax^b$ ise, $\ln y = \ln a + b \ln x$ olur. Böylece $\ln y \rightarrow \ln x$ çizimi bir doğru verir, bu doğrunun eğimi " b ", kesim noktası da " $\ln a$ " dır. Bu tip bir fonksiyonu çizmek için **log-log eksenleri** kullanılmalıdır.

Soru

Şekil 3.8

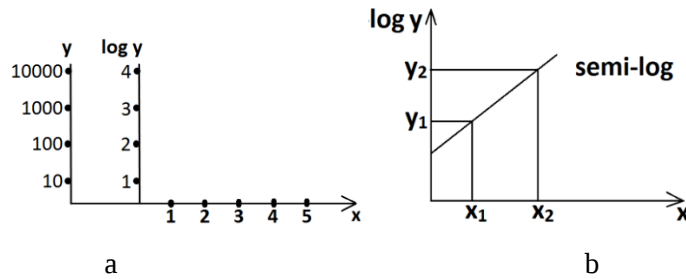
y ile x arasındaki ilişki Şekil 3.8 deki gibi ise y - x bağıntısını bulunuz (orijinden geçiyor).

3.2.1 Logaritmik Koordinatlar

Gerek üstel, gerekse kuvvet fonksiyonlarında bir doğru elde etmek için her iki tarafın logaritması alınır. Üstel fonksiyonlarda sol taraf logaritmik olur, kuvvet fonksiyonlarında ise her iki taraf logaritmik olur. Bunları Kartezyen koordinat sisteminde çizmek için verilen değerlerin logaritmalarını almak gerekir. *Bazı hallerde bu logaritma alma işlemi, çizimden daha çok zaman almaktadır.* Bu nedenle, **semi-log** veya **log-log** denilen ölçekler türetilmiştir. Koordinat sistemleri logaritmik olarak hazırlandığından, biz doğrudan sayının logaritması yerine kendisini koyar ve çizimi ona göre yaparız. Bunu bir örnekle açıklayalım:

$$y = a e^{bx} \quad \text{ise} \quad \ln y = \ln a + b x \quad \text{olur.}$$

Burada $\ln y$ (veya $\log y$) ‘yi, x ’ e karşı bir semi-log grafiğinde çizmeğe çalışalım,



Şekil 3.9

Böylece "**log y**" ye **paralel** bir eksen çizip onun da üzerine tam karşılığı olan y değerlerini yazalım (Şekil 3.9a). Şimdi y -ekseni logaritmik olup, eğer biz " y " nin değerini olduğu gibi ordinata koyarsak adeta logaritmasını almış gibi oluyoruz. Yukarıdaki örnekte (Şekil 3.9 b) koordinat sistemi bir **semi-log** örneğidir, çünkü apsis bu durumda lineerdir.

Eğer bir kuvvet fonksiyonu varsa ($y = a x^b$ gibi) ve bunu lineer halde çizmek istersek, o zaman

$$\ln y = \ln a + b \ln x \quad \text{elde ederiz.}$$

$\ln y$ ile $\ln x$ arasındaki ilişki lineer olup, eğim "**b**", kesim noktası da " **$\ln a$** " olacaktır. Bu durumda hazırlanan grafik kâğıdında **hem ordinat hem de apsis logaritmik** ölçekte olup, bu grafiğe **log-log** grafiği denilir. Gerek **semi-log**, gerekse **log-log** çizimlerde eğimler aşağıdaki şekilde bulunur.

$$y = e^{bx}$$

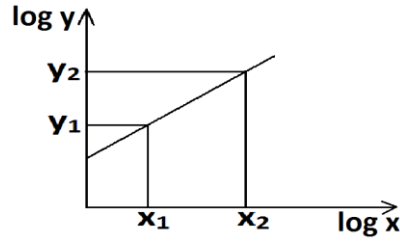
şeklinde bir fonksiyonda

$$\ln y = \ln a + b x$$

$$b = \frac{\ln y_2 - \ln y_1}{x_2 - x_1} = \text{eğim},$$

$$\ln a = \ln y_1 - b x_1 \text{ veya } \ln a = \ln y_2 - b x_2 = \text{kesim noktası}$$

$y = a x^b$ formundaki bir fonksiyon da Şekil 3.10' da görüldüğü gibi çizilir.



Şekil 3.10 log-log grafiği

burada,

$$b = \frac{\ln y_2 - \ln y_1}{\ln x_2 - \ln x_1} = \frac{\ln(y_2/y_1)}{\ln(x_2/x_1)} = \frac{\log(y_2/y_1)}{\log(x_2/x_1)} \rightarrow \text{eğim}$$

$$\begin{aligned} \ln a &= \ln y_1 - b \ln x_1 \text{ veya} \\ &= \ln y_2 - b \ln x_2 \rightarrow \text{kesim noktası} \end{aligned}$$

Bu kurallara çok dikkat etmek gerekir, çünkü öğrenciler bunları lineer koordinat sistemleriyle karıştırmaktadırlar! Dolayısıyla beklenmedik yanlış sonuçlar elde edebilirler.

Örnek 11.5 Verileri **semi-log** ve **log-log** grafiğinde çizerek onlara denklem uydurmak [30]

F, **t** 'ye karşı çizildiğinde ($t_1=15$, $F_1=0.298$) ve ($t_2=30$, $F_2=0.0527$)

a) **semi-log** grafik üzerinde

b) **log-log** grafik üzerinde lineer bir ilişki elde edilmektedir.

Buna göre her iki durum için **F** ile **t** arasındaki bağıntıyı bulunuz.

Çözüm:

a) Semi-log için $F = a e^{b t}$ şeklinde olması gerekir veya $\ln F = \ln a + b t$ 'dir.

$$b = \frac{\ln(F_2/F_1)}{t_2 - t_1} = \frac{\ln(0.0527/0.298)}{30 - 15} = -0.1155$$

$$\ln a = \ln F_1 - b t_1 = \ln(0.298) + (0.1155)(15) = 0.5218 \Rightarrow a = 1.685$$

Buna göre $F = 1.685 e^{-0.1155 t}$ ilişkisi vardır.

b) log-log için:

$$F = a t^b \text{ veya } \ln F = \ln a + b \ln t$$

$$b = \frac{\ln(F_2/F_1)}{\ln(t_2/t_1)} = \frac{\ln(0.0527/0.298)}{\ln(30/15)} = -2.50$$

$$\ln a = \ln F_1 - b \ln t_1 = \ln(0.298) + 2.5 \ln(15) = 5.559 \rightarrow a = 260$$

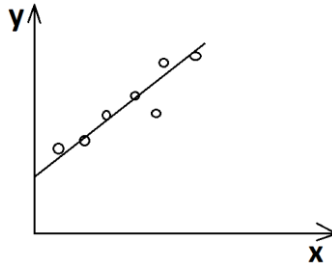
Böylece $F = 260t^{-2.50}$ olur.

MATLAB’de semi-log ve log-log skalaları zaten vardır ve gerek kuvvet, gerekse üstel bağıntıların kolayca çizilebileceği daha önce görülmüştü.

3.3 Dağınık Verilere Doğru Uydurma

Genellikle laboratuvarlarda veya diğer yerlerde yapılan ölçümlerde bağımsız olarak ölçülen değişkenle bağımlı değişken arasındaki ilişki herhangi bir koordinat sisteminde çizildiğinde ya bir doğru (veya doğruya yakın), ya da bir eğri verir. Doğrusal gözükten veriler bile aynı doğru üzerinde olmayabilir, bazı sapmalar gösterir. Şekil 3.11 de görüldüğü gibi bütün veri noktaları aynı doğru üzerine düşmemektedir. Ama genel trend (eğilim) ilişkinin bir doğru ile ifade edilebileceğini gösteriyor!

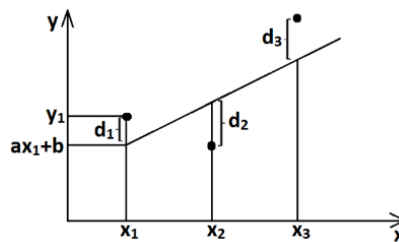
İşte böyle dağınık verilere *uyan en uygun doğruyu* ancak **en küçük kareler yöntemiyle** çizmek mümkündür. Bu yöntemi, aşağıda örneklerle göreceğiz.



Şekil 3.11 Dağınık verilere bir doğru uydurma

3.3.1 En Küçük Kareler Yöntemi (The Least Squares Method)

Bu istatistiksel yöntemde amaç, **dağınık olan veri noktalarını temsil eden en uygun doğruyu** çizmektir [20,30,39]. Şekil 3.12’yi dikkate alalım:



Şekil 3.12 En küçük kareler yönteminin uygulanma prensibi

Sadece 3 veri noktası (**koyu siyah noktalar**) verilmiş olsun. Biz, bu noktaları temsil eden doğruyu göz kararı da çizebilirdik, fakat bunun **bilimsel** hiç bir anlamı olamaz. Çünkü 100 kişiye verseniz 100'ü de farklı birer doğru çizer, gözüne ne kestirmişse!

Eğer doğrunun denklemi $y = a_1 x_i + a_0$ ise her ölçülen x_i için bir y_i bulunmuş olur. "y" nin çizilen doğru üzerindeki değeri $a_1 x_i + a_0$ olacaktır. Böylece sapma, gerçek veri değerinden (d_i) kadar uzakta olacaktır (burada $i=1, 2, 3, \dots$). Buna göre sapma $d_i = (a_1 x_i + a_0) - y_i$ kadardır. Burada d_i pozitif veya negatif olabilir. Burada amaç, " d_i " leri minimuma indirmektedir. Diyelim ki " n " tane noktamız var: $(x_1, y_1), (x_2, y_2), \dots (x_i, y_i) \dots (x_n, y_n)$. Bu durumda " n " tane d_i olacaktır. En uygun doğru, " **d_i " lerin karelerinin toplamını minimuma indirgeyen doğrudur**". Bu şekilde **en uygun doğrunun** " a_1 " ve " a_0 " değerlerini bulabiliriz. Şöyle ki

$$\Phi(a_0, a_1) = \sum_{i=1}^n d_i^2 = \sum_{i=1}^n (a_1 x_i + a_0 - y_i)^2$$

olsun. Bu ifadenin *minimumunu* bulmak için yukarıdaki $\Phi(a_0, a_1)$ fonksiyonunun hem " a_1 " ve hem de " a_0 " a göre kısmi türevleri alınıp sıfıra eşitlenir. Elde edilen iki denklem, " a_1 " ve " a_0 " için çözülürse, o zaman " a_1 " ve " a_0 " için en uygun değerleri, başka bir deyimle bütün veri noktalarını temsil eden en uygun doğrunun denklemi bulunmuş olur.

En küçük kareler yönteminde ölçüm değerleri olan y_i ile işlemler sonucunda elde edilecek olan fonksiyon değerleri arasındaki farkın minimum olması aranır. Ancak “-“ ve “+” farklar birbirlerini şans eseri de olsa götüreceğinden farkların toplamının çok küçük olması doğru ile verilerin hemen hemen üst üste çakıştığı anlamına gelmez. Böyle bir yanılgıdan kurtulmak için farkların *karelerinin toplamının minimum* yapılması gerekir. Bu nedenle y_i ile $f(x_i)$ değerleri arasındaki farkların karelerinin toplamı alınır.

En küçük kareler yönteminde ölçüm değerleri olan y_i ile işlemler sonucunda elde edilecek olan fonksiyon değerleri arasındaki farkın minimum olması için önce y_i ile $f(x_i)$ değerleri arasındaki farkların toplamının karesi alınır.

$$\sum_{i=1}^n [f(x_i) - y_i]^2 = \sum_{i=1}^n [a_0 + a_1 x_i - y_i]^2$$

Burada elde edilecek olan farkların karelerinin toplamının a_0 ve a_1 katsayılarına göre minimum olması gerekir. Bundan dolayı yukarıda elde edilen eşitliğin a_0 ve a_1 katsayılarına göre *kısmi* türevleri alınarak sıfıra eşitlenirse

$$\frac{d \sum_{i=1}^n [a_0 + a_1 x_i - y_i]^2}{da_0} = 2 \sum_{i=1}^n [a_0 + a_1 x_i - y_i] = 0$$

$$\frac{d \sum_{i=1}^n [a_0 + a_1 x_i - y_i]^2}{da_1} = 2 \sum_{i=1}^n [a_0 + a_1 x_i - y_i] x_i = 0 \quad \text{olur.}$$

Yukarıdaki denklemler biraz düzenlenirse,

$$\begin{aligned} a_0 n + a_1 \sum_{i=1}^n x_i &= \sum_{i=1}^n y_i \\ a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 &= \sum_{i=1}^n x_i y_i \end{aligned}$$

elde edilir. Ortaya çıkan bu iki bilinmeyenli iki lineer denklem, matris ve vektör formunda yazılırsa

$$\begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}$$

olur.

Bundan sonra yapılması gereken, bu denklem takımından bilinmeyen a_0 ve a_1 katsayılarını bulmaktır. Katsayıları bulma işlemi **Cramer** yöntemi, ya da *ek-matris* veya *Gauss eleme* yöntemlerinden birinin kullanıldığı ve yahut da matrisin tersini almak yoluyla gerçekleştirilerek yukarıdaki eşitlik aşağıdaki gibi de yazılabilir.

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}$$

Eğer beklenen fonksiyon ikinci dereceden bir fonksiyon olsaydı bu durumda fonksiyonun karşılığı;

$$f(x) = a_0 + a_1 x + a_2 x^2$$

şeklinde tanımlanmış olurdu. Bu durumda katsayıları temsil eden değerlerin sayısı 3'e çıkacak ve bunun sonucu olarak da en küçük kareler yönteminde kullanılacak olan matrislerin satır sayısı üçe çıkmış olacaktır. Bu durum aşağıda görülmektedir.

$$\begin{bmatrix} n & \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^4 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n x_i^2 y_i \end{bmatrix}$$

Eğer fonksiyonun karşılığı daha genel bir hale getirilerek **n'inci** dereceden bir polinom olarak tanımlanacaksa aşağıda verilen eşitlik kullanılmalıdır.

$$\underbrace{\begin{bmatrix} n & \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \cdots & \sum_{i=1}^n x_i^m \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 & \cdots & \sum_{i=1}^n x_i^{m+1} \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^4 & \cdots & \sum_{i=1}^n x_i^{m+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^n x_i^m & \sum_{i=1}^n x_i^{m+1} & \sum_{i=1}^n x_i^{m+2} & \cdots & \sum_{i=1}^n x_i^{2m} \end{bmatrix}}_{\mathbf{A}} \times \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n x_i^2 y_i \\ \vdots \\ \sum_{i=1}^n x_i^m y_i \end{bmatrix}}_{\mathbf{b}}$$

Bu durumda vektör-matris operasyonundan $a = A^{-1} \times b$ olur ve **a** vektörü $(a_0 \ a_1 \ \dots \ a_m)$ kolayca bulunabilir.

MATLAB’de eğri uydurma işlemi **polyfit** komutu kullanılarak gerçekleştirilir. Bu komutun kullanımına ait form aşağıda görülmektedir.



Bu komutun kullanımı ile elde edilen sonucun doğruluğu fonksiyonu tanımlayacak olan polinomun derecesine bağlıdır. Eğer bu değer **n=1** olarak seçilecek olursa elde edilecek polinoma ait ifade lineer olacak demektir. Buna göre **n=2** olduğunda ise kuadratik bir form elde edilecek demektir. **polyfit**’in kullanımına ait örnek bir uygulama ile aşağıda görülmektedir.

Tablo 3.2’de x ve y değerlerinin değişimi

x	1	4	9	13	21
y	2	13	22	28	43

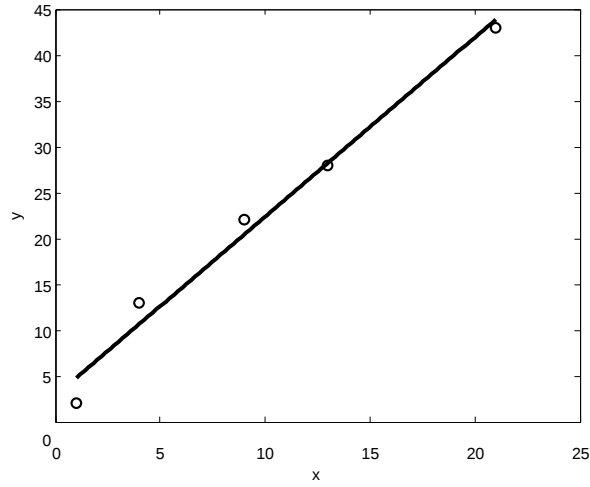
Öncelikle yukarıda verilen tabloda yer alan değerlere göre ortaya çıkacak eğrinin **birinci dereceden** polinoma göre elde edilmesini gerçekleştirelim. Bunun için gerekli olan örnek MATLAB programı ve elde edilen sonuçları karşılaştırmalı olarak aşağıda görülmektedir.

Görüldüğü gibi burada elde edilen **ikinci dereceden** polinom sonucu biraz daha doğrudan ziyade parabolleşerek daha doğru sonuç vermektedir. Bu işlem, sahip olunan veri uzunluğu **n** ile tanımlandığında eğri uydurmada kullanılabilecek olan polinom formun maksimum derecesi **n-1** oluncaya kadar arttırılabilir. Yukarıda verilen örnek için **n** değeri 5 olduğuna göre eğer, uydurmada kullanılabilecek olan polinomun maksimum derecesi 4 olacaktır. Buna ait örnek MATLAB programı ve uygulama sonucu aşağıda görülmektedir:

Kod 3.3

```
% verilere doğru denklemi uydurma n=1
x=[1 4 9 13 21];
y=[2 13 22 28 43];
P=polyfit(x,y,1) % birinci dereceden polinom
f=poly2str(P,'x')
x2=linspace(x(1),x(end));
for k=1:length(x2)
    y2(k)=polyval(P,x2(k));
end
plot(x,y,'o',x2,y2,'-') % sonuç
P =1.9506 2.8738 % doğru denkleminin katsayıları
f=1.9506 x + 2.8738 % doğru denklemi
```

İlgili grafik Şekil 3.13'te verilmiştir.



Şekil 3.13

Kod 3.4

İkinci derece polinom için gerekli program ve ilgili grafik (Şekil 3.14) aşağıdadır:

% verilere kuadratik denklem uydurma, n=2

```
x=[1 4 9 13 21];
```

```
y=[ 2 13 22 28 43];
```

```
P=polyfit(x,y,2) % ikinci dereceden polinom
```

```
f=poly2str(P,'x')
```

```
x2=linspace(x(1),x(end));
```

```
for k=1:length(x2)
```

```
    y2(k)=polyval(P,x2(k));
```

```
end
```

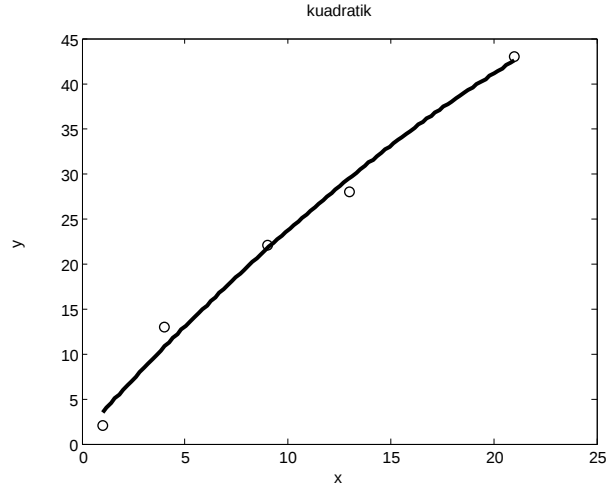
```
plot(x,y,'o',x2,y2,'-')
```

Sonuç

```
P = -0.0268 2.5416 0.9924
```

```
f = -0.02678 x^2 + 2.5416 x + 0.99239
```

ikinci derece polinom için grafik ise aşağıdadır.



Şekil 3.14

Kod. 3.5

Son olarak 4. dereceden polinomu deneyelim.

% verilere 4.dereceden denklem uydurma, n=4

```
x=[1 4 9 13 21];
```

```
y=[ 2 13 22 28 43];
```

```
P=polyfit(x,y,4) % dördüncü dereceden polinom
```

```
f=poly2str(P,'x')
```

```
x2=linspace(x(1),x(end));
```

```
for k=1:length(x2)
```

```
    y2(k)=polyval(P,x2(k));
```

```
end
```

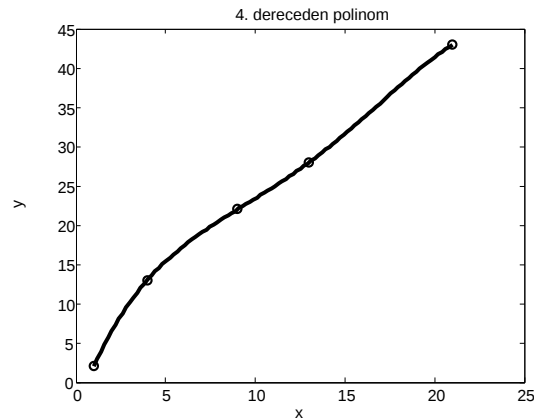
```
plot(x,y,'o',x2,y2,'-')
```

Sonuç:

```
P = -0.0006 0.0340 -0.6153 6.0830 -3.5011
```

```
f = -0.00064338 x^4 + 0.034038 x^3 - 0.61529 x^2 + 6.083 x - 3.5011
```

Grafik (Şekil 3.15) ise aşağıdaki gibidir:



Şekil 3.15

Buradan görüleceği gibi $n=4$ polinomu verilere daha iyi uymaktadır.

Daha kısa yoldan lineer denklemin katsayılarını ($\mathbf{a}_1$ ve $\mathbf{a}_0$) tayin etmek için ikinci ve daha pratik bir yol da aşağıdaki gibidir [22,30]

Bunun için öncelikle aşağıdaki tarifleri yapalım:

$$S_x = \frac{1}{n} \sum_{i=1}^n x_i; \quad S_{xx} = \frac{1}{n} \sum_{i=1}^n x_i^2$$

$$S_y = \frac{1}{n} \sum_{i=1}^n y_i; \quad S_{xy} = \frac{1}{n} \sum_{i=1}^n x_i y_i$$

$$y = \mathbf{a}_1 x + \mathbf{a}_0$$

ifadesinin bütün noktaları temsil eden en iyi doğru olduğunu kabullenirsek

$$a_1 = \frac{S_{xy} - S_x S_y}{S_{xx} - (S_x)^2}, \quad a_0 = \frac{S_{xx} S_y - S_{xy} S_x}{S_{xx} - (S_x)^2}$$

ifadeleriyle bulunur.

Bu son yaklaşımla ilgili bir örnek, durumu daha açıklayıcı kılacaktır.

Örnek 3.6 *En küçük kareler yönteminin kullanımı*

P ile **t** arasında aşağıdaki gibi bir bağıntı vardır [30]

$$P = \frac{1}{m t^{1/2} + r}$$

(Burada "**m**" ve "**r**" sabit sayılar olup belirlenmesi istenmektedir).

Alınan veriler:

P	0.279	0.194	0.168	0.120	0.083
t	1.0	2.0	3.0	5.0	10.0

En küçük kareler yöntemini kullanarak "**m**" ve "**r**" in değerlerini bulunuz.

Çözüm:

Yukarıda verilen ifade, aşağıdaki şekilde yazılabilir.

$$\frac{1}{P} = m t^{1/2} + r$$

Böylece $1/P$ ile $t^{1/2}$ arasında lineer bir ilişki olması gerekir.

$y=1/P$ ve $x=t^{1/2}$ diyelim.

y	3.584	5.155	5.952	8.333	12.048
x	1.0	1.414	1.732	2.236	3.162

$y=mx+r$ den

$$S_x = 1.909, \quad S_y = 7.014, \quad S_{xx} = 4.200, \quad S_{xy} = 15.582$$

$$\Rightarrow m = \frac{S_{xy} - S_x S_y}{S_{xx} - (S_x)^2} = 3.94 \quad \text{ve}$$

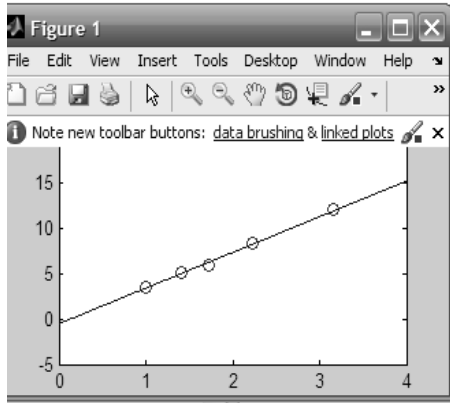
$$r = -0.517, \quad \text{Böylece}$$

$$\Rightarrow P = \frac{1}{3.94t^{1/2} - 0.517} \quad \text{bulunur.}$$

Bu problemin MATLAB ile çözümü ve ilgili grafik (Şekil 3.16) aşağıda verilmiştir.

Kod: 3.6

```
y=[3.584 5.155 5.952 8.333 12.048];
x=[1.0 1.414 1.732 2.236 3.162];
P=polyfit(x,y,1)
xx=0:.01:4;
yy=polyval(P,xx);
plot(x,y,'or',xx,yy)
```



Şekil 3.16 MATLAB ile en küçük kareler yöntemi kullanarak grafik çizimi

3.3.2 Regresyon Katsayısı

Eğri uydurmada kullanılacak olan polinomun verilere uygunluk derecesi r ile tanımlanan regresyon katsayısı ile ortaya konur. Elde edilen regresyon katsayısı $0 < r \leq 1$ aralığında bir değerdir. Eğer bu değer $r \approx 0$ ise uydurulan fonksiyonun iyi olmadığı, $r \approx 1$ ise de uydurulan fonksiyonun çok iyi olduğu anlaşılır. Regresyon katsayısının hesabı için öncelikle ölçüm sonucundan elde edilen sayısal değerlerin aritmetik ortalaması bulunur.

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

Sonra ölçüm değerleri ve uydurulan fonksiyona ait hata hesabı için aşağıdaki işlemler gerçekleştirilir.

$$\delta_y = \sum_{i=1}^n [y_i - \bar{y}]^2$$

ve

$$\delta_f = \sum_{i=1}^n [f(x_i) - \bar{y}]^2$$

Elde edilen bu değerlere göre regresyon katsayısının karşılığı,

$$r = \frac{\delta_f}{\delta_y} \quad \text{şeklinde bulunur.}$$

Özet

- Lineer olmayan bir fonksiyonu lineer halde çizmek, bilinmeyen parametreleri (**a** ve **b**) tayin etmeyi kolaylaştırmaktadır.
- İki büyüklük arasındaki ilişki **(miktar 1)=a₁(miktar 2)+a₀** şeklinde ise o zaman **(miktar 1)** in **(miktar 2)** ye karşı çiziminde bir doğru elde ederiz..
- Mühendislikte sık sık karşımıza çıkan lineer olmayan iki tip fonksiyon vardır. Bunlar **üstel** (*exponential*, $y=ae^{bx}$ gibi) ve **kuvvet** (*power-law*, $y=ax^b$ gibi) fonksiyonlarıdır.
- Grafik çizimlerini kolaylaştırmak için semi-log ve log-log ölçekleri (skalaları) türetilmiştir.
- $y=ae^{bx} \Rightarrow \log y = \log a + bx$ ifadesinde **y-x** çizimi semi-log grafiğinde, $y=ax^b \Rightarrow \log y = \log a + b \log x$ ifadesinde de **x-y**, log-log grafiğine çizilir. Bu durumlarda daima lineer bir denklem elde edilir.
- Dağınık verilere denklem uydurmak için en küçük kareler yöntemi kullanılır. Bu, hem doğru, hem de eğriler için geçerlidir..
- Regresyon katsayısı **r** ile verilir. Burada

$$r = \frac{\delta_f}{\delta_y} \text{ 'dir. Ayrıca, } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

$$\delta_y = \sum_{i=1}^n [y_i - \bar{y}]^2 \text{ ve } \delta_f = \sum_{i=1}^n [f(x_i) - \bar{y}]^2 \text{ olarak tarif edilmiştir.}$$

PROBLEMLER

1. Aşağıda verilen fonksiyonları nasıl çizelim ki bir doğru elde edebilelim?

- $y^2 = \sqrt[3]{x^2} + 23$
- $3/y^2 = a(x^2 + 2)^2 + b$
- $y^3 = (a x^4 + b)$
- $\cos y = \frac{1}{(a x + b)^2}$
- $y = a e^{bx}$
- $y = a x^b$

2. Aşağıdaki bağıntıları bulunuz.

- y** ile **t** arasındaki bağıntı dikdörtgen (Kartezyen) koordinat sisteminde bir doğru veriyor
- y-t log-log** grafiğinde bir doğru veriyor.
- y-t, semi-log** grafiğinde bir doğru veriyor.

- d) $(y^2 + 4) - (t^3 - 2)$ arasındaki bağıntı **semi-log** grafiğinde doğru veriyor
- e) $1/y - (t + 5)$ arasında **log-log** grafiğinde bir doğru veriyor

3. Aşağıdakileri nasıl çizelim ki bir doğru elde edip "a" ve "b" yi bulalım?

- a) $y = a e^{b t^2}$
- b) $y = a t^b$
- c) $y^3 = e^{(a t^2 - b)}$
- d) $1/y = a (t^2 + 3)^{-b}$

BÖLÜM 4 - LİNEER OLMAYAN CEBİRSEL DENKLEMLERİN KÖKLERİNİN BULUNMASI

Amaç

Bu bölümün bitiminde,

- *Lineer olmayan cebirsel denklemlerin*
 - *Grafik yöntemle*
 - *Nümerik Yöntemle*
 - *Basit İterasyon*
 - *Newton-Raphson*
 - *Yarıya Bölme*
 - *Regula-Falsi*
 - *Secant*

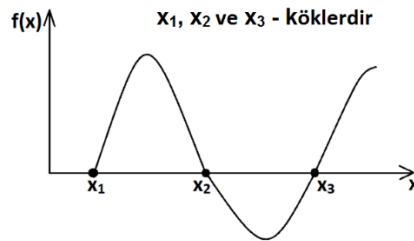
çözümlerini öğrenmiş olacağız

Eğer verilen denklem lineer ise (yani bilinmeyen sadece ve sadece 1 kuvvetine sahipse) böyle denklemlerin çözümü o kadar zor değildir. Örneğin, $5x+3=0$ veya $ax+by=4$ denklemleri lineerdir, çünkü değişkenler (x ve y) birinci derecedendir. Lineer olmayan denklemlerin kökleri ise biraz daha zor bulunur. Örneğin, $x^3 + x^{1/2} + e^{-x} = 0$ lineer olmayan bir denklemdir. Bunun köklerini, analitik yolla, direkt olarak bulmak mümkün değildir. Onun için aşağıda bahsedeceğimiz tekniklerden birisini kullanılarak böyle lineer olmayan denklemlerin kökleri bulunabilir. İki genel yöntem vardır: **grafiksel** ve **nümerik**

4.1 Grafik Yöntem

$f(x)=0$ şeklinde bir denklem verilmişse $f(x)$, x' e karşı çizilir. Burada x' e çeşitli değerler vermekle buna karşılık gelen $f(x)$ ' ler hesap edilir. Sonra da $f(x)$ -x eğrisi çizilir. Apsisi kesen noktalarda $f(x)=0$ olduğuna göre, x-eksenindeki kesim noktalarındaki " x_i " ler çözümü verecektir.

Şekil 4.1' de grafik yöntemle kök bulmanın mantığı gösterilmiştir.



Şekil 4.1 Grafik yöntemle kök bulmak

Yukarıdaki grafikte (Şekil 4.1) x_1 , x_2 ve x_3 , bu denklemin köklerini vermektedir. Eğer verilen denklem $f(x)=0$ şeklinde değilse, denklemi $f(x)=0$ şekline sokmak gerekir. Örneğin $x^2 + e^{-x} = \sin x$ şeklinde verilmiş ise bunu $f(x)=x^2 + e^{-x} - \sin x = 0$ şeklinde yazıp ondan sonra işleme başlamak gerekir.

Diyelim ki,

$$\left(P + \frac{a}{V^2}\right)(V - b) = RT$$

van der Waals denklemini verilmiş olsun. **Verilen V** (Hacim) ve **T** (Mutlak sıcaklık) değerleri için **P'** yi bulmak kolaydır (buradaki **a** ve **b** sabitleri gazın cinsine bağlı termodinamik sabitler olup bilinmektedir).

Fakat **verilen P ve T** değerleri için **V'** yi bulmak hiç de o kadar kolay değildir. Çünkü **V'** ye göre lineer olmayan kübik bir denklem elde edeceğiz.

O zaman

$$f(V) = \left(P + \frac{a}{V^2} \right) (V - b) - RT = 0$$

şeklinde yazılıp **f(V)** ' yi **V** ye karşı çizip **V'** nin çözüm için değerlerini bulabiliriz.

4.2 Nümerik Yöntemler

Kullanılacak yöntem

- a) Tek kök veya tüm köklerin bulunmasına
- b) Köklerin gerçek veya sanal olmasına
- c) Kökler için yaklaşık değer bulunup bulunmamasına

bağlıdır. Bazıları tek denklem, bazıları ise denklem sistemleri için kullanılır.

Aşağıda, nümerik olarak köklerin bulunmasına ait

- a) *Basit İterasyon Yöntemi (Ardışık Yerleştirme Yöntemi)*
- b) *Newton-Raphson Yöntemi*
- c) *Yarıya Bölme Yöntemi (Bisection Method)*
- d) *Regula -Falsi Yöntemi*
- e) *Secant Yöntemi*

yöntemleri örneklerle ele alınacaktır.

4.2.1 Basit İterasyon Yöntemi

Verilen fonksiyonu $x=g(x)$ şekiline dönüştürmek lazım. $f(x)=0$ şeklinde verilmişse bunu söylenen forma çevirmek gerekir.

Genel formül:

$$x_{k+1}=g(x_k) \quad \text{şeklindedir.}$$

$$g'(x), \quad g(x)' \text{in türevi olsun.}$$

$$|g'(x)| < 1 \text{ ise köke yaklaşım vardır.}$$

$$|g'(x)| \leq 1 \text{ ise köke yaklaşım vardır am yavaştır.}$$

$$|g'(x)| > 1 \text{ ise köke yaklaşım yoktur.}$$

Bu nedenle kök bulmaya kalkmadan önce $g'(x)$ hesap edilmelidir.

$$|\varepsilon_t| = \left| \frac{x_{k+1} - x_k}{x_{k+1}} \right| < \varepsilon_k$$

oluncaya kadar işleme devam edilir. Burada ε_k 'nın verilen toleransı ifade ettiğini hatırlayalım.

Örnek 4.1 Basit iterasyon yöntemiyle kök bulma

$x^3 - 2x^2 - 7 = 0$, denklemini, $x_0 = 2.5$ ve $\varepsilon_k = 0.01$ için iterasyon yöntemi ile çözünüz. (x_0 , ilk tahmini değeri gösteriyor)

Çözüm:

$$x = \underbrace{\left(\frac{x^3-7}{2}\right)^{1/2}}_{g(x)} \Rightarrow \underbrace{\frac{1}{2}\left(\frac{x^3-7}{2}\right)^{-1/2} \times 3x^2}_{g'(x)} \Rightarrow g'(2.5) = 1.5961 > 1.$$

Dolayısıyla yakınsama yoktur.

Öyle ise değişik açıdan çözüme yaklaşalım:

$x^3 = 2x^2 + 7$ olsun. O zaman

$$x = (2x^2 + 7)^{1/3} \Rightarrow \underbrace{\frac{1}{3}(2x^2 + 7)^{-2/3} \times 2x}_{g'(x)} \Rightarrow g'(2.5) = 0.23 < 1 \text{ yakınsama vardır.}$$

$x_1 = (2x_0^2 + 7)^{1/3} = 2.6916$. Aynı şekilde sırasıyla

$$\left. \begin{array}{l} x_2 = 2.7802 \\ x_3 = 2.8214 \end{array} \right\} \varepsilon_{t_{32}} = \frac{2.8214 - 2.7802}{2.8214} = 0.0146$$

$$x_4 = 2.8406 \Rightarrow \varepsilon_{t_{43}} = \frac{2.8406 - 2.8214}{2.8406} = 0.0068 < \varepsilon_k$$

Sonuç olarak kökün istenen civardaki değeri $x=2.8406$ olarak bulunmuş olur.

4.2.2 Newton-Raphson (N-R) Yöntemi

Bir fonksiyonun Taylor serisinde x_{k+1} civarında açılımı

$$f(x_{k+1}) = f(x_k) + f'(x_k)(x_{k+1} - x_k) + O(x_{k+1} - x_k)^2$$

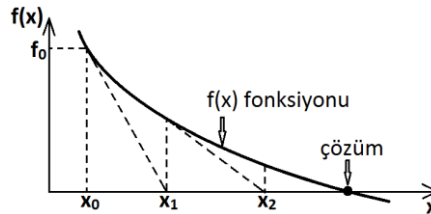
şeklindedir.

Son terimi ihmal edersek ve x_{k+1} kök ise;

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

şeklinde yazılabilir. Burada $f'(x_k)$, x_k 'daki türevi, x_{k+1} de bulunan yeni değerdir. Yöntemin grafiksel olarak mantığı aşağıdaki Şekil 4.2' de verilmiştir.

Burada x_0 ilk tahmin değeri olup f_0 , dan çizeceğimiz teğet x_1 ; x_1 den çizeceğimiz teğet x_2 verecek. Öyle ki eğrinin, yani $f(x)$ fonksiyonun, x eksenini kestiği yere kadar bu işleme devam edilir ki bu da bize istenen kökü verecektir.



Şekil 4.2 Newton-Raphson Yönteminin Grafiksel Olarak Mantığı

Örnek 4.2 N-R yöntemiyle kök bulma

$x = e^{-x}$ denkleminin, Newton-Raphson Yöntemi ile kökünü bulunuz.

Çözüm:

$$\begin{aligned} f(x) &= x - e^{-x} \\ f'(x) &= 1 + e^{-x} \end{aligned} \quad \text{şeklinde yazılır. Burada amaç } f(x)=0 \text{ veren } x' \text{ i bulmaktır.}$$

$x=0$ ise $f(x)<0$, $x=1$ ise $f(x)>0$.

Demek ki kök 0 ile 1 arasındadır.

$x_0=0.2$ diyelim

$$f(0.2) = 0.2 - e^{-0.2} = -0.6187$$

$$f' = 1 + e^{-0.2} = 1.8187$$

$$\Rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0.2 - \frac{-0.6187}{1.8187} = 0.5402$$

Böylece devam edersek

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 0.5670$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 0.5671$$

x_2 ile x_3 arasındaki değer son derece küçük olduğuna göre $\{|x_3 - x_2| \leq 0.0001\}$, x_3 'ün çözüm olduğunu kabul edebiliriz.

Aynı problemi MATLAB' in **solve** komutu ile çözelim:

Kod 4.1

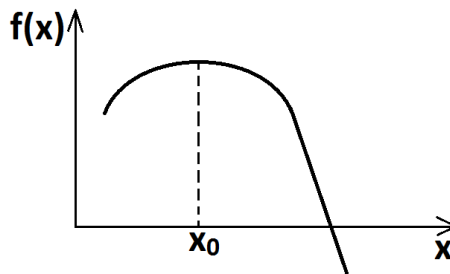
```
eq='x=exp(-x)'
```

```
solve(eq)
```

```
x=0.5671
```

Soru

Aşağıda verilen Şekil 4.3'ü göz önüne alalım



Şekil 4.3 Newton-Raphson yönteminin uygulanabilirliği

Eğer, ilk tahmini değer x_0 ' ı gösterilen yerden (*maksimum noktası*) alırsak, *Newton-Raphson Yöntemi* ile çözüme ulaşabilir miyiz [30]?

Örnek 4.3 Bir polinomun kökünün *N-R yöntemi*yle bulunması

$y = x^3 - 2x^2 - 7 = 0$, denkleminin $x_0 = 1.8$ civarındaki kökünü bulunuz. $\epsilon_k = 0.001$ olsun.

Çözüm:

$$x_{k+1} = x_k - \frac{y(x_k)}{y'(x_k)} \text{ ve } y' = 3x^2 - 4x \text{ 'dir.}$$

$$k = 0, \quad y(x_0) = -7.648, \quad y'(1.8) = 2.52, \quad x_1 = x_0 - \frac{y(x_0)}{y'(x_0)} = 4.84$$

Yeni bulunan x_1 değerinden x_2 , ondan da x_3 değeri bulunur ve böyle devam edilirse;

$$k = 1 \Rightarrow x_2 = 3.6708$$

$$k = 2 \Rightarrow x_3 = 3.0682$$

$$k = 3 \Rightarrow x_4 = 2.8768$$

$$k = 4 \Rightarrow x_5 = 2.8576$$

$$k = 5 \Rightarrow x_6 = 2.8574 \text{ } \} \text{ kök olarak kabul edilir, çünkü } \varepsilon_t = |-0.00006| < 0.001$$

4.2.3 Yarıya Bölme Yöntemi (Bisection Method)

a ve **b** iki reel sayı olsun. $f(a) \cdot f(b) < 0$ ise kök **[a,b]** aralığındadır.

$$x_0 = \frac{x_a + x_b}{2}, \quad f(x_0) \approx 0 \text{ ise kök } x_0 \text{ 'dır.}$$

$f(x_0) \neq 0$ ise işleme aşağıdaki gibi devam edilir.

$$\left. \begin{array}{ll} f(x_a) \cdot f(x_0) < 0, & x_a = x_0 \\ f(x_a) \cdot f(x_0) > 0, & x_b = x_0 \end{array} \right\} \begin{array}{l} f(x_0) \approx 0 \text{ olduğunda veya} \\ \varepsilon_t < \varepsilon_k \text{ ise işleme son verilir.} \end{array}$$

Örnek 4.4 Yarıya bölme yöntemiyle kök bulma

Aşağıdaki denklemi **yarıya bölme** yöntemi ile çözünüz.

$$x^3 - 2x^2 - 7 = 0, \quad x_a = 2, \quad x_b = 3 \text{ olsun.}$$

Çözüm:

$$f(a) \cdot f(b) = -14 < 0$$

$$x_0 = \frac{2+3}{2} = 2.5, \quad x_0 = x_a$$

$$f(2.5) \cdot f(3) = -7.7 \Rightarrow x_0 = \frac{2.5+3}{2} = 2.75, \quad x_0 = x_a$$

$$f(2.75) \cdot f(3) = -2.65 \Rightarrow x_0 = \frac{2.75+3}{2} = 2.875$$

$$f(2.875) \cdot f(3) = 0.465 \Rightarrow x_0 = \frac{2.875+3}{2} = 2.9375$$

$$f(2.875) \cdot f(2.9375) = 0.2532 \Rightarrow x_0 = \frac{2.875+2.938}{2} = 2.906$$

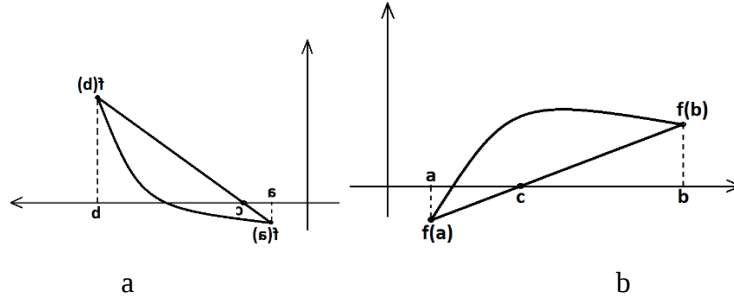
$$f(2.875) \cdot f(2.90625) = 0.132 \Rightarrow x_0 = \frac{2.875+2.90625}{2} = 2.891$$

$$\varepsilon_t = \frac{2.906-2.891}{2.906} = 0.005 < \varepsilon_k \text{ Böylece kök: 2.906'dır.}$$

MATLAB kullandığımızda ise (**solve** komu ile) $x = 2.857$ bulunmuştur.

4.2.4 Regula-Falsi, R-F, (Yer Değiştirme) Yöntemi

Bu yöntemin esası aşağıda Şekil 4.4a ve 4.4b de verilmiştir.



Şekil 4.4

$f(a) \cdot f(b) < 0$ ise kök $[a, b]$ aralığındadır. c , $[a, b]$ aralığında bir noktadır. Buna göre;

$f(a) \cdot f(c) < 0$ ise kök $[a, c]$ aralığında

$f(a) \cdot f(c) > 0$ ise kök $[b, c]$ aralığındadır.

$$c = \frac{bf(a) - af(b)}{f(a) - f(b)}$$

ifadesinden hesaplanan c için $f(c) \approx 0$ ise çözüm " c "dir veya $|\varepsilon_t| < \varepsilon_k$ ise işlem durdurulur.

$\varepsilon_t = \frac{c_{k+1} - c_k}{c_{k+1}}$ ile verilmiştir.

İşlem adımları şöyledir:

- 1) **a** ve **b** değerleri $f(a) \cdot f(b) < 0$ için seçilir.
- 2) **c** yukarıdaki denklemden hesaplanır.
- 3) $f(c) \approx 0$ ise işlem tamam, değilse $f(a) \cdot f(c) < 0$ ise $c=b$, $f(a) \cdot f(c) > 0$ ise $c=a$ olur.

Örnek 4.5 R-F yöntemiyle kök bulma

Daha önceki örneklerde verilen $x^3 - 2x^2 - 7 = 0$ denklemini bu yöntemle çözelim. **a=2** , **b=3** ve $\varepsilon_k = 0.01$ olsun.

Çözüm:

$$f(a) \cdot f(b) = f(2) \cdot f(3) = -14$$

$$c = \frac{bf(a) - af(b)}{f(a) - f(b)} = c = \frac{3f(2) - 2f(3)}{f(2) - f(3)} = 2.78$$

$$f(c) \cdot f(a) = f(2) \cdot f(2.78) = 6.8 \Rightarrow c_1 = b$$

$$c_2 = \frac{2.78f(2) - 2f(2.78)}{f(2) - f(2.78)} = 2.905$$

$$f(2) \cdot f(2.905) = -4.46 \Rightarrow c_2 = a$$

$$c_3 = \frac{2.905f(2) - 2f(2.905)}{f(2) - f(2.905)} = 2.295$$

$$f(2) \cdot f(2.295) = 2.5128 > 0 \Rightarrow a = 2.295 = c_3$$

$$c_4 = \frac{2.905f(2.295) - 2.295f(2.905)}{f(2.295) - f(2.905)} = 2.8411$$

$f(c_4) = -0.2108 \approx 0$. Böylece $c_4 = 2.8411$ kök olarak kabul edilebilir.

4.2.5 Secant Yöntemi

Newton-Raphson (N-R) yönteminde türev almak zor olabilir. Öyle durumlarda türev geriye doğru sonlu fonksiyonlar yöntemiyle hesaplanır.

N-R 'da $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ ifadesi

Secant'da $x_{k+1} = x_k - \frac{f(x_k)(x_{k-1} - x_k)}{f(x_{k-1})f(x_k)}$ ifadesi geçerlidir.

Örnek 4.6 Secant yöntemiyle kök bulma

$f(x) = e^{-x} - x = 0$ ise denklemin kökü nedir? $x_{k-1} = 0$, $x_k = 1$ olsun,

$f(1) = 0.6321$, $f(0) = 1$

$$x_{k+1} = 1 - \frac{-0.6321(0 - 1)}{1 - (-0.6321)} = 0.6127$$

$x - e^{-x} = 0$, $x_{k-1} = 0$, $x_k = 1$ (başlangıç), $\varepsilon_k = 0.01$

$$x_{k+1} = x_k - \frac{f(x_k)(x_{k-1} - x_k)}{f(x_{k-1})f(x_k)}$$

$x_2 = 0.5638$, $x_{k-1} = 1$, $x_k = 0.5638$

$x_3 = 0.5670$, $x_{k-1} = 0.5638$, $x_k = 0.5670$

$$x_4 = 0.5671 \Rightarrow \varepsilon_t = \frac{0.5671 - 0.5670}{0.5671} = 0.00017 < \varepsilon_k$$

Böylece kök $x_4 = 0.5671$ olarak alınabilir. Daha önce diğer yöntemlerle bulduğumuz değere yakındır.

Özet

Lineer olamayan, tek bilinmeyenli denklemlerin köklerini bulmak için iki genel yöntem vardır, *Grafiksel ve Nümerik*.

Grafiksel yöntemde fonksiyonun grafiği çizilir, apsisi kestiği noktalar köklerdir.

Nümerik yöntemler ise

- a) Basit İterasyon Yöntemi (Ardışık Yerleştirme Yöntemi)
- b) Newton-Raphson Yöntemi
- c) Yarıya Bölme Yöntemi (Bisection Method)
- d) Regula -Falsi Yöntemi
- e) Secant Yöntemi

diye sınıflara ayrılır. Bunlar içinde en çok kullanılanı *Newton-Raphson yöntemi*dir.

Ayrıca, MATLAB'in **solve** ve **fzero** komutlarını da kullanarak lineer olmayan karmaşık denklemlerin köklerini bulabiliriz.

PROBLEMLER

1) Cevaplayınız

- a) $x^2 - e^{-x} = 0$ çözümünü grafik yöntemle bulunuz.
- b) $\sin x + 5x = 0$ lineer mi, değil mi? Neden?
- c) $a \times y - y = 0$ lineer mi, değil mi? Neden?
- d) $5y + 8x = 14x + 3y$ lineer mi, değil mi? Neden?

2) Aşağıdaki denklemleri **[a,b]** aralığındaki kökünü

- a) Basit İterasyon
- b) Yarıya Bölme
- c) Regula-Falsi

yöntemleri ile bulunuz.

- I. $x^2 - 3\cos x + 4\tan x = 0$, $[0,1]$, $\varepsilon_k = 0.01$
- II. $x - 1 + e^{-2x} = 0$, $[0,1]$, $\varepsilon_k = 0.01$
- III. $x \ln x - 1 = 0$, $[0,2]$, $\varepsilon_k = 0.01$

3) Problem 2'deki kökleri

- a) fzero
- b) solve

komutları ile bulunuz.

4) $\frac{x}{4} - \ln(2x) = 0$ denkleminin $x_0 = 14$ ve $\varepsilon_k = 0.01$ için kökünü N-F yöntemiyle bulunuz.

5) Aşağıdakilerin köklerini **solve** komutu ile bulunuz.

- a) $x^5 + 2x^3 - \frac{1}{2}x^2 + 5x = 4$
- b) $x^3 - 3x^2 - 4x = 2$
- c) $x^5 - x^4 + x^3 = 17$

6) $\varepsilon_k = 0.01$ alınarak denklemlerin belirtilen noktalara (x_0) yakın köklerini bulunuz.

- a) $x^3 + 2x^2 - x = 0$, $x_0 = -2.6$
- b) $x^5 - 3x^2 + \exp(-x) = 0$, $x_0 = 0.5$
- c) $\exp(x) - x^4 - 4 = 0$, $x_0 = 2$

BÖLÜM 5 - LINEER DENKLEM SİSTEMLERİ

Amaç

Bu bölümün bitiminde,

- Lineer denklem sistemlerinin Gauss Eliminasyon yöntemiyle çözümü
- Lineer denklem sistemlerinin vektör-matris yöntemiyle MATLAB’de çözümü

konularını öğrenmiş olacağız

Giriş

Mühendislik ve diğer alanlarda karşınıza sıkça çıkan denklem sistemleri lineer olanlardır. Aşağıda genel olarak ifade edilen bir lineer denklem sistemi ilgili sembollerle gösterilmiştir. Lineer denklem sistemleri ile uğraşırken vektör ve matris operasyonlarını iyi bilmek gerekir.

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{array} \right\} \begin{array}{l} [A] = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \quad \Leftarrow \text{Katsayı matrisi} \\ \\ x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \end{array}$$
$$Ax = b \Rightarrow \boxed{x = A^{-1} \times b} \quad \text{çözüm vektörü}$$

5.1 Lineer Denklem Sistemlerinin Çözümü

Gauss Eleme Yöntemi

Denklemler üzerinde işlemler (*katsayılarla çarpma, diğer bir denkleme ilave etme, vb*) yapılarak bilinmeyenlerden biri elenir. Bu şekilde devam edilerek tek bilinmeyenli tek denklem kalıncaya kadar devam edilir. Elde edilen değer, geriye doğru yerine konularak diğer tüm bilinmeyenler bulunur.

Örneğin;

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{array} \right\} \text{Basit denklem sistemini bu yöntemle çözelim.}$$

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 = b_1 / -a_{21} \\ a_{21}x_1 + a_{22}x_2 = b_2 / a_{11} \\ \hline -a_{11}a_{21}x_1 - a_{12}a_{21}x_2 = -b_1a_{21} \\ +a_{11}a_{21}x_1 + a_{22}a_{11}x_2 = b_2a_{11} \\ \hline (a_{22}a_{11} - a_{12}a_{21})x_2 = a_{11}b_2 - b_1a_{21} \end{array} \right\} \begin{array}{l} \text{ilk denklem “} -a_{21} \text{”,} \\ \text{ikinci denklem “} -a_{11} \text{” ile çarpılır.} \\ \text{İki denklem toplanır ve } x_2 \text{ için çözülür. Sonra da } x_2 \\ \text{değeri yerine konularak } x_1 \text{ için aşağıdaki değer} \\ \text{bulunur.} \end{array}$$

$$x_2 = \frac{a_{11}b_2 - b_1a_{21}}{a_{22}a_{11} - a_{12}a_{21}}$$

$$x_1 = \frac{a_{22}b_2 - b_2a_{12}}{a_{11}a_{22} - a_{21}a_{12}}$$

Genel olarak, pivotleme yöntemi kullanırsak;

$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$ / (a_{21}/a_{11}) Pivot denklemi, a_{11} = Pivot katsayısı

:

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

$$\frac{a_{21}x_1 + \frac{a_{12}a_{21}}{a_{11}}x_2 + \dots + \frac{a_{1n}}{a_{11}}a_{21} = b_1 \frac{a_{21}}{a_{11}}}{(1)}$$

(1) denklem (a_{21}/a_{11}) ile çarpıldıktan sonra, (2) denklemden çıkarılırsa;

$$\underbrace{(a_{22} - \frac{a_{12}a_{21}}{a_{11}})}_{a'_{22}}x_2 + \dots + \underbrace{(a_{2n} - \frac{a_{21}}{a_{11}}a_{1n})}_{b'_2}x_n = b_2 - b_1 \frac{a_{21}}{a_{11}} \quad \text{elde edilir.}$$

$$a'_{22}x_2 + a'_{23}x_3 + \dots + a'_{2n}x_n = b'_2$$

Durum şöyle; eleme işlemi ile (n-1) denklem kalır. Bu şekilde tek denklem ve tek bilinmeyen kalıncaya kadar devam edilir. Aşağıdaki basamak sistemine dikkat edersek en sonunda tek bilinmeyen (yani x_n) ve tek denklem kalıyor.

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a'_{22}x_2 + a'_{23}x_3 + \dots + a'_{2n}x_n = b'_2$$

$$a'_{33}x_3 + a'_{34}x_4 + \dots + a'_{3n}x_n = b'_3$$

:

$$a'_{nn}x_n = b'_n \Rightarrow x_n = \frac{b'_n}{a'_{nn}}$$

Buradan da geriye doğru gidilerek x_{n-1} , x_{n-2} , ..., x_1 bulunur.

Genel olarak

$$x_i = \frac{b_i - \sum_{j=1 \rightarrow i}^n a'_{ij}x_j}{a'_{ii}}$$

şeklindedir.

Örnek 5.1 Lineer denklem sisteminin Gauss-Eleme yöntemiyle çözümü

Verilen

$$3x_1 = 14 - 2x_2$$

$$12x_1 + 13x_2 = +6x_3 = -40$$

$$8x_2 + 28 = 3x_1 - 9x_3$$

lineer denklem sistemini Gauss-Eleme yöntemiyle çözünüz.

Çözüm:

$x_1 \quad x_2 \quad x_3$ -1/4 ile çarpıp 1. satır ile topla
1 ile çarpıp 1. satır ile topla

$$3 \quad 2 \quad 0 \quad 14$$

$$12 \quad 13 \quad 6 \quad 40$$

$$\underline{-3 \quad 8 \quad 9 \quad -28}$$

$$3 \quad 2 \quad 0 \quad 14$$

$$0 \quad -5/4 \quad -3/2 \quad 4$$

$$0 \quad 10 \quad 9 \quad -14$$

5/40 ile çarpıp 2. satır ile topla

$$3 \quad 2 \quad 0 \quad 14$$

$$0 \quad -5/4 \quad -3/2 \quad 4$$

$$\underline{0 \quad 0 \quad -3/8 \quad 9/4}$$

$$\frac{-3}{8}x_3 = \frac{9}{4} \Rightarrow x_3 = -6$$

$$\frac{-5}{4}x_2 - \frac{3}{2}(-6) = 4 \Rightarrow x_2 = 4$$

$$3x_1 + 2 \times 4 + 0 = 14 \Rightarrow x_1 = 2$$

Örnek 5.2 Lineer denklem sisteminin MATLAB ile çözümü

Örnek 5.1' de verilen problemi şimdi de MATLAB ile çözünüz.

Çözüm:

$$\underbrace{\begin{bmatrix} 3 & 2 & 0 \\ 12 & 13 & 6 \\ -3 & 8 & 9 \end{bmatrix}}_A \underbrace{\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}}_x = \underbrace{\begin{Bmatrix} 14 \\ 40 \\ -28 \end{Bmatrix}}_b$$

$$\mathbf{x} = \mathbf{A}^{-1} \times \mathbf{b}$$

Buradan; $\mathbf{x} = \begin{Bmatrix} 2 \\ 4 \\ -6 \end{Bmatrix}$. Görüldüğü gibi MATLAB ile kolayca sonuca ulaşıldı.

Özet

Bu bölümde lineer denklem sistemleri tanıtıldıktan sonra bunların Gauss eleme yöntemiyle nasıl çözüldüğü örneklerle anlatılmıştır. Ayrıca MATLAB’de vektör-matris kavramı kullanılarak bu denklem sistemlerinin çok çabuk olarak nasıl çözülebileceği yine örneklerle gösterilmiştir.

PROBLEMLER

1) Aşağıdaki denklem sistemlerini

- a) Gauss-Eleme Yöntemiyle
b) MATLAB fonksiyonlarını kullanarak çözünüz

a) $x_1 + 4x_2 - x_3 + 4x_4 = 4$
 $x_1 - 3x_2 + x_3 - x_4 = 5$
 $3x_1 + 2x_2 + 4x_3 - x_4 = 7$
 $2x_1 - 2x_2 - x_3 + 4x_4 = 3$

b) $x_1 - 0.5x_2 + 0.7x_3 = 1$
 $0.7x_1 + x_2 + 0.3x_3 = 0.5$
 $-x_1 + 1.4x_2 + 1.5x_3 = 2$

c) $7x_1 + 3x_2 + x_3 = 21$
 $6x_2 + 3x_3 + 7x_4 = -3$
 $x_1 + 5x_2 + 4x_4 = 11$
 $3x_1 + 2x_2 + 4x_4 = 3$

d) $2.4x_1 + 0.5x_2 + 3.9x_3 + 0.7x_4 = 5.7$
 $x_1 + 1.2x_2 + 3x_3 + 4x_4 = -9.2$
 $2x_1 - 7.1x_2 + 4.1x_3 = 2$
 $4.2x_1 - 7.2x_2 + 1.2x_3 - 8.1 = 0$

2) Aşağıda katsayı matrisi ve sağ taraftaki vektörü verilen ifadelerin köklerini MATLAB ile bulunuz.

a) $A = \begin{bmatrix} 10 & 4 & 6 & 8 \\ 2 & 4 & 3 & 7 \\ 3 & 3 & 11 & 3 \\ 2 & 7 & 6 & 7 \end{bmatrix}$ $b = \begin{bmatrix} -2 \\ 10 \\ 8 \\ -7 \end{bmatrix}$

b) $A = \begin{bmatrix} 1.02 & -0.41 & -0.25 \\ -0.41 & 1.13 & -0.14 \\ -0.25 & -0.14 & 1.21 \end{bmatrix}$ $b = \begin{bmatrix} 0.61 \\ 2.55 \\ 1.78 \end{bmatrix}$

c) $A = \begin{bmatrix} 1.02 & 0.41 & 0.25 \\ 0.41 & 1.13 & 0.14 \\ 0.25 & 0.14 & 1.21 \end{bmatrix}$ $b = \begin{bmatrix} 0.51 \\ 1.55 \\ 2.78 \end{bmatrix}$

d) $A = \begin{bmatrix} 10.9 & 1.2 & 2.1 & 0.9 \\ 1.2 & 11.2 & 1.5 & 2.5 \\ 2.1 & 1.5 & 9.8 & 1.3 \\ 0.9 & 2.5 & 1.3 & 12.1 \end{bmatrix}$ $b = \begin{bmatrix} -7 \\ 5.3 \\ 10.3 \\ 24.6 \end{bmatrix}$

3) Aşağıdaki denklem sistemlerini MATLAB ile çözünüz.

a) $x_1 + x_2 + 2x_3 + 5x_4 = 2$

$$3x_1 + 4x_2 + 2x_3 + 4x_4 = -5$$

$$2x_1 + 9x_2 + 3x_3 + 2x_4 = 2$$

$$2x_1 + x_2 + 2x_3 - 3x_4 = -4$$

b) $6x_1 + 2x_2 + 2x_3 = -2$

$$2x_2 + \frac{2}{3}x_3 + \frac{1}{3}x_4 = -3.8$$

$$x_1 + 2x_2 - 2x_3 = 0$$

c) $x_1 + 4x_2 - x_3 + 4x_4 = 4$

$$1.5x_1 - x_2 - 4.5x_3 = 9.5$$

$$-2.5x_1 - 1.5x_2 + x_3 = -7$$

d) $2.5x_1 - 4x_2 - 2x_3 - 3x_4 = -20.29$

$$3x_1 - 3x_2 - 4.3x_3 + 3x_4 = 8.89$$

$$2x_1 - 4x_2 - 3.25x_3 - x_4 = 4.84$$

$$x_1 - 5x_2 + 3.3x_3 - 20x_4 = -14.01$$

e) $x_1 + 2x_2 - 12x_3 + 8x_4 = 27$

$$5x_1 + x_2 + 7x_3 - 2x_4 = 4$$

$$-3x_1 + 7x_2 + 9x_3 + 5x_4 = 11$$

$$6x_1 - 12x_2 - 8x_3 + 3x_4 = 49$$

f) $1.02x_1 - 0.25x_2 - 0.3x_3 = 0.515$

$$-0.25x_1 + 0.14x_2 + 1.21x_3 = 2.78$$

$$-0.41x_1 - 1.13x_2 - 0.15x_3 = 1.555$$

g) $4x_1 + 0.24x_2 - 0.08x_3 - 0.16x_4 = 8$

$$0.09x_1 + 3x_2 - 0.15x_3 - 12x_4 = 7$$

$$0.04x_1 - 0.08x_2 + 4x_3 + 0.06x_4 = 20$$

$$0.02x_1 + 0.06x_2 + 0.04x_3 - 10x_4 = 1$$

h) $-1.05x_1 + 3.96x_2 + 1.15x_3 = -2.71$

$$3.96x_1 - 1.01x_2 = 9.5$$

$$-3.09x_1 - 1.15x_2 + 3.96x_3 = 5.98$$

i) $x_1 + x_2 - 2x_3 = 1$

$$3x_1 + 2x_2 + 4x_3 = 2$$

$$3x_1 + x_2 + x_3 = 4$$

$$2x_1 - 3x_2 + x_3 = -1$$

$$-3x_1 + x_2 + x_5 = 0$$

j) $x_1 + x_2 - x_3 + x_4 - x_5 = 3$

$$2x_1 + x_3 - 2x_4 = 2$$

$$-3x_1 + x_2 + x_3 = -1$$

$$2x_1 + x_2 - x_3 + x_4 - x_5 = 0$$

k) $2x_1 - x_2 + 2x_3 + x_4 = 0$

$$x_1 + x_2 - x_3 + x_4 = 0$$

$$x_2 + 3x_3 - 5x_5 = 0$$

4) Verilen lineer denklem sistemini

$$10x_1 + 2x_2 - x_3 = 27$$

$$-3x_1 - 6x_2 + 2x_3 = -61.5$$

$$x_1 + x_2 + 5x_3 = -21.5$$

a) Gauss eleme yöntemi ile

b) MATLAB komutları ile (**inv** fonksiyonu kullanarak) çözünüz.

c) Bulduğunuz değerleri denklemlerde yerine koyup sonuçları kontrol ediniz.

5) Verilen denklem sistemi için

$$-3x_2 + 7x_3 = 2$$

$$x_1 + 2x_2 - x_3 = 3$$

$$5x_1 - 2x_2 = 2$$

- a) MATLAB ile Determinantı hesaplayınız.
- b) Gauss-eleme yöntemi ile sistemi çözünüz.
- c) Bulduğunuz değerleri denklemlerde yerine koyup sonuçları kontrol ediniz.

6) Verilen

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85 \quad (1)$$

$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3 \quad (2)$$

$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4 \quad (3)$$

Denklem sistemini *Gauss-Eleme Yöntemi* ile çözünüz.

BÖLÜM 6 - NÜMERİK TÜREV VE NÜMERİK İNTEGRASYON

Amaç

Bu bölümün bitiminde,

- Nümerik olarak türev almayı
- Nümerik olarak integral almayı
 - Yamuklar yöntemiyle
 - Simpson parabolik yöntemiyle

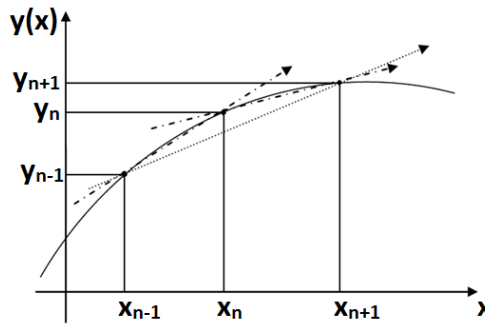
uygulamalı olarak öğrenmiş olacağız

6.1 Nümerik Türev

Türev, bir fonksiyonun, bağlı olduğu değişkenlere göre *değişim hızının* bir ölçüsüdür. Bir $y(x)$ fonksiyonunun $x=a$ noktasındaki türevi sayısal olarak aşağıdaki şekillerde tanımlanabilir [20].

- İleri Farklar : $y'(x_n) = \frac{y(x_{n+1}) - y(x_n)}{x_{n+1} - x_n}$
- Geri Farklar : $y'(x_n) = \frac{y(x_n) - y(x_{n-1})}{x_n - x_{n-1}}$
- Merkezi Farklar : $y'(x_n) = \frac{y(x_{n+1}) - y(x_{n-1}))}{x_{n+1} - x_{n-1}}$

Dikkat edilecek olursa türev, alınacak noktaya ek olarak ileri farklar yöntemi bir adım *sonrasını*, geri farklar yöntemi bir adım *öncesini* ve merkezi farklar yöntemi de *her ikisine* ait bilgiyi gerektirir. Bundan dolayı, söz konusu yöntemler kullanılarak elde edilen sonuçlar arasında farklılıklar görülmesi muhtemeldir.



Şekil 6.1 İleri, geri ve merkezi farklar yöntemleri ile nümerik türev alma mantığı

Örneğin, Şekil 6.1’de, x_n noktasına ait türevin değeri, oklar ile gösterildiği gibi elde edilen türev değerleri farklıdır. İleri veya geri farklar yöntemi ile hesaplanan sayısal türevin değeri, analitik değerinden farklılık gösterdiğini ve yöne göre değiştiğini kaydedelim. Merkezi farklar yöntemi ise ortalama bir sonuç üretir. Buna göre, $y(x) = x^2 + 9$ fonksiyonunun $x=1.5$ noktasındaki sayısal türevini üç yöntemle bulunuz ve analitik sonuç ile karşılaştırıp irdeleyiniz.

6.2 Nümerik İntegrasyon

Birçok mühendislik problemlerinde öyle karmaşık ifadeler elde edilir ki bunların integrasyonu *analitik* olarak hemen hemen imkânsızdır. Örneğin, reaktör tasarımında (ideal piston akışlı reaktör, PAR için);

$$V = F_{A_0} \int_0^x \frac{dx}{-r_A},$$

Burada,

V = reaktör hacmi

F_{A_0} = A nın molar akış debisi

x = dönüşüm (<1)

$-r_A$ = reaksiyon hızı

Reaksiyonun hız ifadesi ($-r_A$) basit olabildiği gibi çok karmaşık da olabilir.

Diyelim ki

$$-r_A = k C_{A_0}^2 \left(\frac{(1-x)^2}{(1+\epsilon x)^2} \right)$$

olsun. Burada k , C_{A_0} ve ϵ birer sabittir. O zaman yukarıdaki denklemden reaktör hacmini bulmak için

$$V = F_{A_0} \int_0^x \frac{(1+\epsilon x)^2}{(1-x)^2} dx$$

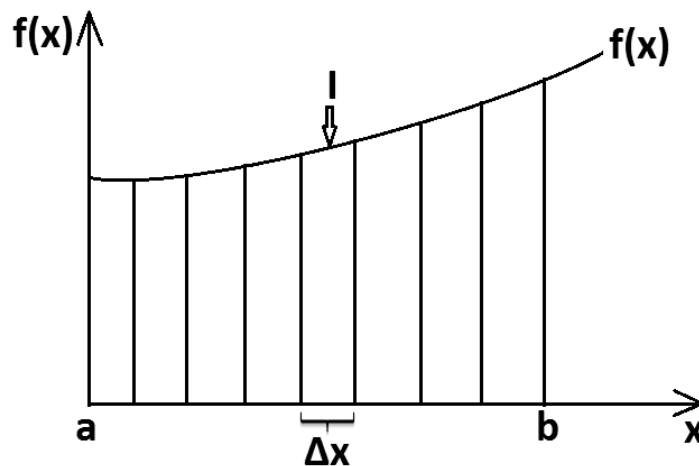
olur. Bunun gibi karmaşık ifadeleri ancak nümerik olarak integre edebiliriz.

Genel olarak

$$I = \int_a^b f(x) dx$$

şeklinde bir integral düşünelim. Diyelim ki $f(x)$ çok karmaşık bir ifade ve bunu *analitik* olarak integre edemiyoruz. O zaman nümerik integrasyon yöntemlerine başvurmamız gerekecektir.

İntegrasyon, toplamadan başka bir şey değildir. Şekil 6.2'yi göz önüne alalım:



Şekil 6.2 İntegrasyon

Şekil 6.2 deki integrasyonun değeri, **a** ve **b** ile sınırlanmış eğrinin altındaki alandır.

Küçük aralıklara Δx_i dersek eğrinin altındaki alan

$$\sum_{i=1}^{\infty} f_i(x) \Delta x_i = \int_a^b f(x) dx \text{ olur.}$$

Nümerik integrasyon için çeşitli yöntemler mevcuttur [22]. Biz bunlardan en yaygın kullanılanları göreceğiz.

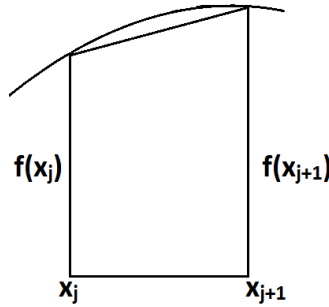
6.2.1 Yamuklar Yöntemi (Trapezoids Rule)

İntegral, genel itibariyle, fonksiyonu oluşturan eğrinin altındaki alan olduğunu belirtmiştik. Yamuklar yönteminde, eğrinin altında kalan alanı hesaplamak için, alan belli aralıklara bölünür. Aşağıda Şekil 6.3’de görüldüğü gibi, her aralık için bir yamuk oluşturulur fakat oluşturulan her yamuk için pozitif veya negatif *hesaplanmayan* alan kalır. Oluşturulan bu yamuğun alanı:

$$A_j = \frac{1}{2} (f(x_j) + f(x_{j+1})) (x_{j+1} - x_j)$$

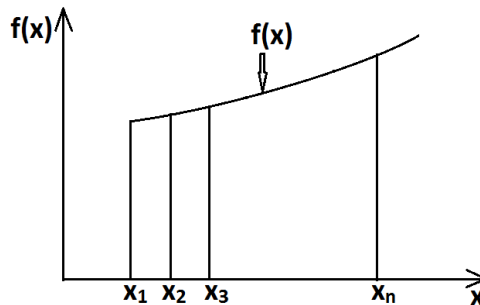
$$A = \sum_{j=1}^n A_j$$

Burada dikkat edilmesi gereken nokta, toplam sembolündeki ‘**n**’ ifadesi aralık sayısını göstermektedir. Unutulmamalı ki bir fonksiyonu ‘**n**’ parçaya ayırmak için ‘**n+1**’ tane noktanın seçilmesi gerekir.



Şekil 6.3 Yamuk kuralı

Eğri altındaki alan küçük yamuklara bölünerek Şekil 6.4’te görüldüğü x_1 den x_n ’e kadar integre edilir.



Şekil 6.4

Şekildeki alan çok sayıda küçük yamuklara bölünerek bu yamukların toplan alanı aşağıdaki ifadeyi verir.

$$I = \int_{x_1}^{x_n} f(x)dx = \frac{1}{2} \left[(x_2 - x_1)(f_1 + f_2) + (x_3 - x_2)(f_2 + f_3) + \dots + (x_{i+1} - x_i)(f_i + f_{i+1}) + \dots + (x_n - x_{n-1})(f_{n-1} + f_n) \right]$$

Basit bir kısaltmadan sonra

$$I = \frac{1}{2} \left[\sum_{i=1}^{n-1} (x_{i+1} - x_i)(f_{i+1} + f_i) \right]$$

Eğer $h = x_{i+1} - x_i = \text{sabit}$ ise

$$I = \frac{h}{2} \left[f_1 + f_n + 2 \sum_{j=2}^{n-1} f_j \right] \quad \text{olur.}$$

Eğer veri noktaları, yani (x_i, y_i) , verilmişse, sadece o veri noktalarını yerine koymakla integrasyonu hesaplayabiliriz, *ayrıca bir grafik çizmeye gerek yoktur.* (İleride bununla ilgili örnek verilecektir).

Örnek 6.1 Yamuklar yöntemiyle integrasyon hesaplama

$$f(x) = x^3 + 2$$

fonksiyonunun $[2,5]$ aralığındaki alanını hesaplayınız.

Çözüm:

1. Öncelikle alanı kaç parçaya ayıracağımıza ya da aralıklar arası mesafeye karar vermemiz gerekir.

Seçeceğimiz yol, oluşturacağımız MATLAB fonksiyonunu direkt olarak etkileyecektir. Bu örnek için alanı $n=1, 10$ ve 100 parçaya böldüğümüzü düşünüp hepsi için ayrı ayrı sonucu hesaplayalım ve sonuçları karşılaştıralım.

2. Alan hesabı için gerekli MATLAB algoritmasının oluşturulması.

- I. Verilen n sayısı için aralıklar arası mesafenin hesaplanması ($\Delta x = \frac{5-2}{n}$)
- II. Girilen n sayısı için $n+1$ tane x koordinatlarının bulunması
- III. Bulunan her x koordinatı için f fonksiyonun değerinin bulunması
- IV. $i=1,2,\dots,n$ için A_i alanlarının hesaplanması
- V. Toplam alanın hesaplanması $A=A_1+ A_2+\dots+ A_n$

3. Oluşturulan algoritmaya göre MATLAB ortamında fonksiyonun yazılması .

Aşağıda $n=1$ için verilen **Kod 6.1** de sırasıyla:

1. $n=1$ tanımlamasından sonra $n=1$ için dx : aralıklar arası uzaklık bulunuyor.
2. **Kolon** operatörü kullanılarak, $[2,5]$ aralığında x değerleri dx aralıklarla hesaplanıyor.
3. Her x değeri için karşılık gelen $f(x)$ değerleri hesaplanıyor.
4. **for** döngüsü kullanılarak her bir parça için alan hesaplanıyor.
5. Son olarak bulunan bu değerler **sum** komutu ile toplanıyor.

$n=1$ işlemlerinde x , f ve A matrislerinin boyutları küçük olduğu için işlemler sonucunda değerleri gösterilmiştir. Diğer n değerleri için sadece toplam alan gösterilecektir.

Aşağıda $n=10$ ve $n=100$ bulunan sonuçlar verilmiştir.

$n=10 \rightarrow A=158.7225$

$n=100 \rightarrow A=158.2547$

Sonuçlar incelendiğine, sonuçların $n=1$ den çok farklı olduğu görülecektir. Bunun nedeni, $n=1$ için oluşturulan yamuk kübik bir fonksiyon için çok fazla hata içermektedir. Fakat $n=10$ ve $n=100$ sonuçları birbirine yakındır. n sayısı arttıkça sonuç belli bir sayıya yaklaşacaktır. Bu sayı, bu problemin analitik sonucudur. Burada karar verilmesi gereken, sonuca ne kadar yakın olmak istendiğidir. n sayısı arttıkça sonuçtaki hata azalır fakat problemin çözüm süresi de artar. Üstelik n sayısı arttıkça sonuçlar arası fark da gittikçe azalır.

Karşılaştırmak için aynı problemin, daha büyük n değerleri için bulunan nümerik sonuçları ile analitik sonuçları aşağıda verilmiştir.

$n=1000$	$A=158.2502$
$n=100000$	$A=158.2500$
analitik sonuç	$A=158.2500$

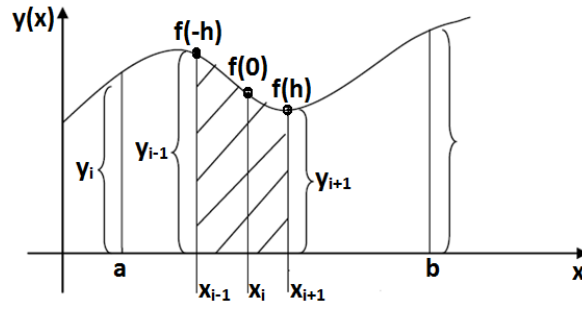
```
>> n=1;
>> dx=(5-2)/n;
>> x=2:dx:5
x =
    2    5
>> f=x.^2+2;
>> for i=1:n
    A(i)=1/2*(f(i)+f(i+1))*(x(i+1)-x(i));
end
>> sum(A)
A =
    205.5000
```

Kod 6.1

6.2.2 Simpson Parabolik Yöntemi

Simpson yöntemi ile herhangi bir fonksiyona ait alan hesabı gerçekleştirilirken a ve b olarak sınırları belirlenen x eksenindeki aralığın h olarak adlandırılan eşit uzaklıkta fakat çift sayıdan oluşan n adet parçalara bölünmesi gerekir. Yamuk yönteminde olduğu gibi bu parçaların adedi ne kadar fazla olursa elde edilecek olan alan hesabının doğruluğu da o kadar yüksek olur. Ayrıca vurgulamak gerekirse Simpson yöntemi işlemler sonucunda elde edilen değerin doğruluğu bakımından Yamuk yöntemine göre daha iyi sonuç verir.

Aşağıdaki Şekil 6.5'i dikkate alarak [20] bu yöntemle ilgili son uygulama denklemini türetelim.



Şekil 6.5 Simpson Parabolik Yönteminin grafiksel gösterimi

x_{i-1} ile x_{i+1} arasında kalan eğri $y_{i-1} = f(-h)$, $y_i = f(0)$ ve $y_{i+1} = f(h)$ noktalarından geçen $f(x) = ax^2 + bx + c$ doğrusu ile tanımlanır. Burada bilinmeyen a , b ve c katsayıları aşağıdaki eşitliklerden bulunabilir.

$$y_{n-1} = f(-h) = ah^2 - bh + c$$

$$y_n = f(0) = c$$

$$y_{n+1} = f(h) = ah^2 + bh + c$$

Buradan,

$$c = y_n, \quad b = \frac{y_{n+1} - y_{n-1}}{2h} \quad \text{ve} \quad a = \frac{1}{2h^2} (y_{n+1} + y_{n-1} - 2y_n)$$

olarak bulunur. $y(x)$ fonksiyonunun x_{i-1} ve x_{i+1} arasındaki değeri,

$$\int_{x_{i-1}}^{x_{i+1}} y(x) dx = \int_{-h}^h f(x) dx$$

şeklinde tanımlanabildiğine göre,

$$\begin{aligned} \int_{-h}^h (ax^2 + bx + c) dx &= \left[a \frac{x^3}{3} + b \frac{x^2}{2} + cx \right]_{-h}^h \\ &= \left[a \frac{h^3}{3} + b \frac{h^2}{2} + ch \right] - \left[-a \frac{h^3}{3} + b \frac{h^2}{2} - ch \right] \\ &= 2a \frac{h^3}{3} + 2ch \end{aligned}$$

Katsayıların değerleri yerine konulduğunda integrale ait aşağıdaki ifade elde edilir.

$$\int_{-h}^h (ax^2 + bx + c) dx = 2 \frac{1}{2h^2} (y_{n+1} + y_{n-1} - 2y_n) \frac{h^3}{3} + 2y_n h = \frac{h}{3} (y_{n-1} + 4y_n + y_{n+1})$$

Bir dizinin integrali alınırken bu yönteme ait kullanılabilecek olan formül yapısı aşağıdaki gibi tanımlanabilir.

$$\int_a^b y(x) dx = \sum_{i=1}^{(m-1)/2} \int_{x_{2i-1}}^{x_{2i+1}} f(x) dx$$

Formül, ortak olan değere göre tekrar düzenlenebilir.

$$\int_a^b y(x)dx = \frac{h}{3}(y_1 + 4(y_2 + y_4 + \dots + y_{1m-1}) + 2(y_3 + y_5 + \dots + y_{m-2}) + y_m)$$

Dolayısıyla Genel Denklem

$$Alan = \frac{h}{3}(y_0 + y_n + \underbrace{4 \sum_{i=1,3,5,\dots}^{n-1} y_i}_{\text{tekler}} + \underbrace{2 \sum_{i=2,4,\dots}^{m-1} y_i}_{\text{çiftler}})$$

şeklinde olur.

Böylece integral işlemi $y_1 = y(a)$ ve $y_m = y(b)$ değerleri ile birlikte m adet nokta kullanılarak gerçekleştirilmiş olur. Bu yöntemin örnek bir uygulamasını Yamuk yönteminde kullanılan fonksiyon için gerçekleştirelim:

$y = e^{-x^2}$ fonksiyonunu göz önüne alalım

Verilen fonksiyona ait sınır değerleri yine Yamuk yönteminde kullanıldığı gibi $a = 0$, $b = 1$ ve parça aralığı da $h = 0.25$ olarak tanımlandığında Simpson yöntemine ait kullanılacak olan formül aşağıdaki gibi olacaktır.

$$Alan \cong \frac{h}{3}[y_1 + 4(y_2 + y_4) + 2(y_3) + y_5]$$

Buna göre fonksiyona ait değer hesaplamaları Simpson yöntemine ait formülde yerlerine yazılırsa;

$$Alan \cong \frac{0.25}{3}[1 + 4(0.9394 + 0.5698) + 2(0.7788) + 0.3679] = 0.7469 \text{ birim kare}$$

şeklinde verilen sınırlar arasındaki fonksiyona ait alan bulunmuş olur. Bu işlemlerin MATLAB'de gerçekleştirilişine ait örnek uygulama aşağıda görülmektedir.

Aynı fonksiyona ait integrali analitik olarak hesaplasaydık sonuç **0.7468** birim kare olacaktı.

Örnek 6.2 Simpson yöntemiyle nümerik integrasyonu bulma

$\int_0^1 \frac{\sin x}{\sqrt{x^3+1}} dx$, $n = 4$ ve $h = \frac{1-0}{4} = 0.25$ için integrali hesaplayınız.

Çözüm:

i	x_i	$f(x_i) = y_i$	
0	0	0	İlk
1	0.25	0.2455	Tek
2	0.50	0.4520	Çift
3	0.75	0.5717	Tek
4	1	0.5950	Son

$$A = \frac{0.25}{3}[0 + 0.5950 + 4(0.2455 + 0.5717) + 2(0.4520)] = 0.3973$$

Örnek 6.3 Yamuk ve Simpson yöntemlerinin beraber kullanımı

Sıcaklıkla (T), değişen molar ısı kapasiteleri (C_p) aşağıdaki şekilde ölçülmüştür [30].

T(°C)	20	50	80	110	140	170	200	230
Cp(J/mol °C)	28.95	29.13	29.30	29.48	29.65	29.82	29.99	30.16

3 mol gazı 20 °C den 230 °C ye kadar ısıtmak için gerekli entalpiyi (ΔH) bulunuz.

Çözüm:

$$\Delta H = \int_{20}^{230} NC_p dT \quad \text{dir.}$$

burada N=gazın mol sayısı

n=çift olduğu için 1 ve 2 noktaları arasında *yamuk kuralını*, 2 den 8 e kadar da *Simpson kuralını* kullanalım. Böyle bir uygulama şart değildir, fakat her iki metodu da aynı problemde kullanmak için bu yol seçilmiştir.

$$\Delta T = T_i - T_{i-1} = 30 \text{ } ^\circ\text{C}$$

$$\Delta H = \left[\underbrace{\frac{h}{2} (C_{p_1} + C_{p_2})}_{\text{Yamuk}} + \underbrace{\frac{h}{3} \left(C_{p_2} + C_{p_8} + 4 \sum_{j=3,5} C_{p_j} + 2 \sum_{i=4,6} C_{p_i} \right)}_{\text{Simpson}} \right] N$$

Yukarıda verilen değerleri yerine koyarsak

$$\Delta H = 18620 \text{ J}$$

bulunur.

Özet

Bu bölümde, özellikle laboratuvarlarda elde edilen ve oluşturulan tablolarda (fonksiyon olmadan) nümerik olarak nasıl türev ve integral alacağını gözden geçirdik. Nümerik integrasyonda önemli iki yöntemi ele aldık. Bunlar, *Yamuk* ve *Simpson* yöntemleri idi.

PROBLEMLER

- 1) $\int_0^6 \frac{x}{1+x} dx$ integralini yamuk yöntemi ile bulunuz (n=12).
- 2) $\int_0^4 \frac{x^2}{1+x} dx$ integralini yamuk yöntemi ile bulunuz (n=15).
- 3) $\int_1^4 \frac{\ln x}{1+x^2} dx$ integralini Simpson yöntemi ile bulunuz (n=10).
- 4) $\int_0^7 (1+x^3)^{3/2} dx$ integralini Simpson yöntemi ile bulunuz (n=10).
- 5) $\int_0^\pi \frac{1}{3+\sin x} dx$ integralini Simpson yöntemi ile bulunuz (n=7).
- 6) $\int_2^4 x^3 \log(x^2) dx$ integralini Simpson yöntemi ile bulunuz (n=5).
- 7) Yukarıda verilen problemlerin hepsini (Örnek 6.3 dahil) MATLAB ile çözüp sonuçları karşılaştırınız.

BÖLÜM 7 - DİFERANSİYEL DENKLEMLERİN ÇÖZÜMLERİ

Amaç

Bu bölümün bitiminde,

- Diferansiyel denklemlerin çözümünü
 - Euler yöntemiyle,
 - Runga-Kutta yöntemiyle
- Diferansiyel denklemlerin MATLAB'deki **dsolve** komutu ile analitik çözümlerini
- Başlangıç-değer ve sınır-değer problemlerinin çözümlerini

uygulamalı olarak öğrenmiş olacağız

7.1 Nümerik Çözümler

Diferansiyel denklemlerin genel olarak analitik çözümleri ya yoktur ya da çözümleri çok zahmetlidir. Özellikle, lineer olmayan diferansiyel denklemlerin kapalı çözümlerini bulmak hemen hemen imkânsızdır. Böyle durumlarda diferansiyel denklemlerin nümerik çözümlerine başvurulur. Aşağıda klasik iki nümerik yöntemi göreceğiz, **Euler** ve **Runga-Kutta**. Daha sonra da ilgili diferansiyel denklemlerin MATLAB ile nümerik ve hatta analitik çözümlerini göreceğiz. Diferansiyel denklemlerin genel çözümleri yerine özel çözümlerini bulmak istersek o zaman *başlangıç* ve/veya *sınır şartlarını* kullanmamız gerekecektir. Aşağıdaki örneklerde bu şartları kullanacağız.

7.1.1 Euler Tek Adım Yöntemi

$$\frac{dy}{dx} = f(x, y)$$

verilmiş olsun. x_0 ve y_0 başlangıç değerleri olup bilinmektedir.

$$\begin{aligned} y_1 &= y_0 + \Delta x f(x_0, y_0) & \text{burada, } x_1 &= \Delta x \text{ alırsak (ki } \Delta x \text{ küçük bir adım)} \\ y_2 &= y_1 + \Delta x f(x_1, y_1) & x_2 &= 2\Delta x \\ y_3 &= y_2 + \Delta x f(x_2, y_2) & x_3 &= 3\Delta x \\ &\vdots & & \\ y_{i+1} &= y_i + \Delta x f(x_i, y_i) & x_{i+1} &= (i+1)\Delta x \end{aligned}$$

Örnek 7.1 Euler yönteminin bir tanktaki konsantrasyonu (derişimi) bulmada kullanılması

Bir tanktaki konsantrasyon (x) zamana göre aşağıdaki şekilde değişmektedir.

$$\tau \frac{dx}{dt} = 1 - x$$

Burada τ = tankın zaman sabitesi olup 1 olarak kabul edilmiştir.

Çözüm:

$$\tau \frac{dx}{dt} = 1 - x$$

$t = 0$ iken $x_0 = 0$ ve $\Delta t = 0.05$ olsun

$$\frac{dx}{dt} = 1 - x$$

$$x_1 = x_0 + \Delta t f(x_0, t_0) \quad t_1 = \Delta t$$

$$x_2 = x_1 + \Delta t f(x_1, t_1) \quad t_2 = 2\Delta t$$

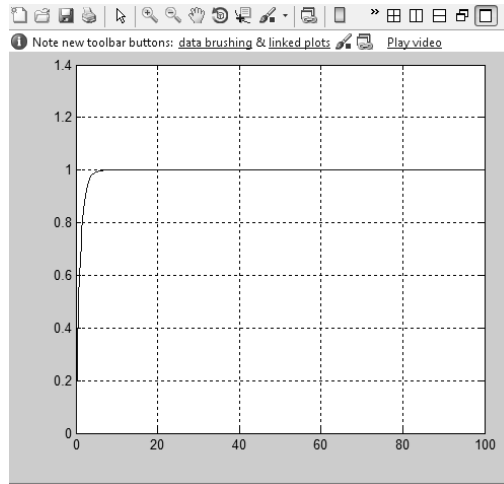
$$x_3 = x_2 + \Delta t f(x_2, t_2) \quad t_3 = 3\Delta t$$

Buna göre $t=0.1, 0.2, 0.4$ ve 0.8 saatlerdeki konsantrasyonları bulunuz.

Kod 7.1

MATLAB programı ile çözüm:

```
function dydt=Ornek71(t,y)
tspan=[0 100];
y0=0
dydt=1-y
[t,y]=ode45('Ornek71',tspan,y0)
plot(t,y(:,1))
```



Şekil 7.1 Örnek 7.1'in çözümü

(Cevap: MATLAB ile çözümünde sırasıyla $y=0.098$, $y=0.185$, $y=0.337$ ve $y=0.560$)

Aynı problemin analitik çözümü ile (ki kolayca elde edilebilir) $t=0.8$ için $x=0.551$ bulunur ki bu da Euler yönteminin iyi bir yaklaşım yaptığını göstermektedir.

Aynı problemin analitik çözümü de aşağıdaki kodla bulunabilir:

Kod 7.2

```
function [t,y]=Ornek72_1(t,y)
y=dsolve('Dy=1-y, y(0)=0')
y = 1-exp(-t)
```

$$y = \text{inline}(1 - \exp(t))$$

$$y(t) = 1 - \exp(-t)$$

$$y(0.1) = 0.0952$$

$$y(0.2) = 0.1813$$

$$y(0.4) = 0.3297$$

$$y(0.8) = 0.5507$$

değerleri bulunmuş olur. Böylece bu karşılaştırma ile Euler yönteminin ne kadar geçerli olduğu anlaşılmış oluyor.

7.1.2 Runga-Kutta (R-K) Yöntemi

Bu yöntem, özellikle *kimyasal reaksiyon mühendisliğinde, proses kontrolde, ısı ve kütle transferinde* sıkça kullanılmakta olup oldukça kuvvetli bir yöntemdir. Birinci mertebeden

$$\frac{dy}{dt} = f(t, y)$$

diferansiyel denklemi verilmiş olsun.

Adım adım çözüm:

$$y_{n+1} = y_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

olur.

Burada,

$$k_1 = hf(t_n, y_n)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(t_n + h, y_n + k_3)$$

olarak verilmiştir

Buna 4. mertebeden Runga-Kutta Yöntemi denir.

Örnek 7.2 4. Mertebeden R-K yönteminin diferansiyel denklem çözümünde kullanımı

Aşağıda verilen diferansiyel denklemi 4. mertebeden R-K ile çözünüz.

$$\frac{dy}{dx} = y + x^2 - 1 \quad y(0) = 0$$

Çözüm:

Problemin MATLAB çözümü (Kod 7.3) aşağıda verilmiştir.

1. Fonksiyon ve ilk y değeri tanımlanıyor.
2. **for** döngüsü ile bir sonraki aşamada kullanılacak y değeri hesaplanıyor.

3. Döngü içinde yukarıda verilen denklemlere göre k_1 , k_2 , k_3 , k_4 ve bir sonraki adım için yeni y değeri hesaplanıyor.

Bulunan sonuç analitik sonuca ($y=21905$) yakındır.

```
% Runga-Kutta
f=inline('y+x^2-1')
y=0;
h=.1;
for x=0:0.1:10
    k1=f(x,y);
    k2=f(x+h/2,y+k1/2*h);
    k3=f(x+h/2,y+k2/2*h);
    k4=f(x+h,y+k3*h);
    y=y+(1/6)*(k1+2*k2+2*k3+k4)*h;
end
disp(y)
y= 2.4220e+004
Kod 7.3
```

Örnek 7.3 Birinci mertebeden bir reaksiyon hız ifadesinin Runga-Kutta yöntemiyle çözümü

A $\rightarrow$ R reaksiyonu birinci mertebeden olup, reaksiyon hızı $\frac{dN_A}{dt} = -3N_A$ ile verilmiştir ($k=3$ saat $^{-1}$),

N_A =mol sayısı, $N_{A_0}=1$ mol ($t=0$ da)

Buna göre dördüncü mertebeden Runga-Kutta metodunu kullanarak 0.2 ve 0.4 saat sonraki "A"nın mol sayısını bulunuz.

Çözüm:

$N_A=y$ diyelim.

$$-\frac{dy}{dt} = 3y \Rightarrow f(t, y) = -3y \quad \Delta h = 0.2 \quad \text{saat olsun.}$$

$$k_1 = \Delta h f(t_0, y_0) = 0.2(-3 \cdot 1) = -0.6$$

$$k_2 = \Delta h f\left(t_0 + \frac{\Delta h}{2}, y_0 + \frac{k_1}{2}\right) = 0.2\left[-3\left(1 - \frac{0.6}{2}\right)\right] = -0.42$$

$$k_3 = \Delta h f\left(t_0 + \frac{\Delta h}{2}, y_0 + \frac{k_2}{2}\right) = 0.2\left[-3\left(1 - \frac{0.42}{2}\right)\right] = -0.474$$

$$k_4 = \Delta h f(t_0 + \Delta h, y_0 + k_3) = 0.2[-3(1 - 0.474)] = -0.316$$

$$y_1 = 1 + \frac{-0.6 - 0.42 - 0.474 - 0.316}{6} = 0.5494 \quad \text{mol}$$

Aynı şekilde

$$y_2 = 0.302 \quad 0.4 \text{ saat için}$$

$$y_3 = 0.166 \quad 0.6 \text{ saat için}$$

$$y_4 = 0.091 \quad 0.8 \text{ saat için}$$

Bu denklemin analitik çözümü

$$y = y_0 e^{-kt}$$

Buna göre

$$t=0.2 \text{ saat ise } y=0.5488$$

$$t=0.4 \text{ saat ise } y=0.3012 \quad \text{bulunur ki bu da Runge-Kutta ile bulduğumuz değerlere çok yakındır.}$$

Bu problemin MATLAB çözümüne ait ifadeler aşağıda verilmiştir

Kod 7.4

```
y=dsolve('-Dy=3*y,y(0)=1')
```

```
y = exp(-3*t)
```

```
y=inline('exp(-3*t)')
```

```
y =
```

```
Inline function:
```

```
y(t) = exp(-3*t)
```

```
y(.2) =0.5488
```

```
y(.4)= 0.3012
```

7.2 Analitik Çözümler

Adi Diferansiyel Denklemlerin (ODE) Analitik (Sembolik) olarak çözümü

Mademki diferansiyel denklemlerin nümerik çözümlerine göz atmaktayız, bu arada MATLAB ile bu denklemleri sembolik (exact) olarak nasıl çözüleceğine bir göz atmanın da faydalı olacağı kanaatindeyiz

Diferansiyel Denklemlerin Bilgisayar Çözümleri

Birçok matematik problemini, bilgisayarın desteğini alarak kolayca çözebiliriz. Bunun için matematik desteği veren uygun bir bilgisayar programı kullanmalıyız. Bu amaca yönelik birçok program ticari olarak üretilmiştir, **MATLAB**, **MATHCAD**, **Mathematica**, **MAPLE** gibi. Bunlardan en yaygın kullanılanlardan birisi **MATLAB** programıdır ki biz de bunu kullanıyoruz.

Bunun için aşağıdaki kurallara dikkat etmeliyiz.

i) MATLAB, **y** gibi bir fonksiyonun varsayılan değişkenini **x** değil **t** olarak kabul etmektedir. Yani diferansiyel denklemimizi yazarken, serbest değişken için **t** kullanmalıyız.

Not: Şayet, değişken olarak x (veya başka bir harf) kullanmak istersek, bunu, komut içinde virgül ile ayrılmış iki ' (kesme) işareti arasında belirtmeliyiz.

ii) y' türev fonksiyonu için **Dy, y''** (2. mertebeden türev fonksiyonu) için **D2y, y'''** (3. mertebeden türev fonksiyonu) için **D3y** yazmalıyız.

iii) MATLAB, denklemleri **sembolik** olarak çözdüğünden, denklemleri ve gerekirse özel değerleri iki '(kesme) işareti arasına yazmalıyız. Birden fazla ifade yazılacaksa, ayrıca olarak aralara virgül (,) koymalıyız.

dsolve Komutu

t bağımsız değişkenine bağlı y gibi bir fonksiyon ve türevlerinden oluşan sembolik ifadeye karşılık gelen diferansiyel denkleminin **genel** ve istenirse tanımlanmış başlangıç/sınır değerlere karşılık gelen **özel** çözümlerini bulmaya yarar.

Genel ifade

dsolve ('diferansiyel denklem', 'özel değer1', 'özel değer2', ...)

komutuyla yazılan diferansiyel denklemin özel değer1, özel değer2, ... özel değerlerine karşılık gelen **özel çözümünü** buluruz.

MATLAB programlama dilinde birinci mertebeden diferansiyel denklem çözümlerini yaparken diferansiyel denklemi aşağıdaki şekle koyar ve **dsolve** komutu ile çözeriz.

$$y' = dy/dx = g(x, y)$$

Örnek 7.4 **dsolve** komutu ile genel çözüm bulma

$y' = y^2$ ise t değişkenine bağlı diferansiyel denklemin genel çözümünü bulunuz.

Çözüm:

dsolve ('Dy=y^2')

ans = $y = -1/(t-C1)$

Burada **C1** integrasyon sabitidir. Sınır ve/veya başlangıç şartları ile değeri bulunabilir.

Örnek 7.5 **dsolve** komutu ile genel çözüm bulma

$y' = y^2 + 1$ ise t değişkenine bağlı diferansiyel denklemin genel çözümünü bulun.

Çözüm:

dsolve ('Dy=y^2+1')

ans = $y = \tan(t+C1)$

Örnek 7.6 **dsolve** komutu ile x 'e göre denklem çözme

$y' = y/x$ ise çözümü bulunuz.

Çözüm:

dsolve ('Dy=y/x', 'x')

ans= $\ln y = \ln x + C1$ olur

Örnek 7.7 *Sınır-değer problemi*

$$\frac{d^2 y}{dx^2} - y = 0, \quad y(0) = 3 \text{ ve } y'(1) = -5$$

ise özel çözümü bulunuz (*sınır-değer problemi*)

Çözüm:

y=dsolve('D2y-y=0', 'y(0) = 3', 'Dy(1) = -5')

y =(3*exp(-1)-5)/(exp(1)+exp(-1))*exp(t)+(5+3*exp(1))/(exp(1)+exp(-1))*exp(-t)

simplify(y)

ans =(3*exp(t)-5*exp(t+1)+5*exp(1-t)+3*exp(2-t))/(exp(2)+1)

pretty(ans)

3 exp(t) - 5 exp(t + 1) + 5 exp(1 - t) + 3 exp(2 - t)

exp(2) + 1

Örnek 7.8 *İkinci mertebeden bir diferansiyel denklemin genel çözümü*

$$y'' + 5y' + 4y = 5x + 3$$

diferansiyel denkleminin genel çözümünü bulunuz.

Çözüm:

y = desolve (D2y + 5*Dy + 4*y = 5*x +3', 'x')

y = C1*exp(-x)+ C2*exp (-4*x)+5/4*x-13/16

Örnek 7.9 *Diferansiyel denklem sistemlerinin çözümü*

Aşağıdaki t'ye bağlı denklem sistemini çözünüz.

$$x' = x + 2y - z$$

$$y' = x + z$$

$$z' = 4x - 4y + 5z$$

Çözüm:

[x, y, z] = dsolve ('Dx = x + 2y - z', 'Dy = x + z', 'Dz = 4x - 4y + 5 z')

x = 2*C1*exp(2*t)-2*C1*exp(t)-C2*exp(3*t)+2*C2*exp(2*t)-1/2*C3*exp(3*t)+1/2*C3*exp(t)

y = 2*C1*exp(t)-C1*exp(2*t)+C2*exp(3*t)-C2*exp(2*t)+1/2*C3*exp(3*t)-1/2*C3* exp(t)

z = -4*C1*exp(2*t)+4*C1*exp(t)+4*C2*exp(3*t)-4*C2*exp(2*t)-C3*exp(t)+2*C3*exp(3*t)

Örnek 7.10 x 'e göre diferansiyel denklemi çözme

$$\frac{dy}{dx} = 4x + 9y \text{ diferansiyel denkleminin genel çözümünü bulunuz.}$$

Çözüm:

$$\text{çözüm} = \text{dsolve}('Dy=4x+9y', 'x')$$

$$\text{çözüm} = -9-4x+\exp(x)*C1$$

Örnek 7.11 x 'e göre özel çözüm bulma

Örnek 7. 10 için $y(0) = 5$ başlangıç koşulu ile özel çözümünü bulunuz

Çözüm:

$$\text{çözüm} = \text{dsolve}('Dy=4x+9y', 'y(0)=5', 'x')$$

$$\text{çözüm} = -9-4x+5*\exp(x)$$

Örnek 7.12 Başlangıç-değer probleminin **dsolve** ile çözümü

$$\frac{dy}{dx} = 3y^2 \text{ ve } y(0) = 4 \text{ başlangıç değer probleminin özel çözümünü bulunuz.}$$

$$\text{çözüm} = \text{dsolve} (Dy=3*y^2, 'y(0)=4', 'x')$$

$$\text{çözüm} = y = -4/(12*x-1)$$

Örnek 7.13 x 'e göre genel çözüm bulma

$y' + 4y = 3x$ diferansiyel denkleminin genel çözümünü bulunuz.

Çözüm:

$$y = \text{dsolve}('Dy+4*y=3*x', 'x')$$

$$y = C1 * \exp(-4x) + 3/4 * x - 3/16$$

7.2.1 Çoklu Diferansiyel Denklemler

Biz burada **iki bağımlı diferansiyel denklemi** göz önüne alıp bunları aynı anda *Runga-Kutta* ile nasıl çözeceğimizi örnekle ortaya koyacağız.

$$y' = f(t, y, z)$$

$$z' = g(t, y, z)$$

denklemlerin çözümüne bakalım:

$$y_{n+1} = y_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

$$z_{n+1} = z_n + \frac{l_1 + 2l_2 + 2l_3 + l_4}{6}$$

Eğer h =integrasyonda küçük bir adım ise

$$k_1 = hf(t_n, y_n, z_n)$$

$$l_1 = hg(t_n, y_n, z_n)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{l_1}{2}\right)$$

$$l_2 = hg\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{l_1}{2}\right)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}, z_n + \frac{l_2}{2}\right)$$

$$l_3 = hg\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}, z_n + \frac{l_2}{2}\right)$$

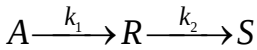
$$k_4 = hf(t_n + h, y_n + k_3, z_n + l_3)$$

$$l_4 = hg(t_n + h, y_n + k_3, z_n + l_3)$$

k_i ve l_i 'lerin küçük indekslieleri hesaplandıktan sonra büyüklere doğru gidilmelidir.

Bu durumu, kimya mühendisliğinde ardışık reaksiyon sistemine uygulayarak örnekleyelim [40-45]

Örnek 7.14 *Ardışık kimyasal reaksiyonların konsantrasyonlarının tayini*



ardışık seri reaksiyonlarının birinci mertebeden olduğunu kabul edelim.

$k_1=3 \text{ saat}^{-1}$, $k_2=1 \text{ saat}^{-1}$, $t=0$ iken $N_{A_0}=1 \text{ mol}$, ve $N_{R_0} = N_{S_0} = 0$ olsun.

a) 0.2 saatlik artışlarla **A**, **R** ve **S** in konsantrasyonlarını Runge-Kutta metodunu kullanarak bulunuz.

Burada $h=0.2$ saat olsun.

b) Bu neticeleri analitik yoldan elde edilen N_A , N_R ve N_S konsantrasyonları ile kıyaslayınız.

Çözüm:

$N_A=y$, $N_R=z$ diyelim

$$\frac{dN_A}{dt} = -3N_A \Rightarrow \frac{dy}{dt} = -3y$$

$$\frac{dN_R}{dt} = 3N_A - N_R \Rightarrow \frac{dz}{dt} = 3y - z$$

Denklem sistemleri elde edilir.

Böylece

$$f(t, y, z) = -3y$$

$$g(t, y, z) = 3y - z$$

olur.

$$t=0 \text{ iken } y_0=1, \quad z_0=0$$

$$k_1 = 0.2[(-3)(1)] = -0.6$$

$$l_1 = 0.2[3.1 - 0] = 0.6$$

$$k_2 = 0.2\left[(-3)\left(1 + \frac{-0.6}{2}\right)\right] = -0.42$$

$$l_2 = 0.2\left[(3)\left(1 - \frac{0.6}{2}\right) - \frac{0.6}{2}\right] = 0.36$$

$$k_3 = 0.2\left[(-3)\left(1 - \frac{0.42}{2}\right) + 0\right] = -0.474$$

$$l_3 = 0.2\left[3\left(1 - \frac{0.42}{2}\right) - \left(\frac{0.36}{2}\right)\right] = 0.438$$

$$k_4 = 0.2\left[-3\left(1 - \frac{0.472}{1}\right) + 0\right] = -0.3156$$

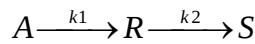
$$l_4 = 0.2[3(1 - 0.472) - 0.438] = 0.228$$

$$y_1 = 1 + \frac{-0.6 + 2(-0.42) + 2(-0.474) + (0.3156)}{6} = 0.5494 \quad \text{mol}$$

$$z_1 = 0 + \frac{0.6 + 2(0.36) + 2(0.438) + 0.228}{6} = 0.4040 \quad \text{mol}$$

Madde denkleğinden $N_{S_1} = 0.0466 \text{ mol}$ olarak bulunur.

Şimdi bunu analitik yolla çözelim:



$$\frac{dN_A}{dt} = -3N_A \Rightarrow N_A = N_{A_0} e^{-k_1 t}$$

$$\frac{dN_R}{dt} = k_1 N_A - k_2 N_R \Rightarrow \frac{dN_R}{dt} + k_2 N_R = k_1 N_A = k_1 N_{A_0} e^{-k_1 t}$$

Bu denklem birinci mertebeden lineer bir denklemdir ve

$$\frac{dy}{dx} + P(x)y = Q(x) \text{ şeklindedir.}$$

Burada,

$$N_R = y, \quad P(x) = k_2, \quad Q(x) = k_1 N_{A_0} e^{-k_1 t} \text{ 'dir.}$$

İntegrasyon faktörünü kullanırsak

Genel çözüm:

$$y = e^{-\int P(x)dx} \left[\int e^{\int P(x)dx} \cdot Q(x)dx + K \right]$$

K = integrasyon sabiti

olur.

Bunları yerine koyduktan sonra integrasyon yapıldığında

$$N_R = \left(\frac{k_1 N_{A_0}}{k_2 - k_1} \right) e^{-k_1 t} + K e^{-k_2 t}$$

elde edilir.

t=0 $N_R=0$ ve $N_S=0$, $N_{A_0}=1$ olduğuna göre

$$K = -\frac{k_1 N_{A_0}}{k_2 - k_1} \quad \text{bulunur.}$$

$$N_R = \frac{k_1 N_{A_0}}{k_2 - k_1} e^{-k_1 t} + e^{-k_2 t}$$

olur.

$N_A + N_R + N_S = 1$ olduğuna göre

$$N_S = N_{A_0} \left[1 - \left(\frac{k_2}{k_2 - 1} \right) e^{-k_1 t} + \left(\frac{k_1}{k_2 - k_1} \right) e^{-k_2 t} \right]$$

ifadesi ile elde edilir.

Şimdi t=0.2 olarak alınırsa

$N_A=0.5488$ ve $N_R=0.4049$ bulunur ki bu da *Runga-Kutta* ile elde edilen ile hemen hemen aynıdır.

Bu problemin MATLAB programı ile grafiksel çözümü aşağıda verilmiştir (Şekil 7.2)

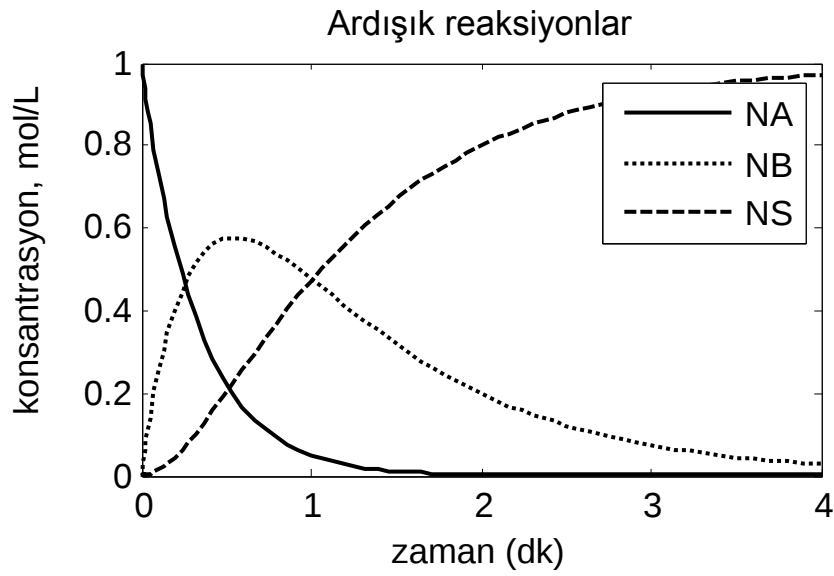
Kod 7.5

```
function dNdt=Ornek14(t,N)
% dNA/dt=-k1NA
% dNR/dt=k1NA-k2NR
% dNS/dt=k2NR
% NA= N(1), NR=N(2), NS=N(3) olsun
global k1 k2
dNdt=[-k1*N(1);k1*N(1)-k2*N(2);k2*N(2)];
% Aşağıdaki koşum (Run file)dosyasını yazalım
clc, clf, clear, k1=3; k2=1;
tspan=[0 4];
```

```

N0=[1 0 0];
[t,N]=ode45('Ornek14',tspan,N0)
plot(t,N(:,1),t,N(:,2),t,N(:,3))
legend('NA','NR','NS')
xlabel('zaman (dk)')
ylabel('konsantrasyon, mol/L')
title('Ardışık reaksiyonlar')

```



Şekil 7.2 Ardışık reaksiyonların grafiksel çözümü

Özet

Bu bölümde, mühendislik ve fen uygulamalarında son derece önemli olan diferansiyel denklemlerin *Euler* ve *Runga-Kutta yöntemleri*yle nasıl çözüleceği tartışıldı. Sembolik (analitik) çözüm için MATLAB’de ***dsolve*** komutunu kullanarak tek veya çoklu diferansiyel denklemlerin çözümleri örneklerle verildi. Ayrıca başlangıç-değer ve sınır-değer problemlerinin MATLAB’de nasıl çözüldüğü gösterildi.

BÖLÜM 8 - KİMYA MÜHENDİSLİĞİNDE ÖNEMLİ BAZI PROSES VE BİRİMLERİNİN MODELLENMESİ

Amaç

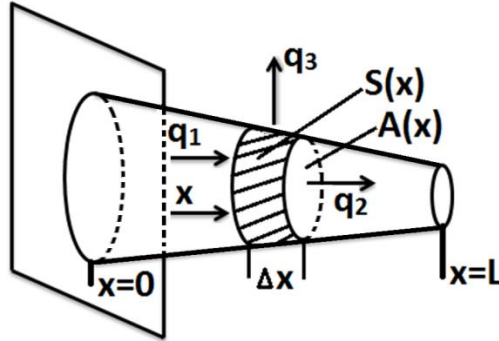
Bu bölümün bitiminde,

- Bazı proses birimlerinde, örneğin
 - Kanatçıklar (ısı transferi)
 - Buharlaştırıcılar
 - Reaktörler (Kesikli, TKSR ve PAR)
- Denge Proseslerinde
 - Tek kademeli
 - Ters akımlı çok kademeli

sistemlerin nasıl modellendiğini öğrenmiş olacağız.

8.1 Kanatçıklardan Isı Transferi Modeli

Küçük bir kontrol hacmi alalım. Bu hacmin kesiti $A(x)$, yüzey alanı da $S(x)$ olsun. Bu durum Şekil 8.1’de gösterilmiştir [46-48].



Şekil 8.1 Kanatçıklardaki (fins) ısı transferi

Burada kanatçığın kesitinin x istikametinde değiştiğini kabul edelim. Kontrol hacmine kondüksiyonla (iletim) gelen ısı q_1 , çıkan q_2 ve konveksiyonla (taşınım) dışarı yayılan ısıya da q_3 dersek:

$$q_1 = q_2 + q_3$$

yazılabilir. Yatışkın halde

$$kA \frac{dT}{dx} \Big|_{x+\Delta x} - kA \frac{dT}{dx} \Big|_x - hS(T - T_\infty) = 0 \text{ olur.}$$

T_∞ = çevre sıcaklığı olup sabittir

$P(x)$ =kanatçığın çevresi

$$S(x) = P(x)\Delta x$$

$$\lim_{\Delta x \rightarrow 0} \frac{kA \frac{dT}{dx} \Big|_{x+\Delta x} - kA \frac{dT}{dx} \Big|_x}{\Delta x} - hP(T - T_\infty) = 0$$

$$\frac{d}{dx} \left(kA \frac{dT}{dx} \right) - hP(T - T_\infty) = 0 \text{ olur.}$$

Kanatçıktaki sıcaklık dağılımını bulmak için, bu diferansiyel denklemi çözmek gerekir. Eğer

$\Theta = T - T_\infty$, k =sabit ve A da değişmiyorsa o zaman yukarıdaki diferansiyel denklem

$$\frac{d^2 \Theta}{dx^2} - \frac{hP}{kA} \Theta = 0 \text{ haline dönüştürülebilir. Bu denklemin çözümü **iki sınır şartı** gerektirir.}$$

Sınır şartları:

Üç tip sınır şartı ile karşılaşabiliriz:

a) $T = T_0 \quad @ \ x = 0$

$T = T_L \quad @ \ x = L$

b) $T = T_0 \quad @ \ x = 0$

$\frac{dT}{dx} = 0 \quad @ \ x = L \quad (\text{Uçta izolasyon var})$

c) $T = T_0 \quad @ \ x = 0$

$-k \frac{dT}{dx} = h(T - T_\infty) \quad @ \ x = L \quad (\text{Uçtan dışarıya konveksiyonla ısı transferi oluyor})$

Yukarıda verilen ikinci dereceden diferansiyel denklem için

Genel çözüm:

$$\Theta = c_1 e^{mx} + c_2 e^{-mx}$$

burada, $m^2 = \frac{hP}{kA}$

Hiperbolik fonksiyon halinde ise

$$\Theta = A \cdot \cosh mx + B \cdot \sinh mx \quad \text{olur.}$$

$$\left(\cosh x = \frac{e^x + e^{-x}}{2}, \quad \sinh x = \frac{e^x - e^{-x}}{2} \right)$$

olduğunu hatırlayınız!

“b” sınır şartlarını kullanırsak

$$\frac{\Theta}{\Theta_0} = \frac{T - T_\infty}{T_0 - T_\infty} = \frac{e^{mx}}{1 + e^{2mL}} + \frac{e^{-mx}}{1 + e^{-2mL}}$$

bu sınır şartlarını kullandıktan sonra

$$\frac{\Theta}{\Theta_0} = \frac{\cosh [m(L - x)]}{\cosh mL} \quad \text{bulunur.} \quad \Theta = T - T_\infty \quad \text{olduğunu hatırlayınız.}$$

Kanatçıktan transfer olan ısı

$$q = \int_s h[T - T_\infty]dS = \int_s h\Theta dS$$

veya

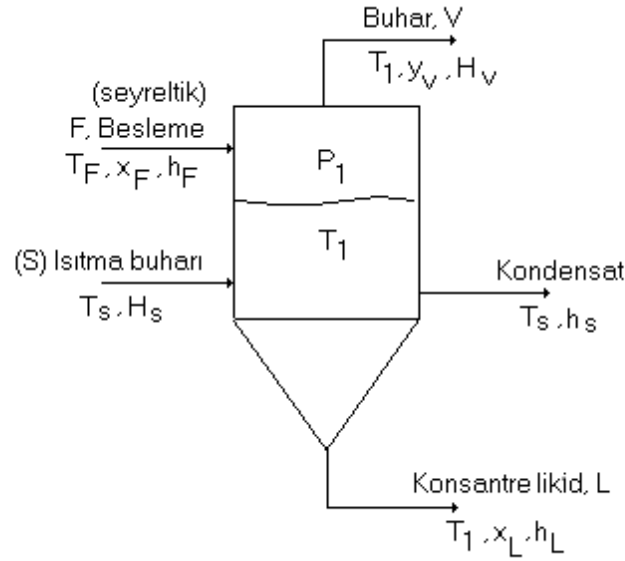
$$q = -kA \frac{dT}{dx} \Big|_0$$

denkleminde bulunur. Eğer $T=f(x)$ fonksiyonu belli ise bu son denklemin kullanılması daha uygundur.

8.2 Buharlaştırıcı Modeli

Buharlaştırıcılar (evaporatörler), seyreltik olan bir çözeltiyi daha derişik bir hale getirmek için kullanılan bir ısı deęiřtiriciden başka bir řey deęildir. Burada çözücü ile çözünen arasında büyük **bir kaynama noktası farkı** vardır [49].

Buharlaştırıcılar için genel denklikleri Şekil 8.2 üzerinde görelim:



Şekil 8.2 Tek Kademeli Buharlaştırıcı

S = İstimin besleme debisi

T_F = Besleme sıcaklığı

x_F = Besleme konsantrasyonu

h_F = Besleme entalpisi

F = Besleme debisi

T_S = Isıtma istiminin sıcaklığı

H_S = Isıtma istiminin entalpisi

h_S = Kondensatın entalpisi

T_1 = Buharlaştan çözeltinin sıcaklığı

h_L = Konsantre çözelti entalpisi

L = Konsantre çözeltinin debisi

x_L = Konsantre çözeltinin konsantrasyonu

y_v = Anahtar bileşenin buhar mol fraksiyonu

$$q = U \cdot A \cdot \Delta T, \quad \Delta T = T_s - T_1$$

$$\lambda = H_s - h_s \text{ (İstimin gizli buharlaşma ısısı)}$$

Şimdi hem kütle, hem de enerji denkliklerini yazalım:

$$\text{Yatışkın halde: } F = L + V \quad (\text{total denklik})$$

$$\text{Çözünen komponent denkliği: } F \cdot x_F = L \cdot x_L \quad (y_v=0)$$

Enerji denkliği:

$$F \cdot h_F + S \cdot H_s = L \cdot h_L + V \cdot H_v + S \cdot h_s \quad (\text{Yatışkın halde})$$

veya

$$F \cdot h_F + S \cdot \lambda = L \cdot h_L + V \cdot H_v \quad (\lambda = H_s - h_s \text{ olduğunu hatırlayınız})$$

Transfer olan ısı:

$$q = S \cdot (H_s - h_s) = S \cdot \lambda \text{ olur.}$$

Bu durumu nümerik bir problemle örnekleyelim.

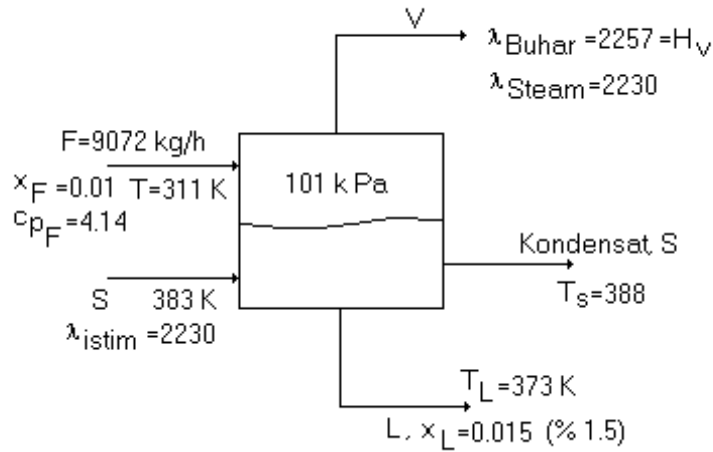
Örnek 8.1 Buharlaştırıcı hesaplamaları [30]

Tek kademeli bir buharlaştırıcıya 9072 kg/saat lık bir debi ile % 1 (ağırlık) tuz çözeltisi devamlı olarak beslenmektedir. Giriş sıcaklığı 311 K olan besleme akımı % 1.5 'a kadar buharlaştırılacaktır.

Buharlaştırıcıdaki basınç 1 atm olup ve ısıtma buharı da 143.3 kPa da doymuş olarak gelmektedir.

Toplam ısı transferi katsayısı $U=1074 \text{ W/m}^2\text{K}$ dir. Buharlaştırıcıdan çıkan buharın miktarı ile gerekli ısıtma yüzeyini "A" ve istim miktarı "S" yi hesaplayınız.

Çözelti seyreltik olduğu için kaynama noktası suyunkine eşdeğer kabul edilebilir. Giren akımın spesifik ısı kapasitesi $C_{pF}=4.14 \text{ kJ/kg K}$ olarak alınacaktır.



Şekil 8.3 Örnek 8.1'e ait akış diyagramı

Çözüm:

Toplam kütle denkliği:

$$F=V+L \quad 9072=V+L$$

Çözünen bileşen için denklik : $F \cdot x_F = L \cdot x_L$

$$9072(0.01)=L(0.015)$$

Bu iki denklemden, $L= 6048 \text{ kg/ saat}$

$$V= 3024 \text{ kg/saat bulunur.}$$

Kaynama noktası:373 K

Referans sıcaklığımız 373 K olsun

$$h_F = c_{p_F} (T_F - T_1), \quad h_L = 0 \quad \text{alırsak}$$

$$F \cdot h_F + S \cdot H_s = L \cdot h_L + V \cdot H_v + S \cdot h_s; \quad H_v = \lambda_{buhar}$$

$$9072(4.14)(311-373) + S(2230) = 3048 (2257)$$

$$S = 4108 \text{ kg/saat}$$

$$q = S \cdot \lambda_{istim} = 4108(2230) 1000/3600 = 2544000 \text{ W}$$

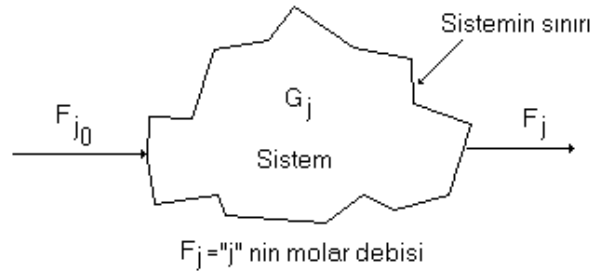
$$\Delta T = T_s - T_1$$

$$q = U \cdot A \cdot T = 1704 (A) (383 - 373)$$

$$A = 149.3 \text{ m}^2$$

8.3 Reaktörlerin Modellenmesi

Kimyasal reaksiyonların içinde olduğu kaplara **reaktör** denir. Bunlar, *kesikli* (batch), *yarıkesikli* (semi-batch) veya *sürekli* (continuous) reaktörler olarak akış itibariyle üçe ayrılır. Reaksiyon söz konusu olduğu için bu tip sistemlerde hem toplam kütle denkliği, hem de bileşen denklüklerini yazabiliriz. Aşağıdaki Şekil 8.4 de verilen sistemi göz önüne alalım. Burada herhangi bir "j" bileşeni için mol denkliğini yazabiliriz. "j" reaktant olabildiği gibi ürün de olabilir.



Şekil 8.4 Reaksiyonun gerçekleştiği bir sistemde mol denkliği

$$\text{Giren} + \text{Üretilen} - \text{Çıkan} = \text{Biriken}$$

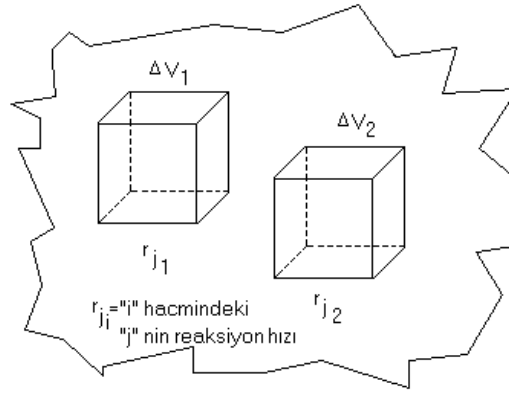
$$F_{j0} + G_j - F_j = \frac{dN_j}{dt}$$

$$V = \text{sistemin hacmi ve } r_j = \text{reaksiyon hızı ise } G_j = V \cdot r_j \text{ olacaktır.}$$

"r_j" sistemin her yerinde aynı olmayabilir!

Eğer reaksiyon hızı her tarafta aynı değilse o zaman sistemi çok küçük ΔV hacimlerine bölmemiz

gerekir. Her küçük hacimde reaksiyon hızının sabit olduğunu kabul edip, sonra da elde edilen değerler bütün sistem için integre ederiz. Şekil 8.5' e bakalım.



Şekil 8.5 Reaktörlerde Dağılık Parametre Sistemi

$$\begin{aligned}
 \Delta G_1 &= r_{j_1} \cdot \Delta V_1 \\
 \Delta G_2 &= r_{j_2} \cdot \Delta V_2 \\
 &\vdots \\
 \Delta G_j &= r_{j_i} \cdot \Delta V_i
 \end{aligned}
 \left\{ \begin{array}{l} \sum_{j=1}^M \Delta G_j = G_j = \sum_{i=1}^M r_{j_i} \cdot \Delta V_i \quad (M \text{ küçük hacimlerin sayısı}) \end{array} \right.$$

Eğer

$$M \rightarrow \infty \text{ ise } G_j = \int_V r_j \cdot dV$$

olur.

Böylece yukarıdaki sistem için

$$F_{j_0} + \int_V r_j \cdot dV - F_j = \frac{dN_j}{dt}$$

elde edilir.

Bu genel denklemden yola çıkarak herhangi bir **reaktörün modelini** yazabiliriz.

8.3.1 Kazan Tipi Kesikli Reaktör (KR) (Batch Reactor)

$$F_{j_0} = F_j = 0 \quad (\text{Giren ve çıkan yok})$$

$$\int r_j \cdot dV = \frac{dN_j}{dt} \text{ şekline indirgenmiş olur. } j \text{ bileşeninin, } A \text{ reaktantı olduğunu}$$

düşünürsek ve eğer $V = \text{sabit}$ ise

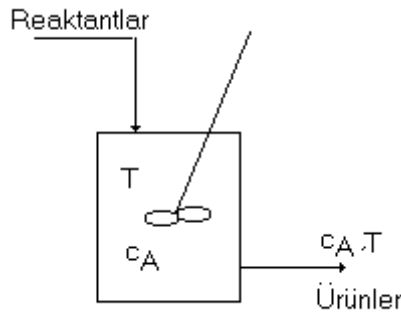
$$\frac{1}{V} \frac{dN_A}{dt} = -r_A$$

$$\frac{d(N_A/V)}{dt} = \frac{dC_A}{dt} = -r_A \quad (\text{A reaktant ise reaksiyon hızı } "-r_A" \text{ olur})$$

$$\frac{dC_A}{dt} < 0 \text{ olduğuna lütfen dikkat ediniz.}$$

8.3.2 Tam Karıştırmalı Sürekli Reaktör (TKSR) (Continuous-Stirred-Tank-Reactor, CSTR)

Bu tip ideal reaktörlerde % 100 'lük bir karışma olduğu için reaktörün her tarafında hem konsantrasyon hem de sıcaklık aynıdır. Ayrıca çıkan ürünün konsantrasyon ve sıcaklığı ile reaktördeki konsantrasyon ve sıcaklık aynıdır (Şekil 8.6).



Sekil 8.6 TKSR

$$F_{j0} + \int r_j \cdot dV - F_j = \frac{dN_j}{dt}$$

Genel denklemini göz önüne alırsak (yatışkın halde sağ taraf sıfır olur)

Burada birikim olmadığına ve F_{j0} ile $F_j \neq 0$ olduğuna göre

$$F_{j0} - F_j = -r_j V \rightarrow V = \frac{F_{j0} - F_j}{-r_j}$$

model denklemini elde ederiz.

Eğer x =dönüşüm ise $F_j = F_{j0} (1 - x_j)$ olduğuna göre,

$$V = \frac{F_{j0} \cdot x_j}{-r_j} \text{ de yazılabilir.}$$

"j" reaktant ise

$$-r_j > 0 \quad \text{ve} \quad (F_{j0} - F_j) > 0$$

"j" ürün ise

$$-r_j < 0 \quad \text{ve} \quad (F_{j0} - F_j) < 0$$

$$F_j = C_j \cdot v$$

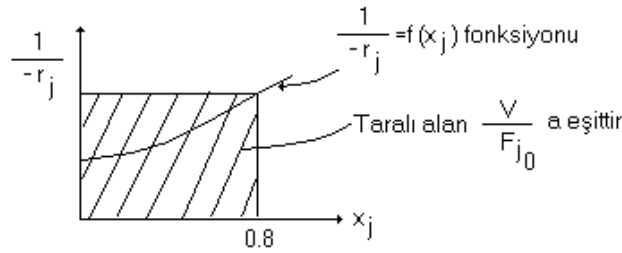
Burada, v hacimsel debidir.

Bu tip reaktörler daha ziyade **sıvı faz** reaksiyonlarında kullanılır. Reaksiyonun hız denklemini biliniyorsa belli bir dönüşüm için V hesap edilebilir. Çoğu zaman reaksiyon hızı, dönüşüm, " x_j " nin fonksiyonu olarak ifade edilir.

Dönüşüme göre verilen denklem düzenlenirse

$$\frac{V}{F_{j0}} = \frac{1}{-r_j} \cdot x_j \text{ yazılabilir. Eğer } \frac{1}{-r_j}, x_j \text{ 'ye karşı çizilirse } \textbf{pozitif mertebedeki} \text{ bir reaksiyon için}$$

Şekil 8.7 deki grafik elde edilir.



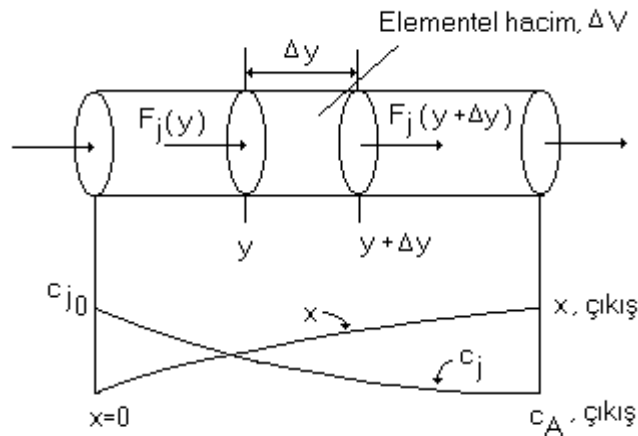
Şekil 8.7 TKSRe hacminin **Levenspiel** grafiğinden bulunması

Örneğin, burada **taralı alan** F_{j0} ile çarpılırsa % 80 dönüşüm için gerekli reaktör hacmini bulunmuş olur

8.3.3 Piston Akışlı Reaktör (PAR)

(**Plug-Flow-Reactor, PFR**)

İdeal olan piston akışlı boru tipli reaktörlerde, konsantrasyon reaktör boyunca değiştiği için, TKSRe'nin aksine, küçük bir reaktör hacmi seçip mol denkleğini o **kontrol veya elementel hacim** üzerinde yapıp sonra da bütün reaktör boyunca integre etmemiz gerekiyor (Şekil 8.8).



Şekil 8.8 Piston Akışlı Reaktör (PAR)

Yatışkın halde,

$$\frac{dN_j}{dt} = 0$$

Küçük elementel ΔV hacmi üzerinde "j" nin mol denkliğini yazarsak

$$\int_{\Delta V} r_j \cdot dV = r_j \cdot \Delta V$$

Burada $\Delta V = A \cdot \Delta y$ olup 'A' boru kesitidir.

$$\Rightarrow F_j(y) + r_j \cdot \Delta V - F_j(y + \Delta y) = 0$$

$$F_j(y) + r_j \cdot A \cdot \Delta y - F_j(y + \Delta y) = 0$$

$$F_j(y + \Delta y) - F_j(y) = r_j \cdot A \cdot \Delta y$$

(Her iki tarafı Δy 'ye bölüp ve Δy 'yi sıfıra götürürsek)

$$\lim_{\Delta y \rightarrow 0} \frac{F_j(y + \Delta y) - F_j(y)}{\Delta y} = r_j \cdot A$$

$$\frac{dF_j}{dy} = r_j \cdot A \rightarrow \frac{dF_j}{dV} = r_j$$

$$\rightarrow V = \int \frac{dF_j}{r_j} = \int \frac{d(v \cdot c_j)}{r_j} = v \cdot \int_{c_{j0}}^{c_j} \frac{dc_j}{r_j}$$

$c_j = c_{j0} (1 - x_j)$ olduğuna göre $V = v \cdot c_{j0} \int_0^{x_j} \frac{dx_j}{-r_j}$ olur. Burada v =hacimsel debi olup sabit kabul edilmiştir.

Bu son denklemde $-r_j = f(x_j)$ dir. Bu integrasyonu yapabilmek için " - r_j " nin x_j ile nasıl değiştiğini bilmek lazımdır. Bu ilişki çok karmaşık ise o zaman V (reaktör hacmini) bulmak için analitik olarak integrasyonu yapamayabiliriz. Bu durumda, önceki bölümlerde gördüğümüz nümerik integrasyon yönteminden (**Euler, Runga-Kutta, Simpson**, vs.) birini kullanmak suretiyle gerekli integrasyon yapılabilir.

Örnek 8.2 Reaktör hacminin bulunması

A--->B Reaksiyonun hız denklemi $-r_A = k \cdot C_A$ olsun (1. mertebe).

Eğer bu reaksiyon ideal piston akışlı bir reaktörde gerçekleşiyorsa reaktör hacmi

$$V = v \int_{c_{A_0}}^{c_A} \frac{dc_A}{-kc_A} = \frac{v}{-k} \left| \ln c_A \right|_{c_{A_0}}^{c_A} = \frac{v}{k} \ln \frac{c_{A_0}}{c_A}$$

denkleminde bulunur. Sabit sıcaklıkta hız sabiti (daha doğrusu hız katsayısı) sabittir. Çünkü **k** sadece sıcaklığın bir fonksiyonudur. (Katalitik reaksiyonlarda **k**'nın, katalizörün de aynı zamanda bir fonksiyonu olduğunu burada belirtelim).

V'yi **x** cinsinden bulunuz.

Çözüm

$$x = \frac{c_{A_0} - c_A}{c_{A_0}} = 1 - \frac{c_A}{c_{A_0}}$$

x=dönüşüm

$$\rightarrow \frac{c_A}{c_{A_0}} = 1 - x$$

Yukarıdaki denklem

$$V = -\frac{v}{k} \ln(1 - x) \quad \text{şeklinde de yazılır.}$$

Genel olarak, x dönüşüm ise

$$V = F_{A_0} \int_0^x \frac{dx}{-r_A}$$

olduğunu bu kez de siz türetiniz.

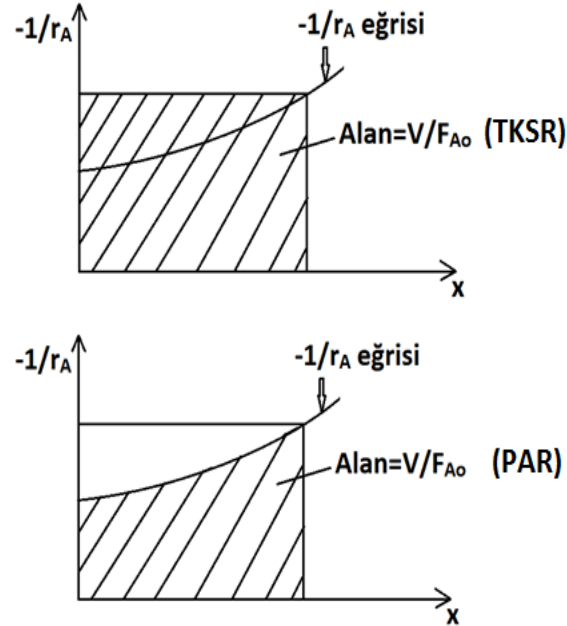
Böylece özetlersek:

$$V = \frac{1}{-r_A} \frac{dN_A}{dt} \quad (\text{Kesikli kazan tipi reaktör})$$

$$V = \frac{F_{A_0}}{-r_A} x_A \quad (\text{Tam karıştırmalı sürekli reaktör})$$

$$V = F_{A_0} \int_0^x \frac{dx}{-r_A} \quad (\text{Piston akışlı reaktör})$$

Eğer $-r_A$ çok karmaşık ise o zaman Şekil 8.9 da görüldüğü gibi $1/-r_A$, x_A ' ya karşı çizilir ve grafiksel olarak reaktör hacmi bulunur.



Şekil 8.9 TKSR ve PAR hacimlerinin grafiksel olarak bulunması

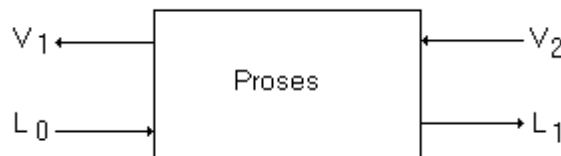
Aynı dönüşüm için $(V)_{TKRS} > (V)_{PAR}$ olduğu grafikten de açık olarak görülmektedir. Daha fazla bilgi için örneğin, Fogler [31] ve Levenspiel' in [42] alanında çok iyi olan kitaplarına başvurabilirsiniz.

8.4 Denge Prosesleri

8.4.1 Tek Kademeli Denge Prosesleri

İki ayrı fazı temas haline getirdiğimizde eğer çözünen komponentlerden biri her iki faza da

dağılabiliyorsa o komponent dengeye varıncaya kadar iki fazda belli konsantrasyonlarda yer alır. Bu durum, *ekstraksiyon, distilasyon ve absorpsiyonun* temelini oluşturur. (A+B+C) komponentlerinden oluşan iki fazı göz önüne alalım (Şekil 8.10).



Şekil 8.10 Denge prosesi

Total denklik: $L_0 + V_2 = L_1 + V_1 = M$

(A) denkliği: $L_0 x_{A0} + V_2 y_{A2} = L_1 x_{A1} + V_1 y_{A1} = M x_{AM}$

(C) denkliği: $L_0 x_{C0} + V_2 y_{C2} = L_1 x_{C1} + V_1 y_{C1} = M x_{CM}$

X_{AM} = M deki "A"nın kütleli fraksiyonu

Bu üç denklemi çözmek için ayrıca komponentler arasında denge ilişkilerinin de bilinmesi gerekir (Aşağıda örneklerle göreceğiz).

8.4.2 Gaz-Sıvı Sistemler

"A" nın V fazı içinde çözündüğünü düşünelim (V = hava olsun).

"A" nın yine L fazında da çözündüğünü kabul edelim (L = su olsun)

Havanın sudaki çözünürlüğünü ihmal edersek A sadece iki faz arasında belli bir değere gelinceye kadar dağılacaktır.

Buna göre L' = suyun miktarı

V' = havanın miktarı ise

$$L' \left(\frac{X_{A_0}}{1 - X_{A_0}} \right) + V' \left(\frac{y_{A_2}}{1 - y_{A_2}} \right) = L' \left(\frac{X_{A_1}}{1 - X_{A_1}} \right) + V' \left(\frac{y_{A_1}}{1 - y_{A_1}} \right)$$

Yukarıdaki denklemi çözmek için y_{A_1} ile x_{A_1} arasında dengeyi veren **Henry yasasını** kullanmak gerekiyor.

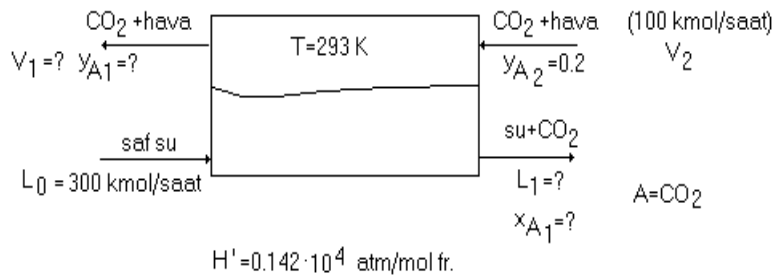
$$y_{A_1} = H' x_{A_1}$$

Burada H' Henry sabitidir.

Örnek 8.3 Gaz-sıvı sistemler

293 K de CO₂ içeren hava su ile temasa getiriliyor. CO₂ suda çözünüyor. Tek kademeli sistemde

meydana gelen bir absorpsiyonda çıkış gazı ile çıkan sıvı dengede olduğu kabul ediliyor. Gazın giriş debisi 100 kmol/saat ve % 20 CO₂ içermektedir. Suyun giriş debisi 300 kmol/saat olduğuna göre çıkan gaz ve sıvının miktarları ve bileşimlerini bulunuz.



Şekil 8.11 Örnek 8.3'e ait akış diyagramı

$$L' = L_0 = 300 \text{ kmol}$$

$$V' = V_1(1 - y_{A_2}) = 100(1 - 0.2) = 80 \text{ kmol/saat}$$

CO₂ için balansı yazarsak

$$300\left(\frac{0}{1-0}\right) + 80\left(\frac{0.2}{1-0.2}\right) = 300\left(\frac{x_{A_1}}{1-x_{A_1}}\right) + 80\left(\frac{y_{A_1}}{1-y_{A_1}}\right)$$

Elimizde bir denklem ve iki bilinmeyen var (x_{A_1} ve y_{A_1}). İkinci bir denkleme daha ihtiyacımız var ki bu da Henry yasasından gelen ilişkidir.

$$y_{A_1} = 0.142 \cdot 10^4 x_{A_1}$$

verilmiş olsun.

" y_{A_1} " in bu değerini yukarıda yerine koyarsak

$$x_{A_1} = 1.41 \cdot 10^{-4}$$

$$y_{A_1} = 0.2$$

$$L_1 = \frac{L'}{1 - x_{A_1}} \cong 300 \text{ kg/saat}$$

$$V_1 = \frac{V'}{1 - y_{A_1}} \cong 100 \text{ kg/saat}$$

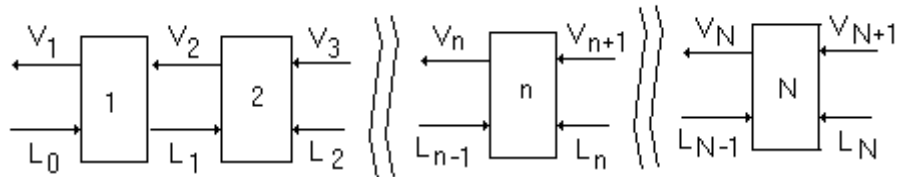
Böylece giren ve çıkan akımlar ve bileşimler **hemen hemen aynı kalmaktadır**. Bu da CO₂' in sudaki **çözünürlüğünün az** olduğunu göstermektedir.

8.4.3 Ters Akımlı Çok Kademeli Prosesler

Biraz önce verdiğimiz örnekte tek kademeli temasta transfer olan maddenin miktarı çok azdı. Endüstride bir fazdan diğer faza daha çok madde aktarmak için tek kademe yerine **çok kademeli sistemler** kullanılır.

Her kademeyi terkeden fazların **termodinamiksel dengede** olduğu kabul edilir. Burada çıkaracağımız denklemler çok kademeli (*multiple-stage*) operasyonların (*distilasyon, ekstraksiyon, absorpsiyon, vb.*) hepsi için geçerlidir.

İki fazlı ters akım prensibine göre çalışan bir sistem aşağıdaki Şekil 8.12' de şematik olarak verilmiştir.



Şekil 8.12 Ters akım prensibine göre çalışan çok kademeli denge prosesleri

$$L_0 + V_{N+1} = L_N + V_1 = M \quad (\text{Toplam denklik})$$

"n" ci kademede V_n ve L_n denge halinde olduklarını bir kez daha vurgulayalım.

Komponent denklikleri :

Transfer olan bileşenin L ve V fazındaki konsantrasyonları sırasıyla x ve y ile gösterelim.

$$L_0 x_0 + V_{N+1} y_{N+1} = L_N x_N + V_1 y_1 = M x_M$$

İlk "n" kademe için denklik:

$$L_0 + V_{n+1} = L_n + V_1 \quad (\text{Toplam})$$

$$L_0 x_0 + V_{n+1} y_{n+1} = L_n x_n + V_1 y_1$$

Bu denklemi y_{n+1} için çözersek

$$y_{n+1} = \frac{L_n x_n}{V_{n+1}} + \frac{V_1 y_1 - L_0 x_0}{V_{n+1}}$$

ifadesi elde edilir ki bu da y_{n+1} ile x_n arasındaki ilişkiyi göstermektedir. Buna *işletme doğrusu* denilir.

Yani, V deki y_{n+1} ile L deki x_n arasında (birbirlerini ters yönde geçen iki akım) çok önemli bir ilişkidir.

Örnek 8.4 Gaz-Sıvı sistemler

Havanın içinde %1 mol aseton bulunmaktadır ve bu asetonlu hava çok kademeli bir kuleye gelip orada su ile temas ettikten sonra, asetonun % 90 ı suda tutulmaktadır. Gelen (aseton+havanın) debisi 30 kmol/saat, suyunki ise 90 kmol/saattir. Ayrıca asetonun iki faz arasındaki dağılımı $y_A = 2.53 x_A$ dır.

Acaba kaç teorik kademeye ihtiyacımız vardır ki bu asetonun % 90'ını havadan suya geçirebileyim?

Çözüm:

$$y_{N+1} = 0.01 \quad x_{A_0} = 0 \quad V_{N+1} = 30 \text{ kmol/saat} \quad L_0 = 90 \text{ kmol/saat}$$

Giren aseton:

$$y_{N+1} V_{N+1} = 0.01(30) = 0.3 \text{ kmol/saat}$$

Giren hava:

$$(1 - y_{A_{N+1}}) V_{N+1} = (1 - 0.01)30 = 29.7 \text{ kmol/saat}$$

$$V_1 \text{ deki aseton: } 0.1 (0.3) = 0.03 \text{ kmol/saat}$$

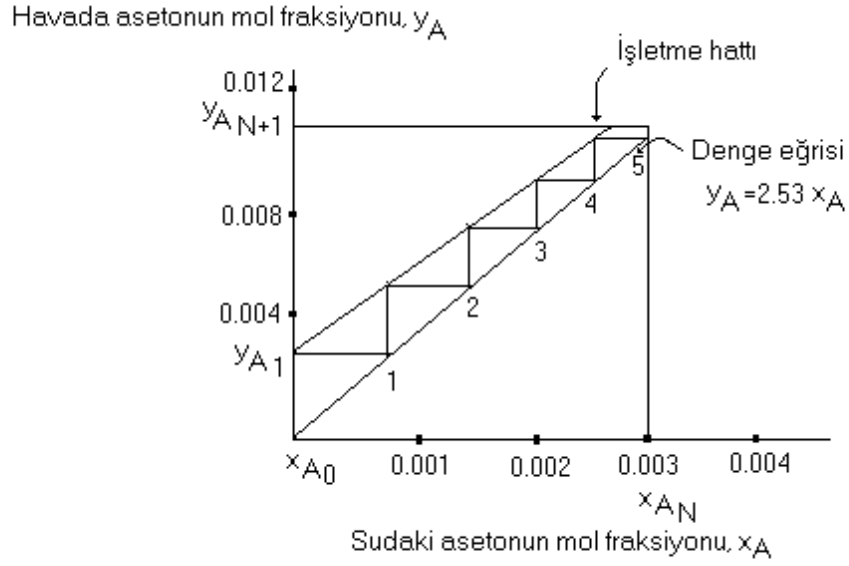
$$L_N \text{ deki aseton: } (0.9) (0.3) = 0.27 \text{ kmol/saat}$$

$$V_1 = 29.7 + 0.03 = 29.73 \text{ kmol/saat}$$

$$L_N = 90 + 0.27 = 90.27$$

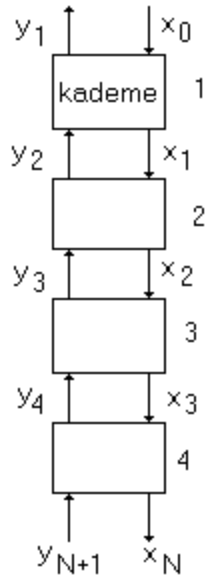
$$x_{A_N} = \frac{0.27}{90.27} = 0.003$$

Çözüm grafiksel olarak Şekil 8.13’ de gösterilmiştir.



Şekil 8.13

Böylece gerekli olan teorik kademe sayısı 5.2 bulunur. Pratikte tabiatıyla bu 6 kademe kabul edilir ve absorpsiyon kolonu ona göre tasarlanır. Şekil 8.14 kademelere ait akış diyagramını göstermektedir.



Şekil 8.14

Özet

Bu bölümde, önemli bazı proses birimlerinin (kanatçıklar, buharlaştırıcılar, reaktörler) modellenmesi yapıldıktan sonra tek-kademeli ve çok-kademeli denge prosesleri (distilasyon, ekstraksiyon, absorpsiyon, vd) için modelleme yapılmış ve örnekler verilmiştir.

PROBLEMLER

- 1) Şeker, suda çözünürken t dakika sonra çözünmeden kalan A miktarı $dA/dt = -kA$ ($k>0$) diferansiyel denklemi ile verilmiştir.
 - a) Eğer şekerin %25'i 1 (bir) dakikada çözünürse, şekerin yarısı ne kadar sürede çözünür?
 - b) Aynı problemi $dA/dt = -kA^2$ hız ifadesi için çözünüz
- 2) Bir nükleer güç santralindeki bir kaza, çevrede, doğal olarak azalan radyoaktif materyal ile kirlenmiş bir alan bıraktı. Radyoaktif materyalin başlangıç miktarı şimdi 15 **su** (güvenlik birimi) ve 5 ay sonraki miktarı 10 **su**'dur.
 - a) t ay sonra geri kalan radyoaktif materyalin $A(t)$ miktarını (**su** birimi olarak) veren bir formül çıkarınız.
 - b) 8 ay sonra ne kadar radyoaktif materyal kalacaktır?
 - c) $A = 1$ **su** ise insanlar o alana güvenle dönebilirler. Buna göre insanların o alana güvenle dönebilmeleri için kazadan sonra kaç ay geçmelidir?
- 3) Öğleden önce cinayete kurban giden bir şahsın cesedi, 70°F'lık sabit sıcaklıkta tutulan bir odada bulunuyor. Öğleyin (12:00) vücudun sıcaklığı 80°F ve öğleden sonra saat 13:00'de 75°F dir. Ölüm zamanında vücudun sıcaklığının 98.6°F olduğunu ve onun Newton'un soğuma yasasına göre soğuduğunu kabul edersek ölüm zamanı nedir?

- 4) İnce uzun bir çubuk (kanatçık = *fin*) için ısı transfer denkliği aşağıdaki şekilde yazılabilir

$$\frac{d^2T}{dx^2} + h(T_a - T) = 0$$

burada,

T : sıcaklık

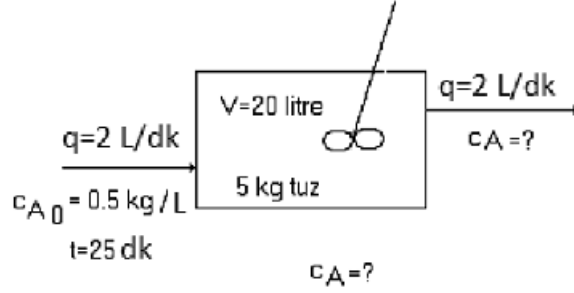
x : çubuk boyunca uzaklık

h : ısı transfer katsayısı

T_a : çevrenin sıcaklığı

- a) $T(x=0) = 55$, $T(x=20) = 250$ ise özel çözümü bulunuz
 - b) 10 metrelik bir çubuk ve $T_a = 20$ °C, $T(x=0) = 40$ °C, $T(x=10) = 200$ °C, $h = 0.02$ için, $x = 2, 4, 6, 8$ noktalarındaki sıcaklık değerlerini bulunuz.
- 5) Aşağıda tanımlanmış olayın matematiksel modelini diferansiyel denklem şeklinde yazınız.
 - a) **P** nüfuslu bir şehirde, bir söylentiyi işiten bireylerin **N** sayısının zamana oranı henüz söylentiye işitmeyenlerin sayısı ile orantılıdır.
 - b) **P** nüfuslu bir şehirde, bir bulaşıcı hastalığa yakalanmış bireylerin **N** sayısının zamana oranı, hastalığa yakalanmışlarla hastalığa yakalanmamış olanların sayısının çarpımı ile orantılıdır

- 6) 20 litre hacminde olan bir tankta başlangıçta 5 kg tuz bulunmaktadır. Ağzına kadar tuz çözeltisi ile dolu olan bu tanka dakikada debisi 2 litre olan ve konsantrasyonu 0.5 kg/L çözelti devamlı olarak verilmektedir. Tankı terkeden sıvının da debisi yine dakikada 2 litre olup, tank iyice karıştırılmaktadır. Buna göre 25 dakika sonra tanktaki tuzun konsantrasyonunu hesaplayınız (Şekil 8.15).



Şekil 8.15

BÖLÜM 9 - MİKROSKOBİK VE MAKROSKOBİK DENKLİKLER

Amaç

Bu bölümün bitiminde,

- Mikroskopik denklüklerde süreklilik denkleminin çıkarılışı
- Makroskopik denklüklerden
 - Süreklilik denklemleri (total kütle)
 - Komponent denklükleri
 - Enerji denklüğü

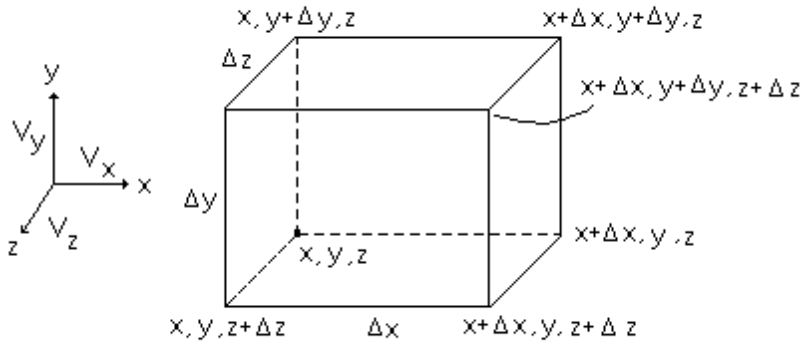
konularının nasıl modellendiğini öğrenmiş olacağız.

9.1 Mikroskopik Denklükler

Burada türeteceğimiz denklemler, aynı zamanda küçük değişikliklerle hem akışkanlar mekaniğinde **momentum transferinde** hem de **ısı transferinde** enerji denklüklerinde kullanılır. Böylece de taşınım olayları arasında (*momentum, ısı, kütle*) ne kadar büyük **analoji** (*benzerlik*) olduğunu da vurgulamış oluyoruz.

9.1.1 Süreklilik Denklemleri

Bir akışkan sistemi içinde Şekil 9.1 de görüldüğü gibi çok küçük bir *kontrol hacmi* seçelim,



Şekil 9.1 Mikroskopik hacimde kütle denklüğü

Şekilde görüldüğü gibi, kontrol hacminin boyutları Δx , Δy ve Δz olsun

Total hacim: $\Delta x \Delta y \Delta z$

Total kütle: $\rho \Delta x \Delta y \Delta z$

Kontrol hacmi içinde kütlenin değişme hızı: $\frac{\partial}{\partial t}(\rho \Delta x \Delta y \Delta z)$

→

V hızı vektör olup v_x , v_y ve v_z bileşenlerinden (komponentlerinden) meydana gelmiştir.

x - yüzünden giren kütlenin debisi: $\left| \rho v_x \Delta y \Delta z \right|_x$

y - yüzünden giren kütlenin debisi: $\left| \rho v_y \Delta x \Delta z \right|_y$

z - yüzünden giren kütlenin debisi: $\left| \rho v_z \Delta x \Delta y \right|_z$

x - yüzünden çıkan: $\left| \rho v_x \Delta y \Delta z \right|_{x+\Delta x}$

y - yüzünden çıkan: $\left| \rho v_y \Delta x \Delta z \right|_{y+\Delta y}$

z - yüzünden çıkan: $\left| \rho v_z \Delta x \Delta y \right|_{z+\Delta z}$

Kütlenin korunumu:

$$\frac{\partial \rho}{\partial t} (\Delta x \Delta y \Delta z) = \left| \rho v_x \Delta y \Delta z \right|_x + \left| \rho v_y \Delta x \Delta z \right|_y + \left| \rho v_z \Delta x \Delta y \right|_z - \left(\left| \rho v_x \Delta y \Delta z \right|_{x+\Delta x} + \left| \rho v_y \Delta x \Delta z \right|_{y+\Delta y} + \left| \rho v_z \Delta x \Delta y \right|_{z+\Delta z} \right)$$

Her iki tarafı $\Delta x \Delta y \Delta z$ ' ye bösek ve bu boyutları sıfıra götürsek.

Örneğin,

$$\lim_{\substack{\Delta x \rightarrow 0 \\ \Delta y \rightarrow 0 \\ \Delta z \rightarrow 0}} \frac{\left| \rho v_x \right|_{x+\Delta x} - \left| \rho v_x \right|_x}{\Delta x} = - \frac{\partial \rho v_x}{\partial x}$$

Diğerleri için de aynı işlemi yaparsak

$$\frac{\partial \rho}{\partial t} = - \frac{\partial \rho v_x}{\partial x} - \frac{\partial \rho v_y}{\partial y} - \frac{\partial \rho v_z}{\partial z}$$

Yatışkın halde süreklilik denklemini elde etmek için

$$\frac{\partial \rho v_x}{\partial x} = \rho \frac{\partial v_x}{\partial x} + v_x \frac{\partial \rho}{\partial x} \quad \text{yazılabilir.}$$

Diğer terimler içinde aynı işlemi yaparsak

$$\frac{\partial \rho}{\partial t} = v_x \frac{\partial \rho}{\partial x} + v_y \frac{\partial \rho}{\partial y} + v_z \frac{\partial \rho}{\partial z} = -\rho \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \right)$$

$$\frac{D\rho}{Dt} = -\rho(\nabla \cdot \mathbf{v})$$

Burada, $\frac{D\rho}{Dt}$ = Büyük türev (substantial derivative)

$\nabla \cdot \mathbf{v}$ = Hız gradienti olarak tarif edilmektedir.

Vektör Notasyonları

Kartezyen sistemde hız

$$\vec{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$$

(burada $\mathbf{i}$, $\mathbf{j}$ ve $\mathbf{k}$ birim vektörler)

$$\text{grad} \xi = \nabla \xi = \frac{\partial \xi}{\partial x} \mathbf{i} + \frac{\partial \xi}{\partial y} \mathbf{j} + \frac{\partial \xi}{\partial z} \mathbf{k}$$

$$\mathbf{i} \cdot \mathbf{i} = \mathbf{j} \cdot \mathbf{j} = \mathbf{k} \cdot \mathbf{k} = 1$$

$$\mathbf{i} \cdot \mathbf{j} = \mathbf{i} \cdot \mathbf{k} = \mathbf{j} \cdot \mathbf{k} = 0 \quad \nabla = \text{del}$$

Örneğin,

$$\vec{v} \cdot \nabla \xi = v_x \frac{\partial \xi}{\partial x} + v_y \frac{\partial \xi}{\partial y} + v_z \frac{\partial \xi}{\partial z} \quad \text{olur.}$$

$$\nabla \mathbf{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}$$

Süreklilik denklemi vektör notasyonunda aşağıdaki şekilde yazılabilir:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \rho \cdot \vec{v}$$

Eğer ρ zamanla değişmeyip sabit kalıyorsa

$$\frac{\partial \rho}{\partial t} = 0 \rightarrow -\rho \nabla \cdot \vec{v} = 0$$

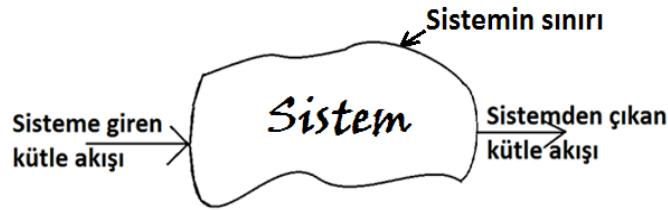
veya

$$\nabla \cdot \vec{v} = 0$$

Genelde süreklilik için kullanılan bu denklem, daha önce de söylediğimiz gibi, *aynı yaklaşımla*, ısı transferinde *enerji korunumu yasası* ve akışkanlar mekaniğinde *momentum korunumu yasasını* da türetebiliriz (Bunlar, ilgili derslerde daha önce türetilmiştir!).

9.2 Makroskobik Denklikler

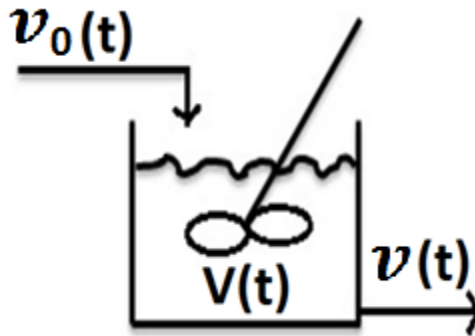
9.2.1 Toplam Kütle Denkliği (Süreklilik Denklemi)



Şekil 9.2 Süreklilik denkleminin türetimine ait sistem

giren - çıkan = biriken (hepsi birim zamanda, Şekil 9.2)

Aşağıdaki Şekil 9.3 'ü göz önüne alalım.



Şekil 9.3

v = hacimsel debi (m^3/dk , ft^3/sn gibi)

ρ = yoğunluk (kg/m^3 , lb/ft^3 gibi)

$V(t)$ = Tankın "t" anındaki hacmi

Yukarıdaki prensibe göre

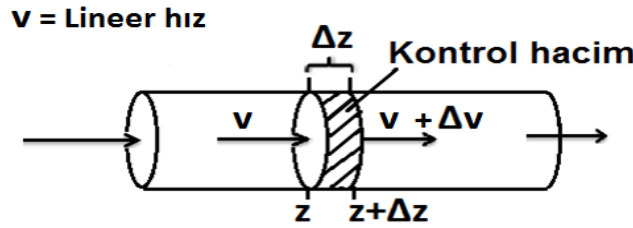
$$v_0 \rho_0 - v \rho = \frac{d(\rho V)}{dt}; \quad \text{Burada birikim: } \frac{d(\rho V)}{dt}$$

Eğer yoğunluk değişimi yoksa $v_0 - v = \frac{dV}{dt}$ olacaktır.

Silindirik bir tüpte aynı prensibi kullanırsak burada iki değişken söz konusu:

a) t = zaman, b) z = borunun uzunluğu.

Süreklilik denklemini uygulamak için küçük bir dilim seçip ona göre denkliği yazmak gerekir. Sistem, *dağınık parametre sistemidir* (Şekil 9.4).



Şekil 9.4 Bir borudaki süreklilik denkleminin türetilmesi

$$\text{giren} = vA\rho|_z \cdot \Delta t, \quad \text{çıkan} = vA\rho|_{z+\Delta z} \cdot \Delta t$$

v =hız, A =borunun kesiti olup sabittir.

Biriken= $d/dt[(A\rho\Delta z)]\Delta t$ burada, Δt geçen küçük zaman dilimi

$$\begin{aligned} \Delta t[vA\rho|_z - vA\rho|_{z+\Delta z}] &= \frac{\partial(A\rho\Delta z)\Delta t}{\partial t} \Rightarrow \lim_{\Delta z \rightarrow 0} \frac{vA\rho|_z - vA\rho|_{z+\Delta z}}{\Delta z} = \frac{\partial(A\rho)}{\partial t} \\ &\Rightarrow -\frac{\partial(v\rho)}{\partial z} = \frac{\partial\rho}{\partial t} \Rightarrow \\ &\frac{\partial\rho}{\partial t} + \frac{\partial(v\rho)}{\partial z} = 0 \end{aligned}$$

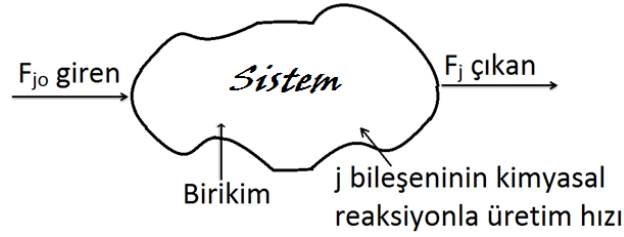
Bu son çıkarılan denklem *süreklilik denkleminin diferansiyel ifadesidir*. Eğer **yatışkın hal** (*steady-state*) söz konusu ise, yani zamanla bir değişim yoksa, o zaman $\frac{\partial\rho}{\partial t}$ terimi sıfır olur ki neticede $\frac{d(v\rho)}{dz} = 0$ şekline indirgenmiş olur. Eğer üç boyutlu (x, y, z yönlerini) dikkate alırsak o zaman $\nabla(v\rho) = 0$ olur ki genelde **bu vektör notasyonu** taşınım olayları dersinde karşınıza çıkacaktır[47].

Burada

$$\nabla = \frac{\partial(\quad)}{\partial x} i + \frac{\partial(\quad)}{\partial y} j + \frac{\partial(\quad)}{\partial z} k \text{ olup buna Laplace operatörü denilmektedir.}$$

9.2.2 Komponent (Bileşen) Denklikleri

Kimya mühendisliğinde sık sık karşımıza çıkan, bileşen (komponent) denklikleridir. Herhangi bir j bileşenini göz önüne alalım.



Şekil 9.5

$$\boxed{\text{Giren}-\text{çıkan}+\text{üretilen}=\text{biriken}}$$

Bu denkleme lütfen dikkat edelim. Buradaki üretim hızı mutlaka *pozitif* olacak diye bir kayıt yoktur. Eğer tüketim varsa o zaman o terim denkleme *negatif* işaretle girecektir. Aynı şekilde **biriken** pozitif olabileceği gibi negatif de olabilir!

$$v_0 C_{j_0} - v C_j + r_j V = \frac{d(v C_j)}{dt} \quad (j \text{ komponent denkliği})$$

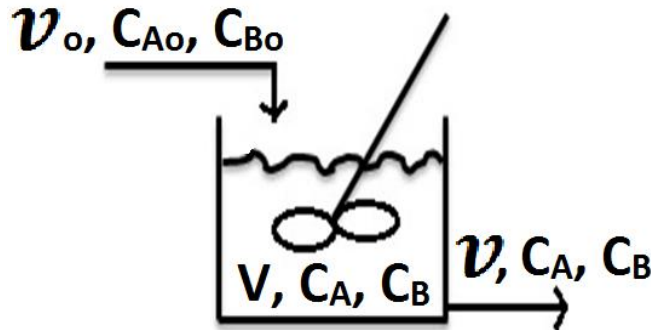
Burada,

$$v_0 C_{j_0} = F_j \quad v C_j = F_j$$

Bu durumu aşağıda iki örnekle açıklayalım.

Örnek 9.1 Bileşen denklikleri

Çok iyi karıştırılmış kesikli bir tank reaktöründe $A \xrightarrow{k} B$ birinci mertebeden bir reaksiyon oluşmaktadır. Şekil 10.5 e bakarak A ve B bileşenleri için gerekli bileşen denkliklerini yazınız.



Şekil 9.6

Çözüm:

Reaksiyon hızı $-r_A = kC_A$

"A"nın tank içindeki birikim hızı $= \frac{d(VC_A)}{dt}$

"B"nin tank içindeki birikim hızı $= \frac{d(VC_B)}{dt}$

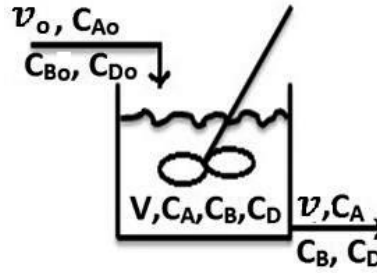
A -denkliği : $v_0C_{A_0} - vC_A - Vkc_A = \frac{d(VC_A)}{dt}$

B -denkliği : $v_0C_{B_0} - vC_B + Vkc_A = \frac{d(VC_B)}{dt}$

Not: Eğer "j" reaktant ise hız $[-r_j]$ dir.
Eğer "j" ürün ise hız $[+r_j]$ dir.

Örnek 9.2 Ardışık reaksiyonlar

$A \xrightarrow{k_1} B \xrightarrow{k_2} D$ reaksiyonu tam karıştırmalı sürekli reaktörde (TKSR) gerçekleştirilmektedir. Reaksiyonlar birinci mertebeden ise bileşenler için ayrı ayrı denklemleri yazınız (Şekil 9.7)



Şekil 9.7

Çözüm:

A: $v_0C_{A_0} - vC_A - k_1C_A = \frac{d(VC_A)}{dt}$

B: $v_0C_{B_0} - vC_B + k_1C_A - k_2C_B = \frac{d(VC_B)}{dt}$

D: $v_0C_{D_0} - vC_D + k_2C_B = \frac{d(VC_D)}{dt}$

9.2.3 Enerji Denkliği



Şekil 9.8 Enerji denkleminin çıkarılışı

Yukarıdaki sistem için enerji denklemini yazarsak

$$v_0 \rho_0 (U_0 + K_0 + \Phi_0) - v \rho (U + K + \Phi) + (Q_G + Q) - (W + v \rho - v_0 \rho_0) = \frac{d}{dt} [(U + K + \Phi) V \rho]$$

U = İç enerji

K = Kinetik enerji

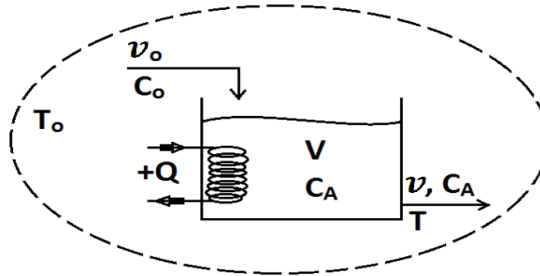
Φ = Potansiyel enerji

W = Mil (Shaft) işi

P = Sistem basıncı

P_0 = Sisteme giren akışın basıncı

Aşağıdaki Şekil 9.9'u dikkate alalım.



Şekil 9.9 Enerji denkleği sistemi

$A \xrightarrow{k} B$ birinci mertebeden bir reaksiyonun gerçekleştiğini düşünelim.

ΔH = reaksiyon ısısı / mol A; $Q_G = -\Delta H$ dır.

Bu örnekte $W = 0$, $K = \Phi = 0$ kabul edilebilir. O zaman

$$v_0 \rho_0 U_0 - v \rho U + Q_G + Q - v \rho \frac{P}{\rho} + v_0 \rho_0 \frac{P_0}{\rho_0} = \frac{d(\rho V U)}{dt}$$

$$\Rightarrow v_0 \rho_0 (U_0 + P_0 V^*) - v \rho (U + P V^*) + Q_G + Q = \frac{d(\rho V U)}{dt}$$

$$\frac{1}{\rho} = \bar{V} \text{ (spesifik hacim); } \quad h = U + P \bar{V} = \text{entalpi olduğuna göre}$$

$$v_0 \rho_0 h_0 - v \rho h + Q - \Delta H V k C_A = \frac{d(\rho V h)}{dt}$$

olur.

$$C_p = \left(\frac{\partial h}{\partial T} \right)_p, \quad C_v = \left(\frac{\partial U}{\partial T} \right)_v, \quad h = C_p T \text{ dir. Sıvı ve katılar için } C_p \approx C_v \text{ dir.}$$

$$\rho C_p \frac{d(VT)}{dt} = \rho C_p (v_0 T_0 - v T) + Q - \Delta H V k C_A$$

Eğer sistemin sıcaklığı ve hacmi sabit ise denklemin sol tarafı sıfır olur. Bu model denklemlerin uygulama alanları, özellikle *kimyasal reaksiyon mühendisliği* dir [40-45].

Özet

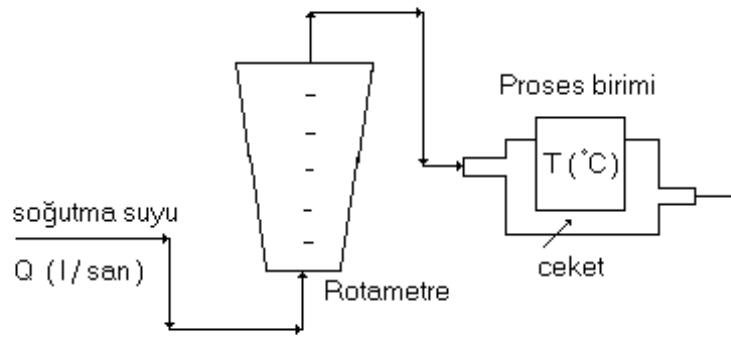
Bu bölümde, mühendislikte matematiğin en yaygın olarak kullanıldığı alanların mikroskobik denklemler olduğu belirtildikten sonra mikroskobik açıdan süreklilik denklemi türetilmiştir. Aynı yolla momentum ve enerji denklemlerinin de türetilebileceği belirtildi. Daha sonra da makroskobik (*global*) denklemler toplam kütle, komponentler ve enerji için türetilmiştir.

PROBLEMLER

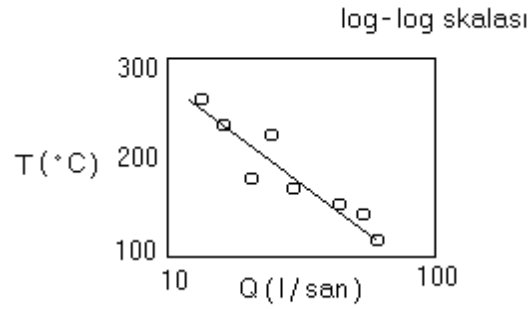
- 1) 1000000 nüfuslu bir şehirde birgün içme suyundaki *phenylethylamine* hakkında şüpheli bir söylenti yayılmaya başladı. Bir hafta içinde 100000 kişi bu söylentiye duydu. Söylentiye işitenlerin sayısının artış hızının o söylentiye işitmeyenlerin sayısı ile orantılı olduğunu varsayınız. Şehrin nüfusunun yarsının bu söylentiye duyması ne kadar süre gerektirir?
- 2) 25 yaşındaki bir bayan yıllık 30000 dolar ile başlayan bir maaşla mühendis oluyor. Onun $S(t)$ maaşı t yıl sonra $S(t) = 30e^{t/20}$ bin dolara üstel olarak artıyor. Bu arada maaşının %10' u sürekli olarak bir emeklilik hesabına yatırılıyor ve bunun yıllık %5' lik oranla sürekli olarak faizi birikiyor.
 - a) t yıl sonra emeklilik hesabında biriken $A(t)$ miktarı için gerekli diferansiyel denklemi elde ediniz.
 - b) $A(40)$, yani onun 65 yaşında emekli olduğunda alacağı miktarı bulunuz.
- 3) Bir proses biriminde sıcaklığın kontrolü için ceketten soğuk su Q (litre / sn) debisi ile geçirilmektedir (Şekil 9.10a). Prosesin sıcaklığı (T) ile soğuk su debisi (Q) arasındaki ilişki son

derece karmaşık olup **ampirik** bir formül bulma yoluna gidilmiştir. T ile Q arasındaki ilişki Kartezyen (dikdörtgen) koordinatlarında semi-log ve lineer sistemde çizildiğinde birer eğri, fakat log-log kâğıdına çizildiğinde Şekil 9.10b’ de görüldüğü gibi bir lineerlik vermektedir. En küçük kareler metoduyla (EKM) çizilen doğrudaki ($Q_1=25$, $T_1=210$) ve ($Q_2=40$, $T_2=120$) bulunmuştur. Buna göre;

- Q ile T arasındaki ampirik ilişkiyi bulunuz.
- "a" şıkında bulduğunuz ilişkiye dayanarak, sıcaklık 175 °C olduğunda gerekli Q’ yu bulunuz.



Şekil 9.10a



Şekil 9.10b

BÖLÜM 10 - KİMYA MÜHENDİSLİĞİNDEN SEÇİLMİŞ ÇÖZÜMLÜ ÖRNEKLER

Amaç

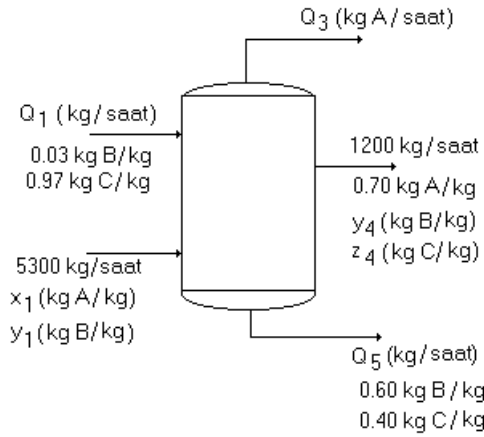
Kimya mühendisliği alanında **İngilizce** çeşitli kaynaklardan seçilen problemler ve onların çözümleri ile ilgili bilgi ve beceriyi geliştirmek

Bu bölümde daha önceki bölümlerde kapsadığımız konulara ilişkin çeşitli problemler **literatürden** (kaynaklara bakınız) seçilmiş ve çözümleri ile burada verilmiştir. Burada amaç, daha önceden öğrenilen kavramların uygulanabilmesidir.

Eğer burada bir zorlukla karşılanırsa o zaman daha önceki bölümleri tekrar gözden geçirmek gerekebilir.

Örnek 10.1

Aşağıdaki şekilde çalışan bir distilasyon kolonu verilmiş olsun. Bu akış diyagramına bakarak:



Şekil 10.1 Örnek 10.1 e ait akış diyagramı

- Kaç tane **bağımsız** madde denkliği yazılabilir?
- Kaç tane bilinmeyen akış hızı veya mol fraksiyonu verilmesi lazım ki diğer bilinmeyenler hesap edilebilsin?
- Q_1 ve x_1 verildiğini kabul edersek öyle denklemler sistemi yazınız ki sırasıyla her denklem çözüldüğünde *sadece bir bilinmeyen* içersin.

Çözüm

- Bilinmeyenlerin sayısı= 7

(Q_1 , x_1 , y_1 , Q_3 , y_4 , z_4 ve Q_5)

3 komponent (A, B, C) olduğu için 3 bağımsız denklik yazılabilir. İki tane de fraksiyon denklemi vardır:

$$x_1 + y_1 = 1$$

$$y_4 + z_4 = 1 - 0.7$$

b) Böylece 7 bilinmeyene karşı 5 denklem var. **İki bilinmeyen** daha verilmesi gerekir ki diğer bilinmeyenler bulunabilsin.

c) Örneğin Q_1 ve x_1 verilmiş olsun.

$$y_1 = 1 - x_1$$

A -denkliği : $5300 \cdot x_1 = 1200 \cdot 0.7 + Q_3 \rightarrow Q_3$ bulunur.

Total denklik:

$$Q_1 + 5300 = Q_3 + 1200 + Q_5 \rightarrow Q_5 \text{ bulunur.}$$

B -denkliği: $Q_1 \cdot 0.03 + 5300 \cdot x_1 = 1200 \cdot y_4 + 0.6 \cdot Q_5 \rightarrow y_4$ bulunur.

$$z_4 + y_4 = 1 - 0.7 \rightarrow z_4 \text{ bulunur.}$$

Bu örnekten görüldüğü gibi **denklemlerin yazım sırası, çözümünü ne kadar kolaylaştıracağı gibi, yanlış sıralama da çözümünü o kadar zorlaştırır.**

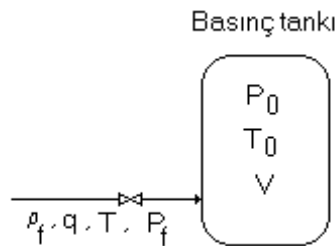
Diyelim ki 7 bilinmeyen içeren 7 denklem verilmiş olsun. Bunu, denklemlerin sırası ne olursa olsun MATLAB paket programları ile kolayca çözebileceğimizi daha önceki bölümlerde belirtmiştik.

Örnek 10.2

Hacmi V , başlangıç basıncı ve sıcaklığı P_0 ve T_0 olan bir çelik kaba, sıcaklığı T , yoğunluğu ρ , basıncı P_f ve debisi q olan bir gaz verilmektedir (Şekil 10.2)

a) Kabin içindeki basınç ile zaman arasındaki *matematiksel bağıntıyı* bulunuz.

b) Kap, P_t basıncına kadar dayanıklı ise maksimum ne kadar süre ile içeriye gaz vermek gerekir ki bir **patlama olmasın?**



Şekil 10.2 Örnek 10.2'ye ait akış diyagramı

Çözüm

Bu, yarı-kesikli (semi-batch) bir sistemdir. Kabın belli bir dayanıklılığı var. Vanayı açınca kabın basıncı nasıl değişir?

Zaman ile bu basınç (P) arasındaki bağıntıyı bulmak için yine kütle korunumu prensibini uygulamak gerekir. Sistemimizi basınç tankı olarak seçersek:

$$a) \frac{d(\rho V)}{dt} = \rho_f \cdot q \quad ; \quad V = \text{sabit olduğundan}$$

$$V \frac{d\rho}{dt} = \rho_f \cdot q$$

$$\int_{\rho_0}^{\rho} d\rho = \rho_f \frac{q}{V} \int_0^t dt \rightarrow \rho - \rho_0 = \rho_f \frac{q}{V} t$$

veya

$$\rho = \rho_0 + \rho_f \frac{q}{V} t$$

$$PV = nRT \Rightarrow P = \left(\frac{n}{V} \right) RT, \quad \text{Eğer aynı cins gaz kaba geliyorsa}$$

$$P = \frac{(M \cdot W)n}{V} R'T$$

$$P = \rho \cdot R' \cdot T, \quad \text{burada } R' = \frac{R}{M \cdot W}, \quad M.W = \text{Gazın mol ağırlığı}$$

$$\rho = \frac{P}{R'T}$$

olduğuna göre

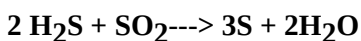
$$P = P_0 + \rho_f \cdot \frac{q}{V} \cdot R' \cdot T \cdot t \quad \text{burada } q = \text{debi}$$

T= Kabın içindeki sıcaklık. Gerçek gazlar durumunda kabın sıcaklığı da artacaktır (adiyabatik sıkışma gibi!)

$$b) t = \frac{(P_t - P_0)V}{\rho_f \cdot q \cdot R' \cdot T} \quad \text{olur.} \quad P_t < P_f \text{ dir.}$$

Örnek 10.3

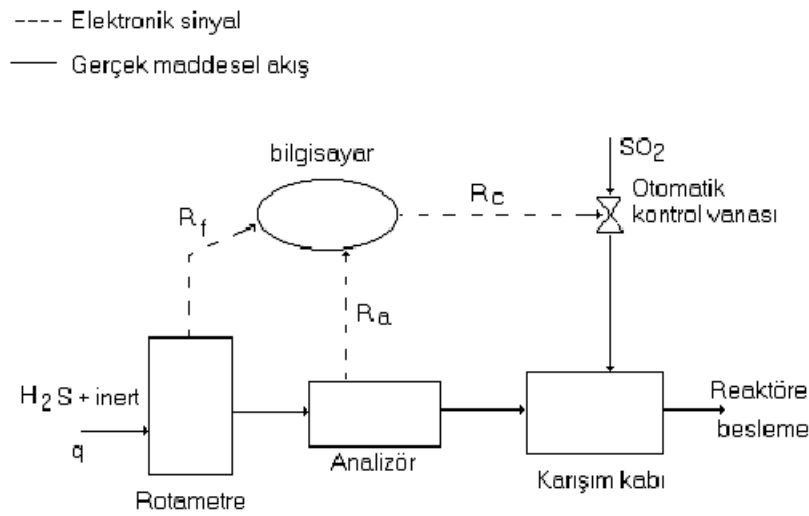
H₂S ve inert gazlar içeren bir akım ve saf SO₂ içeren ikinci bir akım



reaksiyonu ile S ün geri kazanıldığı gibi bir reaktöre besleniyor. H_2S ' in SO_2 ye oranının stokiyometrik olması için besleme hızı bilgisayar ile ayarlanıyor (Şekil 10.3).

a) Birinci akım % 85 mol H_2S içeriyor ve reaktöre 400 kmol/saat debi ile giriyor ise böyle bir şemada SO_2 'nin molar akış debisi ne olmalıdır?

b) Bilgisayar kullanarak SO_2 akış debisi otomatik olarak ayarlanmak isteniyor. H_2S akımının debisi doğrudan doğruya q molar akış hızı ile orantılı R_f sinyalini ileten elektronik bir akımmetre ile ölçülecektir. 100 kmol/saat akış hızı ile $R_f=20$ milivolttur. Bu proste H_2S in mol kesri R_a sinyalini ileten termal bir kondüktivite dedektörü ile ölçülüyor. Bilgisayar, SO_2 hattındaki akış kontrol vanasına R_c sinyalini iletir. R_c ve SO_2 akış hızı arasındaki ilişki, 100 kmol molar akış debisine karşılık $R_c=50$ milivolt olup doğru orantılıdır. H_2S in mol fraksiyonu x ile R_c arasındaki bağıntıyı bulunuz



Şekil 10.3 Örnek 10.3'e ait akış diyagramı

Çözüm

a) Boyutsal analiz denklemini kurarsak

$$\frac{400 \text{ kmol}}{\text{saat}} \left| \frac{0.85 \text{ kmol } H_2S}{1 \text{ kmol}} \right| \left| \frac{1 \text{ kmol } SO_2}{2 \text{ kmol } H_2S} \right| = 170 \frac{\text{kmol}}{\text{saat}} SO_2 \text{ gerekir.}$$

b) $x=H_2S$ in mol fraksiyonu ise

$$q = a \cdot R_f$$

$$R_f=20 \text{ mV iken } q=100 \text{ kmol/saat} \rightarrow a=5 \text{ veya } q = 5 \cdot R_f$$

$$n_{SO_2} = b \cdot R_c \rightarrow R_c = 50 \text{ iken, } n_{SO_2} = 100 \Rightarrow b = 2 \text{ veya } n_{SO_2} = 2 \cdot R_c$$

$$n_{SO_2} = 5 \cdot R_f \cdot \frac{x}{2} \Rightarrow 2 \cdot R_c = 5 \cdot R_f \cdot \frac{x}{2} \Rightarrow R_c = 1.25 \cdot R_f \cdot x \text{ bulunur.}$$

Aşağıdaki örneği çözmeden önce açık ve kapalı sistemler için termodinamiğin birinci yasasını yazalım.

$$\text{Biriken} = \text{giren} - \text{çıkan} + \text{üretilen} - \text{tüketilen}$$

$$Mc_v \frac{dT_s}{dt} = mc_p (T_i - T_s) + Q - W \quad (\text{Açık sistem})$$

$$Mc_v \frac{dT_s}{dt} = Q - W \quad (\text{Kapalı sistem})$$

M : Sistemin kütlesi (Toplam kütle)

c_v : Sistemin spesifik ısı kapasitesi

T_s : Sistemin sıcaklığı

m : Giren kütle miktarı

T_i : Giren kütle sıcaklığı

T_s : Sistemin sıcaklığı

Q : Sisteme verilen ısı

W : Sistemin *yaptığı* iş

Kapalı sistemde ısı alışverişi var, fakat kütle girişi veya çıkışı yoktur. Her iki duruma birer örnek verelim.

Örnek 10.4

Çok iyi karıştırılmış kazan tipi bir reaktör elektrik ile ısıtılmaktadır. Reaktantlar reaktöre doldurulduktan sonra 25 °C dan 250 °C'ye kadar ısıtılıyor. Aşağıdaki verileri kullanarak ısıtma için gerekli zamanı bulunuz?

Reaktantlar: kütle=1.5 kg $C_v = 0.9 \text{ cal/g.}^\circ\text{C}$

Isıtma hızı $Q=500 \text{ Watt (J / s)}$; Reaktör kütlesi=3 kg; $C_v = 0.12 \text{ cal/g.}^\circ\text{C}$

Çözüm

Kapalı bir sistem olup mekanik iş de söz konusu olmadığından

$$Mc_v \frac{dT_s}{dt} = Q \quad (\text{kapalı sistem})$$

$$t=0 \quad T_s=25^\circ\text{C}$$

$$t=t_f \quad T_s=250^\circ\text{C}$$

$$\begin{aligned} Mc_v &= \sum M_i c_{v_i} \\ &= (1500)0.9 + (3000)0.12 = 1740 \text{ cal / }^\circ\text{C} \\ &= 7150 \text{ joule / }^\circ\text{C} \end{aligned}$$

$$\frac{Mc_v \int_{25}^{250} dT_s}{Q} = \int_0^{t_f} dt \Rightarrow t_f = \frac{225Mc_v}{Q}$$

$$t_f = \frac{225^\circ\text{C}(7150 \text{ J}/^\circ\text{C})}{500 \text{ J}/\text{sn}} = 3270 \text{ sn} = 53.7 \text{ dk}$$

Not:

a) katı ve sıvılarda $c_v \approx c_p$ olduğunu, halbuki gazlarda $c_p - c_v = R$ olduğunu da burada hatırlatalım.

b) Sistem dışı doğru iş yapıyorsa **W pozitif**, sistem üzerine iş yapılıyorsa **W negatif** değer alır.

(Bazı kitaplarda bunun ters işaretlisine de rastlamak mümkün!!).

Örnek 10.5

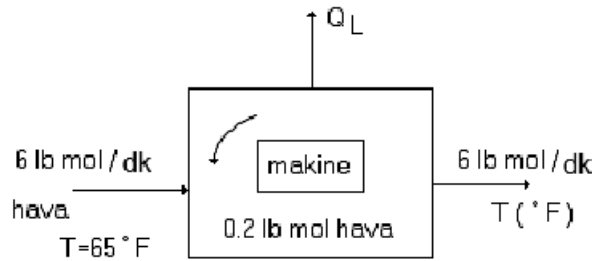
Çalışmakta olan bir makine 8530 Btu/dk (1 Btu=250 cal) ısı üretmektedir. Bu makine hava ile soğutuluyor. Makine içinde hava çok iyi sirküle edilerek, çıkan havanın sıcaklığı sistemin içindeki havanın sıcaklığına eşittir. Hava 65 °F'da makineye 6 lbmol/dk debi ile verilmektedir. Çıkan hava da yine 6 lbmol / dk' dır. Makinenin içinde daima 0.2 lbmol hava sirküle ediliyor (Şekil 10.5) Buna göre eğer makine 65 °F de iken çalışmaya başlarsa,

a) Yatışkın halde ulaşacağı sıcaklığı hesaplayınız. $C_v(\text{hava için})=5 \text{ Btu} / \text{lbmol } ^\circ\text{F}$

b) **Zaman- sıcaklık** arasındaki ilişkiyi gösterir *diferansiyel denklemi* türetiniz.

c) Ne kadar zamanda yatışkın hale ulaşabiliriz?

Kaybolan ısı $Q_L=33(T_s-65)$ ifadesi ile veriliyor.



Şekil 10.5 Örnek 10.5'ye ait sistem

Çözüm

a) İçerdeki sıcaklık artarsa kayıp olan ısı miktarı da artar. Bu zıt vektörler arasında denge

kurulacaktır. Sistem açık bir sistemdir. Sistem sınırları değişmediği için mekanik iş de yoktur.

$$Mc_v \frac{dT_s}{dt} = mc_p (T_i - T_s) + Q$$

$$Q = Q_{\text{üretilen}} - Q_{\text{kaybolan}}$$

Yatışkın hale ulaşınca $\frac{dT_s}{dt} = 0$ olur.

$$mc_p(T_s - 65) = 8530 - 33(T_s - 65) ; \quad m = 6 \text{ lb mol / dk}, \quad c_p = 7 \text{ Btu / lb mol.}^\circ\text{F}$$

$$m \cdot c_p = (6 \times 7) = 42 \text{ Btu / }^\circ\text{F}$$

Buradan

$$T_s = 178.8^\circ\text{F} \text{ bulunur.}$$

(Eğer giren havanın debisi artırılırsa yatışkın hal sıcaklığı azalır mı, artar mı? Hesapla gösteriniz!)

b) $Mc_v \frac{dT_s}{dt} = mc_p(T_i - T_s) + Q$ ifadesini çözelim:

$$Mc_v \frac{dT_s}{dt} = mc_p(T_i - T_s) + 8530 - 33(T_s - 65)$$

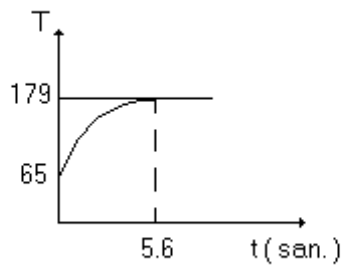
$$M = 0.2 \text{ lbmol}, \quad C_v = 5, \quad mc_p = 42 \text{ olduğuna göre}$$

$$\frac{dT_s}{dt} = -74.9T + 13400 \text{ diferansiyel denklemi elde edilir. Yatışkın halde } \frac{dT_s}{dt} = 0 \text{ alınırsa } T = 178.8^\circ\text{F} \text{ bulunur.}$$

c)

$$\int_{65}^{178.8} \frac{dT}{-74.9T + 13400} = t_f$$

Bu ifadenin MATLAB ile integrasyonundan $t_f = 0.093 \text{ dk}$ veya $t_f = 5.6 \text{ sn.}$ Bu yatışkın hale geliş durumu Şekil 10.6 'da gösterilmiştir.



Şekil 10.6

Örnek 10.6

Kazan tipi kesikli bir reaktörde $A \rightarrow B$ reaksiyonu gerçekleştirilmektedir. Reaksiyon hızı

$$r_A = 0.2 V C_A$$

ile verilmektedir. (V =reaksiyon karışımının hacmi.) Başlangıçtaki A 'nın konsantrasyonu

$C_{A_0} = 0.1 \text{ mol / L}$ dir. A nın % 90'ının reaksiyona girmesi için gerekli zamanı hesaplayınız.

Çözüm:

Biriken=-tüketilen (Kesikli) ; $N_A = V \cdot C_A$ ve V =sabit olduğuna göre

$$\frac{dN_A}{dt} \cdot \frac{1}{V} = -r_A,$$

veya

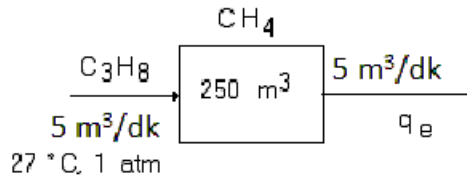
$$\frac{d(V \cdot C_A)}{dt} = -0.2 \cdot V \cdot C_A$$

$$\int_{C_{A_0}}^{C_A} \frac{dC_A}{C_A} = -0.2 \cdot t \rightarrow \ln \frac{C_A}{C_{A_0}} = -0.2 \cdot t$$

$$\ln(0.1) = -0.2 \cdot t \quad t = 11.5 \text{ sn olur.}$$

Örnek 10.7

Saf metanla dolu olan ve hacmi 250 m^3 olan bir yakıt tankına aniden $5 \text{ m}^3 / \text{dk}$ lık debi ile *propan* verilmektedir (*step input*). Tanktan sürekli alınan gaz da yine $5 \text{ m}^3 / \text{dk}$ dır. Tanktaki metan mol fraksiyonunun % 1 e düşmesi için ne kadar süre ile propan verilmelidir. Bütün gazlar 27°C ve 1 atmosfer basınçta bulunmaktadır (Şekil 10.7).



Şekil 10.7 Örnek 10.7'ye ait sistem

Çözüm

Sıcaklık ve basınç sabit olduğuna göre giren ve çıkan gazların mol sayıları eşittir.

$$q_i = q_e = \frac{5 \text{ m}^3}{\text{dak}} \left| \frac{273}{300} \right| \left| \frac{1}{22.4} \right| = 0.203 \text{ kmol / dk}$$

(1 mol gaz, 0 °C' de ve 1 atm basınçta 22.4 litre hacim kaplar.)

Tanktaki gazın mol sayısına "**n**" dersek

$$n=250(273/300)/22.4=10.2 \text{ kmol}$$

Metan bileşeni için gerekli denkliği yazarsak

Biriken= - Çıkan

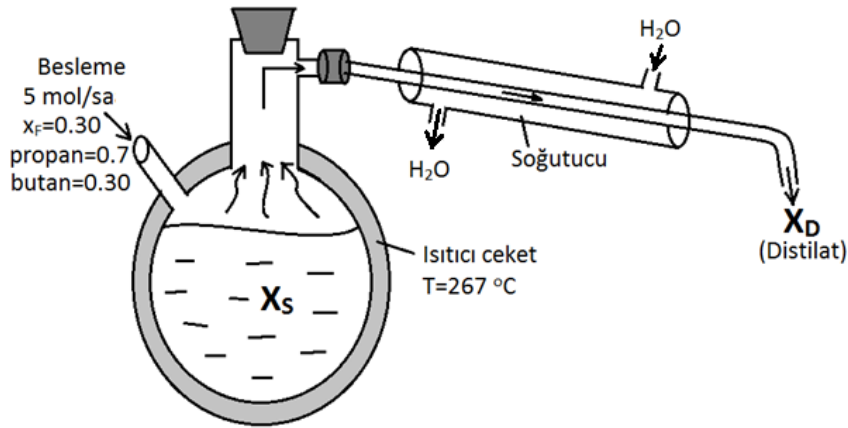
x = metanın tank içindeki mol fraksiyonu olsun.

$$\frac{d(nx)}{dt} = -qx$$

$$10.2 \int_1^{0.01} \frac{dx}{x} = -0.203 \cdot t \rightarrow t = 230 \text{ dk}$$

Örnek 10.8

Başlangıçta 10 mol propan-bütan karışımını içeren bir distilasyon kabında, bütanın mol fraksiyonu 0.3 olup bu kaba gelen besleme hızı 5 mol / saat ve bütanın mol kesri de gene 0.3 dür. Distilasyondan alınan distilatın mol fraksiyonu x_D ile, sistemdeki x_S arasındaki bağıntı $x_D = \frac{x_S}{1 + x_S}$ şeklindedir. Burada x_S distilasyon **kabındaki** bütanın mol fraksiyonudur. x_S in 0.3 den 0.4 e yükselmesi için ne kadar zaman gerekir? Çıkan distilat miktarının da 5 mol / saat olduğu kabul ediliyor (Şekil 10.8)



Şekil 10.8 Örnek 10.8'e ait cihaz

Çözüm

Bütan denkleğini küçük bir zaman aralığı (Δt) için yazarsak

$$\text{Giren} = 5(0.3)\Delta t$$

$$\text{Çıkan} = 5x_D \Delta t$$

$$\text{Biriken} = 10 \cdot x_s \Big|_{t+\Delta t} - 10 \cdot x_s \Big|_t = 10 \cdot \Delta x_s$$

$$10 \Delta x_s = 1.5 \Delta t - 5 \cdot x_D \cdot \Delta t, \text{ her iki taraf } \Delta t \text{ ye bölünür, ve } \Delta t \text{ sıfıra götürülürse}$$

$$\lim_{\Delta t \rightarrow 0} \frac{10 \cdot \Delta x_s}{\Delta t} = 1.5 - 5x_D$$

$$\frac{10 dx_s}{dt} = 1.5 - 5 \cdot x_D$$

veya

$$\frac{dx_s}{dt} = 0.15 - 0.5 \cdot x_D \rightarrow \frac{dx_s}{dt} = 0.15 - 0.5 \frac{x_s}{1+x_s} \text{ diferansiyel denklemi elde edilir.}$$

(x_D , x_s ten büyük olabilir mi? Cevaplayınız).

$$\rightarrow \int_{0.3}^{0.4} \frac{dx_s}{0.15 - 0.5 \frac{x_s}{1+x_s}} = t \quad (\text{Runga-Kutta metodu}) \text{ ile veya analitik integrasyonla}$$

ve yahut MATLAB ile $t=5.85$ saat bulunur.

b) Yatışkın hale varmak için, yani x_s in sabit olması için, değerinin ne olması lazım ve ne kadar zamanda bu değere ulaşır?

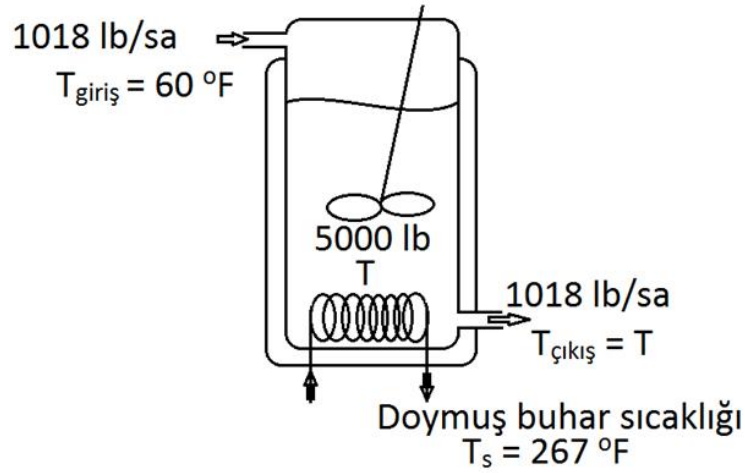
$$\frac{dx_s}{dt} = 0.15 - 0.5 \frac{x_s}{1+x_s}. \text{ Yatışkın halde değişim yok, yani } \frac{dx_s}{dt} \text{ sıfır olur.}$$

O zaman $x_s=0.428$ bulunur.

$$\text{Gerekli zaman da } \int_{0.3}^{0.428} \frac{dx_s}{0.15 - 0.5 \frac{x_s}{1+x_s}} = t \Rightarrow t = 21.7 \text{ saat olarak bulunur.}$$

Örnek 10.9

Bir ısıtıcıda başlangıçta 5000 lb yağ 60 °F da bulunmaktadır. Bu ısıtıcıya saatte 1018 lb yağ verilmekte, 1018 lb ısınmış yağ ısıtıcıyı terk etmektedir. Isı transferinin $Q=h(T_s-T)$ ile verildiğini düşünerek ne kadar sürede çıkan yağın sıcaklığı 90 °F a ulaşır? Motor gücü=1 beygir, verimlilik= 0.75, $h=291$ Btu/ saat °F olup (UA) ya eşittir. Yağın ısı kapasitesi 0.5 Btu / lb°F dır (Şekil 10.9)



Şekil 10.9 Örnek 10.9'a ait cihaz

Çözüm

Herhangi bir andaki yağ miktarı sabit (5000 lb). Burada sadece enerji denkliği yazılacaktır. Verilmesi gereken istim miktarının veya Q ' nün miktarı bilinmelidir.

Küçük bir Δt için; $Q = h(T_s - T)\Delta t = 291(267 - T)\Delta t$

çıkan enerji= $1018 \cdot 0.5(T - 60)$

Gireni yazalım: Termodinamikte referans bir sıcaklık alınır ve hesaplamalar ona göre ifade edilir. $T_{\text{ref}} = 60$ °F alınırsa yağla giren entalpi sıfır olur.

Biriken= $5000 \cdot 0.5(T - 60)_{t+\Delta t} - 5000 \cdot 0.5(T - 60)_t$

$W = 0.75 \text{ hp} \cdot \frac{0.76 \text{ Btu}}{1 \text{ hp} \cdot \text{s}} \left| \frac{3600 \text{ s}}{h} \right| = 1910 \text{ Btu / saat}$

$5000 \cdot 0.5(T - 60)_{t+\Delta t} - 5000 \cdot 0.5(T - 60)_t = 291(267 - T)\Delta t + 1910 \cdot \Delta t - 1018 \cdot 0.5(T - 60)\Delta t$

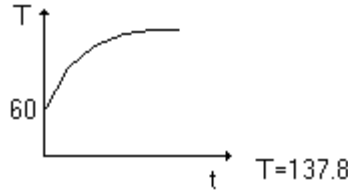
her iki yan Δt ye bölünür ve limit alınır

$2900 \frac{dT}{dt} = 291(267 - T) + 1910 - 509(T - 60)$

elde edilir.

$\frac{dT}{dt} 44.1 - 0.32 \cdot T \rightarrow \int_{60}^{90} \frac{dT}{44.1 - 0.32 T} = t \Rightarrow t = 1.52 \text{ saat}$

Yatışkın hale varmak için gerekli zamanı bulmak istersek o zaman $\frac{dT}{dt} = 0$ alınır ve $T=137.8$ °F bulunur. Bu sıcaklığa varmak için gerekli zaman $t = 27.3$ saattir. Bu zaman-sıcaklık ilişkisi aşağıdaki şekilde gösterilmiştir.



Şekil 10.10 Örnek 10.9'e ait grafik

Örnek 10.10

CO ve Cl₂ aktif odun kömürü beraberinde CO+Cl₂---->COCl₂ reaksiyonuna göre fosgen elde ediliyor.

T=303.8 K de reaksiyon hızı $R_f (mol / dak) = \frac{8.75C_{CO}C_{Cl_2}}{(1 + 58.6C_{Cl_2} + 34.3C_{COCl_2})^2}$ denklemi ile verilmektedir.

Burada C_i= konsantrasyon

a) 3 litrelik hacmi olan bir reaktörde % 60 CO, % 40 Cl₂ içeren gaz T=303.8 K ve P=1 atm. yukarıdaki reaksiyon oluşmaktadır. Odun-kömürü tarafından kaplanan hacim ihmal edilmektedir. c_p (t), fosgenin herhangi bir t anındaki konsantrasyonu ise c_{CO} ve C_{Cl₂} ' ün c_p cinsinden değerini bulunuz.

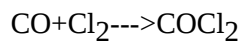
b) c_p için diferansiyel denklği yazarak

$$\frac{dc_p}{dt} = \frac{2.92(0.02407 - c_p)(0.01605 - c_p)}{(1.941 - 24.3c_p)^2}$$
 ifadesini türetiniz. Bu denklem için başlangıç şartını belirtiniz.

c) "b" şıkkında verilen denklemden yola çıkarak limitleyen reaktantın % 75 inin reaksiyona girebilmesi için gerekli olan zamanı hesaplayınız.

d) "c" şıkkında elde edilen ifadeyi analitik olarak integre etmek mümkün. Fakat istenilen bu ifadeyi nümerik olarak bilgisayar programı yazarak hesaplayınız. (Simpson kuralını uygularken nokta sayısını i) 5, ii) 21 ve iii) 51 alınız. Buradan elde edilen değerleri analitik netice ile karşılaştırınız.

Çözüm



$$a) \frac{3 \text{ litre}}{303.8 \text{ K}} \left| \frac{273 \text{ K}}{22.4 \text{ litre}} \right| = 0.12035 \text{ mol gaz}$$

$$\left. \begin{aligned} (c_{CO})_0 &= 0.6(0.12035)/3 = 0.02407 \text{ mol / L} \\ (c_{Cl})_0 &= 0.4(0.12035)/3 = 0.01605 \text{ mol / L} \end{aligned} \right\} \text{Başlangıç konsantrasyonları}$$

$$c_{CO} = 0.02407 - c_p(t)$$

$$c_{Cl_2} = 0.01605 - c_p(t)$$

$$\begin{aligned} \text{b) } \frac{d(Vc_p)}{dt} &= \frac{8.75 c_{CO} c_{Cl_2}}{(1 + 58.6 Cl_2 + 34.3 c_p)^2} \\ &= \frac{2.92(0.02407 - c_p)(0.01605 - c_p)}{(1.941 - 24.3 c_p)^2}, \quad V=3 \text{ litre} \end{aligned}$$

c) Burada kısıtlı (limitleyen) reaktant Cl_2 .

% 75 inin reaksiyona girmesi demek $c_p = 0.01605(0.75) = 0.01204$

$$t = \frac{1}{2.92} \int_0^{0.01204} \frac{(1.941 - 24.3 c_p)^2}{(0.02407 - c_p)(0.01605 - c_p)} dc_p = 90.435 \text{ dk.}$$

d) Gerekli program Q-Basic ile yazılıp hesap edilebilir. Biz "c" şıkında MATLAB kullanarak integrasyonu verilen değerler arasında nümerik olarak elde ettik.

Örnek 10.11

CO_2 içeren bir gaz, iyi karıştırılmalı kesikli bir absorberde su ile temas edilmektedir. CO_2 in sudaki çözünürlüğü $c_A = \frac{P_A}{H_A}$ (Henry Yasası) ile verilmiştir. c_A (mol / L) = CO_2 in sudaki konsantrasyonu, H_A = Henry sabiti

CO_2 absorbsiyon hızı $r_a \left(\text{mol} / \text{cm}^2 \cdot \text{s} \right) = k(c_A^* - c_A)$ ifadesi ile verilmiştir.

c_A = Sıvıdaki CO_2 in gerçek konsantrasyonu

$$c_A^* = \text{Doymuş haldeki konsantrasyon} \left(c_A^* = \frac{P_A}{H_A} \right)$$

Gaz fazı P (atm) total basınçta olup içindeki CO_2 in mol fraksiyonu y_A , saf suyun da hacmi V (cm^3) olsun. Sıvının çok iyi karıştırıldığını, yani homojen olduğunu, absorbe olan CO_2 in çok az olduğunu P, V ve y_A nın sabit olduğunu kabul ediniz.

a) CO₂ için diferansiyel denkliği yazınız ve bu ifadenin integrasyonu ile $c_A(t) = c_A^*[1 - \exp(-kSt/V)]$ olduğunu gösteriniz. Burada S = ara yüzey alanı

b) P=20 atm, V(sıvı hacmi)=5 litre, tank çapı=10 cm, y_A=0.3, H_{CO₂}=9230 atm/(mol/cm³)
k=0.02 cm/s olduğuna göre, c_A=0.62 mol/L olması için ne kadar zaman geçmelidir?

Çözüm

a) CO₂ in sıvı içindeki mol sayısı (herhangi bir t zamanında)=V · c_{CO₂}

Gaz fazından sıvı faza geçen CO₂ için denkliği yazarsak c_A = c_{CO₂}

$$\frac{d(Vc_A)}{dt} = kS(c_A^* - c_A) \Rightarrow \frac{dc_A}{dt} = \frac{kS}{V}(c_A^* - c_A)$$

$$\int_0^{c_A} \frac{dc_A}{c_A^* - c_A} = \frac{kS}{V}t + \ln(c_A^* - c_A) \Big|_0^{c_A} = -\frac{kS}{V} \cdot t$$

$$\frac{c_A^* - c_A}{c_A^*} = e^{-kSt/V} \Rightarrow c_A^* - c_A = c_A^* e^{-kS/V \cdot t}$$

$$\Rightarrow c_A = c_A^* \left[1 - e^{-\frac{kS}{V}t} \right]$$

$$b) c_A^* = \frac{P_A}{H_A} = \frac{20 \cdot 0.3 \text{ atm}}{9230 \text{ atm/mol/cm}^3} = 6.5 \cdot 10^{-4} \text{ mol/cm}^3$$

$$0.62 = 0.65 \left[1 - e^{-\frac{0.02785t}{5000}} \right]$$

$$\Rightarrow t = 9795 \text{ sn.} = 2.72 \text{ saat}$$

BÖLÜM 11 - MATEMATİKSEL MODELLEME İLE İLGİLİ ÇÖZÜLECEK BAZI PROBLEMLER

(Burada verilen problemler [30,48-52] referanslarından adapte edilmiştir)

Amaç

- Kitapta öğrenilen bilgi ve becerileri, derlenmiş olan bir seri problemler üzerinde test etmek
- MATLAB'in, özellikle, diferansiyel denklemlerin çözümüne uygulanmasını pekiştirmektir.

- 1) $dy/dx + 1 = 2y$, $y(1) = 1$ denkleminin özel çözümü bulunuz.
- 2) $y'' + 3y' + 2y = 0$ diferansiyel denkleminin önce genel çözümünü sonra da $y(0)=1$ ve $y'(0)$ başlangıç değerleri için özel çözümünü bulunuz.
- 3) Aşağıdaki başlangıç-değer problemlerini çözünüz.
 - a) $\frac{dy}{dx} = ye^x$, $y(0) = 2e$
 - b) $\frac{dy}{dx} = 6e^{2x-y}$, $y(0) = 0$
- 4) Aşağıdaki başlangıç-değer problemlerini Euler, Runge-Kutta ve MATLAB sembolik matematik ile çözünüz. ($h=0.1$ ve $t_{\text{son}}=1$ alınız)
 - a) $y'' + 2y' + 4y = 0$ $y(0) = 2$, $y'(0) = 0$
 - b) $y' = x + 2y$ $y(0) = 1$
 - c) $y'' + ty = 0$ $y(0) = 1$, $y'(0) = 0$
- 5) $3y'' - y' + 3y = \sin(2t)$ $y(0) = 1$, $y'(0) = 0$ diferansiyel denklemini çözünüz.
- 6) Problem 2'yi **dsolve** komutu ile sembolik olarak çözünüz ve $t=6$ 'daki değerini **ode45** komutu ile bulunanla karşılaştırınız.
- 7) $x^2y'' + xy' + x^2y = 0$ $y(1) = 1$, $y(8) = 0$ sınır-değer problemini **dsolve** komutu ile çözerek $x=3$ 'teki y değerini bulunuz. Bu değeri, **ode45** fonksiyonu kullanarak bulduğunuz değerle karşılaştırınız.
- 8) Aşağıdaki denklem sistemini $h=0.1$ için Runge-Kutta yöntemiyle çözünüz.
$$\begin{aligned} y_1' &= -y_2 + \sin(xy_3) & y_1(0) &= 1 \\ y_2' &= -y_1^2 & y_2(0) &= 0 \\ y_3' &= -y_3 - y & y_3(0) &= 1 \end{aligned}$$

9) Aşağıda verilen denklem sistemlerini çözünüz.

a) $\frac{dx}{dt} = -x + y$ $x(0) = 1$

b) $\frac{dy}{dt} = -5x + 3y$ $y(0) = 3$

c) $\frac{dx}{dt} = x - y + 4z$ $x(0) = 1$

d) $\frac{dy}{dt} = 3x + 2y - z$ $y(0) = 7$

e) $\frac{dz}{dt} = 2x + y - z$ $z(0) = 3$

10) $\frac{dy}{dt} = y - t$ diferansiyel denkleminin $y(0)=2$ başlangıç değeri ile $t=0$ 'dan $t=5$ 'e kadar nümerik olarak çözümünü bulunuz ve t 'ye karşı y 'i çiziniz.

11) $\frac{d^2y}{dt^2} + y = 9$ 2.mertebeden diferansiyel denkleminin $y(0)=1$ ve $y'(0)=0$ başlangıç değerleri için $t=0$ ve $t=5$ aralığında nümerik çözümünü bulunuz ve t 'ye karşı bağımlı değişkenin MATLAB ile grafiğini çiziniz.

KAYNAKLAR

- [1] Chapra, S. C., *Applied Numerical Methods with MATLAB for Engineers and Scientists*, 2E, McGraw-Hill, 2008.
- [2] Chapra S.C. ve Canale, R.P, *Numerical Methods for Engineers*, 6E, McGraw-Hill, 2010.
- [3] Gültekin, S. *Mühendislik ve Fen Uygulamalarında MATLAB ile Nümerik Analiz*, Papatya yayıncılık, 2012.
- [4] Mathews ve Fink, *Numerical Methods Using MATLAB*, Prenhice Hall, 1999.
- [5] Karris, S.T., *Numerical Analysis Using MATLAB and Spreadssheets*, 2E, Orchard, 2004.
- [6] Altıntaş, A., *MATLAB ve Genel Uygulamaları*, Değişim Yayıncılık, 2006.
- [7] Davis ve Sigmon, *MATLAB Primer*, 7E, Chapman and Hall/CRC, 2005.
- [8] Hahn ve Valentine, *Essential MATLAB for Engineers and Scientists*, 3E, Elsevier, 2007.
- [9] Driscoll, T.B., *Learning MATLAB*, SIAM, 2009.
- [10] Kalechman, M., *Practical MATLAB Applications for Engineers*, CRS, 2009.
- [11] Cutlip, M., *Problem Solving in Chemical and Biochemical Engineering with Polymath, Excel and MATLAB*, Prentice-Hall, 2007.
- [12] Knight, A., *Basic of MATLAB and Beyond*, CRC, 2000.
- [13] Quarteroni ve Saleri, *Scientific Computing with MATLAB and Octave*, 2E, Springer, 2006.
- [14] Colgren, R., *Basic MATLAB, Simulink and Stateflow*, American Inst. Aeronautics, 2007.
- [15] Chau, P., *Chemical Process Control- a First Course with MATLAB*, CRC, 2001.
- [16] Griffiths, D.F., *Introduction to MATLAB*, University of Dundee, 2001.
- [17] Uysal, M., *MATLAB- Matematiksel Uygulamalar ve Mühendislik Uygulamaları*, Beta, 2004.
- [18] Beers, K.J., *Numerical Methods for Chemical Engineering Applications in MATLAB*, Cambridge, 2007.
- [19] Fausett, L., *Applied Numerical Analysis Using MATLAB*, Addison Wesley, 2007.
- [20] Çankaya ve Akgün, *MATLAB ile Meslek Matematiği*, Seçkin Yayıncılık, 2008.
- [21] Smith, D., *Engineering Computation with MATLAB*, 1E, Addison Wesley, 2009.
- [22] Carnahan, B.H., Luther, A., Wilkes, J. O., *Applied Numerical Methods*, Wiley, 1969.
- [23] Greenspan, D., *Introduction to Numerical Analysis and Application*, Markham, 1971.
- [24] Constantinides, A., *Numerical Methods for Chemical Engineers with MATLAB Applications*, Prentice Hall, 1999.
- [25] Kiusalaas, J., *Numerical Methods in Engineering with MATLAB*, Cambridge, 2005.
- [26] İnan, A., *MATLAB Kılavuzu*, 2E, Paptya Yayımcılık, 2007.
- [27] Levenspiel, O., *Chemical Reaction Engineering*, 3E, Wiley, 1999.
- [28] Smith, J. M., *Chemical Engineering Kinetics*, 3E, McGraw-Hill, 1981.
- [29] Himmelblau, D ve ark., *Basic Calculations in Chemical Engineering*, 7E, Prentice Hall, 2004.
- [30] Felder, R.M., Rousseau, R.W., Bullard, L.G., *Elementary Principles of Chemical Processes*, 4E, Wiley, 2016.

- [31] Fogler, H. S., *Essential of Chemical Reaction Engineering*, Prentice Hall, 2011.
- [32] Downey, A.B., *Physical Modelling MATLAB*, 2008.
- [33] Spiegel, C., *PEM Fuel Cell- Modelling and Simulation Using MATLAB*, Elsevier, 2008.
- [34] Lipsman ve Rosenberg, *A Guide to MATLAB for Beginners and Experienced Users*, Cambridge, 2001.
- [35] Manassah, J. T., *Elementary Mathematical Computational Tools for Electrical and Computer Engineers Using MATLAB*, CRC, 2001.
- [36], Mathworks, *MATLAB and SIMULINK*, Student Version, Prentice Hall, 2007.
- [37] Holzbecher, E., *Environmental Modelling Using MATLAB*, Springer, 2007.
- [38] Gilat, A., *MATLAB- An Introduction with Applications*, Wiley, 2004.
- [39] Gültekin, S. (Editor) *Kimya Mühendisliğine Giriş*, Papatya Yayıncılık, 2014.
- [40] Froment ve Bischoff, *Chemical Reactor Analysis and Design*, 3E, Wiley, 2011.
- [41] Holland ve Anthony, *Fundamentals of Chemical Reaction Engineering*, 2E, Prentice Hall, 1989.
- [42] Butt, J. B., *Reaction Kinetics and Reactor Design*, Dekker, 2E, 1999.
- [43] Davis, M.E. ve Davis, R.J. *Fundamentals of Chemical Reaction Engineering*, 2E, McGraw-Hill, 2003.
- [44] Harriot, P., *Chemical Reactor Design*, Dekker, 2003.
- [45] Fogler, H.S., *Elements of Chemical Reaction Engineering*, 4E, Prentice Hall, 2004.
- [46] Welty, J. R., Wicks, C. E., Wilson, R. E., *Fundamentals of Momentum, Heat and Mass Transfer*, 3E, Wiley, 1984.
- [47] Bird, R.B., Stewart, W.E., Lightfoot, E. N., *Transport Phenomena*, 2E, Wiley, 2002
- [48] Rasmuson, A., Andeson, B., Olson, L., Anderson, R., *Mathematical Modeling in Chemical Engineering*, Cambridge University Press, 2014
- [49] Geankoplis, C.J., *Transport Processes and Unit Operations*, 4E, Prentice Hall, 2003.
- [50] Boyce ve DiPrima, *Elementary Differential Equations and Boundary Value Problems*, Wiley, 2001.
- [51] Çengel ve Boles, *Thermodynamics: An Engineering Approach*, 6E, McGraw-Hill, 2010.
- [52] Edwards ve Penny, *Differential Equations and Boundary Value Problems*, 3E, Pearson, 2003.

DİZİN

- Absorbsiyon, 107, 109, 110, 134
Açık sistem, 126
Adi Diferansiyel Denklemler, 36, 86
Adiabatik sıkışma, 124
Aktif Klasör, 7
Ampirik, 121
Analitik Çözümler, 86
Analoji, 112
Ardışık reaksiyonlar, 93, 118
Aritmetik İşlemciler, 8
Basit İterasyon Yöntemi (Ardışık Yerleştirme Yöntemi), 61, 66
Başlangıç Değer, 32
Başlangıç şartları, 2, 87
Başlangıç-değer problemi, 32, 33
Belirli İntegraller, 29
Belirsiz İntegraller, 28
Benzerlik, 112
Benzetim, 7
Bilgisayar, 86, 125
Bir bilinmeyenli denklemler, 30
Buharlaştırıcı Modeli, 96
Büyük türev, 114
Cebirsel Denklemlerin Sembolik Çözümü, 30
Command History, 7
Command Window, 7
Continuous-Stirred-Tank-Reactor, CSTR, 101
Current Directory, 7
Çalışma Alanı, 7
Çizgi Tipi, 17
Çözüm kümesi, 30
Dağınık Verilere Doğru Uydurma, 50
Değişkenler, 9
Denge Prosesleri, 94, 105
Denklem çözme, 30, 87
Denklemin kökleri, 15, 30, 60
Diferansiyel Denklemler, 31, 35, 36, 86, 89
Diferansiyel Denklemlerin Nümerik Çözümü, 35
Diferansiyel Denklemlerin Sembolik Çözümü, 31
Disp komutu, 27
Distilasyon, 105, 107, 110, 122, 130
Diziler, 10
Döngüler (Loops), 26
Ekstraksiyon, 105, 107, 110
Ekstrapolasyon, 16
Eleman- Elemana İşlemleri, 12
En küçük kareler yöntemi, 16, 42, 50, 52, 57, 58
Enerji Denkliği, 119
Enerji korunumu yasası, 115
Euler, 82, 83, 84, 93, 103, 136
Euler Tek Adım Yöntemi, 82
Evaporatörler, 96
Ezplot, 17, 18, 19
Ezplot komutu, 18
Fabrika Tasarımı, 2
Fins, 94
Fonksiyonlar, 9
Fzero komutu, 34, 35
Gauss Eleme Yöntemi, 68
Gaz-Sıvı Sistemler, 106
Geri Farklar, 74
Grafik Yöntem, 60
Grafikler, 7, 17
Griddata, 16
Henry yasası, 106, 107
Hiperbolik fonksiyon, 96
İki bilinmeyenli denklemler, 30
İki Boyutlu (2D) Grafikler, 17
İleri Farklar, 74
İnt komutu, 28, 29
İntegrasyon, 28, 35, 75, 91
İnterp1, 16
İnterp2, 16
İnterpolasyon, 16
İşlemciler, 8, 9
Kanatçıklardan Isı Transferi, 94
Kanatçıklardan Isı Transferi Modeli, 94
Kapalı sistem, 126
Kaynama, 98
Kazan Tipi Kesikli Reaktör (Batch Reactor), 100
Kısmi diferansiyel denklemler, 2, 35
Kimyasal Reaksiyon Mühendisliği, 2
Kinetik, 119
Komponent Denklikleri, 117
Komut Geçmişi, 7
Komut Penceresi, 7
Kontrol hacmi, 94, 112
Koşul İfadeleri, 22
Kök Bulma, 34
Kuvvet Fonksiyonları, 47
Limit, 30, 37, 132
Lineer (Doğrusal) Denklem Sistemleri, 13
Lineer Denklem Sistemlerinin Çözümü, 68
Lineer interpolasyon, 43, 45
Linspace, 11, 53, 54, 55
Logaritmik Koordinatlar, 48
Loglog, 17
Logspace, 11
Makroskobik Denklikler, 115
Mantıksal İşlemciler, 9
MAPLE, 86
Matematiksel Modelleme, 1, 4
MATHCAD, 86
Mathematica, 86
MATLAB© Paket Programı, 7
Matrisin Determinantı, 13
Matrisin Ters, 13
Matrisin Transpozu, 13
Matrisler, 7, 10
Matrislerde Toplama ve Çıkarma, 12

- Matrislerin Çarpılması, 12
 Matrislerin Çarpımı, 12
 M-Dosyaları, 8, 19
 Merkezi Farklar, 74
 Mikroskobik Denklikler, 112
 Model Oluşturma, 7
 Multiple-stage, 107
 Newton 'nun soğuma yasası, 33
 Newton-Raphson Yöntemi, 61, 62, 63, 66
 Nokta Sembolleri, 18
 Nümerik (Sayısal) İşlemler, 7, 34
 Nümerik Çözümler, 82
 Nümerik İntegrasyon, 35, 75
 Nümerik Türev, 74
 Nümerik Yöntemler, 61
 ODE, 31, 35, 86
 Ode45, 35, 36, 83, 93, 136
 Operatörler, 8
 Optimizasyon, 7
 Ordinary Differential Equations, 31
 Parametreler, 4
 Piston Akışlı Reaktör (PAR), 2, 102
 Plot, 17, 42, 53, 54, 55, 83, 93
 Plot komutu, 17
 Plotyy, 17
 Polinomlar, 7, 14
 Polinomların Birbirleriyle Bölümü, 15
 Polinomların Çarpımı, 15
 Polinomların Değerini Bulma, 14
 Polinomların Kökleri, 15
 Polinomların Köklerinin, 15
 Polyval Komutu, 14
 Potansiyel enerji, 119
 Programlama, 7, 19
 Proses Dinamik ve Kontrol, 2
 Quad, 35
 Reaktörlerin Modellenmesi, 99
 Regresyon, 16, 57, 58
 Regresyon Katsayısı, 57
 Regula -Falsi Yöntemi, 61, 66
 Renkler, 18
 Runga-Kutta (R-K) Yöntemi, 84
 Sağa Bölme, 12
 Secant Yöntemi, 61, 66
 Sembolik İşlemler, 7, 28
 Sembolik Limit Alma, 30
 Semi-log, 41, 48, 49, 50, 58, 59, 121
 Semilogx, 17
 Semilogy, 17
 Sınır Değer, 32
 Sınır şartları, 82, 96
 Sınır-değer problemleri, 31, 82, 93
 Simpson Parabolik Yöntemi, 78, 79
 Simpson Yöntemi, 35
 Simülasyon, 7
 Sola Bölme:, 12
 Steady-state, 116
 Step input, 129
 Substantial derivative, 114
 Sum komutu, 77
 Süreklilik Denklemi, 112, 115
 Symbolic Math Toolbox, 28
 Syms komutu, 28
 Tam Karıştırmalı Sürekli Reaktör (TKSR), 2, 101
 Taşınım Olayları, 2
 Termodinamiğin birinci yasasını, 126
 Ters Akımlı Çok Kademeli Prosesler, 107
 Toolbox, 28
 Toplam Kütle Denkliği (Süreklilik Denklemi), 115
 Total denklik, 97
 Türev, 74
 Üssel fonksiyonlar, 48
 Van der Waals denklemi, 60
 Vektör Notasyonları, 114
 Vektör Oluşturma, 10
 Vektör ve Matris İşlemleri, 12
 Vektörler, 10, 13
 Veri Analizi, 7
 Verilere Polinom Uydurmak, 16
 Workspace, 7
 Yamuklar Yöntemi (Trapezoids Rule), 76
 Yarıya Bölme Yöntemi (Bisection Method), 61, 64, 66
 Yatışkın halde, 94, 97, 103, 113, 127, 131

Matlab İle Kimya Mühendisliğinde Matematiksel Modelleme

Проф., докт. Селахаттин Гүлтекин

Техникалык редактор: *Ш. Даулятов*

Мукабанын дизайны: *Жошкун Кала*

Кыргыз-Түрк «Манас» университетинин Башкаруу кеңешинин жана Басма комиссиянын чечими боюнча
басылды. Басууга 06.07.2017 кол коюлду.

Китептин өлчөмү А4. Нускасы 200.

КТМУнун Басмакана бөлүмү.

Бишкек ш., Жал к.р., КТМУнун Ч. Айтматов атындагы кампусу.