

FİZ433 FİZİKTE BİLGİSAYAR UYGULAMALARI
(DERS NOTLARI)

Hazırlayan:

Prof.Dr. Orhan ÇAKIR

Ankara Üniversitesi, Fen Fakültesi, Fizik Bölümü

Ankara, 2017

İÇİNDEKİLER

1. LİNEER OLMAYAN DENKLEMLERİN KÖKLERİNİN BULUNMASI I/II

2. LİNEER DENKLEM SİSTEMLERİNİN ÇÖZÜLMESİ I/II

3. UYGUN EĞRİNİN BULUNMASI VE INTERPOLASYON I/II

4. SAYISAL İNTEGRAL HESAPLARI I/II

5. DİFERENSİYEL DENKLEMLERİN SAYISAL ÇÖZÜMLERİ I/II

6. BENZETİM I/II

7. FİZİKTE SEMBOLİK HESAPLAMA I/II

EKLER

KAYNAKLAR

KONU 1

LİNEER OLMAYAN DENKLEMLERİN KÖKLERİNİN BULUNMASI

Fen ve mühendislikte $y=f(x)$ şeklindeki bir denklemin köklerinin bulunmasıyla ilgili bir çok uygulama vardır. Gerçel köklerin bulunması için tekrarlamalı yapıdaki yöntemler kullanılacaktır. Örnek bir denklem $y=x^3-x^2-4x+4$ olarak yazılabilir. Burada uygulanacak yöntemler, sırasıyla Grafiksel Yöntem, Arama Yöntemi, İkili Arama Yöntemi, Kiriş Yöntemi, Newton-Raphson Yöntemi ve Sekant Yöntemidir. Bu yöntemlerin fizikteki uygulama alanlarına örnek, elektrostatik problemleri, hidrolik problemleri, vb. verilebilir. Bölümün sonunda örnek problemler yer almaktadır.

Grafiksel Yöntem

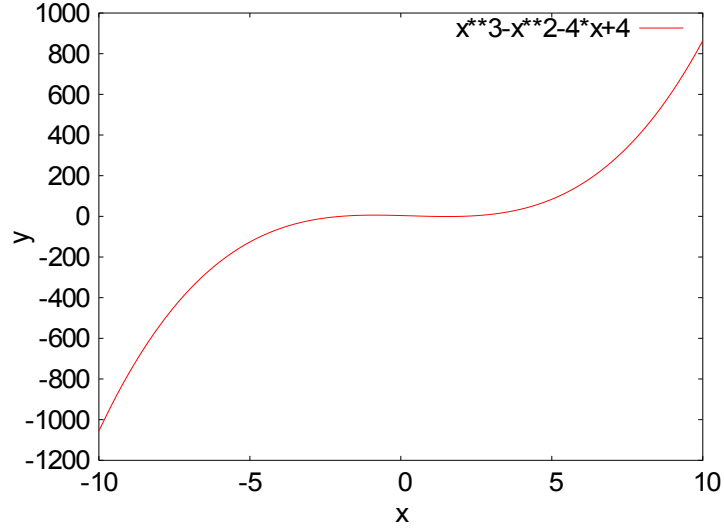
Bir denklemin kökleri, $y=f(x)=0$ biçimindeki denklemini sağlayan x değerleridir. Grafiksel yöntemde bir denklemin kökleri, $y=f(x)$ fonksiyonunun x eksenini kesen yerdeki x değerleri olarak bulunabilir. Örnek olarak, $y=f(x)=x^3-x^2-4x+4$ denklemini ele alalım. Burada grafik çizmek için Gnuplot programı (Williams, 1986) kullanılabilir. Program çalıştırıldıktan sonra açılan pencerede

```
gnuplot> set xlabel "x" ↵
```

```
gnuplot> set ylabel "y" ↵
```

```
gnuplot> plot x**3-x**2-4*x+4 ↵
```

yazılır. Burada ↵ işareti bilgisayar klavyesinde Enter veya Return tuşunun simgesidir. Fonksiyonun grafiği standart x sınırları (-10:10) aralığında görüntülenir, Şekil 1.1.



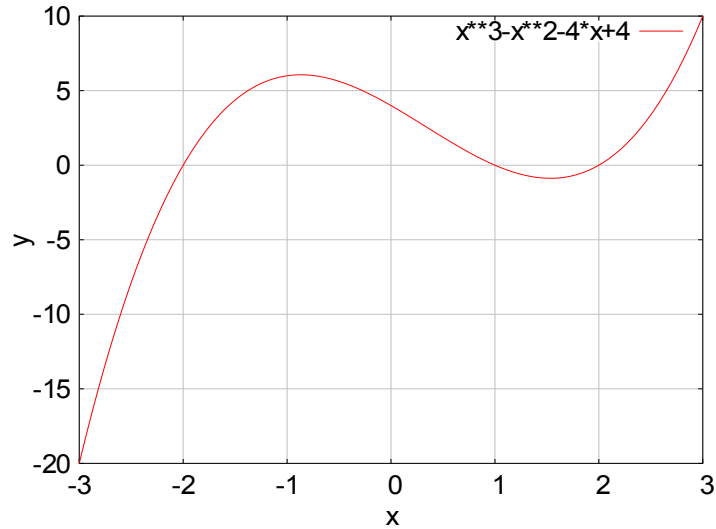
Şekil 1.1 $y=x^3-x^2-4x+4$ fonksiyonunun grafiği

Buradan kökleri bulmak biraz zor görünmektedir, fakat fonksiyonun x eksenini kesim yerleri tahmin edilebilir. Daha iyi bir tahmin için x eksenini veya y eksenini sınırlandırılabilir. Ayrıca şablon çizgileri de çizdirilirse daha kolaylık olacaktır. Bunun için örneğin, eksen isimleri, x bölge sınırlandırması ve şablon çizgileri aşağıdaki gibi yapılabilir.

```
gnuplot> set xrange [-3:3] ↵
```

```
gnuplot> set grid ↵
```

```
gnuplot> replot ↵
```



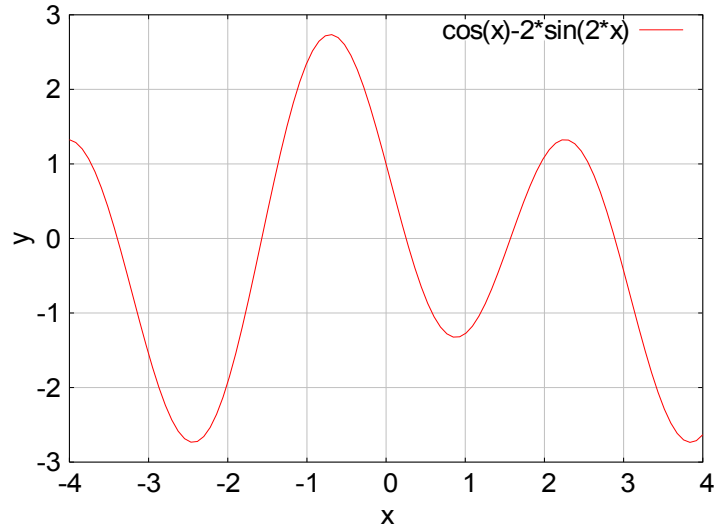
Şekil 1.2 $-3 < x < 3$ aralığında $y=x^3-x^2-4x+4$ fonksiyonunun grafiği

Şekil 1.2'den denklemin üç gerçel kökünün $x=-2$, $x=1$ ve $x=2$ de olduğu görülür. Bu x değerleri $f(x)=0$ denklemini sağlarlar.

Benzer şekilde $f(x)=\cos(x)-2\sin(2x)$ fonksiyonunun -4 ile 4 arasındaki köklerini bulabiliriz.

```
gnuplot> set xrange [-4:4] ↵
```

```
gnuplot> plot cos(x)-2*sin(2*x) ↵
```



Şekil 1.3 $y=\cos(x)-2\sin(2x)$ fonksiyonunun grafiği

Şekil 1.3'den görüldüğü gibi, burada kökler x 'in tamsayı değerleri değildir o nedenle daha ayrıntılı grafik çizmek gerekebilir. Yine bir tahminle köklerin bu aralıkta $(-4:4)$ 5 tane olduğunu ve bunların x_1, x_2, x_3, x_4, x_5 ile gösterildiğinde,

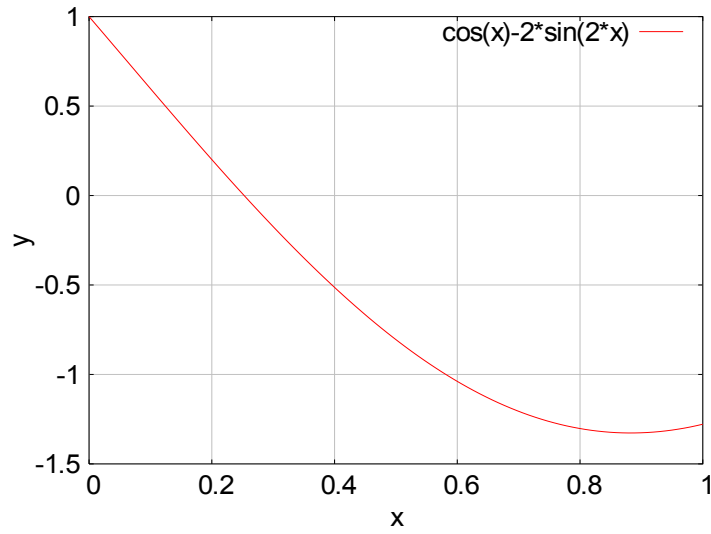
- x_1 : -4 ile -3 arasında
- x_2 : -2 ile -1 arasında
- x_3 : 0 ile 1 arasında
- x_4 : 1 ile 2 arasında
- x_5 : 2 ile 3 arasında

olduğunu söyleyebiliriz. Burada köklerin sadece birini daha ayrıntılı bulmak istersek x bölgesini kök civarına sınırlandırmalıyız. Örneğin, pozitif ilk kökü (x_3) bulmaya çalışırsak bunun için grafik çizilmesinde

```
gnuplot> set xrange [0:1] ↵
```

```
gnuplot> replot ↵
```

yazılabilir. Bunun sonucunda grafik



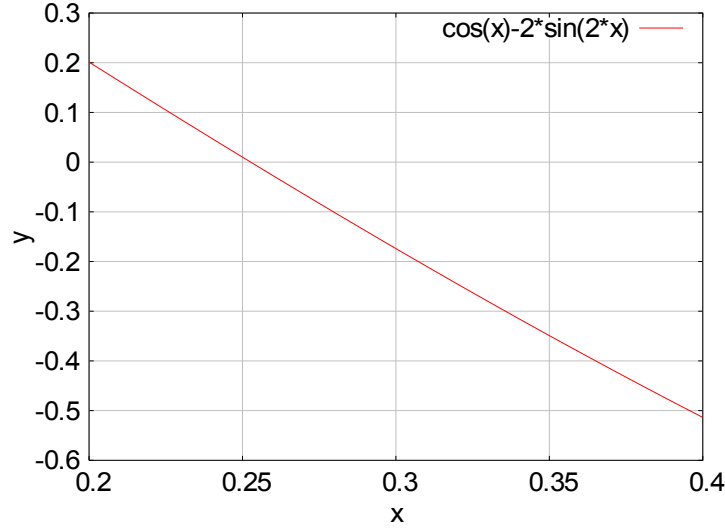
Şekil 1.4 $0 < x < 1$ aralığında $y = \cos(x) - 2\sin(2x)$ fonksiyonunun grafiği

şeklinde olacaktır, buradan x_3 kökünün 0.2 ile 0.4 arasında olduğunu söyleyebiliriz, Şekil 1.4. Daha ayrıntılı bir inceleme yapmak üzere grafiği 0.2 ile 0.4 arasında çizmek için

```
gnuplot> set xrange [0.2:0.4] ↵
```

```
gnuplot> replot ↵
```

yazılır ve Şekil 1.5 elde edilir. Buradan da kökün 0.25 civarında olduğunu söyleyebiliriz.



Şekil 1.5 $0.2 < x < 0.4$ aralığında $y = \cos(x) - 2\sin(2x)$ fonksiyonunun grafiği

Bu kök için bu grafiğin de yeterli duyarlılıkta olmadığını görürüz. Eğer yeteri kadar zaman harcarsak bu kökü belli bir duyarlılığa kadar bulabilme imkanımız olur. Bu türlü köklerin bulunmasında grafik yöntemden daha çok sayısal teknikler kullanılmaktadır, ve daha hızlı çözüme gidilmektedir.

Bilgisayar programlama dilleri kullanılarak ikinci dereceden bir denklemin kökleri kolaylıkla bulunabilir. Bunun için yazılmış örnek FORTRAN ve C programları aşağıda verilmiştir. Bu programlarda a, b, ve c katsayıları gerçel sayı olarak girilmekte, önce gerçel kökün olup olmadığına bakılmakta ve gerçel değil ise uyarı verilir program sonlandırılmaktadır. Eğer diskriminant $d > 0$ ($d^2 = b^2 - 4ac$) ise köklerin bulunmasına gidilmektedir. Bu programlar $ax^2 + bx + c = 0$ denkleminin iki kökünü (x_1, x_2) gerçel sayı olarak vermektedir. Programların derlenmesi ve çalıştırılması işletim sistemine bağlı olarak bazı farklılıklar gösterebilir. Örneğin, Windows altında FORTRAN ve C/C++ derleyicileri için sırasıyla, serbest yazılımlar olan G77, FORCE 2.0 (Guedes, 1999) ve GCC, Bloodshed Dev-C++ (Laplace, 1989) derleyicileri kullanılabilir. Fizikte karşımıza her zaman ikinci dereceden denklemler çıkmayabilir, ve kökleri de analitik hesaplanamayabilir. Bu durumda sayısal tekniklere başvurmak gerekmektedir.

- **İkinci Dereceden Bir Denklemin Gerçel Köklerini Hesaplayan Programlar**

- a) **FORTRAN Programı**

```
real a,b,c,d,x1,x2
```

```

write(*,*)"katsayilari girin"

read(*,*) a,b,c

d=b**2-4.*a*c

if(d.lt.0.) then

write(*,*)"gercel kok yok !"

stop

endif

x1=(-b+sqrt(d))/(2.*a)

x2=(-b-sqrt(d))/(2.*a)

write(*,*)"Kokler=",x1,x2

end

```

b) C Programı

```

#include<stdio.h>

#include<math.h>

main(){

float a,b,c,d; float x1,x2;

printf("katsayilari girin\n");

scanf("%f,%f,%f",&a,&b,&c);

d=b*b-4.*a*c;

if(d<0.) printf("gercel kok yok !\n");

else{

x1=(-b+sqrt(d))/(2.*a);

x2=(-b-sqrt(d))/(2.*a);

printf("Kokler=%f,%f\n",x1,x2); } }

```


Burada kaynak (source) dosyası fortran derleyicisi (g77) ile derlenip (compile), sonra bağlanıyor (link) ve çalıştırılabilir (run) dosyası oluşturulur, EK-1. Bu çalıştırılabilir dosya çalıştırılır ve ekranda sonuç görünür. İstenirse program hem çalıştırılır hem de çıktı bilgisi, bulunulan dizinde başka bir dosyaya yönlendirilebilir. Bu durumda sonuç ekrana yazılmaz, oluşturulan dosyanın içindedir.

Arama Yöntemi

Bu yöntem, yukarıda anlatılan grafiksel yöntemdeki bölge bölge kök arama şeklindeki yöntemin sayısal olarak bilgisayar programlama dillerinden birinin kullanılarak yapılması esasına dayanır.

Arama yönteminde:

- x -ekseni boyunca birkaç aralık belirlenir. Bu aralıklar o şekilde belirlenir ki bunlar x 'in değer aralığı içinde olsun ve seçilen bölgede tek kökün bulunması için yeteri kadar küçük olsun.
- Fonksiyon her aralığın başında ve sonunda hesaplanır, eğer fonksiyonun bir aralığın iki uç noktasındaki değerleri zıt işaretli ise fonksiyonun bu aralıkta x eksenini kestiği bilinir. Bu durumda kök bu aralıkta olacaktır.
- Eğer fonksiyonun işareti aralıkların iki uç noktası arasında değişirse denklemin kökünün bulunması daha hassas yapılabilir. Bunun için x eksenini daha alt bölgelere ayrılabilir ve bir alt bölgeden diğerine geçerken çok küçük adımlar hazırlanabilir.
- Eğer fonksiyonun işareti aralığın iki uç noktasında değişmezse, sonraki aralığa geçilir.

Örneğin, fonksiyon $f(x)=x^3-\cos(2x)$ olsun. Bu denklemin -1 ile 1 arasındaki kökleri ile ilgilenelim. Bunun için bu aralığı 0.1 adımlarla 20 aralığa bölelim. Bu aralıkların uç noktalarında fonksiyonun değerini hesaplayan FORTRAN ve C programları aşağıda verilmiştir.

- **Arama Yöntemi ile İlgili Programlar**

a) FORTRAN programı

```
real x,fx  
  
write(*,*)"x   f(x)"  
  
do 10 i=1,5
```

```

x=(i*1.-3.)/2.
fx=x*x*x-cos(2.*x)
write(*,*)x,fx
10  continue
    end

```

b) C Programı

```

#include<stdio.h>
#include<math.h>
main(){
float x,fx;
printf("x f(x)\n");
for(x=-1;x<=1;x+=0.5){
fx=x*x*x-cos(2.*x);
printf("%f %f\n",x,fx); } }

```

Oluşturulan program çıktı dosyası bir metin düzenleyici kullanılarak açılabilir ve içindeki bilgiler görülebilir. Sonuç aşağıdaki gibi olacaktır.

| <u>x</u> | <u>f(x)</u> |
|----------|--------------|
| -1. | -0.583853126 |
| -0.5 | -0.665302277 |
| 0. | -1. |
| 0.5 | -0.415302277 |
| 1. | 1.41614687 |

Buradan $f(x)=0$ denkleminin çözümünde $x=0$ da bir kök ve $x=0.5$ ile $x=1$ arasında bir kök bulunmaktadır. Bu değerin daha iyi belirlenmesi için x adımlarını daha azaltmalıyız. Örneğin, adımlar $\Delta x=0.1$ olmak üzere $x=0.5$ ile $x=1$ aralığında tekrar kök aramak üzere programın döngü kısımları değiştirilebilir. Bu durumda da $x=0.6$ ile $x=0.7$ aralığında fonksiyonun işareti değişmektedir. Daha da iyi sonuç elde etmek için $x=0.6$ ile $x=0.7$ aralığını 10 eşit parçaya bölebiliriz. Bu durumda artım $\Delta x=0.01$ aralıklarla olacaktır. Bunun sonucunda x 'in $x=0.64$ ile $x=0.65$ arasındaki bir değerinde fonksiyonun işaret değiştirdiğini ve kökün bu aralıkta olduğunu buluruz. Bu son işlemde istenen kök %1 duyarlılıkla bulunmuştur.

Bu yöntemde başta büyük aralıklar seçmek birbirine yakın köklerin kaybedilmesine veya atlanılmasına, çok küçük aralıklar seçmek de hesaplama bakımından uzun süreceğinden çok iyi sonuçlar elde edilemeyebilir. Ancak basitliği bakımından yöntem, bazı durumlarda tercih edilebilir.