KMU138

BİLGİSAYAR PROGRAMLAMA

TEMEL SAYISAL YÖNTEMLER ALGORITMALARI, IKIYE BÖLME YÖNTEMI,

GAUSS ELIMINASYONU, ADD ÇÖZÜMÜ

Kaynaklar

1.Pratap, R. “Getting Started with MATLAB: A Quick Introduction for Scientists and Engineers”Oxford University Press, 2010.

2.Hunt, B.R., Lipsman, L.R. and Rosemberg J. M. “A guide to MATLAB for Beginners and ExperiencedUsers"Cambridge University Press, 2001.

3.Kubat, C. “MATLAB Yapay Zeka ve Mühendislik Uygulamaları” İkinci Baskı, Pusula Yayıncılık, 2014McGraw Hill, International Edition 2012.

İKİYE BÖLME YÖNTEMİ

Açıklama: Genel olarak, f (x) xl'den xu'ya kadar aralıkta gerçek ve sürekli ise ve f (xL) ve f (xU) zıt işaretlere sahipse,
f (xL) f (xU) <0
o zaman xl ve xu arasında en az bir gerçek kök vardır.
Artımlı arama yöntemleri, işlevin işaretini değiştirdiği bir aralığı bularak bu gözlemden yararlanır. Daha sonra işaretin yeri değişir (ve sonuç olarak kök), aralığı bir dizi alt aralığa bölerek daha kesin olarak tanımlanır. Bu alt aralıkların her biri, işaret değişikliğini bulmak için aranır. İşlem tekrarlanır ve alt tahminler daha ince artışlara bölünerek kök tahmini rafine edilir.
Alternatif olarak ikili doğrama, aralık yarığı veya Bolzano'un yöntemi olarak adlandırılan ikiye bölme yöntemi, aralığın her zaman yarıya bölüneceği bir tür artımlı arama yöntemidir. Bir fonksiyon bir aralıktaki işareti değiştirirse, orta noktadaki fonksiyon değeri değerlendirilir. Kökün yeri daha sonra içinde işaret değişikliğinin meydana geldiği alt aralığın orta noktasında yer aldığı belirlenir. İşlem, rafine edilmiş tahminler elde etmek için tekrarlanır. İkiye bölme hesaplaması için basit bir algoritma aşağıda verilmiştir.

İKİYE BÖLME YÖNTEMİ

İlk olarak, f(x) fonksiyonunun belirtilen aralıkta kökü olup olmadığı [$f(x_a)^* f(x_ü) < 0$] kontrol edilir. Şart sağlıyorsa kök vardır. Çünkü fonksiyonlar zıt işaretlidir.

- [$f(x_a) * f(x_ü) > 0$] ise kök yoktur.
- [$f(x_a) * f(x_ü) = 0$] ise kök x_a ya da $x_ü$

İlk iterasyonda, belirtilen fonksiyon aralığının orta noktası tespit edilir.

$$x_o = \frac{x_a + x_ü}{2}$$

Kök [x_a, x_o] ya da [x_o, $x_ü$] aralığından birisinde olmalıdır

- $f(x_a)^* f(x_o) < 0$ ise kök [x_a, x_o] aralığında
- $f(x_o)^* f(x_ü) < 0$ ise kök [x_o, $x_ü$] aralığında
- $f(x_o)^* f(x_ü) = 0$ ise kök x_o 'dur

KAYNAK:
http://bilisim.kocaeli.edu.tr/dosyalar/Dosyalar/DersNotlari/4-denklem-cozumleri-ogr.pdf
Erişim Tarihi:06.02.2020

İKİYE BÖLME YÖNTEMİ

```
xyz=input('Enter your function: ','s')
xl=input('Enter xl:')
xu=input('Enter xu:')
tol=input('Enter stopping criterion:')
kmax=input('Enter max iter:')
f=inline(xyz);
fid=fopen('bisection.txt','w');
fprintf(fid,'Roots of Equation f(x)=x^3-12*x^2+47*x-60 \n\n');
fprintf(fid,'iter     xl       xu       xr       f(xl)       f(xu)       f(xr) \n');
fprintf(fid,'-------------------------------------------------------------------\n');
for i=1:kmax
   ya=f(xl);
   yb=f(xu);
   xr=0.5*(xl+xu);
   yr=f(xr);
   fprintf(fid,'%4.1f %7.4f %7.4f %7.4f   %7.4f   %7.4f   %7.4f\n',i,xl,xu,xr,ya,yb,yr);
   if ya*yr<0
      xu=xr;
   else xl=xr;
   end
   if abs(yr)<tol;
      break
   end
end
fclose(fid);
disp('The root of the equation is')
xr
```

GAUSS ELIMINASYONU

Satır indirgeme olarak da bilinen Gauss eliminasyonu, lineer denklem sistemlerini çözmek için lineer cebirde bir algoritmadır. Genellikle karşılık gelen katsayı matrisi üzerinde gerçekleştirilen bir işlem dizisi olarak anlaşılır. Bir matriste satır küçültme gerçekleştirmek için, matrisin sol alt köşesi mümkün olduğunca sıfırla dolana kadar matrisi değiştirmek için bir dizi temel satır işlemi kullanılır. Bu işlemleri kullanarak, bir matris her zaman bir üst üçgen matrise ve aslında sıra kademeli formda olan bir matrise dönüştürülebilir. Satır azaltma işlemi, ilk satır işlemlerini kullanır ve iki kısma ayrılabilir. İlk bölüm (bazen ileri eleme olarak da adlandırılır), belirli bir sistemi sıralı forma indirgemekte, bu da hiç çözümün, benzersiz bir çözümün veya sonsuz sayıda çözümün olup olmadığını söyleyebilmektedir. İkinci kısım (bazen geri ikame olarak da adlandırılır), çözüm bulunana kadar sıra işlemlerini kullanmaya devam eder; başka bir deyişle, matrisi azaltılmış sıralı kademeli forma sokar.

GAUSS ELIMINASYONU

$$A=\begin{bmatrix}144 & 12 & 1\\ 64 & 8 & 1\\ 25 & 5 & 1\end{bmatrix}, X=\begin{Bmatrix}x_1\\ x_2\\ x_3\end{Bmatrix}, B=\begin{Bmatrix}279.2\\ 177.2\\ 106.8\end{Bmatrix} \rightarrow [AB]=\begin{bmatrix}144 & 12 & 1 & 279.2\\ 64 & 8 & 1 & 177.2\\ 25 & 5 & 1 & 106.8\end{bmatrix}$$

$$[64 \quad 8 \quad 1 \quad 177.2]-\left(\frac{64}{144}\right)[144 \quad 12 \quad 1 \quad 279.2]$$

$$= [0 \quad 2.6667 \quad 0.55555 \quad 53.1111]$$

$$[25 \quad 5 \quad 1 \quad 106.8]-\left(\frac{25}{144}\right)[144 \quad 12 \quad 1 \quad 279.2]$$

$$= [0 \quad 2.9167 \quad 0.8264 \quad 58.3278]$$

$$[AB]=\begin{bmatrix}144 & 12 & 1 & 279.2\\ 0 & 2.6667 & 0.55555 & 53.1111\\ 0 & 2.9167 & 0.8264 & 58.3278\end{bmatrix}$$

GAUSS ELIMINASYONU

$$[0 \quad 2.9167 \quad 0.8264 \quad 58.3278] - \left(\frac{2.9167}{2.6667}\right)[0 \quad 2.6667 \quad 0.55555 \quad 53.1111]$$
$$= [0 \quad 0 \quad 0.2188 \quad 0.2376]$$

$$\begin{bmatrix} 144 & 12 & 1 \\ 0 & 2.6667 & 0.55555 \\ 0 & 0 & 0.2188 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 279.2 \\ 53.1111 \\ 0.2376 \end{Bmatrix}$$

$$0.2188x_3 = 0.2376 \quad \rightarrow \quad x_3 = 1.0859$$
$$2.6667x_2 + 0.55555x_3 = 53.1111 \quad \rightarrow \quad x_2 = 19.6902$$
$$144x_1 + 12x_2 + x_3 = 279.2 \quad \rightarrow \quad x_1 = 0.2905$$

GAUSS ELIMINASYONU

```
x=zeros(3,1);
AB=[A B]
n=size(A,1);
for i=1:n-1
   P=-(AB(i+1:end,i)/AB(i,i))
   AB(i+1:end,i:end)=AB(i+1:end,i:end)+P*AB(i,i:end)
end
for i=n:-1:1
   x(i,:)=(AB(i,n+1:end)-AB(i,i+1:n)*x(i+1:n,:))/AB(i,i)
end
```

ADD ÇÖZÜMÜ

- 0 ve 2 arasındaki bir zaman aralığı için MATLAB'ın Adi Diferansiyel Denklem (ODE) çözücüsünü kullanarak x (0) = 0 başlangıç koşulu ile aşağıdaki normal diferansiyel denklemi çözün.

$$\frac{dx}{dt} = x + t$$

```
function dxdt=difdenk(t,x)
dxdt=x+t;
```

```
[t,x]=ode23('difdenk',[0 2],0)
plot(t,x)
xlabel('t')
ylabel('x')
```

ADD ÇÖZÜMÜ

t =	x =
0	0
0.0250	0.0003
0.0500	0.0013
0.0810	0.0034
0.1234	0.0079
0.1791	0.0170
0.2496	0.0339
0.3363	0.0634
0.4400	0.1127
0.5615	0.1918
0.7009	0.3146
0.8579	0.5002
1.0319	0.7743
1.2221	1.1716
1.4221	1.7228
1.6221	2.4402
1.8221	3.3606
2.0000	4.3865