NÜMERİK ANALİZ

Bilimsel Hesaplama Matematiği

Nuri ÖZALP

Sabit Nokta ve Fonksiyonel Yineleme

Newton yöntemi ve Steffensen yöntemi, noktaların bir dizisinin

$$x_{n+1} = F(x_n) \quad (n \geq 0) \tag{1}$$

formunda bir formülle hesaplandığı yordamlara birer örnektir. Bu tip bir denklemle tanımlanan bir algoritmaya **fonksiyonel (iterasyon) yineleme** denir. Newton yönteminde

$$F(x) = x - \frac{f(x)}{f'(x)}$$

ile verilirken, Steffensen yönteminde

$$F(x) = x - \frac{[f(x)]^2}{f(x + f(x)) - f(x)}$$

dir.

(1) formülünde

$$\lim_{n\to\infty} x_n = s$$

kabul edelim. s ve F arasındaki ilişki nedir? Eğer F sürekli ise, bu durumda

$$F(s) = F\left(\lim_{n\to\infty} x_n\right) = \lim_{n\to\infty} F(x_n) = \lim_{n\to\infty} x_{n+1} = s$$

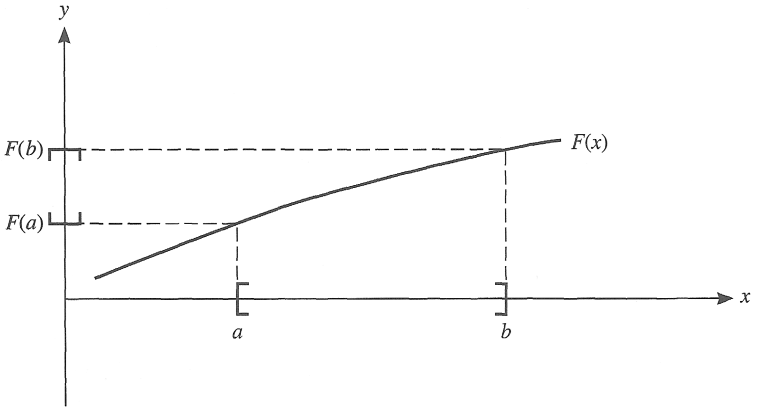
dir. Böylece, $F(s) = s$ olup, bu tipten s ye F nin bir **sabit noktası** denir. Bir sabit noktayı, fonksiyonun ardışık süreçte "kilitlendiği" değer olarak düşünebiliriz.

Bir matematiksel problem, sıklıkla bir fonksiyonun sabit noktasını bulma problemine indirgenebilir. Çok ilgi çekici uygulamalar, diferensiyel denklemler, optimizasyon teorisi ve diğer alanlarda görünmektedir. Genellikle, sabit noktaları aranan F fonksiyonu, bir vektör uzayından bir başkasına bir dönüşümdür. F nin bir kapalı $C \subseteqq \mathbb{R}$ kümesini kendi içine dönüştürdüğü, en basit durumu analiz etmeyi planlıyoruz. İspatlayacağımız teorem **büzülme dönüşümleri** ile ilgilidir. Eğer, F nin tanım bölgesindeki her x ve y noktası için

$$|F(x) - F(y)| \leq \lambda \, |x - y| \tag{2}$$

olacak şekilde, 1 *den küçük* bir λ sayısı varsa, F dönüşümüne (ya da fonksiyonuna) bir **büzülme** denir.

Şekil de gösterildiği gibi, x ve y arasındaki mesafe, büzülme fonksiyonu F yardımıyla, $F(x)$ ve $F(y)$ arasındaki daha kısa bir mesafeye dönüştürülmektedir.



Teorem (Büzülme Dönüşümü)

C reel doğrunun kapalı bir altkümesi olsun. Eğer F, C den C ye bir büzülme dönüşümü ise, bu durumda F tek bir sabit noktaya sahiptir. Dahası, bu sabit nokta bir $x_0 \in C$ başlangıç noktasıyla, Denklem (1) den elde edilen her dizinin limitidir.

Örnek

Aşağıdaki şekilde ardışık olarak tanımlanan $[x_n]$ dizisinin yakınsak olduğunu ispatlayınız.

$$\begin{cases} x_0 = -15 \\ x_{n+1} = 3 - \frac{1}{2}|x_n| \qquad (n \geq 0) \end{cases}$$

Çözüm

$|F(x) - F(y)| = \left|3 - \frac{1}{2}|x| - 3 + \frac{1}{2}|y|\right| = \frac{1}{2}\left||y| - |x|\right| \leq \frac{1}{2}|y - x|$ *olduğundan* $F(x) = 3 - \frac{1}{2}|x|$ *fonkisyonu bir büzülmedir. Sabit noktanın 2 olduğu görülebilir.*

Örnek

$$F(x) = 4 + \frac{1}{3}\sin 2x$$

fonksiyonunun sabit noktasını hesaplamak için Büzülme Dönüşümü Teoremini kullanınız.

Çözüm

Ortalama-Değer Teoreminden, x ve y arasındaki bazı ξ ler için,

$$|F(x) - F(y)| = \tfrac{1}{3}\,|\sin 2x - \sin 2y| = \tfrac{2}{3}\,|\cos 2\xi|\,|x - y| \leq \tfrac{2}{3}\,|x - y|$$

olur. Bu ise f nin $\lambda = 2/3$ ile büzülme olduğunu gösterir. Teoremden, F bir sabit noktaya sahiptir. Başlangıç değeri 4 ile başlayıp, 20 iterasyon ile sabit noktayı hesaplayan bir bilgisayar programı ile $x_{20} = 4.26148\,37$ bulunur.

Bir

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_2 z^2 + a_1 z + a_0 \tag{1}$$

polinomun değerlerinin verimli şekilde hesaplaması için **Horner algoritmasına** gereksinim vardır. Bu algoritma aynı zamanda **içiçe çarpımlar** ve **sentetik bölme** olarak da bilinir. Bu algoritmanın başka amaçlar için de kullanışlı olduğunu göreceğiz. Eğer bir p polinomu ve bir z_0 kompleks sayısı verilmiş ise, Horner algoritması $p(z_0)$ sayısını ve

$$q(z) = \frac{p(z) - p(z_0)}{z - z_0} \tag{2}$$

polinomunu üretir. q polinomunun derecesi p nin derecesinden 1 küçüktür. Bu denklemden

$$p(z) = (z - z_0) q(z) + p(z_0) \tag{3}$$

yazabiliriz.

Bilinmeyen $q(z) = b_0 + b_1 z + \cdots b_{n-1} z^{n-1}$ ve $p(z)$ nin benzer formu Denklem (3) te yazılırsa, z nin her iki taraftaki benzer kuvvetlerinin katsayıları birbirlerine eşitlenebilir. Böylece, aşağıdaki eşitlikler elde edilir:

$$
\begin{aligned}
b_{n-1} &= a_n \\
b_{n-2} &= a_{n-1} + z_0 b_{n-1} \\
&\vdots \\
b_0 &= a_1 + z_0 b_1 \\
p(z_0) &= a_0 + z_0 b_0
\end{aligned}
$$

Horner algoritmasındaki katsayıların hesabı eğer elde yapılacaksa, sıklıkla aşağıdaki düzenleme kullanılır:

$$
\begin{array}{c|ccccc}
 & a_n & a_{n-1} & a_{n-2} & \ldots & a_0 \\
z_0 & & z_0 b_{n-1} & z_0 b_{n-2} & \ldots & z_0 b_0 \\
\hline
 & b_{n-1} & b_{n-2} & b_{n-3} & \ldots & \boxed{b_{-1}}
\end{array}
$$

Kutu içindeki sayı $p(z_0) = b_{-1}$ i sağlar.

Örnek

$p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2$ olmak üzere $p(3)$ ü hesaplamak için Horner algoritmasını kullanınız.

Çözüm

Hesaplamayı yukarıda önerildiği gibi düzenleyelim.

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 3 & & 3 & -3 & 12 & 21 \\ \hline & 1 & -1 & 4 & 7 & \boxed{19} \end{array}$$

Böylece $p(3) = 19$ *olup,*

$$p(z) = (z-3)(z^3 - z^2 + 4z + 7) + 19$$

yazabiliriz.

Horner algoritması aynı zamanda **deflasyon** (daralma) için de kullanılır. Bu, bir polinomdan bir lineer çarpanı ayırma işlemidir. Eğer z_0, p polinomunun bir kökü ise, $z-z_0$, p nin bir çarpanı olup, tersi de doğrudur. p nin geriye kalan kökleri $p(z)/(z-z_0)$ ın $n-1$ tane köküdür.

Örnek

Önceki örnekteki polinomu, 2 nin köklerden biri olduğu gerçeğini kullanarak çarpanlarına ayırınız.

Çözüm

Yukarıda açıklanan hesaplama düzeninin aynısını kullanalım:

$$
\begin{array}{c|ccccc}
 & 1 & -4 & 7 & -5 & -2 \\
2 & & 2 & -4 & 6 & 2 \\
\hline
 & 1 & -2 & 3 & 1 & \boxed{0}
\end{array}
$$

Böylece,

$$z^4-4z^3+7z^2-5z-2=(z-2)(z^3-2z^2+3z+1)$$

Horner algoritmasının üçüncü bir uygulaması, herhangi bir nokta komşuluğunda Taylor açılımını bulmak içindir. $p(z)$, Denklem (1) deki gibi olsun ve

$$\begin{aligned} p(z) &= a_n z^n + a_{n-1} z^{n-1} + \cdots + a_0 \\ &= c_n (z - z_0)^n + c_{n-1} (z - z_0)^{n-1} + \cdots + c_0 \end{aligned}$$

denklemindeki c_k katsayılarını aradığımızı kabul edelim. Kuşkusuz, Taylor Teoremi $c_k = p^{(k)}(z_0)/k!$ olduğunu söylemektedir, fakat daha kullanışlı bir algoritmayı araştıra-lım. $p(z_0) = c_0$ olduğuna dikkat edersek, p polinomuna z_0 noktasıyla Horner algoritması uygulandığında bu katsayı elde edilir. Algoritma ayrıca

$$q(z) = \frac{p(z) - p(z_0)}{z - z_0} = c_n (z - z_0)^{n-1} + c_{n-1} (z - z_0)^{n-2} + \cdots + c_1$$

polinomunu verir. Bu ise, $c_1 = q(z_0)$ olduğundan, q polinomuna z_0 noktasıyla Horner algoritması uygulanarak, ikinci katsayı olan c_1 in elde edilebileceğini gösterir. Bu işlem tüm c_k katsayıları bulununcaya kadar tekrarlanır.

Örnek

Önceki örnekteki polinomun $z_0 = 3$ komşuluğunda Taylor açılımını bulunuz.

Çözüm. İşlemler aşağıdaki gibi düzenlenebilir:

$$
\begin{array}{c|ccccc}
 & 1 & -4 & 7 & -5 & -2 \\
3 & & 3 & -3 & 12 & 21 \\
\cline{2-6}
 & 1 & -1 & 4 & 7 & \boxed{19} \\
3 & & 3 & 6 & 30 & \\
\cline{1-5}
 & 1 & 2 & 10 & \boxed{37} & \\
3 & & 3 & 15 & & \\
\cline{1-4}
 & 1 & 5 & \boxed{25} & & \\
3 & & 3 & & & \\
\cline{1-3}
 & \boxed{1} & \boxed{8} & & &
\end{array}
$$

Kutu içindeki sayılar polinomun c_k katsayıları olup, buradan $p(z) = (z-3)^4 + 8(z-3)^3 + 25(z-3)^2 + 37(z-3) + 19$ dır

Yukarıda açıklanan algoritmayı **tam Horner algoritması** olarak adlandırıyoruz. Bunu çalıştıran önkod, c_k katsayıları a_k girdi katsayılarının *üzerine yazılacak* şekilde düzenlenmiştir.

girdi $n, \ (a_i : 0 \leq i \leq n), z_0$
$k = 0$ **dan** $n - 1$ **döngü**
  $j = n - 1$ **den** k **ya** -1 **adımla döngü**
    $a_j \leftarrow a_j + z_0 a_{j+1}$
  döngü sonu
döngü sonu
çıktı $(a_i : 0 \leq i \leq n)$

Şimdi Newton yönteminin bir polinoma nasıl uygulanacağına dair her şeye sahibiz. İterasyonun

$$z_{k+1} = z_k - \frac{f(z_k)}{f'(z_k)}$$

denklemi ile tanımlandığını hatırlayalım. Eğer bu bir p polinomuna uygulanırsa, Horner algoritmasıyla birleştirilen güçlü bir yöntem $p(z)$ ve $p'(z)$ yi hesaplamak için kullanılabilir. Eğer, tam Horner algoritmasında sadece iki adım kullanılırsa, $c_0 = p(z_0)$ ve $c_1 = p'(z_0)$ ı elde edeceğimizi gördük. Bu iki adım önkodda birleştirilebilir. Ayrıca, girdi katsayılarına iterasyonun ardışık adımlarında ihtiyaç duyulacağından, bunları üzerlerine yazmaktan vazgeçiyoruz. p, Denklem (1) deki formuyla, ve z_0 verilmek üzere, $\alpha = p(z_0)$ ve $\beta = p'(z_0)$ ı üreten önkod şu şekildedir:

girdi n, $(a_i : 0 \leq i \leq n), z_0$
$\alpha \leftarrow a_n$
$\beta \leftarrow 0$
$k = n - 1$ **den** 0 **a** -1 **adımla döngü**
    $\beta \leftarrow \alpha + z_0 \beta$
    $\alpha \leftarrow a_k + z_0 \alpha$
döngü sonu
çıktı α, β

Eğer bu önkodu $\texttt{horner}(n,\ (a_i : 0 \leq i \leq n), z_0, \alpha, \beta)$ olarak adlandırırsak, bu durumda, verilen polinom için z_0 başlangıçlı, M adımlı Newton yöntemi için önkod şu şekilde olabilir:

girdi $n,\ (a_i : 0 \leq i \leq n), z_0, M, \varepsilon$
$j = 1$ **den** M **ye döngü**
  $\texttt{horner}(n,\ (a_i : 0 \leq i \leq n), z_0, \alpha, \beta)$ **çağır**
  $z_1 \leftarrow z_0 - \alpha/\beta$
  çıktı α, β, z_1
  eğer $|z_1 - z_0| < \varepsilon$ **ise dur**
  $z_0 \leftarrow z_1$
döngü sonu

Örnek

Yukarıdaki örnekteki polinom için $z_0 = 0$ dan başlayarak Newton yöntemini uygulayınız.

Çözüm

Öncelikle $z_0 = 0$ *ı kullanarak, biraz önce açıklanan algoritmayla* $p(0) = -2$ *ve* $p'(0) = -5$ *değerlerini hesaplarız.* z *nin yeni değeri* $z_1 = z_0 - \frac{p(z_0)}{p'(z_0)} = 0 - \frac{-2}{-5} = -0.4$ *olur. Diğerleri*

k	$p(z_k)$	$p'(z_k)$	z_k
1	-2.00000	-5.00000	-0.40000
2	1.40160	-12.77600	-0.29029
3	1.46322	-10.17322	-0.27591
4	0.00226	-9.86030	-0.27568
5	0.00000	-9.85537	-0.27568

z_k *nın* -0.27568 *köküne hızla yakınsadığına dikkat ediniz.*