NÜMERİK ANALİZ

Bilimsel Hesaplama Matematiği, Gazi Kitabevi 2012

Nuri ÖZALP

BİLGİSAYAR ARİTMETİĞİ

Kayan Noktalı Sayılar ve Yuvarlama Hataları

Desimal sistemde

$$427.325 = 4 \times 10^2 + 2 \times 10^1 + 7 \times 10^0 + 3 \times 10^{-1} + \times 2 \times 10^{-2} \times 5 \times 10^{-3}$$

$-\pi$ sayısı

$$-\pi = -3.14159\ 26535\ 89793\ 23846\ 26433\ 8...$$

dir. Burada yazılan en son 8 rakamı, 8×10^{-26} ya karşılık gelir.
İkilik sistemde tipik bir sayı, detaylı yazılımı ile, örneğin

$$\begin{aligned}(1001.11101)_2 &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &\quad + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5}\end{aligned}$$

olur. Bu, desimal gösterimde 9.90625 reel sayısına denktir.

1/10 gibi basit bir sayı bile herhangi bir ikilik makinede tam olarak yüklenemez, çünkü bu sayı sonsuz bir ikilik ifade gerektirir:

$$\frac{1}{10} = (0.0\ 0011\ 0011\ 0011\ ...)_2 \tag{1}$$

Örneğin, 0.1 i bir 32-bitlik bilgisayara okutursak ve sonra 40 desimal noktalı çıktı alırsak, aşağıdaki sonucu elde ederiz:

0.10000 00014 90116 11938 47656 25000 00000 00000

Yuvarlama

$$\begin{aligned} 0.1735 &\longleftarrow 0.1735499 \\ 1.000 &\longleftarrow 0.9999500 \\ 0.4322 &\longleftarrow 0.4321609 \end{aligned}$$

Eğer x, onun n-rakam yaklaşımı olan $\tilde{x}$ a yuvarlanırsa, bu durumda

$$|x - \tilde{x}| \leq \tfrac{1}{2} \times 10^{-n} \tag{2}$$

olur.

Kesme

Eğer x bir desimal sayı ise, ona **yutulmuş** veya **kesilmiş** n-rakam yaklaşımı, basitçe n. den sonraki tüm rakamların atılarak elde edildiği $\hat{x}$ sayısıdır. Böylece,

$$|x - \hat{x}| < 10^{-n} \tag{3}$$

dir. x ve $\hat{x}$ arasındaki ilişki; $x - \hat{x}$ nın ilk n hanede sıfır olması ve $0 \leq \delta < 1$ olmak üzere, $x = \hat{x} + \delta \times 10^{-n}$ olmasıdır. Buradan,

$$|x - \hat{x}| = |\delta| \times 10^{-n} < 10^{-n}$$

olup, bu da (3) eşitsizliğidir

Normalleştirilmiş Bilimsel Gösterim

Desimal sistemde herhangi bir reel sayı **normalleştirilmiş bilimsel gösterimle** ifade edilebilir. Bunun anlamı; tüm rakamlar desimal noktanın sağında kalacak şekilde ve ilk rakam sıfır olmayacak şekilde desimal nokta kaydırılır ve 10 un uygun kuvvetleri kullanılır. Örneğin

$$\begin{aligned} 732.5051 &= 0.7325051 \times 10^{3} \\ -0.005612 &= -0.5612 \times 10^{-2} \end{aligned}$$

gibi. Genel olarak r, $\frac{1}{10} \leq r < 1$ aralığında bir sayı ve n de bir tamsayı (pozitif, negatif veya sıfır) olmak üzere, sıfırdan farklı bir x sayısı

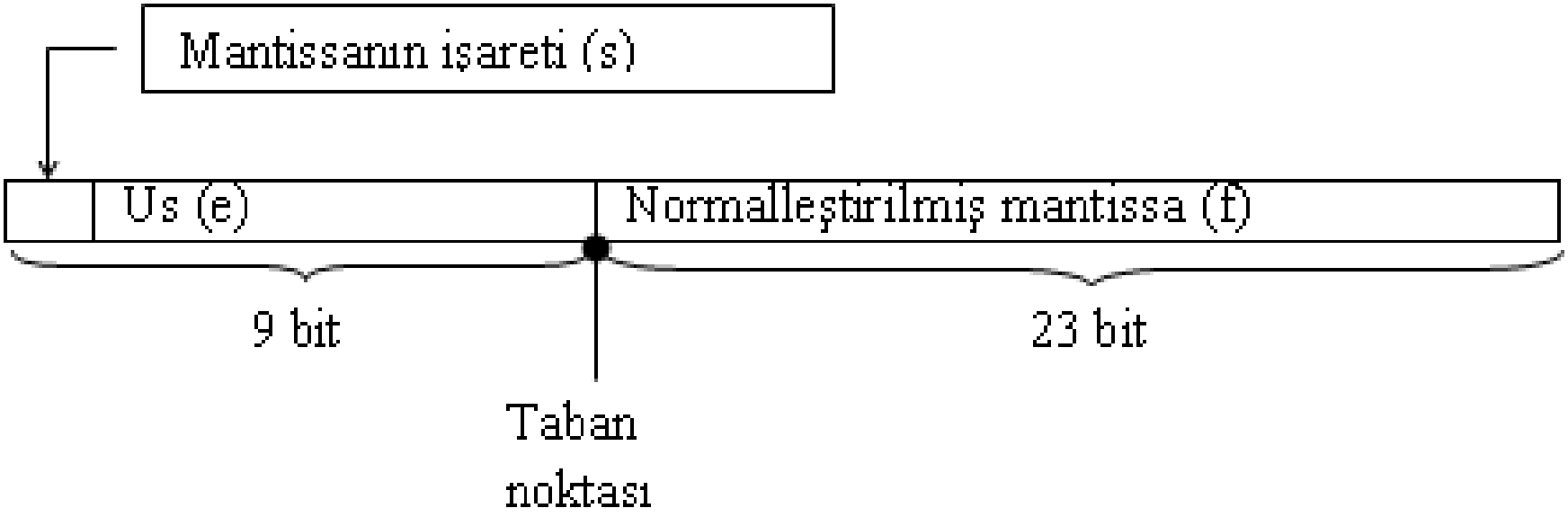
$$x = \pm r \times 10^{n}$$

formunda temsil edilebilir. Kuşkusuz, eğer $x = 0$ ise, bu durumda $r = 0$ olup, diğer tüm durumlarda, r verilen aralıkta kalacak şekilde n yi ayarlayabiliriz.

32-bitlik bir bilgisayarda tek-duyarlı reel sayı için bir kayan-noktalı sayı temsili üç alana bölünür.
Bir kelimeyi oluşturan bitler, sıfırdan farklı bir $x = \pm q \times 2^m$ reel sayısının temsilinde aşağıdaki şekilde düzenlenir:

x reel sayısının işareti	1 bit
üs kuvveti (e tamsayısı)	8 bit
mantissa kısmı (f reel sayısı)	23 bit

$x = \pm q \times 2^m$ reel sayısı **sol-kaymalı normalleştirilmiş ikili sayı** olarak yazılabilir öyle ki mantissadaki sıfırdan farklı ilk bit ikilik noktanın hemen önündedir. Yani $q = (1.\mathrm{f})_2$ dir. Bu bit her zaman 1 kabul edildiğinden, yüklemeye gerek kalmaz. Mantissa $1 \leq q < 2$ aralığındadır. Kelimede mantissa için ayrılan 23 bit f den 23 bit yüklemek için kullanılabilir. Bunun anlamı ise, makinenin kayan-noktalı sayıları için 24 bitlik mantissaya sahip olacağıdır.


Mantissanın işareti (s)
Us (e)
Normalleştirilmiş mantissa (f)
9 bit
23 bit
Taban
noktası

sıfırdan farklı normalleştirilmiş makine sayıları, değerleri aşağıdaki gibi yeniden kodlanan bit alanlarıdır:

$$q = (1.\mathrm{f})_2 \text{ ve } m = \mathrm{e} - 127$$

olmak üzere

$$x = (-1)^{\mathrm{s}} q \times 2^m \tag{5}$$

dir. Burada, $1 \leq q < 2$ ve q daki en anlamlı bit 1 olup, açık olarak yüklenmez. Ayrıca, s de x in işaretini (pozitif: bit 0, negatif: bit 1) temsil eden bittir. $m = e - 127$, 8-bitlik üs ve f de x reel sayısının 23- bitlik kesirli kısmıdır ki baştaki açık 1 bit ile anlamlı rakam alanı $(1.\square\square\square\cdots\square\square\square)_2$ yi verir.
Eşitlik (5) formunda ifade edilen bir reel sayıya **normalleştirilmiş kayan-noktalı formdadır** denir.

$|m|$ nin 8 bitten fazla olmaması kısıtlamasının anlamı

$$0 < \mathrm{e} < (11\ 111\ 111)_2 = 2^8 - 1 = 255$$

olup, $\mathrm{e} = 0$ ve $\mathrm{e} = 255$ değerleri ± 0, $\pm\infty$ ve `NaN` (Sayı Değil) özel durumları için ayrılmıştır. $m = -127$ olduğundan $-126 \leq m \leq 127$ almaktayız ki 32-bitlik bilgisayarda, $2^{-126} \approx 1.2 \times 10^{-38}$ e kadar küçük ve $(2 - 2^{-23})2^{127} \approx 3.4 \times 10^{38}$ e kadar büyük olan sayıları işleyebilir. Bu ise bazı bilimsel hesaplamalar için yeterince geniş bir alan olmayıp, bu ve diğer nedenlerle programlarımızı bazen **çift-duyarlı** yazmak zorunda kalırız.

Yakın Makine Sayıları

Şimdi, 32 bitlik bilgisayarda verilen bir pozitif x reel sayısına, yakın bir makine sayısı ile yaklaşmanın sonucu oluşan hatayı inceleyelim.

$$x = q \times 2^m \qquad 1 \leq q < 2 \qquad -126 \leq m \leq 127$$

kabul edelim. x e en yakın makine sayısının ne olduğunu soralım. Öncelikle, a_i ler 0 veya 1 olmak üzere,

$$x = (1.a_1 a_2 ... a_{23} a_{24} a_{25} ...)_2 \times 2^m$$

yazalım. Yakın bir makine sayısı basitçe $a_{24}a_{25}...$ uzantı bitlerini atarak elde edilebilir. Bu yordama genellikle **yutma** denir. Sonuç

$$x_- = (1.a_1 a_2 ... a_{23})_2 \times 2^m$$

dir. x_- reel doğru üzerinde x in solunda kalır.

Bir başka yakın makine sayısı x in sağında kalır. Bu sayı ise **yuvarlama** ile elde edilir; yani uzantı bitlerini önceki gibi atarak, fakat en son kalan a_{23} bitini bir birim artırarak elde edilir. Bu sayı da

$$x_+ = \left((1.a_1a_2...a_{23})_2 + 2^{-23}\right) \times 2^m$$

dir. x_- ve x_+ nın x e daha yakın olanı bilgisayarda x i temsil etmek için seçilir.

Eğer x, x_- ile daha iyi temsil ediliyorsa,

$$|x - x_-| \leq \tfrac{1}{2} |x_+ - x_-| = \tfrac{1}{2} \times 2^{m-23} = 2^{m-24}$$

olur. Bu durumda **bağıl hata** aşağıdaki şekilde sınırlıdır:

$$\left| \frac{x - x_-}{x} \right| \leq \frac{2^{m-24}}{q \times 2^m} = \frac{1}{q} \times 2^{-24} \leq 2^{-24}$$

İkinci durumda, x, x_+ ya x_- den daha yakın olup,

$$|x - x_+| \leq \tfrac{1}{2} |x_+ - x_-| = 2^{m-24}$$

dür. Aynı analiz bağıl hatanın 2^{-24} den daha büyük olamayacağını gösterir.

Mutlak ve Bağıl Hatalar

Bir x reel sayısına bir başka x^* sayısı ile yaklaşıldığında, **hata** $x - x^*$, **mutlak hata**

$$|x - x^*|$$

ve **bağıl hata**

$$\left|\frac{x - x^*}{x}\right|$$

dir.

Çıkarmada duyarlılık kaybı

$$\begin{aligned} x &= 0.37214\ 78693 \\ y &= 0.37202\ 30572 \\ x-y &= 0.00012\ 48121 \end{aligned}$$

Eğer bu hesap beş-rakam mantissalı bir desimal bilgisayarda gerçekleştirilirse, bu durumda

$$\begin{aligned} \mathrm{fl}(x) &= 0.37215 \\ \mathrm{fl}(y) &= 0.37202 \\ \mathrm{fl}(x)-\mathrm{fl}(y) &= 0.00013 \end{aligned}$$

olur. Bu durumda, bağıl hata oldukça büyüktür:

$$\left|\frac{x-y-[\mathrm{fl}(x)-\mathrm{fl}(y)]}{x-y}\right| = \left|\frac{0.00012\ 48121-0.00013}{0.00012\ 48121}\right| \approx \%4$$

Hemen Hemen Eşit Niceliklerin Çıkarılması

$$y \longleftarrow \sqrt{x^2+1}-1$$

atama deyimi, x in küçük değerleri için çıkarma sadeleştirmesi ve duyarlılık kayıbı içerir. Fonksiyonu

$$y = (\sqrt{x^2+1}-1)\left(\frac{\sqrt{x^2+1}+1}{\sqrt{x^2+1}+1}\right) = \frac{x^2}{\sqrt{x^2+1}+1}$$

şeklinde tekrar yazalım. Böylece, farklı bir atama deyimi

$$y \longleftarrow x^2/(\sqrt{x^2+1}+1)$$

Teorem (Duyarlılık Kaybı)

Eğer x ve y, $x > y$ olacak şekilde pozitif, normalleştirilmiş kayan-noktalı ikilik makine sayıları, ve

$$2^{-q} \leq 1 - \frac{y}{x} \leq 2^{-p}$$

ise, bu durumda $x - y$ çıkarmasında en fazla q ve en az p tane anlamlı ikilik bit kaybolur.

Kararlı ve Kararsız Hesaplamalar

Eğer bir sayısal sürecin bir adımında yapılan küçük hatalar ardışık adımlarda büyüyorsa ve hesaplamanın tamamındaki duyarlılığı ciddi olarak azaltıyorsa, bu sayısal süreç **kararsızdır** denir.

$$\begin{cases} x_0 = 1 \qquad x_1 = \frac{1}{3} \\ x_{n+1} = \frac{13}{3}x_n - \frac{4}{3}x_{n-1} \qquad (n \geq 1) \end{cases} \tag{1}$$

Bu indirgeme bağıntısının

$$x_n = \left(\frac{1}{3}\right)^n \tag{2}$$

dizisini oluşturduğu kolayca görülebilir. (1) dizisi hızla ıraksar (2) ise yakınsar.

Sayısal kararsızlığın bir başka örneği

$$y_n = \int_0^1 x^n e^x dx \qquad (n \geq 0) \tag{3}$$

sayılarının hesaplanması ile elde edilir. y_{n+1} i tanımlayan integrale kısmi integrasyon uygularsak

$$y_{n+1} = e - (n+1)y_n \tag{4}$$

indirgeme bağıntısını elde ederiz. Buradan ve $y_0 = e - 1$ açık gerçeğinden y_1 i elde ederiz:

$$y_1 = e - y_0 = e - (e-1) = 1$$

(4)-bağıntısını kullanarak, 32 `bitlik` bir bilgisayarda $y_2, y_3, \ldots, y_{15}$ i hesaplarsak üç tanesinin sonucu

$$\begin{aligned} y_2 &= 0.71828\ 17 \\ y_{11} &= 1.42245\ 3 \\ y_{15} &= 39711.43 \end{aligned}$$

olup, bunlar *doğru* olamaz, çünkü Denklem (3) den açıkça y dizisi $y_1 > y_2 > \cdots > 0$ ve $\lim_{n\to\infty} y_n = 0$ ı sağlar. (Gerçekten, $0 < x < 1$ için x^n ifadesi monoton olarak 0 a yakınsar.) Bunu bildiğimiz için, Denklem (4) den, $\lim_{n\to\infty}(n+1)y_n = e$ olduğunu görürüz.

Sayfa 62 B.Prob. 1-3, Sayfa 70 B.Prob. 1