Hatalar ve Bilgisayar Aritmetiği

Analitik yollardan çözemediğimiz birçok matematiksel problemi sayısal yöntemlerle bilgisayarlar aracılığı ile çözmeye çalışırız. Bu şekilde Sayısal yöntemler kullanarak elde ettiğimiz sonuçlar çeşitli tipteki hatalardan etkilenmiş olabilir. Çözümdeki hatayı tespit etmek için öncelikle hatanın kaynağını teşhis etmemiz gerekmektedir. Başlıca hata kaynaklarını şu şekilde sıralayabiliriz.

1- **Kullanılan Datadaki Hatalar:** Fizik ve mühendislik problemlerinde deney ve ölçümlerle elde edilmiş veriler kullanılır. Deney ve ölçüm ise kullanılan malzeme ve ortamdan kaynaklanan hatalara açıktır. Bu tür hatalar elde edilen datanın da hatalı olmasına sebep olur.

2- **Yuvarlama Hataları:** Bazı rasyonel sayılar ve tüm irrasyonel sayılar sonsuz sayıda ondalık basamakla ifade edilirler. Fakat bu sayılar bilgisayarda temsil edilirken ancak sonlu sayıda basamak kullanılabileceğinden geriye kalan kısım yuvarlama hatalarını oluşturmaktadır.

3- **Kesme Hataları:** Örneğin sonsuz terime sahip bir seri ifadesini bir yaklaşım olarak kullanmak istediğimizde sonlu sayıda terimden faydalanırız. Geriye kalan terimler ise hata terimi ya da kesme hatası olarak adlandırılır.

4- **İnsan kaynaklı Hatalar:** Yanlış formülasyon, hatalı yazım, yanlış kodlama ve bunun gibi hatalar insan kaynaklı hatalardır.

5- **Matematiksel Modellemede Basitleştirme:** Gerçek hayatta karşılaştığımız problemlerin birer matematik modelini kurarak onlara çözüm üretmeye çalışırız. Kurduğumuz model her zaman birebir olmaz ve bazı basitleştirmeler içerir. İşte bu basitleştirmeler bizi gerçek çözümden biraz uzaklaştırarak hataya sebebiyet verirler.

2-tabanlı Gösterilim

IEEE 754-2008 standardına göre 64-bit bir (2-tabanlı) sistemde reel sayıların gösterilimi

$$(-1)^s\, 2^{c-1023}\, (1+f)$$

şeklindedir. Burada s sayının işaretini (ilk 1-bit), c karakteristik sayıyı (sonraki 11-bit), f ise fraksiyon yani kesirli kısmı (son 52-bit) gösterir. Örneğin,

0 10000000011 1011100100010000000000000000000000000000000000000000

makine sayısının ilk biti 0 olduğundan $s = 0$ yani sayı pozitif, karakteristik kısım ise (10000000011) yani ondalık olarak,

$$c = 1 \cdot 2^{10} + 0 \cdot 2^9 + \cdots + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1024 + 2 + 1 = 1027$$

bu durumda sayının üstel kısmı $2^{1027-1023} = 2^4$. Son 52-bit ise,

1011100100010000000000000000000000000000000000000000

olduğundan,

$$f = 1 \cdot 2^{-1} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-8} + 1 \cdot 2^{-12}$$

Sonuç olarak, bu makine sayısı

$$(-1)^s\, 2^{c-1023}(1+f) = (-1)^0 \cdot 2^{1027-1023} \cdot (1+f) = 27.56640625$$

ondalık sayısını gösterir.

Bununla birlikte bu sayıdan bir küçük makine sayısı

0 10000000011 1011100100001111111111111111111111111111111111111111,

ve bir büyük makine sayısı ise

0 10000000011 1011100100010000000000000000000000000000000000000001.

dır. Yani bu şu demektir bizim orijinal makine sayımız sadece 27.56640625 ondalık sayısını göstermekle kalmaz 27.56640625 ile bir küçük makine sayısı arasındaki sayıların yarısını ve 27.56640625 ile bir büyük makine sayısı arasındaki sayıların yarısını da gösterir. Daha açık olmak gerekirse,

[27.5664062499999982236431605997495353221893310546875,

27.5664062500000017763568394002504646778106689453125) aralığındaki tüm reel sayılar tek bir makine sayısı ile gösterilir.

Bu sistemde gösterilebilecek en küçük pozitif sayı $s = 0$, $c = 1$, and $f = 0$ ‘a karşı gelen

$$2^{-1022} \cdot (1 + 0) \approx 0.22251 \times 10^{-307},$$

Sayısı ve en büyük sayı ise $s = 0$, $c = 2046$, and $f = 1 - 2^{-52}$ ‘a karşı gelen

$$2^{1023} \cdot (2 - 2^{-52}) \approx 0.17977 \times 10^{309}.$$

sayısıdır. Yani, hesaplamalarda ortaya çıkan

$$2^{-1022} \cdot (1 + 0)$$

dan daha küçük sayılar **underflow** hatası verir ve genellikle sıfır kabul edilir. Diğer taraftan

$$2^{1023} \cdot (2 - 2^{-52})$$

den daha büyük sayılar ise **overflow** hatası verir ve hesaplamanın durmasına sebep olur.

Sıfır sayısı için iki gösterilim vardır ; $s = 0$, $c = 0$ and $f = 0$ veya $s = 1$, $c = 0$ and $f = 0$.

Ondalık Makine Sayıları ve Hatalar

2’li sistemde sonlu sayıda sayının tüm reel sayıları temsil etmesi beraberinde bir takım hataları da getirmektedir. Bu durumu ondalık sistem üzerinde incelemeye çalışalım. Normalize edilmiş makine sayılarını k-basamaklı ondalık sistemde

$$\pm 0.d_1 d_2 \cdots d_k \times 10^n$$

biçiminde göstereceğiz. Burada, $1 \le d_1 \le 9$, $0 \le d_i \le 9$, $i = 1,2,\dots,k$ ve n bir tamsayıdır.

Herhangi bir pozitif reel sayı normalize biçimde,

$$0.d_1 d_2 \dots d_k d_{k+1} d_{k+2} \dots \times 10^n$$

şeklinde yazılabilir. Bu sayıyı k-basamaklı bir makine sayısına çevirmenin iki türlü yolu vardır:

a) Yuvarlama: $5 \le d_{k+1} \le 9$ ise d_k bir artırılır ve kesirli kısımda $d_{k+1} d_{k+2} \dots$ basamakları sayıdan atılır, yani sayı

$$0.d'_1 d'_2 \dots d'_k \times 10^n$$

halini alır. d_k bir artırıldığında diğer basamaklarda ve hatta üstel kısım *n* 'de bile değişim olabilir. Burada

$$d'_k = \begin{cases} d_k & , \quad 0 \leq d_k \leq 4 \\ d_k + 1 & , \quad 5 \leq d_k \leq 9 \end{cases}$$

olarak belirlenir.

b) Kesme: Kesirli kısımda $d_{k+1}d_{k+2} \ldots$ basamakları sayıdan atılır başka bir işlem yapılmaz, yani sayı

$$0.d_1 d_2 \ldots d_k \times 10^n$$

halini alır.

Örnek: π irrasyonel sayısının normalize ondalık biçimi,

$$\pi = 0.314159265 \ldots \times 10^1$$

dır. Bu sayıyı 5-basamaklı bir makine sayısına kesme ve yuvarlama ile çevirirsek,

$$fl(\pi) = 0.31415 \times 10^1 \text{ (kesme ile)}$$

$$fl(\pi) = 0.31416 \times 10^1 \text{ (yuvarlama ile)}$$

Burada $fl(\,)$ gösterilimi makine sayısı olduğunu belirtmek için kullanılmıştır.

Her iki yaklaşımda da ortaya çıkan hatayı belirlemek için aşağıdaki tanımları kullanacağız.

Tanım: Eğer p^* , p sayısına bir yaklaşımı temsil ediyorsa,

$$\text{Mutlak Hata} = |p - p^*|$$

ve

$$\text{Bağıl Hata} = \frac{|p-p^*|}{|p|} \quad , \; p \neq 0$$

olarak tanımlanır.

Bağıl hata, mutlak hataya göre yaklaşımın doğruluğu hakkında daha doğru bilgi verir.

Örnek: Aşağıdaki yaklaşımlar için mutlak ve bağıl hatayı belirleyiniz.

(a) $p = 0.3000 \times 10^1$ and $p^* = 0.3100 \times 10^1$;

(b) $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$;

(c) $p = 0.3000 \times 10^4$ and $p^* = 0.3100 \times 10^4$.

Çözüm:

(a) $p = 0.3000 \times 10^1$ and $p^* = 0.3100 \times 10^1$ için mutlak hata 0.1, ve bağıl hata $0.333\overline{3} \times 10^{-1}$ dir.

(b) $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$ için mutlak hata 0.1×10^{-4}, ve bağıl hata $0.333\overline{3}\times 10^{-1}$ dir.

(c) $p = 0.3000 \times 10^4$ and $p^* = 0.3100 \times 10^4$ için mutlak hata 0.1×10^3, ve bağıl hata $0.333\overline{3}\times 10^{-1}$ dir.

Bu örnekte görüldüğü gibi her üç şıkta da bağıl hata $0.333\overline{3}\times 10^{-1}$ olmasına rağmen mutlak hata oldukça farklı değerler almaktadır. Bunun sebebi bağıl hatanın sayının büyüklüğünü de gözönüne almasıdır. Dolayısıyla bağıl hata daha anlamlı sonuç vermektedir.

Örnek: Farz edelim ki, $x = 5/7$ ve $y = 1/3$ olsun. 5-basamak bir sistemde kesme uygulayarak $x + y$, $x - y$, $x \times y$ ve $x \div y$. değerlerini hesaplayınız. Mutlak ve bağıl hataları belirleyiniz.

Çözüm: Bu sayıların ondalık gösterilimi

$$x = \frac{5}{7} = 0.\overline{714285} \quad , \qquad y = \frac{1}{3} = 0.\overline{3}$$

ve 5-basamak makine gösterilimleri ise kesme ile,

$$fl(x) = 0.71428 \times 10^0 \quad , \qquad fl(y) = 0.33333 \times 10^0$$

dır. Bu durumda

$$x + y = fl(\,fl(x) + fl(y)\,) = fl(\,0.71428 \times 10^0 + 0.33333 \times 10^0\,)$$

$$= fl(\,1.04761 \times 10^0\,)$$

$$= 0.10476 \times 10^1.$$

Gerçek değer ise $x + y = 5/7 + 1/3 = 22/21$ olduğundan, Mutlak hata = $|22/21 - 0.10476 \times 10^1|$ = 0.190×10^{-4}

ve

Bağıl hata = $\frac{|22/21 - 0.10476 \times 10^1|}{|22/21|} = 0.182 \times 10^4$

bulunur. Diğer işlemler ise benzer şekilde,

İşlem	Sonuç	Gerçek değer	Mutlak hata	Bağıl hata
$x + y$	0.10476×10^1	22/21	0.190×10^{-4}	0.182×10^{-4}
$x - y$	0.38095×10^0	8/21	0.238×10^{-5}	0.625×10^{-5}
$x \times y$	0.23809×10^0	5/21	0.524×10^{-5}	0.220×10^{-4}
$x \div y$	0.21428×10^1	15/7	0.571×10^{-4}	0.267×10^{-4}

olarak bulunur.

Örnek: Hata üreten en önemli işlemlerin başında birbirine yakın sayıların çıkartılması gelir. $p = 0.54617$ ve $q = 0.54601$ olsun. 4-basamak aritmetik ile $p - q$ sayısına yuvarlama ve kesme ile yaklaşalım.

a) Yuvarlama ile: $p^* = 0.5462$ ve $q^* = 0.5460$ olduğundan

$$p^* - q^* = 0.5462 - 0.5460 = 0.0002$$

diğer taraftan,

$$p - q = 0.54617 - 0.54601 = 0.00016$$

olduğundan

Bağıl hata = $\left|\frac{0.00016 - 0.0002}{0.00016}\right| = 0.25$

bulunur.

b) Kesme ile: $p^* = 0.5461$ ve $q^* = 0.5460$ olduğundan

$$p^* - q^* = 0.5461 - 0.5460 = 0.0001$$

ve

Bağıl hata = $\left|\frac{0.00016 - 0.0001}{0.00016}\right| = 0.375$

bulunur.

görüldüğü gibi çıkarma işlemi sonucunda bağıl hata oldukça büyük çıktı.

Bazı hesaplamalarda işlemlerin yeniden düzenlenmesiyle yuvarlama hataları azaltılabilir. Bunu iki farklı örnekte görelim.

Örnek : $x^2 + 62.10x + 1 = 0$

denkleminin kökleri yaklaşık olarak

$x_1 = -0.01610723$, $x_2 = -62.08390$

4 basamak hassasiyete sahip bir bilgisayar aritmetiğinde

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} = \frac{-62.10 + 62.06}{2.000} = -0.020$$

Bağıl hata oldukça büyük

$$\frac{|-0.01611 + 0.020|}{|-0.01611|} \approx 2.4 \times 10^{-1}.$$

$$x_2 = \frac{-62.10 - 62.06}{2.000} = -62.10 \qquad \text{bağıl hata} \qquad \frac{|-62.08 + 62.10|}{|-62.08|} \approx 3.2 \times 10^{-4}$$

görüldüğü gibi x_1 hesabında çok yakın iki sayı birbirinden çıkarıldığı için hata da büyük oluyor. Şimdi bu sayıyı farklı bir formülle hesaplayalım.

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}\left(\frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}}\right) = \frac{4ac}{2a\left(-b - \sqrt{b^2 - 4ac}\right)} = \frac{-2c}{\left(b + \sqrt{b^2 - 4ac}\right)}$$

$$fl(x_1) = \frac{-2.000}{(62.10 + 62.06)} = -0.1610 \times 10^{-1}$$

Bu durumda bağıl hata yaklaşık olarak,

$$\left|\frac{-0.01610723 + 0.01610}{-0.01610723}\right| \approx 4.5 \times 10^{-4}$$

daha küçük bulunur.

Örnek: $f(x) = x^3 - 6.1x^2 + 3.2x + 1.5$ fonksiyonunu $x = 4.71$ değerinde 3-basamak aritmetik kullanarak hesaplayınız.

	x	x^2	x^3	$6.1x^2$	$3.2x$
Tam Değer	4.71	22.1841	104.487111	135.32301	15.072
3-basamak (kesme)	4.71	22.1	104.	134.	15.0
3-basamak (yuvarlama)	4.71	22.2	105.	135.	15.1

Yukarıdaki tablodan görüldüğü gibi,

$f(4.71) = 104.487111 - 135.32301 + 15.072 + 1.5 = -14.263899$ (Tam)

$f(4.71) = 104. - 134. + 15.0 + 1.5 = -13.5$ (Kesme ile)

$f(4.71) = 105. - 135. + 15.1 + 1.5 = -13.4$ (Yuvarlama ile)

değerleri bulunur. Bu halde,

Bağıl hata = $\left|\frac{-14.263899+13.5}{-14.263899}\right| \approx 0.05$ (Kesme ile)

Bağıl hata = $\left|\frac{-14.263899+13.4}{-14.263899}\right| \approx 0.06$ (Yuvarlama ile)

şeklindedir. Şimdi bu hataları azaltmaya çalışalım.

Eğer $f(x)$ fonksiyonu aşağıdaki gibi düzenlenirse, işlem sayısı azalacağı gibi hata miktarı da azalır.

$$f(x) = \big((x - 6.1)x + 3.2\big)x + 1.5$$

$f(4.71)$ bu şekilde hesaplanırsa,

Bağıl hata = $\left|\frac{-14.263899+14.2}{-14.263899}\right| \approx 0.0045$ (Kesme ile)

Bağıl hata = $\left|\frac{-14.263899+14.3}{-14.263899}\right| \approx 0.0025$ (Yuvarlama ile)

olarak bulunur. Yani hatada ciddi miktarda azalma olduğu gözlemleniyor.