Bilgisayar sayısistemi, aritmetiği ve hata kaynakları

Temel Kavramlar

Erhan Coşkun

Karadeniz Teknik Üniversitesi

Mart, 2015

Bilgisayar sayı sistemi ve hata

Bu bölümde

- hata kavramı,

Bilgisayar sayı sistemi ve hata

Bu bölümde

- hata kavramı,
- bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar,

Bilgisayar sayı sistemi ve hata

Bu bölümde

- hata kavramı,
- bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar,
- taban dönüşümleri ve ilgili hatalar ile

Bilgisayar sayı sistemi ve hata

Bu bölümde

- hata kavramı,
- bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar,
- taban dönüşümleri ve ilgili hatalar ile
- anlam kaybı hatalarını inceleyeceğiz.

Hata(mutlak)

- x ile gösterilen bir büyüklüğün gerçek değeri ile onu temsil eden x_f değeri arasındaki fark, gerçek değerin x_f ile temsilinde oluşan **mutlak hata** veya x_f deki mutlak hata olarak tanımlanır ve $\Delta x = x - x_f$ notasyonu ile gösterilir.

Hata(mutlak)

- x ile gösterilen bir büyüklüğün gerçek değeri ile onu temsil eden x_f değeri arasındaki fark, gerçek değerin x_f ile temsilinde oluşan **mutlak hata** veya x_f deki mutlak hata olarak tanımlanır ve $\Delta x = x - x_f$ notasyonu ile gösterilir.
- $x = 10.1$ birim uzunluğa sahip sahip olan cismin uzunluğu $x_f = 10$ birim olarak ölçülmüşse, bu ölçüm sonucunda oluşan mutlak hata $\Delta x = 0.1$ değerine eşittir.

Hata(mutlak)

- x ile gösterilen bir büyüklüğün gerçek değeri ile onu temsil eden x_f değeri arasındaki fark, gerçek değerin x_f ile temsilinde oluşan **mutlak hata** veya x_f deki mutlak hata olarak tanımlanır ve $\Delta x = x - x_f$ notasyonu ile gösterilir.
- $x = 10.1$ birim uzunluğa sahip sahip olan cismin uzunluğu $x_f = 10$ birim olarak ölçülmüşse, bu ölçüm sonucunda oluşan mutlak hata $\Delta x = 0.1$ değerine eşittir.
- $x = 1.1$ birim gerçek uzunluğuna sahip olan başka bir cismin uzunluğu $x_f = 1$ birim olarak ölçülmüşse, bu ölçüm sonucunda oluşan mutlak hata da $\Delta x = 0.1$ değerine sahiptir.

Hata(mutlak)

- x ile gösterilen bir büyüklüğün gerçek değeri ile onu temsil eden x_f değeri arasındaki fark, gerçek değerin x_f ile temsilinde oluşan **mutlak hata** veya x_f deki mutlak hata olarak tanımlanır ve $\Delta x = x - x_f$ notasyonu ile gösterilir.
- $x = 10.1$ birim uzunluğa sahip sahip olan cismin uzunluğu $x_f = 10$ birim olarak ölçülmüşse, bu ölçüm sonucunda oluşan mutlak hata $\Delta x = 0.1$ değerine eşittir.
- $x = 1.1$ birim gerçek uzunluğuna sahip olan başka bir cismin uzunluğu $x_f = 1$ birim olarak ölçülmüşse, bu ölçüm sonucunda oluşan mutlak hata da $\Delta x = 0.1$ değerine sahiptir.
- Ancak ikinci ölçümde daha fazla hata yaptığımızı düşünürüz !

Hata(bağıl)

- Hatanın diğer bir ölçüsü ise, yapılan hatanın gerçek değere oranı olarak tanımlanan ve $\varepsilon_b(x) = \Delta x/x, x \neq 0$, notasyonu ile göstereceğimiz **bağıl hatadır**

Hata(bağıl)

- Hatanın diğer bir ölçüsü ise, yapılan hatanın gerçek değere oranı olarak tanımlanan ve $\varepsilon_b(x) = \Delta x/x, x \neq 0$, notasyonu ile göstereceğimiz **bağıl hatadır**
- Buna göre yukarıdaki birinci ölçüm için oluşan bağıl hata $\varepsilon_b(x) = 0.01$

Hata(bağıl)

- Hatanın diğer bir ölçüsü ise, yapılan hatanın gerçek değere oranı olarak tanımlanan ve $\varepsilon_b(x) = \Delta x/x, x \neq 0$, notasyonu ile göstereceğimiz **bağıl hatadır**
- Buna göre yukarıdaki birinci ölçüm için oluşan bağıl hata $\varepsilon_b(x) = 0.01$
- ikincide oluşan hata ise $\varepsilon_b(x) = 0.1$ dir

Hata(bağıl)

- Hatanın diğer bir ölçüsü ise, yapılan hatanın gerçek değere oranı olarak tanımlanan ve $\varepsilon_b(x) = \Delta x/x, x \neq 0$, notasyonu ile göstereceğimiz **bağıl hatadır**
- Buna göre yukarıdaki birinci ölçüm için oluşan bağıl hata $\varepsilon_b(x) = 0.01$
- ikincide oluşan hata ise $\varepsilon_b(x) = 0.1$ dir
- yani ikinci ölçüm sonucu yapılan bağıl hata beklentilerimiz doğrultusunda daha büyüktür.

Hata(bağıl)

- Hatanın diğer bir ölçüsü ise, yapılan hatanın gerçek değere oranı olarak tanımlanan ve $\varepsilon_b(x) = \Delta x/x, x \neq 0$, notasyonu ile göstereceğimiz **bağıl hatadır**
- Buna göre yukarıdaki birinci ölçüm için oluşan bağıl hata $\varepsilon_b(x) = 0.01$
- ikincide oluşan hata ise $\varepsilon_b(x) = 0.1$ dir
- yani ikinci ölçüm sonucu yapılan bağıl hata beklentilerimiz doğrultusunda daha büyüktür.
- O halde bağıl hata günlük hayatta algıladığımız hata ile daha uyumludur.

Yuvarlama hataları

- Sayısal analiz sürecinde oluşan hataların bir kısmı bilgisayar sayı sistemi ile ilgilidir ve bu tür hatalar genelde **yuvarlama hataları** olarak bilinirler.

Yuvarlama hataları

- Sayısal analiz sürecinde oluşan hataların bir kısmı bilgisayar sayı sistemi ile ilgilidir ve bu tür hatalar genelde **yuvarlama hataları** olarak bilinirler.
- Örneğin π sayısının bilgisayarda reel bir sayı için tahsis edilen alana sığdırılması için belirli basamaktan sonraki basamaklarının ihmal edilmesi ile oluşan hata bir yuvarlama hatasıdır.

Yuvarlama hataları

- Sayısal analiz sürecinde oluşan hataların bir kısmı bilgisayar sayı sistemi ile ilgilidir ve bu tür hatalar genelde **yuvarlama hataları** olarak bilinirler.
- Örneğin π sayısının bilgisayarda reel bir sayı için tahsis edilen alana sığdırılması için belirli basamaktan sonraki basamaklarının ihmal edilmesi ile oluşan hata bir yuvarlama hatasıdır.
- $1/3 = 0.333...$ gibi her devirli rasyonel sayının sınırlı bilgisayar bellek alanında temsili esnasında bu tür yuvarlama hataları kaçınılmazdır.

Yuvarlama hataları

- Sayısal analiz sürecinde oluşan hataların bir kısmı bilgisayar sayı sistemi ile ilgilidir ve bu tür hatalar genelde **yuvarlama hataları** olarak bilinirler.
- Örneğin π sayısının bilgisayarda reel bir sayı için tahsis edilen alana sığdırılması için belirli basamaktan sonraki basamaklarının ihmal edilmesi ile oluşan hata bir yuvarlama hatasıdır.
- $1/3 = 0.333...$ gibi her devirli rasyonel sayının sınırlı bilgisayar bellek alanında temsili esnasında bu tür yuvarlama hataları kaçınılmazdır.
- Öte yandan 0.1 sayısıikili sistemin devirli bir sayısıdır ve ikili sistem kullanan bilgisayar sisteminde tam olarak temsil edilemez.

Kesme hataları

- bir fonksiyon yerine $n-$inci dereceden Taylor polinomunun kullanılması(Bölüm 3, Taylor polinomları ve hata)

Kesme hataları

- bir fonksiyon yerine $n-$inci dereceden Taylor polinomunun kullanılması(Bölüm 3, Taylor polinomları ve hata)
- interpolasyon polinomu veya basit fonksiyonlarla yaklaşım(Bölüm 6, interpolasyon)

Kesme hataları

- bir fonksiyon yerine $n-$inci dereceden Taylor polinomunun kullanılması(Bölüm 3, Taylor polinomları ve hata)
- interpolasyon polinomu veya basit fonksiyonlarla yaklaşım(Bölüm 6, interpolasyon)
- sonsuz toplam yerine sonlu bir toplamla integrale yaklaşım(Bölüm 7, Sayısal integrasyon),

Kesme hataları

- bir fonksiyon yerine $n-$inci dereceden Taylor polinomunun kullanılması(Bölüm 3, Taylor polinomları ve hata)
- interpolasyon polinomu veya basit fonksiyonlarla yaklaşım(Bölüm 6, interpolasyon)
- sonsuz toplam yerine sonlu bir toplamla integrale yaklaşım(Bölüm 7, Sayısal integrasyon),
- teğet eğimi(türev) yerine kiriş eğimi(Bölüm 8, Sayısal türev)

Kesme hataları

- bir fonksiyon yerine $n-$inci dereceden Taylor polinomunun kullanılması(Bölüm 3, Taylor polinomları ve hata)
- interpolasyon polinomu veya basit fonksiyonlarla yaklaşım(Bölüm 6, interpolasyon)
- sonsuz toplam yerine sonlu bir toplamla integrale yaklaşım(Bölüm 7, Sayısal integrasyon),
- teğet eğimi(türev) yerine kiriş eğimi(Bölüm 8, Sayısal türev)
- diferensiyel denklemlerde türev yerine sayısal türev kullanılması(Bölüm 9-12) durumunda oluşan hatalar kesme hatalarıdır.

Sayısal analiz ve hata

- Sayısal analiz, yuvarlama ve kesme hatalarını minimize edecek yaklaşımları belirlemeyi amaçlar

Sayısal analiz ve hata

- Sayısal analiz, yuvarlama ve kesme hatalarını minimize edecek yaklaşımları belirlemeyi amaçlar
- Sayısal analizde amaç yuvarlama hatalarını kontrol altında tutan ve "makul" düzeyde kesme hataları ile işlemler yürüten sayısal yöntemler geliştirmek ve problemlerin çözümüne uygulamaktır.

Sayısal analiz ve hata

- Sayısal analiz, yuvarlama ve kesme hatalarını minimize edecek yaklaşımları belirlemeyi amaçlar
- Sayısal analizde amaç yuvarlama hatalarını kontrol altında tutan ve "makul" düzeyde kesme hataları ile işlemler yürüten sayısal yöntemler geliştirmek ve problemlerin çözümüne uygulamaktır.
- Sayısal analiz sürecinde oluşabilecek olan yuvarlama hatalarını anlayabilmek için öncelikle bilgisayar sayı sistemini ve bu sistem üzerindeki aritmetiği yakından inceleyelim.

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- R, reel sayılar kümesi olmak üzere, bilgisayarlar sadece R_f ile göstereceğimiz

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- R, reel sayılar kümesi olmak üzere, bilgisayarlar sadece R_f ile göstereceğimiz
- R nin sonlu elemanlı bir alt kümesini belleklerinde saklayabilir ve bu sayılarla işlem yapabilirler.

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- R, reel sayılar kümesi olmak üzere, bilgisayarlar sadece R_f ile göstereceğimiz
- R nin sonlu elemanlı bir alt kümesini belleklerinde saklayabilir ve bu sayılarla işlem yapabilirler.
- Bilgisayarlar tamsayıları tamsayı formatı(*fixed point format*)

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- R, reel sayılar kümesi olmak üzere, bilgisayarlar sadece R_f ile göstereceğimiz
- R nin sonlu elemanlı bir alt kümesini belleklerinde saklayabilir ve bu sayılarla işlem yapabilirler.
- Bilgisayarlar tamsayıları tamsayı formatı(*fixed point format*)
- kesirli sayıları ise kayan nokta formatı(*floating point format*) adı verilen bir formatta saklarlar.

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- Bilgisayar belleğinde saklanabilen sıfırdan farklı bir kayan noktalı sayı

$$x_f = \pm q_f \times \beta^e$$

biçiminde ifade edilebilen bir sayıdır.

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- Bilgisayar belleğinde saklanabilen sıfırdan farklı bir kayan noktalı sayı

$$x_f = \pm q_f \times \beta^e$$

biçiminde ifade edilebilen bir sayıdır.

- Burada q_f sonlu basamağa sahip olan ve $q_f = 0.d_1 d_2 ... d_n$ biçiminde ifade edilebilen sayının kesir kısmı(mantis veya $d_1 \neq 0$ için normalize edilmiş mantis, $0 \leq d_i < \beta, i = 1, 2, ..., n$)

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- Bilgisayar belleğinde saklanabilen sıfırdan farklı bir kayan noktalı sayı

$$x_f = \pm q_f \times \beta^e$$

biçiminde ifade edilebilen bir sayıdır.

- Burada q_f sonlu basamağa sahip olan ve $q_f = 0.d_1 d_2 ... d_n$ biçiminde ifade edilebilen sayının kesir kısmı(mantis veya $d_1 \neq 0$ için normalize edilmiş mantis, $0 \leq d_i < \beta, i = 1, 2, ..., n$)
- β ise kullanılan sayı sisteminin tabanı(örneğin iki tabanlı sistem için $\beta = 2$,on tabanlı sistem için $\beta = 10$ vb)

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- Bilgisayar belleğinde saklanabilen sıfırdan farklı bir kayan noktalı sayı

$$x_f = \pm q_f \times \beta^e$$

biçiminde ifade edilebilen bir sayıdır.

- Burada q_f sonlu basamağa sahip olan ve $q_f = 0.d_1 d_2 ... d_n$ biçiminde ifade edilebilen sayının kesir kısmı(mantis veya $d_1 \neq 0$ için normalize edilmiş mantis, $0 \leq d_i < \beta, i = 1, 2, ..., n$)
- β ise kullanılan sayı sisteminin tabanı(örneğin iki tabanlı sistem için $\beta = 2$,on tabanlı sistem için $\beta = 10$ vb)
- e ise üs olarak adlandırılır.

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- Bilgisayar belleğinde saklanabilen sıfırdan farklı bir kayan noktalı sayı

$$x_f = \pm q_f \times \beta^e$$

biçiminde ifade edilebilen bir sayıdır.

- Burada q_f sonlu basamağa sahip olan ve $q_f = 0.d_1 d_2 ... d_n$ biçiminde ifade edilebilen sayının kesir kısmı(mantis veya $d_1 \neq 0$ için normalize edilmiş mantis, $0 \leq d_i < \beta, i = 1, 2, ..., n$)
- β ise kullanılan sayı sisteminin tabanı(örneğin iki tabanlı sistem için $\beta = 2$,on tabanlı sistem için $\beta = 10$ vb)
- e ise üs olarak adlandırılır.
- Böylece bir kayan noktalı sayıyı tanımlayabilmek için *işaret, mantis* ve *üs* bilgileri gerekmektedir.

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- Bilgisayar belleğinde saklanabilen sıfırdan farklı bir kayan noktalı sayı

$$x_f = \pm q_f \times \beta^e$$

biçiminde ifade edilebilen bir sayıdır.

- Burada q_f sonlu basamağa sahip olan ve $q_f = 0.d_1 d_2 ... d_n$ biçiminde ifade edilebilen sayının kesir kısmı(mantis veya $d_1 \neq 0$ için normalize edilmiş mantis, $0 \leq d_i < \beta, i = 1, 2, ..., n$)
- β ise kullanılan sayı sisteminin tabanı(örneğin iki tabanlı sistem için $\beta = 2$,on tabanlı sistem için $\beta = 10$ vb)
- e ise üs olarak adlandırılır.
- Böylece bir kayan noktalı sayıyı tanımlayabilmek için *işaret*, *mantis* ve *üs* bilgileri gerekmektedir.
- Mantisdeki noktanın pozisyonu, üs değiştirilmek suretiyle sağa veya sola kaydırılabileceği için *kayan noktalı sayılar* adı verilmektedir.

$$0.2 \times 10^{-1} = 0.02 \times 10^0 = 2.0 \times 10^{-2}$$

- En genel halde β tabanlı sistemde

$$x = \pm q \times \beta^e, q = 0.d_1 d_2 \cdots d_{n-1} d_n d_{n+1} \cdots, d_1 \neq 0$$

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- En genel halde β tabanlı sistemde

$$x = \pm q \times \beta^e, q = 0.d_1 d_2 \cdots d_{n-1} d_n d_{n+1} \cdots, d_1 \neq 0$$

- bilgisayar sayı sistemi bellek sınırlamaları dolayısıyla mantisinde en fazla n basamağa izin veriyorsa bu durumda

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- En genel halde β tabanlı sistemde

$$x = \pm q \times \beta^e, q = 0.d_1 d_2 \cdots d_{n-1} d_n d_{n+1} \cdots, d_1 \neq 0$$

- bilgisayar sayı sistemi bellek sınırlamaları dolayısıyla mantisinde en fazla n basamağa izin veriyorsa bu durumda
- $$q_f = \begin{cases} 0.d_1 d_2 \cdots d_{n-1} d_n & 0 \leq d_{n+1} \ldots < \beta/2 \\ 0.d_1 d_2 \cdots d_{n-1}(d_n + 1) & d_{n+1} \ldots \geq \beta/2 \end{cases}$$

Bilgisayar sayı sistemi, aritmetiği ve ilgili hatalar

- En genel halde β tabanlı sistemde

$$x = \pm q \times \beta^e, q = 0.d_1 d_2 \cdots d_{n-1} d_n d_{n+1} \cdots, d_1 \neq 0$$

- bilgisayar sayı sistemi bellek sınırlamaları dolayısıyla mantisinde en fazla n basamağa izin veriyorsa bu durumda
-

$$q_f = \begin{cases} 0.d_1 d_2 \cdots d_{n-1} d_n & 0 \leq d_{n+1} \ldots < \beta/2 \\ 0.d_1 d_2 \cdots d_{n-1}(d_n + 1) & d_{n+1} \ldots \geq \beta/2 \end{cases}$$

- $x_f = \pm q_f \times \beta^e$ olarak elde edilir.

- Bu durumda x değeri en yakın x_f bilgisayar sayısına yuvarlanmış olur ve $\varepsilon_b(x)$ ise bağıl yuvarlama hatasıdır ve

$$|\varepsilon_b(x)| \leq \frac{1}{2}\beta^{-n+1} \tag{1}$$

- Bu durumda x değeri en yakın x_f bilgisayar sayısına yuvarlanmış olur ve $\varepsilon_b(x)$ ise bağıl yuvarlama hatasıdır ve

$$|\varepsilon_b(x)| \leq \frac{1}{2}\beta^{-n+1} \tag{1}$$

- $\frac{1}{2}\beta^{-n+1}$ sayısına *bilgisayar hassasiyeti*(machine precision) adı verilir .

- Bu durumda x değeri en yakın x_f bilgisayar sayısına yuvarlanmış olur ve $\varepsilon_b(x)$ ise bağıl yuvarlama hatasıdır ve

$$|\varepsilon_b(x)| \leq \frac{1}{2}\beta^{-n+1} \tag{1}$$

- $\frac{1}{2}\beta^{-n+1}$ sayısına *bilgisayar hassasiyeti*(machine precision) adı verilir .
- $>>$ eps(1)
  ans $=$
  2.2204e-016$(= 2^{-52})$
  dir.

Sanal bir sayı sistemi

Sanal bir kayan nokta sistemi örneği üzerinde bilgisayar aritmetiği

- Sanal sistemimizin üs için $-1, 0, 1$ ve mantis için $q = 0.d_1, d_1 = 1, \ldots, 9$ değerlerini alabileceğini kabul edelim.

Sanal bir sayı sistemi

Sanal bir kayan nokta sistemi örneği üzerinde bilgisayar aritmetiği

- Sanal sistemimizin üs için $-1, 0, 1$ ve mantis için $q = 0.d_1, d_1 = 1, \ldots, 9$ değerlerini alabileceğini kabul edelim.
- Sistemimiz tabloda belirtilen noktalar ve negatifleri ile sıfır noktasından oluşur(toplam 55 nokta!):

-1	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1	1	2	3	4	5	6	7	8	9

Table: e(sol sütun) ve Sanal sistem üzerinde kayan noktalı pozitif sayılar

R_f nin özellikleri

- R_f toplama işlemine göre kapalı olmayabilir: Örneğimiz için $5, 6 \in R_f$ fakat $11 \notin R_f$ dir.

R_f nin özellikleri

- R_f toplama işlemine göre kapalı olmayabilir: Örneğimiz için $5, 6 \in R_f$ fakat $11 \notin R_f$ dir.
- R_f de belirli değerden küçük olan sonuçlar sıfır olarak kabul edilir(sistemler bu durumu "undeflow" hatası olarak yorumlarlar):

$$a = 0.01/2 = 0.$$

R_f nin özellikleri

- R_f toplama işlemine göre kapalı olmayabilir: Örneğimiz için $5, 6 \in R_f$ fakat $11 \notin R_f$ dir.
- R_f de belirli değerden küçük olan sonuçlar sıfır olarak kabul edilir(sistemler bu durumu "undeflow" hatası olarak yorumlarlar):

$$a = 0.01/2 = 0.$$

- R_f de toplama işlemine göre birleşme özelliği geçerli olmayabilir:

$$(0.3 + 0.4) + 1 = 2 \neq 1 = 0.3 + (0.4 + 1)$$

Gerçek bir kayan nokta sistemi örneği üzerinde bilgisayar aritmetiği ve hata kaynakları

- R_f *sonlu sayıda elemandan oluşur.*

Gerçek bir kayan nokta sistemi örneği üzerinde bilgisayar aritmetiği ve hata kaynakları

- R_f *sonlu sayıda elemandan oluşur.*
- $>>$ realmax
  ans $=1.7977E+308$

Gerçek bir kayan nokta sistemi örneği üzerinde bilgisayar aritmetiği ve hata kaynakları

- R_f *sonlu sayıda elemandan oluşur.*

- $>>$ realmax
  ans $= 1.7977E + 308$
- O halde kullandığımız bilgisayar $[-1.7977E + 308, 1.7977E + 308]$ aralığı içerisindeki kayan nokta sayılarından oluşan sonlu bir kümeden ibarettir.

Gerçek bir kayan nokta sistemi örneği üzerinde bilgisayar aritmetiği ve hata kaynakları

- R_f *sonlu sayıda elemandan oluşur.*

- $>>$ realmax
  ans $= 1.7977E+308$
- O halde kullandığımız bilgisayar $[-1.7977E+308, 1.7977E+308]$ aralığı içerisindeki kayan nokta sayılarından oluşan sonlu bir kümeden ibarettir.
- Herhangi bir işlemin sonucunun bu aralığın dışarında bir değer üretmesi durumunda sonuç çok büyük sayı(*overflow veya inf*) hatası olarak yorumlanır.

Gerçek bir kayan nokta sistemi örneği üzerinde bilgisayar aritmetiği ve hata kaynakları

- R_f *sonlu sayıda elemandan oluşur.*

- $>>$ realmax
  ans $=1.7977E+308$
- O halde kullandığımız bilgisayar $[-1.7977E+308, 1.7977E+308]$ aralığı içerisindeki kayan nokta sayılarından oluşan sonlu bir kümeden ibarettir.
- Herhangi bir işlemin sonucunun bu aralığın dışarında bir değer üretmesi durumunda sonuç çok büyük sayı(*overflow veya inf*) hatası olarak yorumlanır.
- Örneğin MATLAB/Octave ortamında
  $>>1.797693134862316E+308+1e292$

$$ans = \text{inf}$$

$x = 2.2251E-308$.

Bilgisayar aritmetiği ve hata kaynakları

- >> x=realmin

Bilgisayar aritmetiği ve hata kaynakları

- >> x=realmin
- Bunlara ilaveten belirsizlik adını verdiğimiz tanımlı olmayan (0/0, ∞/∞ gibi) bir işlem sonucu ise *NaN*(Not a Number- sayı değil) olarak kabul edilir.

Bilgisayar aritmetiği ve hata kaynakları

- >> x=realmin
- Bunlara ilaveten belirsizlik adını verdiğimiz tanımlı olmayan (0/0, ∞/∞ gibi) bir işlem sonucu ise *NaN*(Not a Number- sayı değil) olarak kabul edilir.
- *Hiç bir irrasyonel sayı* ($\pi, e, \sqrt{2}$) *veya devirli rasyonel sayı* R_f *de yer almaz.*

Bilgisayar aritmetiği ve hata kaynakları

- *Kayan nokta sayı sisteminde sayılar sıfır noktasının komşuluğunda daha sıkça serpiştirilmiştir ve sonuç olarak işlemlerin gerçekleştirilme sırası farklı yuvarlama hatalarına neden olabilir ve bu durum işlem sonucunu değiştirebilir(Toplama işlemine göre birleşme özelliği olmayabilir).*

Bilgisayar aritmetiği ve hata kaynakları

- *Kayan nokta sayı sisteminde sayılar sıfır noktasının komşuluğunda daha sıkça serpiştirilmiştir ve sonuç olarak işlemlerin gerçekleştirilme sırası farklı yuvarlama hatalarına neden olabilir ve bu durum işlem sonucunu değiştirebilir(Toplama işlemine göre birleşme özelliği olmayabilir).*
- Bu sonucu MATLAB/Octave yazılımlarında kullanılan *eps*() fonksiyonu yardımıyla gözlemleyebiliriz.

Bilgisayar aritmetiği ve hata kaynakları

- *Kayan nokta sayı sisteminde sayılar sıfır noktasının komşuluğunda daha sıkça serpiştirilmiştir ve sonuç olarak işlemlerin gerçekleştirilme sırası farklı yuvarlama hatalarına neden olabilir ve bu durum işlem sonucunu değiştirebilir(Toplama işlemine göre birleşme özelliği olmayabilir).*
- Bu sonucu MATLAB/Octave yazılımlarında kullanılan $eps()$ fonksiyonu yardımıyla gözlemleyebiliriz.
- Bu fonksiyon $eps(x)$ biçimindeki kullanımıyla, x sayısı ile bu sayıya en yakın bilgisayar sistemindeki sayı arasındaki uzaklığı verir.

x	$\lvert x - x_f \rvert$
1×10^{-2}	$1.7347E-018$
1	$2.2204E-016$
1×10^{2}	$1.4211E-014$
1×10^{4}	$1.8190E-012$
1×10^{8}	$1.4901E-008$
1×10^{16}	2
1×10^{32}	$1.8014E+016$

Table: Sayılar ve komşuları arasındaki uzaklık

Bilgisayar aritmetiği ve hata kaynakları

- Sıfır komşuluğundaki sayıların birbirlerine çok yakın olması ve mutlak değerce büyük olan kayan nokta sayıları arasındaki uzaklığın kısmen daha büyük olması sonucu sayıların küçükten büyüğe veya büyükten küçüğe sıralanarak toplanması sonucu farklı sonuçlar elde edilebilmektedir.

Bilgisayar aritmetiği ve hata kaynakları

- Sıfır komşuluğundaki sayıların birbirlerine çok yakın olması ve mutlak değerce büyük olan kayan nokta sayıları arasındaki uzaklığın kısmen daha büyük olması sonucu sayıların küçükten büyüğe veya büyükten küçüğe sıralanarak toplanması sonucu farklı sonuçlar elde edilebilmektedir.
- Örneğin

$$\sum_{i=1}^{N} \frac{1}{i^2} = 1 + 1/2^2 + \cdots + 1/N^2$$

Bilgisayar aritmetiği ve hata kaynakları

- Sıfır komşuluğundaki sayıların birbirlerine çok yakın olması ve mutlak değerce büyük olan kayan nokta sayıları arasındaki uzaklığın kısmen daha büyük olması sonucu sayıların küçükten büyüğe veya büyükten küçüğe sıralanarak toplanması sonucu farklı sonuçlar elde edilebilmektedir.
- Örneğin

$$\sum_{i=1}^{N} \frac{1}{i^2} = 1 + 1/2^2 + \cdots + 1/N^2$$

- $N = 1e5$ için

```
top=0; for i=1:N top=top+1/(i*i); end
```

program parçacığı çalıştırıldığında

$$top = 1.64492406689824$$

- Şimdi de

$$\sum_{i=1}^{N} \frac{1}{(N-(i-1))^2} = 1/N^2 + 1/(N-1)^2 + \cdots + 1/1^2$$

olarak ifade edilen küçükten büyüğe toplamını hesaplayalım:

- Şimdi de

$$\sum_{i=1}^{N} \frac{1}{(N-(i-1))^2} = 1/N^2 + 1/(N-1)^2 + \cdots + 1/1^2$$

olarak ifade edilen küçükten büyüğe toplamını hesaplayalım:

- $$top = 1.64492406689823$$

elde ederiz. Sonuçların farklı olduğunu görüyoruz.

$$\sum_{i=1}^{\infty} \frac{1}{i^2} = \pi^2/6 = 1.64493406684823$$

olduğu dikkate alınırsa küçükten büyüğe toplamın daha doğru sonuç ürettiğini gözlemleriz.

Bilgisayar aritmetiği ve hata kaynakları

- Aynı nedenden dolayı farklı mertebeden sayılarla gerçekleştirilen işlemlerde büyük hatalar oluşabilir. Bu duruma örnek olarak aşağıdaki işlem sonuçlarını inceleyelim:

Bilgisayar aritmetiği ve hata kaynakları

- Aynı nedenden dolayı farklı mertebeden sayılarla gerçekleştirilen işlemlerde büyük hatalar oluşabilir. Bu duruma örnek olarak aşağıdaki işlem sonuçlarını inceleyelim:

İşlem	Yaklaşık hata
$0 + 5E-324 = 4.940656458412465E-324$	$6E-326$
$1E5 \times (1 - 1E-17) = 1E5$	$1E-12$
$1E10 \times (1E16 + 1E-1) = 1E26$	$1E+09$

Table: Farklı mertebeden büyüklüklere sahip sayılarla işlemler ve oluşan hata

Taban dönüşümleri ve ilgili hatalar

- İki tabanlı sistemden on tabanlı sisteme dönüşüm

Taban dönüşümleri ve ilgili hatalar

- İki tabanlı sistemden on tabanlı sisteme dönüşüm
-

$$\begin{aligned}(101101)_2 &= 1\times 2^5 + 0\times 2^4 + 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 \\ &= 32 + 8 + 4 + 1 = 45.\end{aligned}$$

Taban dönüşümleri ve ilgili hatalar

- İki tabanlı sistemden on tabanlı sisteme dönüşüm
-

$$\begin{aligned}(101101)_2 &= 1\times 2^5+0\times 2^4+1\times 2^3+1\times 2^2+0\times 2^1+1\times 2^0\\ &= 32+8+4+1=45.\end{aligned}$$

-

$$\begin{aligned}(101.101)_2 &= 1\times 2^2+0\times 2^1+1\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}\\ &= 5+1/2+1/8=5.625.\end{aligned}$$

Taban dönüşümleri ve ilgili hatalar

- On tabanlı sistemden iki tabanlı sisteme dönüşüm

Taban dönüşümleri ve ilgili hatalar

- On tabanlı sistemden iki tabanlı sisteme dönüşüm
-

$$1964 = c_{10}2^{10} + c_9 2^9 + \cdots c_1 2^1 + c_0 2^0$$

Taban dönüşümleri ve ilgili hatalar

- On tabanlı sistemden iki tabanlı sisteme dönüşüm
- $$1964 = c_{10}2^{10} + c_9 2^9 + \cdots c_1 2^1 + c_0 2^0$$
- $$1964 = 2 \times 982 + c_0$$

Taban dönüşümleri ve ilgili hatalar

- On tabanlı sistemden iki tabanlı sisteme dönüşüm
- $$1964 = c_{10}2^{10} + c_9 2^9 + \cdots c_1 2^1 + c_0 2^0$$
- $$1964 = 2 \times 982 + c_0$$
- Şimdi ise eşitliğin her iki tarafını ikiye bölerek elde edilen

$$982 = c_{10}2^9 + c_9 2^8 + \cdots + c_2 2^1 + c_1$$

ifadesinden de c_1 in 982 nin 2 ye bölümünden elde edilen kalan olduğuna dikkat edelim. Yani

$$982 = 2 \times 491 + c_1$$

Taban dönüşümleri ve ilgili hatalar

$$
\begin{aligned}
1964 &= 982 \times 2 + c_0 \Rightarrow c_0 = 0 \\
982 &= 491 \times 2 + c_1 \Rightarrow c_1 = 0 \\
491 &= 245 \times 2 + c_2 \Rightarrow c_2 = 1 \\
245 &= 122 \times 2 + c_3 \Rightarrow c_3 = 1 \\
122 &= 61 \times 2 + c_4 \Rightarrow c_4 = 0 \\
61 &= 30 \times 2 + c_5 \Rightarrow c_5 = 1 \\
30 &= 15 \times 2 + c_6 \Rightarrow c_6 = 0 \\
15 &= 7 \times 2 + c_7 \Rightarrow c_7 = 1 \\
7 &= 3 \times 2 + c_8 \Rightarrow c_8 = 1 \\
3 &= 1 \times 2 + c_9 \Rightarrow c_9 = 1 \\
1 &= 0 \times 2 + c_{10} \Rightarrow c_{10} = 1
\end{aligned}
$$

Taban dönüşümleri ve ilgili hatalar

$$
\begin{aligned}
1964 &= 982 \times 2 + c_0 \Rightarrow c_0 = 0 \\
982 &= 491 \times 2 + c_1 \Rightarrow c_1 = 0 \\
491 &= 245 \times 2 + c_2 \Rightarrow c_2 = 1 \\
245 &= 122 \times 2 + c_3 \Rightarrow c_3 = 1 \\
122 &= 61 \times 2 + c_4 \Rightarrow c_4 = 0 \\
61 &= 30 \times 2 + c_5 \Rightarrow c_5 = 1 \\
30 &= 15 \times 2 + c_6 \Rightarrow c_6 = 0 \\
15 &= 7 \times 2 + c_7 \Rightarrow c_7 = 1 \\
7 &= 3 \times 2 + c_8 \Rightarrow c_8 = 1 \\
3 &= 1 \times 2 + c_9 \Rightarrow c_9 = 1 \\
1 &= 0 \times 2 + c_{10} \Rightarrow c_{10} = 1
\end{aligned}
$$

- $1964 = (11110101100)_2$ iki tabanlı gösterimi elde edilir.

Taban dönüşümleri ve ilgili hatalar

- Kesirli sayı dönüşümleri

Taban dönüşümleri ve ilgili hatalar

- Kesirli sayı dönüşümleri
- Öncelikle $[[x]]$ notasyonu ile x sayısının tam değerini ve $\{x\}$ ile de kesirli kısmını gösterelim.

Taban dönüşümleri ve ilgili hatalar

- Kesirli sayı dönüşümleri
- Öncelikle $[[x]]$ notasyonu ile x sayısının tam değerini ve $\{x\}$ ile de kesirli kısmını gösterelim.
- $A = 0.125$ sayısının ikili sistemdeki gösterimini belirleyiniz.

Taban dönüşümleri ve ilgili hatalar

- Kesirli sayı dönüşümleri
- Öncelikle $[[x]]$ notasyonu ile x sayısının tam değerini ve $\{x\}$ ile de kesirli kısmını gösterelim.
- $A = 0.125$ sayısının ikili sistemdeki gösterimini belirleyiniz.
- Onlu sistemde

$$A = 0.125 = 1 \times 10^{-1} + 2 \times 10^{-2} + 5 \times 10^{-3}$$

Taban dönüşümleri ve ilgili hatalar

- Kesirli sayı dönüşümleri
- Öncelikle $[[x]]$ notasyonu ile x sayısının tam değerini ve $\{x\}$ ile de kesirli kısmını gösterelim.
- $A = 0.125$ sayısının ikili sistemdeki gösterimini belirleyiniz.
- Onlu sistemde

$$A = 0.125 = 1 \times 10^{-1} + 2 \times 10^{-2} + 5 \times 10^{-3}$$

- Aynı sayıyı iki tabanına göre

$$A = 0.125 = (0.d_1 d_2 d_3 d_4 \ldots) = d_1 \times 2^{-1} + d_2 \times 2^{-2} + d_3 \times 2^{-3} + d_4 \times 2$$

şeklinde ifade ettiğimizi varsayalım

Taban dönüşümleri ve ilgili hatalar

- Kesirli sayı dönüşümleri
- Öncelikle $[[x]]$ notasyonu ile x sayısının tam değerini ve $\{x\}$ ile de kesirli kısmını gösterelim.
- $A = 0.125$ sayısının ikili sistemdeki gösterimini belirleyiniz.
- Onlu sistemde

$$A = 0.125 = 1 \times 10^{-1} + 2 \times 10^{-2} + 5 \times 10^{-3}$$

- Aynı sayıyı iki tabanına göre

$$A = 0.125 = (0.d_1 d_2 d_3 d_4 \ldots) = d_1 \times 2^{-1} + d_2 \times 2^{-2} + d_3 \times 2^{-3} + d_4 \times 2$$

şeklinde ifade ettiğimizi varsayalım

- İfadenin her iki yanını 2 ile çarparsak

$$2A = 0.250 = d_1 + d_2 \times 2^{-1} + d_3 \times 2^{-2} + d_4 \times 2^{-3} + \cdots$$

Taban dönüşümleri ve ilgili hatalar

- Kesirli sayı dönüşümleri
- Öncelikle $[[x]]$ notasyonu ile x sayısının tam değerini ve $\{x\}$ ile de kesirli kısmını gösterelim.
- $A = 0.125$ sayısının ikili sistemdeki gösterimini belirleyiniz.
- Onlu sistemde

$$A = 0.125 = 1 \times 10^{-1} + 2 \times 10^{-2} + 5 \times 10^{-3}$$

- Aynı sayıyı iki tabanına göre

$$A = 0.125 = (0.d_1 d_2 d_3 d_4 \ldots) = d_1 \times 2^{-1} + d_2 \times 2^{-2} + d_3 \times 2^{-3} + d_4 \times 2$$

şeklinde ifade ettiğimizi varsayalım

- İfadenin her iki yanını 2 ile çarparsak

$$2A = 0.250 = d_1 + d_2 \times 2^{-1} + d_3 \times 2^{-2} + d_4 \times 2^{-3} + \cdots$$

- Burada

$$d_1 = [[2A]] = 0$$

ve

Taban dönüşümleri ve ilgili hatalar

- On tabanlı sistemden iki tabanlı sisteme dönüşüm

Taban dönüşümleri ve ilgili hatalar

- On tabanlı sistemden iki tabanlı sisteme dönüşüm
- Her iki yanı tekrar 2 ile çarparak

$$2K_1 = 0.50 = d_2 + d_3 \times 2^{-1} + d_4 \times 2^{-2} + \cdots$$

$$d_2 = [[2K_1]] = 0$$

Taban dönüşümleri ve ilgili hatalar

- On tabanlı sistemden iki tabanlı sisteme dönüşüm
- Her iki yanı tekrar 2 ile çarparak

$$2K_1 = 0.50 = d_2 + d_3 \times 2^{-1} + d_4 \times 2^{-2} + \cdots$$

$$d_2 = [[2K_1]] = 0$$

- Benzer biçimde

$$K_2 = \{2K_1\} = 0.50 = d\_3 \times 2^{-1} + d_4 \times 2^{-2} + \cdots$$

$$2K_2 = 1.0 = d_3 + d_4 \times 2^{-1} + \cdots$$

$$d_3 = [[2K_2]] = 1$$

ve $K_3 = \{2K_2\} = 0$ elde edilir. Böylece kesir kısmı sıfır olana kadar işleme devam ettirilerek elde edilen $d_i, i = 1, 2, \ldots$değerleri kaydedilir. O halde $A = 0.125 = (0.001)_2$ iki tabanlı gösterimi elde edilir.

Taban dönüşümleri ve ilgili hatalar

- On tabanlı sistemden iki tabanlı sisteme dönüşüm

Taban dönüşümleri ve ilgili hatalar

- On tabanlı sistemden iki tabanlı sisteme dönüşüm
- 1964.125 sayısının iki tabanlı gösterimini elde ediniz

Taban dönüşümleri ve ilgili hatalar

- On tabanlı sistemden iki tabanlı sisteme dönüşüm
- 1964.125 sayısının iki tabanlı gösterimini elde ediniz
- Bunun için yapmamız gereken işlem, yukarıdaki iki örneğe ait sonuçları birleştirmektir. On tabanlı sistemde

$$1964.125 = 1\times10^3+9\times10^2+6\times10^1+4\times10^0+1\times10^{-1}+2\times10^{-2}+5\times10^{-3}$$

olacak şekilde ifade edildiği gibi, tam ve kesirli kısımlara karşılık gelen iki tabanlı gösterimler de birleştirildiğinde

$$1964.125 = (1111010100.001)_2 = (0.111101011000 01)_2\times2^{11}$$

elde edilir.

İkili sistemde sayıların bellek gösterimi: enbüyük ve enküçük pozitif sayılar

- *IEEE*754 standartı adı verilen uluslararası standarta göre x_f ile gösterilen sayı, 32 bit formatta

1 bit(üs işareti)	7 bit(Üs)	24 bit (Mantis)

İkili sistemde sayıların bellek gösterimi: enbüyük ve enküçük pozitif sayılar

- *IEEE*754 standartı adı verilen uluslararası standarta göre x_f ile gösterilen sayı, 32 bit formatta

1 bit(üs işareti)	7 bit(Üs)	24 bit (Mantis)

- 64 bit formatta ise

İkili sistemde sayıların bellek gösterimi: enbüyük ve enküçük pozitif sayılar

- *IEEE*754 standartı adı verilen uluslararası standarta göre x_f ile gösterilen sayı, 32 bit formatta

1 bit(üs işareti)	7 bit(Üs)	24 bit (Mantis)

- 64 bit formatta ise
-

1 bit(üs işareti)	10 bit(Üs)	53 bit (Mantis)

bellek alanlarında temsil edilebilir. İşaret bitindeki '0' ise sayının pozitif ve '1' ise negatif olması anlamına gelir. Buna göre 1.964125×10^3 sayısının 32 bit formatta bellek alanına yerleşimi aşağıdaki gösterildiği gibidir:

İkili sistemde sayıların bellek gösterimi: enbüyük ve enküçük pozitif sayılar

- *IEEE*754 standartı adı verilen uluslararası standarta göre x_f ile gösterilen sayı, 32 bit formatta

1 bit(üs işareti)	7 bit(Üs)	24 bit (Mantis)

- 64 bit formatta ise
-

1 bit(üs işareti)	10 bit(Üs)	53 bit (Mantis)

bellek alanlarında temsil edilebilir. İşaret bitindeki '0' ise sayının pozitif ve '1' ise negatif olması anlamına gelir. Buna göre 1.964125×10^3 sayısının 32 bit formatta bellek alanına yerleşimi aşağıdaki gösterildiği gibidir:

-

0	0	0	0	1	0	1	1	**0**	1	1	1	1	0	1	0	1	1	0	0

burada **0**= 0000000000 dır.

- İkili sistemde 32 bit formatında temsil edilebilecek $x_f = \pm(.d_1 d_2 \ldots d_{24}) \times 2^e$ sayısını göz önüne alalım.

- İkili sistemde 32 bit formatında temsil edilebilecek $x_f = \pm(.d_1 d_2 \dots d_{24}) \times 2^e$ sayısını göz önüne alalım.
- Pozitif bir sayı için 7 bitlik üs alanında ikili sisteme göre temsil edilebilecek en büyük üs

$$e = (1111111)_2 = 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 1 \times 2^5 + 1 \times 2^6 = 127.$$

- Ayrıca $d_1 \neq 0$ olmak üzere en büyük mantis değeri

$$(.1_1 1_2 \ldots 1_{24})_2 = 1/2^1 + 1/2^2 + \cdots + 1/2^{24} \approx 1$$

olup, noktadan önceki ilk rakam 1 olarak kabul edilip, depolanmayarak, $(1.1_1 1_2 \ldots 1_{24})_2 \approx 2$ olduğu için en büyük pozitif sayı

$$x_f \cong 2^{127} = 1.701412E + 38 = \boxed{0 \mid 1111111 \mid 111\ldots1}$$

- Ayrıca $d_1 \neq 0$ olmak üzere en büyük mantis değeri

$$(.1_1 1_2 \ldots 1_{24})_2 = 1/2^1 + 1/2^2 + \cdots + 1/2^{24} \approx 1$$

olup, noktadan önceki ilk rakam 1 olarak kabul edilip, depolanmayarak, $(1.1_1 1_2 \ldots 1_{24})_2 \approx 2$ olduğu için en büyük pozitif sayı

$$x_f \cong 2^{127} = 1.701412E + 38 = \boxed{0 \mid 1111111 \mid 111\ldots1}$$

- Benzer olarak en küçük pozitif değer ise üssün alabileceği mutlak değerce en büyük negatif sayıya(-128) karşılık gelir.

Negatif tamsayıların ikili sistemdeki gösterimleri:

$$
\begin{aligned}
1 &= (0000000[1]) = 0\times 2^7+0\times 2^6+0\times 2^5+0\times 2^4+0\times 2^3+0\times 2^2+0\times 2^1+1\times 2^0\\
-1 &= (1111111[1]) = -1\times 2^7+1\times 2^6+1\times 2^5+1\times 2^4+1\times 2^3+1\times 2^2+1\times 2^1+1\times 2^0\\
2 &= (000000[10]) = 0\times 2^7+0\times 2^6+0\times 2^5+0\times 2^4+0\times 2^3+0\times 2^2+1\times 2^1+0\times 2^0\\
-2 &= (111111[10]) = -1\times 2^7+1\times 2^6+1\times 2^5+1\times 2^4+1\times 2^3+1\times 2^2+1\times 2^1+0\times 2^0\\
12 &= (00001[100]) = 0\times 2^7+0\times 2^6+0\times 2^5+0\times 2^4+1\times 2^3+1\times 2^2+0\times 2^1+0\times 2^0\\
-12 &= (11110[100]) = -1\times 2^7+1\times 2^6+1\times 2^5+1\times 2^4+0\times 2^3+1\times 2^2+0\times 2^1+0\times 2^0
\end{aligned}
$$

Negatif tamsayıların ikili sistemdeki gösterimleri:

- $$\begin{aligned} 127 &= (0111111[1]) = 0\times 2^7+1\times 2^6+1\times 2^5+1\times 2^4+1\times 2^3+1\times 2^2+1\times 2^1+1\times 2^0 \\ -127 &= (1000000[1]) = -1\times 2^7+0\times 2^6+0\times 2^5+0\times 2^4+0\times 2^3+0\times 2^2+0\times 2^1+1\times 2^0 \\ -128 &= (10000000) = -1\times 2^7+0\times 2^6+0\times 2^5+0\times 2^4+0\times 2^3+0\times 2^2+0\times 2^1+0\times 2^0 \end{aligned}$$

- Soldan itibaren ilk 1 sayısına kadar olan ve köşeli parantez içerisine alınan ikili sayı grubunun sabit bırakılarak, diğerlerinden 1 lerin 0 ve 0 ların 1 yapılarak pozitif bir tamsayının toplama işlemine göre tersinin elde edildiğine dikkat edelim.

- Soldan itibaren ilk 1 sayısına kadar olan ve köşeli parantez içerisine alınan ikili sayı grubunun sabit bırakılarak, diğerlerinden 1 lerin 0 ve 0 ların 1 yapılarak pozitif bir tamsayının toplama işlemine göre tersinin elde edildiğine dikkat edelim.
- Buna göre 8 bitlik alanda temsil edilebilecek mutlak değerce en büyük negatif sayının -128 olacağı açıktır.

- Soldan itibaren ilk 1 sayısına kadar olan ve köşeli parantez içerisine alınan ikili sayı grubunun sabit bırakılarak, diğerlerinden 1 lerin 0 ve 0 ların 1 yapılarak pozitif bir tamsayının toplama işlemine göre tersinin elde edildiğine dikkat edelim.
- Buna göre 8 bitlik alanda temsil edilebilecek mutlak değerce en büyük negatif sayının -128 olacağı açıktır.
- O halde en küçük pozitif kayan noktalı sayı

$$x_f \cong 2^{-128} = 2.9387358771E-39 = \boxed{1}\boxed{0000000}\boxed{111...1}$$

Negatif tamsayıların ikili sistemdeki gösterimleri:

- İkili sistemde 64 bit formatında temsil edilebilecek kayan nokta sayısını göz önüne alalım.

Negatif tamsayıların ikili sistemdeki gösterimleri:

- İkili sistemde 64 bit formatında temsil edilebilecek kayan nokta sayısını göz önüne alalım.
- Pozitif bir sayı için 11 bitlik üs alanında ikili sisteme göre temsil edilebilecek en büyük üs
  $e = (01111111111)_2 = 1023$ ve en küçük üs ise
  $e = (10000000000)_2 = -1024$ dür.

Negatif tamsayıların ikili sistemdeki gösterimleri:

- İkili sistemde 64 bit formatında temsil edilebilecek kayan nokta sayısını göz önüne alalım.
- Pozitif bir sayı için 11 bitlik üs alanında ikili sisteme göre temsil edilebilecek en büyük üs
  $e = (01111111111)_2 = 1023$ ve en küçük üs ise
  $e = (10000000000)_2 = -1024$ dür.
- Ayrıca en büyük mantis değeri

- $$(.1_1 1_2 \cdots 1_{53})_2 = 1/2^1 + 1/2^2 + \cdots + 1/2^{53} \cong 1$$

- $$(.1_1 1_2 \cdots 1_{53})_2 = 1/2^1 + 1/2^2 + \cdots + 1/2^{53} \cong 1$$

- en büyük pozitif kayan noktalı sayı ise

$$x_f = 2^{1023} = 8.988465674311580E + 307$$

olup bu sayının ikili sistemdeki karşılığı

0	1111111111	111...1

elde edilir.

Negatif tamsayıların ikili sistemdeki gösterimleri:

- Benzer olarak en küçük pozitif değer ise üssün alabileceği mutlak değerce en büyük negatif sayıya(-1024) karşılık gelir ve

$$x_f \cong 2^{-1024} = 5.562684646268004E - 309$$

olup, ikili sistem karşılığı

1	0000000000	111...1

Taban dönüşümü kaynaklı hatalar

- Bu durumun en açık örneği onlu sistemdeki 0.1sayısıdır.

Taban dönüşümü kaynaklı hatalar

- Bu durumun en açık örneği onlu sistemdeki 0.1sayısıdır.
- Bu sayının iki tabanlı sistemdeki karşılığı $(0.0\,(\overline{0011}))_2$ devirli sayısıdır.

Taban dönüşümü kaynaklı hatalar

- Bu durumun en açık örneği onlu sistemdeki 0.1sayısıdır.
- Bu sayının iki tabanlı sistemdeki karşılığı $(0.0\,(\overline{0011}))_2$ devirli sayısıdır.
- Ancak 32 bit formatta bu sayının mantisi için 24 bitlik bir alan tahsis edildiği için bu formattaki gösterimi

$$(0.00011001100110011001100 1)_2$$

dir.

Taban dönüşümü kaynaklı hatalar

- Bu durumun en açık örneği onlu sistemdeki 0.1sayısıdır.
- Bu sayının iki tabanlı sistemdeki karşılığı $(0.0\,(\overline{0011}))_2$ devirli sayısıdır.
- Ancak 32 bit formatta bu sayının mantisi için 24 bitlik bir alan tahsis edildiği için bu formattaki gösterimi

$$(0.00011001100110011001100 1)_2$$

dir.

- bu sayının on tabanlı sistemdeki karşılığını hesaplarsak $9.999996E-002$ elde ederiz.

Taban dönüşümü kaynaklı hatalar

- Bu durumun en açık örneği onlu sistemdeki 0.1sayısıdır.
- Bu sayının iki tabanlı sistemdeki karşılığı $(0.0\,(\overline{0011}))_2$ devirli sayısıdır.
- Ancak 32 bit formatta bu sayının mantisi için 24 bitlik bir alan tahsis edildiği için bu formattaki gösterimi

$$(0.00011001100110011001100 1)_2$$

dir.

- bu sayının on tabanlı sistemdeki karşılığını hesaplarsak $9.999996E-002$ elde ederiz.
- Böylece 0.1 sayısının ikili sistemdeki devirli gösterimi, tahsis edilen bellek alana yerleştirilirken atılması gereken kısmı önemli bir hataya neden olmuştur.

Taban dönüşümü kaynaklı hatalar

- $>> x = 0.1;$

Taban dönüşümü kaynaklı hatalar

- $>> x = 0.1;$
- $>> y = (x + x + x + x + x + x + x + x + x + x)$

Taban dönüşümü kaynaklı hatalar

- $>> x = 0.1;$
- $>> y = (x + x + x + x + x + x + x + x + x + x)$
- $>> z = (1 - y) * 1E20$

Taban dönüşümü kaynaklı hatalar

- $>> x = 0.1;$
- $>> y = (x + x + x + x + x + x + x + x + x + x)$
- $>> z = (1 - y) * 1E20$
- $z = 1.1102E + 004$
  !

Anlamlı basamak kaybı hatası

- Anlamlı basamak sayısı: On tabanlı sistemde eğer $|\varepsilon_b(x)| \leq 5 \times 10^{-k}$ ise x_f yaklaşımı x gerçek değerini $k-1$ anlamlı basamakla temsil etmektedir denir.

Anlamlı basamak kaybı hatası

- Anlamlı basamak sayısı: On tabanlı sistemde eğer $|\varepsilon_b(x)| \leq 5 \times 10^{-k}$ ise x_f yaklaşımı x gerçek değerini $k-1$ anlamlı basamakla temsil etmektedir denir.
- $x_f = 1.1$ yaklaşımı $x = 1$ değerini 0 anlamlı basamakla temsil etmektedir, çünkü

$$|\varepsilon_b(x)| = 1 \times 10^{-1} < 5 \times 10^{-1}$$

Anlamlı basamak kaybı hatası

- Anlamlı basamak sayısı: On tabanlı sistemde eğer $|\varepsilon_b(x)| \leq 5 \times 10^{-k}$ ise x_f yaklaşımı x gerçek değerini $k-1$ anlamlı basamakla temsil etmektedir denir.
- $x_f = 1.1$ yaklaşımı $x = 1$ değerini 0 anlamlı basamakla temsil etmektedir, çünkü

$$|\varepsilon_b(x)| = 1 \times 10^{-1} < 5 \times 10^{-1}$$

- $x_f = 1.23$ yaklaşımı $x = 1.2$ değerini 1 anlamlı basamkla temsil etmektedir, çünkü

$$|\varepsilon_b(x)| = 2.5 \times 10^{-2} < 5 \times 10^{-2}$$

Anlamlı basamak kaybı hatası

- Anlamlı basamak sayısı: On tabanlı sistemde eğer $|\varepsilon_b(x)| \leq 5 \times 10^{-k}$ ise x_f yaklaşımı x gerçek değerini $k-1$ anlamlı basamakla temsil etmektedir denir.
- $x_f = 1.1$ yaklaşımı $x = 1$ değerini 0 anlamlı basamakla temsil etmektedir, çünkü

$$|\varepsilon_b(x)| = 1 \times 10^{-1} < 5 \times 10^{-1}$$

- $x_f = 1.23$ yaklaşımı $x = 1.2$ değerini 1 anlamlı basamkla temsil etmektedir, çünkü

$$|\varepsilon_b(x)| = 2.5 \times 10^{-2} < 5 \times 10^{-2}$$

- $x_f = 1.234$ yaklaşımı $x = 1.232$ değerini 2 anlamlı basamkla temsil etmektedir, çünkü

$$|\varepsilon_b(x)| = 1.6 \times 10^{-3} < 5 \times 10^{-3}$$

Anlamlı basamak kaybı hatası

- Birbirine eşit olan

$$f_1(x) = 1000(log(x+1) - log(x))$$

ve

$$f_2(x) = 1000log((x+1)/x)$$

fonksiyonlarını göz önüne alalım ve bu fonksiyonların $x = 100000$ noktasındaki değerlerini virgülden sonra altı basamağa kadar ve yuvarlama prensibine göre çalışan hesaplayıcıda hesaplayarak sonuçları karşılaştıralım.

Anlamlı basamak kaybı hatası

- Birbirine eşit olan

$$f_1(x) = 1000(log(x+1) - log(x))$$

ve

$$f_2(x) = 1000 log((x+1)/x)$$

fonksiyonlarını göz önüne alalım ve bu fonksiyonların $x = 100000$ noktasındaki değerlerini virgülden sonra altı basamağa kadar ve yuvarlama prensibine göre çalışan hesaplayıcıda hesaplayarak sonuçları karşılaştıralım.

- $x = 100000$ ve $x_1 = log(x+1) = 5.00000434292310$ dir.

Anlamlı basamak kaybı hatası

- Birbirine eşit olan

$$f_1(x) = 1000(log(x+1) - log(x))$$

ve

$$f_2(x) = 1000 log((x+1)/x)$$

fonksiyonlarını göz önüne alalım ve bu fonksiyonların $x = 100000$ noktasındaki değerlerini virgülden sonra altı basamağa kadar ve yuvarlama prensibine göre çalışan hesaplayıcıda hesaplayarak sonuçları karşılaştıralım.

- $x = 100000$ ve $x_1 = log(x+1) = 5.00000434292310$ dir.
- Ancak altı basamaklı ve yuvarlama prensibine göre çalışan hesaplayıcıda $x_1 \cong 5.000004$ olarak kabul edilir.

Anlamlı basamak kaybı hatası

- $$x_2 = log(x) = 5;$$

Anlamlı basamak kaybı hatası

- $$x_2 = log(x) = 5;$$
- $$f_1(x) = 1000(x_1 - x_2) = 0.004;$$

Anlamlı basamak kaybı hatası

- $$x_2 = log(x) = 5;$$
- $$f_1(x) = 1000(x_1 - x_2) = 0.004;$$
- $$f_2(x) = 1000log((x+1)/x) = 1000 \times 0.000043 = 0.043.$$

Anlamlı basamak kaybı hatası

- $$x_2 = log(x) = 5;$$
- $$f_1(x) = 1000(x_1 - x_2) = 0.004;$$
- $$f_2(x) = 1000log((x+1)/x) = 1000 \times 0.000043 = 0.043.$$
- Gerçek sonuç ise 0.043427276862573 dir.

Anlamlı basamak kaybı hatası

- $$x_2 = log(x) = 5;$$
- $$f_1(x) = 1000(x_1 - x_2) = 0.004;$$
- $$f_2(x) = 1000log((x+1)/x) = 1000 \times 0.000043 = 0.043.$$
- Gerçek sonuç ise 0.043427276862573 dir.
- Yuvarlatılmış x_1 değeri 6 anlamlı basamağa sahipken birbirine çok yakın x_1, x_2 sayılarının farkı $x_1 - x_2 = 0.000004$ olarak elde edilir.

Anlamlı basamak kaybı hatası

- $$x_2 = log(x) = 5;$$
- $$f_1(x) = 1000(x_1 - x_2) = 0.004;$$
- $$f_2(x) = 1000log((x+1)/x) = 1000 \times 0.000043 = 0.043.$$
- Gerçek sonuç ise 0.043427276862573 dir.
- Yuvarlatılmış x_1 değeri 6 anlamlı basamağa sahipken birbirine çok yakın x_1, x_2 sayılarının farkı $x_1 - x_2 = 0.000004$ olarak elde edilir.
- Bu yaklaşım gerçek fark değeri olan 0.00000434292310 değerini hiçbir anlamlı basamakla temsil etmez.

Anlamlı basamak kaybı hatası

- $$x_2 = log(x) = 5;$$
- $$f_1(x) = 1000(x_1 - x_2) = 0.004;$$
- $$f_2(x) = 1000log((x+1)/x) = 1000 \times 0.000043 = 0.043.$$
- Gerçek sonuç ise 0.043427276862573 dir.
- Yuvarlatılmış x_1 değeri 6 anlamlı basamağa sahipken birbirine çok yakın x_1, x_2 sayılarının farkı $x_1 - x_2 = 0.000004$ olarak elde edilir.
- Bu yaklaşım gerçek fark değeri olan 0.00000434292310 değerini hiçbir anlamlı basamakla temsil etmez.
- Dolayısıyla fark işlemi anlam basamağı kaybına neden olmuştur

Alıştırmalar

1. $x_f = \pm q \times \beta^e$, $q = 0.d_1 d_2$, $d_{1,2} = 0, 1, \ldots, 5$;$d_1 \neq 0, \beta = 10$ tabanlı ve $e = -1, 0, 1$ üs değerlerine sahip bir R_f sayı sistemi gözönüne alalım ve gerçekleştirilen her işlem sonucunun R_f de olmaması durumunda, sistemde bulunan en yakın sayıya yuvarlatıldığını kabul edelim. Buna göre
   - R_f nin elemanları(sayıları) nelerdir.
   - R_f nin en büyük pozitif ve en küçük pozitif sayıları nelerdir.
   - R_f de "overflow veya inf" hatasına neden olabilecek birer işlem tanımlayınız.
   - R_f kümesinin toplama işlemine göre kapalılık özelliğini sağlamayabileceğini bir örnekle gösteriniz.
   - R_f de toplama işlemine göre birleşme özelliği olamayabileceğini bir örnekle gösteriniz.

Alıştırmalar

2 $(R\ , +, .)$ nın yani Reel sayılar kümesinin bilenen skalerle çarpma ve toplama işlemine göre sağladığı başka hangi özelliklerin $(R_f\ , +, .)$ de geçerli olmayabileceğini düşünüyorsunuz?

3 $x_f = \pm q \times \beta^e$, $q = 0.d_1 d_2 d_3$, $0 \leq d_i < \beta$,formatında yazılabilen kayan noktalı sayılara izin veren sanal bir bilgisayar sistemi gözönüne alalım. Buna göre aşağıdaki tabloda verilen β, e değerleri için sistemde temsil edilebilecek en büyük ve en küçük pozitif tamsayıları belirleyiniz.

4 Soru 3 deki R_f sayı sistemini inceleyerek, hangi tür işlemlerde daha fazla hata oluşabileceğini düşünüyorsunuz?

Alıştırmalar

5 $(R\ , +, .)$ nın yani Reel sayılar kümesinin bilenen skalerle çarpma ve toplama işlemine göre sağladığı başka hangi özelliklerin $(R_f\ , +, .)$ de geçerli olmayabileceğini düşünüyorsunuz?

6 $x_f = \pm q \times \beta^e$, $q = 0.d_1 d_2 d_3$, $0 \leq d_i < \beta$,formatında yazılabilen kayan noktalı sayılara izin veren sanal bir bilgisayar sistemi gözönüne alalım. Buna göre aşağıdaki tabloda verilen β, e değerleri için sistemde temsil edilebilecek en büyük ve en küçük pozitif tamsayıları belirleyiniz.

7 Soru 3 deki R_f sayı sistemini inceleyerek, hangi tür işlemlerde daha fazla hata oluşabileceğini düşünüyorsunuz?

Alıştırmalar

8 $x = 0.d_1 d_2 ... d_n d_{n+1} ... d_m \times \beta^e$,$x_f = 0.d_1 d_2 ... d_n \times \beta^e$ için

$$|\varepsilon_b(x)| = \left| \frac{x - x_f}{x} \right| \leq \frac{1}{2} \beta^{-n+1}$$

olduğunu gösteriniz.(ipucu: Öncelikle

$$|x - x_f| \leq \overbrace{.00...0}^{n \text{ adet}} (\frac{1}{2}\beta) \times \beta^e = (\frac{1}{2}\beta)\beta^{-n-1} \times \beta^e = \frac{1}{2}\beta^{e-n}$$

olduğunu gözlemleyiniz ayrıca β tabanlı sistemde $1/0.1 = \beta^{-1}$ olduğuna dikkat ediniz.)

Alıştırmalar

9 Aşağıda verilen on tabanlı sayıların karşılarında verilen iki tabanlı gösterimlere sahip olduklarını gösteriniz

3	$(11)_2$
8	$(1000)_2$
10	$(1010)_2$
50	$(110010)_2$
100	$(1100100)_2$

Alıştırmalar

10 Aşağıda verilen iki tabanlı sayıların on tabanlı gösterimlerini belirleyiniz

- $(11001)_2$
- $(1001011)_2$
- $(1100000)_2$
- $(100001)_2$

11 Aşağıda verilen kesirli sayıların iki tabanlı gösterimlerinin doğruluğunu kontrol ediniz.$\overline{1001}$ üs çizgi notasyonu sayının devirli sayı olduğunu ifade etmektedir.

- $2.5 \rightarrow (10.1)_2$
- $3.6 \rightarrow (11.\left(\overline{1001}\right))_2$
- $1.25 \rightarrow (1.01)_2$
- $1.125 \rightarrow (1.001)_2$

Alıştırmalar

12 Onlu tabanda 0.2 sayısı 32 bitlik $(0.00110011001100110011001100110011)_2$ gösterimine sahiptir. Elde edilen bu ikilik sistem gösterimini on tabanlı sisteme dönüştürdüğümüzde yine 0.2 sayısını elde ederiz. Bu sonucu kendi işlemlerinizle kontrol ediniz.

13 Bilgisayar sisteminizde 1 ve 2 arasındaki her sayı ile bu sayıya en yakın sayı arasındaki mesafenin birbirine eşit olduğunu eps komutu yardımıyla gözlemleyiniz. Buna göre örneğin $eps(1) = eps(1.5) = eps(1.8)$ olmalıdır.

15 Bilgisayar sisteminizde 2 ile 2 den büyük olan en büyük hangi sayı arasında yer alan bütün sayılar birbirine eşit uzaklıktadırlar?

Alıştırmalar

16 realmax sayısının $(2 - eps) \times 2^{1023}$ olduğunu gözlemleyiniz. Bu durumda 2^{1024} sayı sisteminizde yer alır mı?

17 Aşağıda verilen yaklaşımların karşılarında verilen değerleri kaç anlamlı basamakla temsil ettiğini açıklayınız

- $x_f = 12.36$, $x = 12.345$
- $x_f = 0.0013$, $x = 0.00125$
- $x_f = 0.000011$, $x = 0.0000125$

19 Anlam basamak kaybı hatasının önlenmesi için aşağıdaki fonksiyonlar belirtilen nokta komşuluğunda alternatif olarak nasıl yazılabilir?

- $e^{-x}, x > 0$
- $(-b + \sqrt{b^2 - 4c})/2, b >> 0$(sıfırdan çok büyük), $c \simeq 0$
- $e^{x-y}, x \simeq y$

20 Girilen on tabanlı pozitif tamsayıyı iki tabanlı sayıya dönüştüren bir kod hazırlayınız.