**LEZIONE 1: RAPPRESENTAZIONE DEI REALI IN BASE b, ERRORI DI TRONCAMENTO E DI ARROTONDAMENTO**

Partendo dal seguente fatto, fissata una base b naturale > 1 ogni x ∈ R si può scrivere come  
x = sign(x) (cm cm-1 ... c1 c0 , c-1 c-2 … c-n …)

= sign(x) ( cj bj + c-j b-j)

| parte intera | | parte frazionaria|

dove cj , c-j∈ {0, 1, …., b-1} sono le cifre della rappresentazione di x in base b e gli indici j e -j corrispondono alle potenze della base, potenze positive per la parte intera e negative per la parte frazionaria

Esempi

cj ,c-j ∈ {0, 1, …., 9} base 10

cj ,c-j ∈ {0, 1} base 2

cj ,c-j ∈ {0, 1, 2} base 3

Conviene focalizzarsi su base 10, la base usuale nelle varie culture umane, corrispondente al fatto che contiamo con 10 dita delle mani, anche se nella storia dell’umanità ci sono state altre scelte, ad esempio b=20 (Maya) e b=60 (Sumeri)

**b=2**

la base del calcolatore corrispondente al fatto che l’unità fondamentale di memoria, il bit, può assumere due stati fisici distinti.

Osserviamo che: parte intera ∈ ℕ

parte frazionaria ∈ [0,1]

la parte frazionaria in genere ha infinite cifre e tecnicamente è una serie (somma infinita) a termini non negativi *convergente*.

Facciamo un esempio numerico in base 10 per fissare le idee:

x = +1278.3405…

=(+1)**⋅** {1278+0.3405...}

=(+1)**⋅** {1**⋅**103 + 2**⋅**102 + 7**⋅**101 + 8**⋅**100 + 3**⋅**10-1 + 4**⋅**10-2 + 0**⋅**10-3 + 5**⋅**10-4...}

migliaia centinaia decine unità decimi centesimi millesimi decimillesimi

per far vedere che la serie della parte frazionaria è convergente e quindi rappresenta effettivamente un numero ∈ [0, 1] si usa il criterio del confronto per serie e termini non negativi

infatti c-jb-j <= (b-1)b-j

perché 0 <= c-j <= b-1

quindi c-j b-j <= (b-1)b-j

e tutto si riduce alla serie geometrica di ragione a = b-1 = < 1 perché b > 1

a questo punto conviene ricordarci le proprietà di somma e serie geometrica chiamiamo Sn= dove

Sn = 1 + a + a2 + .... + an

aSn = a + a2 + .... + an+1

quindi

aSn - Sn = (a - 1)Sn = an+1 - 1

cioè Sn = (an+1 - 1) / (a-1) per a != 1

ora an+1 diverge per |a| > 1

mentre an+10, per |a| < 1

allora per |a|<1

aj = Sn = per |a| < 1

Nel nostro caso a = b-1 < 1 allora la serie della parte frazionaria è maggiorata da una serie geometrica convergente (a meno del fattore b-1) e quindi **converge**.

c-j b-j ≤ (b-1)b-j < ∞

(l’indice parte da j=1 e non da j=0 ma questo ovviamente non cambia niente per la convergenza.)

Per vedere che la parte frazionaria sta in [0,1], osserviamo che se tutte le cifre dopo la virgola sono uguali alla cifra massima (periodicità sulla cifra massima) la parte frazionaria è 1.

Vediamolo in base 10:

(0...)10 = 910-j = 910-j = 9(-1) = 9(-1) = 9= 1

Analogamente dimostrare per esercizio che (0.)2 = 12-j = 1

Facciamo alcune osservazioni fondamentali:

* i numeri irrazionali (ad es. ) hanno parte frazionaria infinita (ovvero, hanno infinite cifre dopo la virgola) in qualsiasi base il motivo è che i numeri con parte frazionaria finita in una base sono necessariamente numeri razionali, perché è somma (a meno del segno) della parte intera che è un numero naturale e di una combinazione lineare (i coefficienti sono le cifre delle potenze b-j = 1/bj che sono frazioni)
* i numeri razionali possono avere una parte frazionaria finita o infinita a seconda della base. Ad esempio: = (...)10 = (0.100...)3 infatti = 1 **⋅** 3-1 base 3, ma in base 10 risulta: (...)10 = 3**⋅**10-j = 3**⋅**10-j = = (si vedano i conti per (0.999)10=1)

A questo punto possiamo affrontare la prima questione fondamentale: siccome nel calcolatore per rappresentare un numero reale (tipicamente in base 2) avremo a disposizione una quantità finita di cifre, che ERRORE si fa approssimando un numero reale “tagliando” la parte frazionaria ad n cifre? cioè utilizzando quello che si chiama troncamento ad n cifre della parte frazionaria?

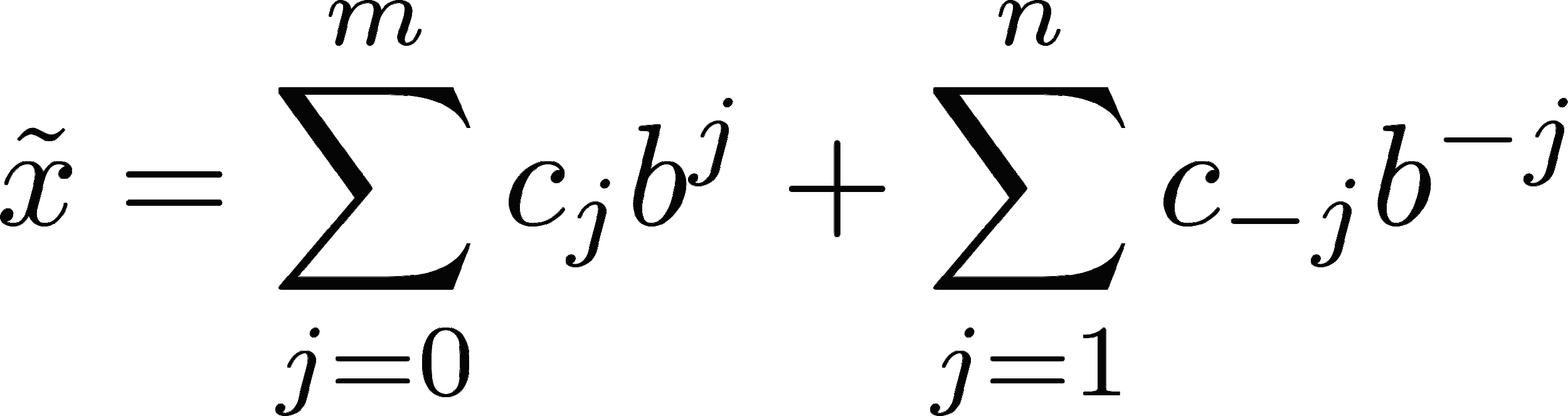
Qui cominciano ad entrare in gioco due concetti chiave e pervasivi del calcolo numerico: il concetto di approssimazione e il concetto di errore

Gli oggetti che si usano (numeri, funzioni, vettori,...) non sono praticamente mai esatti ma sono approssimati a meno di un certo errore, la cosa importante è stimare questi errori e studiare il loro effetto nei calcoli (propagazione degli errori)

Concentriamoci ora su l'errore di troncamento ad n cifre dato

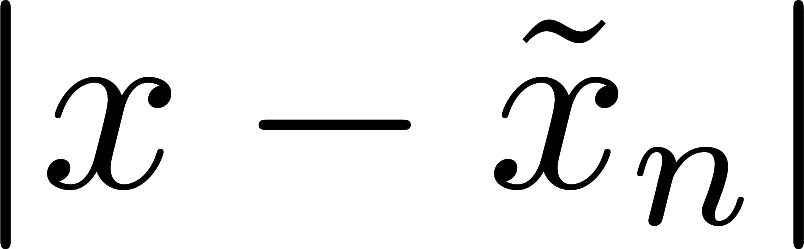
x = sgn(x) { cjbj + c-jb-j}

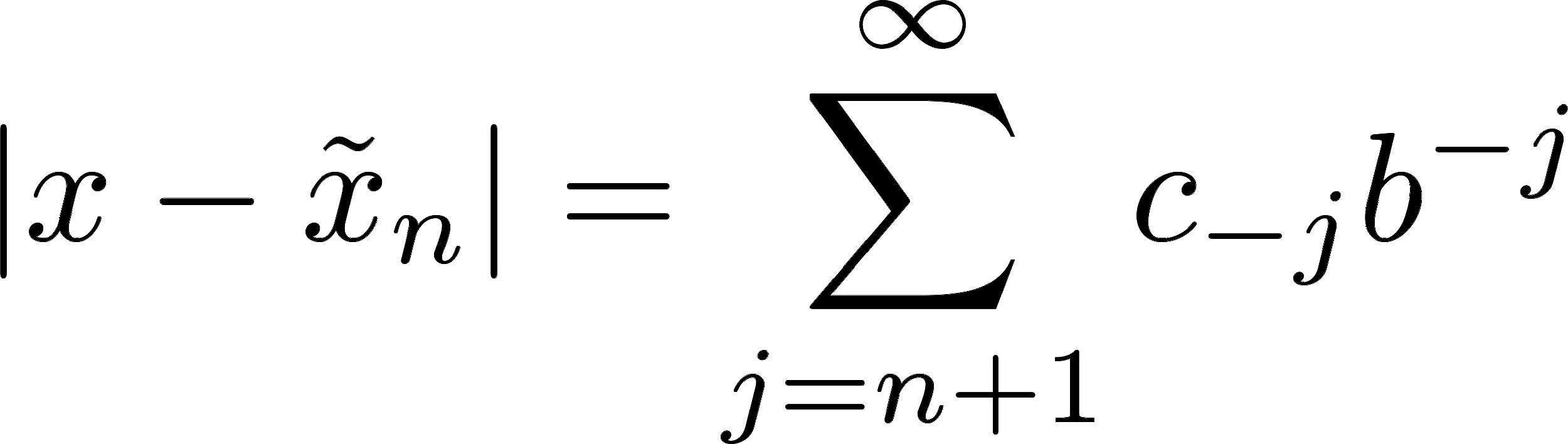
definiamo

[](http://www.texrendr.com/?eqn=%5Ctilde%7Bx%7D%20%3D%20%7B%5Csum%5Climits_%7Bj%3D0%7D%5Em%20c_j%20b%5Ej%20%2B%20%5Csum%5Climits_%7Bj%3D1%7D%5En%20c_%7B-j%7D%20b%5E%7B-j%7D%7D%0)

cioè stesso segno, stessa parte intera e parte frazionaria “tagliata di n cifre”.

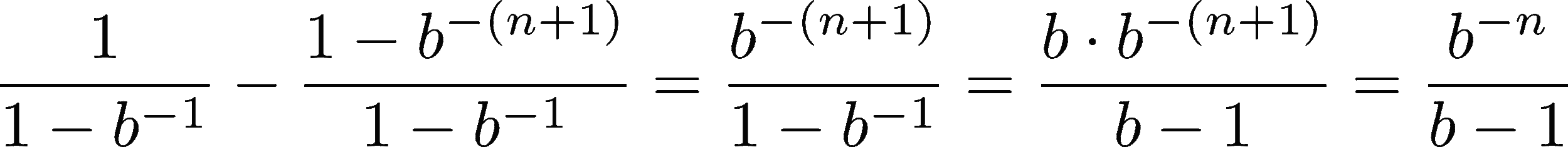
In generale definiamo errore su una quantità a ∈ R approssimata da ã ∈ R (scriviamo ã ≈ a per dire che ã approssima a)

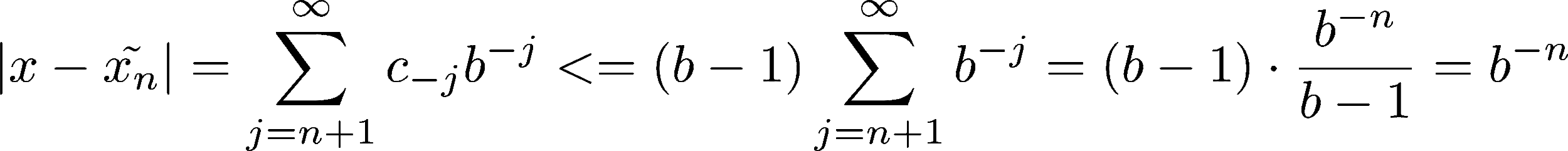
la quantità ERRORE == ,quindi ora cerchiamo di stimare [](http://www.texrendr.com/?eqn=%7Cx-%5Ctilde%7Bx%7D_n%7C%0)

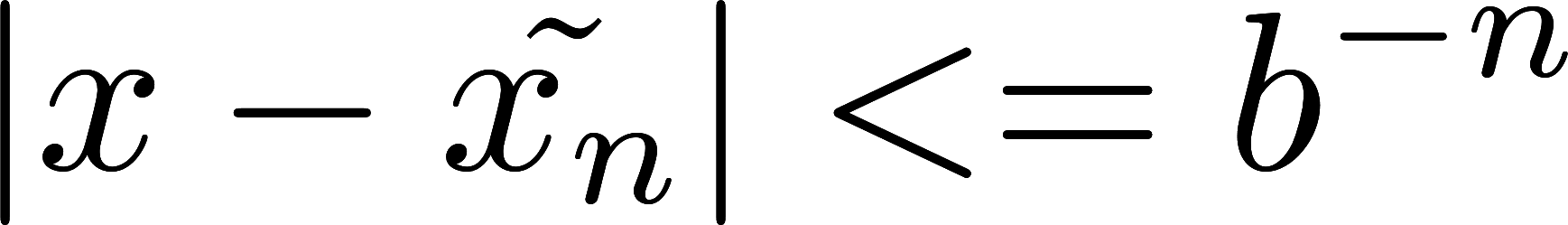
è chiaro che [](http://www.texrendr.com/?eqn=%7Cx-%5Ctilde%7Bx%7D_n%7C%20%3D%20%5Csum%5Climits_%7Bj%3Dn%2B1%7D%5E%5Cinfty%20c_%7B-j%7D%20b%5E%7B-j%7D%0) , e cioè l’errore di troncamento ad n cifre non è altro che il resto della serie che definisce la parte frazionaria.

Andiamo a calcolare questo resto

ragionando come prima, visto che c-j <= b-1 otteniamo c-j b-j <= (b-1) b-j quindi basta calcolare i l resto della serie geometrica per avere una stima.

Ora b-j = b-j - b-j = [](http://www.texrendr.com/?eqn=%5Cfrac%7B1%7D%7B1-b%5E%7B-1%7D%7D%20-%20%5Cfrac%7B1-b%5E%7B-(n%2B1)%7D%7D%7B1-b%5E%7B-1%7D%7D%20%3D%20%5Cfrac%7Bb%5E%7B-(n%2B1)%7D%7D%7B1-b%5E%7B-1%7D%7D%20%3D%20%5Cfrac%7Bb%20%5Ccdot%20b%5E%7B-(n%2B1)%7D%7D%7Bb-1%7D%20%3D%20%5Cfrac%7Bb%5E%7B-n%7D%7D%7Bb-1%7D%0)

da cui [](http://www.texrendr.com/?eqn=%7Cx-%5Ctilde%7Bx_n%7D%7C%20%3D%20%5Csum%5Climits_%7Bj%3Dn%2B1%7D%5E%5Cinfty%20c_%7B-j%7D%20b%5E%7B-j%7D%20%3C%3D%20(b-1)%20%5Csum%5Climits_%7Bj%3Dn%2B1%7D%5E%5Cinfty%20b%5E%7B-j%7D%20%3D%20(b-1)%20%5Ccdot%20%5Cfrac%7Bb%5E%7B-n%7D%7D%7Bb-1%7D%3Db%5E%7B-n%7D%0)

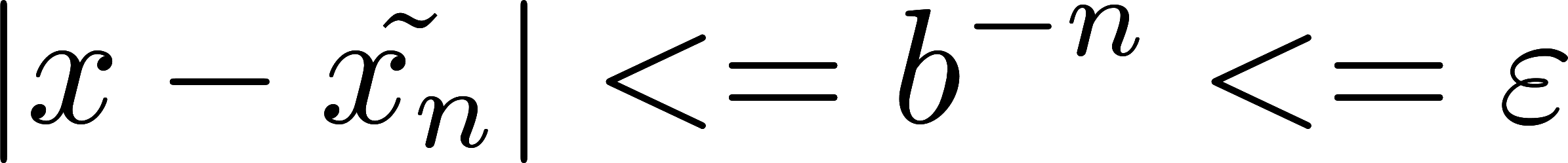
cioè abbiamo ricavato una stima dell’errore di troncamento ed n cifre (in una base generica b) [](http://www.texrendr.com/?eqn=%7Cx-%5Ctilde%7Bx_n%7D%7C%20%3C%3D%20b%5E%7B-n%7D%0)

Attenzione: non abbiamo “calcolato” l’errore, ma lo abbiamo stimato.

Questa è una situazione tipica del calcolo numerico: gli errori di solito non sono noti ma si riesce a ricavare una stima, talvolta rigorosa (disuguaglianze come in questo caso), spesso meno rigorosa ma in grado di dare almeno un ordine di grandezza all’errore.

Il fatto di avere una stima (da sopra) permette comunque di controllare l’errore.

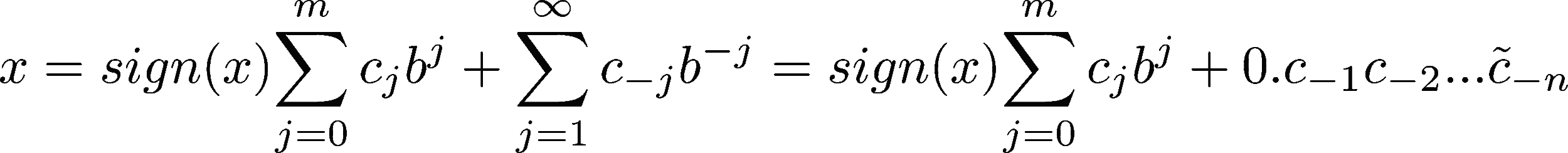
In questo caso la domanda è: quante cifre devo prendere dopo la virgola (cioè, della parte frazionaria) per garantire che l’errore non superi una fissata tolleranza ?

Basterà che la stima dell’errore non superi , infatti [](http://www.texrendr.com/?eqn=%7Cx-%5Ctilde%7Bx_n%7D%7C%3C%3Db%5E%7B-n%7D%3C%3D%5Cvarepsilon%0)

Se questa disuguaglianza *()* è soddisfatta, anche l’errore sarà (la tolleranza ovviamente dipende dal problema applicativo in cui si utilizza l’approssimazione della quantità cercata) risolvendo come si ricava cioè quindi perché l’errore di troncamento sia sicuramente sotto la tolleranza basta prendere il più piccolo numero intero

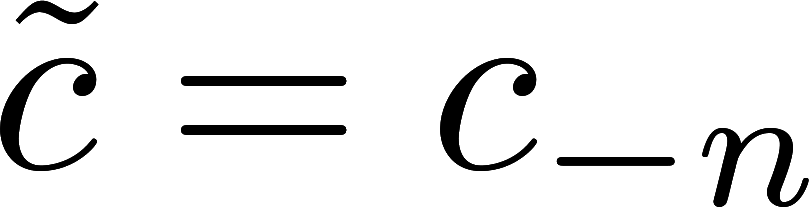
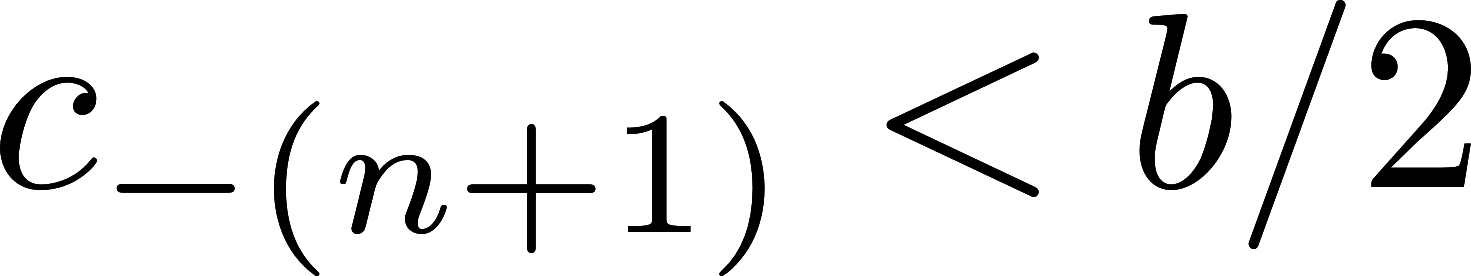
In realtà la tecnica di approssimazione più usata non è il troncamento bensì l’ARROTONDAMENTO.

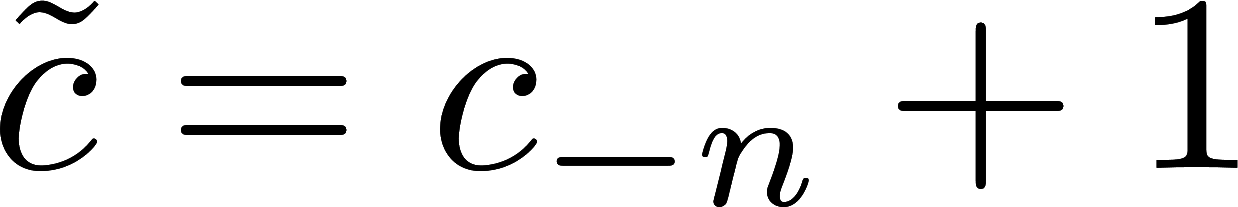
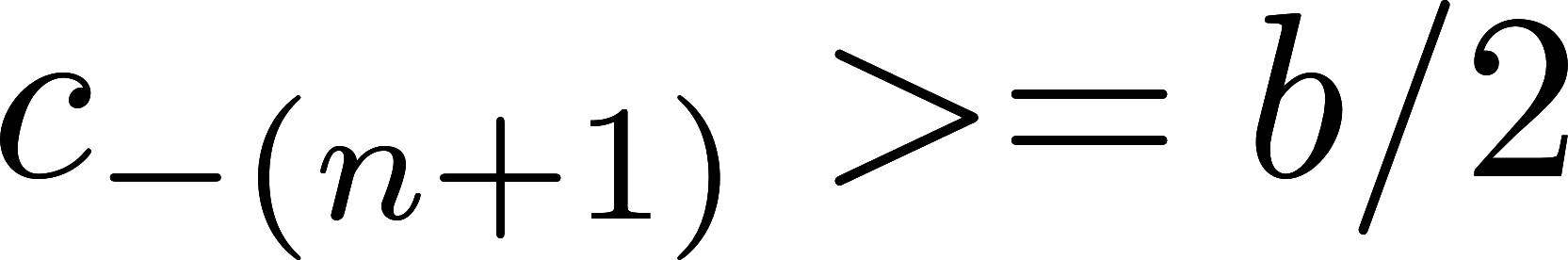
Ricordiamo questa tecnica che tutti già conosciamo dalle scuole, si approssima

[](http://www.texrendr.com/?eqn=x%3Dsign(x)%7B%5Csum%5Climits_%7Bj%3D0%7D%5Em%20c_j%20b%5Ej%20%2B%20%5Csum%5Climits_%7Bj%3D1%7D%5E%5Cinfty%20c_%7B-j%7D%20b%5E%7B-j%7D%7D%20%3D%20sign(x)%7B%5Csum%5Climits_%7Bj%3D0%7D%5Em%20c_j%20b%5Ej%20%2B%200.c_%7B-1%7Dc_%7B-2%7D...%5Ctilde%7Bc%7D_%7B-n%7D%7D%0)

con

dove

[](http://www.texrendr.com/?eqn=%5Ctilde%7Bc%7D%3Dc_%7B-n%7D%0) se [](http://www.texrendr.com/?eqn=c_%7B-(n%2B1)%7D%3Cb%2F2%0) DIFETTO

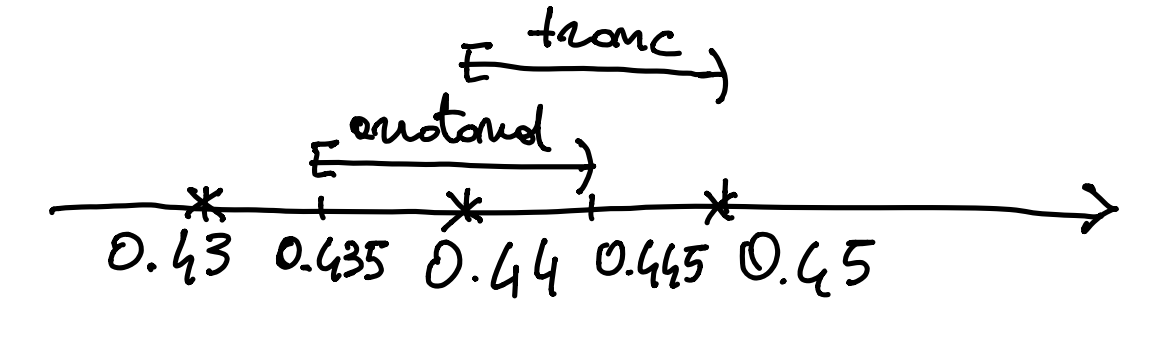
[](http://www.texrendr.com/?eqn=%5Ctilde%7Bc%7D%3Dc_%7B-n%7D%2B1%0) se [](http://www.texrendr.com/?eqn=c_%7B-(n%2B1)%7D%3E%3Db%2F2%0) ECCESSO

cioè si tiene l’n-esima cifra dopo la virgola se la è minore di metà della base (0, 1, 2, 3, 4 in base 10) che si chiama arrotondamento per difetto, invece si aggiunge 1 all’n-esima cifra se la cifra è maggiore o uguale a metà della base (5, 6, 7, 8, 9 in base 10) che sia chiama arrotondamento per eccesso. In questa discussione ci limitiamo alle basi pari che sono quelle più usate nel calcolo (tipicamente la base 10 e la base 2). Nel caso di base pari si può dimostrare che

cioè il massimo errore di arrotondamento ad n cifre dopo la virgola è metà del massimo errore di troncamento ad n cifre (questo non significa che l’errore di arrotondamento sia sempre metà dell’errore di troncamento, come si vede nell’esempio grafico qui sotto).

La dimostrazione che si può fare usando la rappresentazione come serie della parte frazionaria è abbastanza difficile, ci accontenteremo di avere una spiegazione grafica con un esempio numerico.

Consideriamo b=10 e n=2 e prendiamo i seguenti numeri sull’asse reale:



Vediamo quali numeri sono approssimati da 0.44 per troncamento e per arrotondamento a 2 cifre (notiamo che qui abbiamo numeri del tipo **0.** …. che quindi coincidono con la propria parte frazionaria). Per troncamento 0.44 approssimatutti i numeri dell’intervallo [0.44,0.45) che sono del tipo 0.44….(escluso 0.44…=0.45) invece per arrotondamento *approssima?* tutti i numeri dell’intervallo [0.435,0.445) infatti approssima per difetto tutti i numeri del tipo

0.440…

0.441...

0.442...

0.443...

mentre approssima per eccesso tutti i numeri del tipo

0.435…

0.436…

0.437…

0.438…

0.439…

In pratica l’intervallo di troncamento è un *intorno destro* di 0.44 di ampiezza 10-2, mentre l’intervallo di arrotondamento è un intorno simmetrico di raggio (10-2) /2 e ampiezza 10-2. Risulta evidente dal punto di vista grafico che l’estremo superiore degli errori di troncamento è = 10-2 (entrambi sono intervalli semiaperti a destra) mentre il massimo degli errori di arrotondamento e (10-2)/2 e si ottiene per x=0.435.