Σφάλματα Υπολογισμών Αθροισμάτων

$$S_n = 1 + \sum_{k=1}^{n} \frac{1}{k^2+k}$$

Ψευτοκώδικας

```
S = 1
Για k = 1, ..., n
    S = S + 1/(k^2+k)
```

Ακριβής υπολογισμός : $S_n = 1 + \sum_{k=1}^{n} \frac{1}{k^2+k} = 2 - \frac{1}{n+1}$

$S_9 = 2 - \frac{1}{10} = 1.9$, ... , $S_{9999} = 2 - \frac{1}{10.000} = 1.9999$

Διαφορετικός τρόπος υπολογισμού

$$S_n = \frac{1}{1} + \frac{1}{1+1} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \dots + \frac{1}{n(n-1)}$$

Η πρόβλεψη είναι προσεγγιστική. Μπορώ να αθροίσω από το τέλος προς την αρχή

$T_0 = \frac{1}{n(n+1)}$, $T_1 = T_0 + \frac{1}{(n-1)n}$, ... , $T_k = T_{k-1} + \frac{1}{(n-k)(n-k+1)}$

Ψευδοκώδικας Η/Υ

$t = \frac{1}{n(n+1)}$

Για $k = 1, \dots, n-1$

$t = t + \frac{1}{(n-k)(n-k+1)}$

$t = t + 1$

Ε Ξήγηση του φαινομένου

Ας υποθέσουμε ότι έχουμε N αριθμούς a_i, $i = 1, \dots, N$
και για ευκολία ας υποθέσουμε ότι όλοι είναι αριθμοί του μηχανής

Για να υπολογίσουμε το άθροισμα $S_N = \sum_{i=1}^{N} a_i$ χρησιμοποιούμε τον αλγόριθμο

$$\begin{cases} S_1 = a_1 \\ \text{Για } K = 2, \dots, N \\ \quad S_K = S_{K-1} + a_K \end{cases}$$

Στον Η/Υ αν εφαρμόσουμε τον παραπάνω αλγόριθμο θα δημιουργηθούν οι αριθμοί $\tilde{S}_K$ (Η περιορισμένη δομή αριθμού του μηχανής)

$$\tilde{S}_1 = a_1, \quad \tilde{S}_2 = fl(\tilde{S}_1 + a_2), \dots, \quad \tilde{S}_N = fl(\tilde{S}_{N-1} + a_N)$$

Αν u είναι το μέγιστο σφάλμα στρογγυλοποίησης στην αναπαράσταση ενός αριθμού στον Η/Υ ($u = \frac{1}{2} \times 10^{1-k}$ (k - ψηφία ακρίβειας))

$$\tilde{S}_1 = a_1$$

$$\tilde{S}_2 = fl(\tilde{S}_1 + a_2) = (\tilde{S}_1 + a_2)(1+\varepsilon_1) = a_1(1+\varepsilon_1) + a_2(1+\varepsilon_1) \quad , \quad |\varepsilon_1| \leq u$$

$$\tilde{S}_3 = fl(\tilde{S}_2 + a_3) = (\tilde{S}_2 + a_3)(1+\varepsilon_2) = (a_1 + a_2)(1+\varepsilon_1)(1+\varepsilon_2) + a_3(1+\varepsilon_2) \quad , \quad |\varepsilon_2| \leq u$$

$$\tilde{S}_4 = fl(\tilde{S}_3 + a_4) = (\tilde{S}_3 + a_4)(1+\varepsilon_3) = (a_1+a_2)(1+\varepsilon_1)(1+\varepsilon_2)(1+\varepsilon_3) + a_3(1+\varepsilon_2)(1+\varepsilon_3) + a_4(1+\varepsilon_3) \quad , \quad |\varepsilon_3| \leq u$$

$$= (a_1+a_2)(1+\delta_1)^3 + a_3(1+\delta_2)^2 + a_4(1+\delta_3) \quad , \quad |\delta_1|, |\delta_2|, |\delta_3| \leq |u|$$

$$\approx (a_1+a_2)(1+3\delta_1) + a_3(1+2\delta_2) + a_4(1+\delta_3)$$

$$= S_4 + (a_1+a_2)3\delta_1 + a_3 2\delta_2 + a_4\delta_3$$

Αν συνεχίσουμε με αυτό τον τρόπο βλέπουμε ότι

$$\tilde{S}_N \approx S_N + (a_1+a_2)(N-1)J_1 + a_3(N-2)J_2 + \dots + a_{N-1}2J_{N-2} + a_N J_{N-1}$$

με $|J_i| \le u$, $i = 1, \dots, N-1$

Αν οι αριθμοί του αθροίσματος S_N είναι δεκτοί και έχουν διαταχθεί σε φθίνουσα σειρά, τότε τα σφάλματα προσέγγισης που πολ/ζουν τους μεγάλους όρους είναι μεγάλα
(Τα σφάλματα που συσσωρεύονται γίνεται μεγάλα) και πολ/ζουν τους μεγάλους όρους)

Αν όμως οι αριθμοί έχουν διαταχθεί από τον μικρότερο προς το μεγαλύτερο (σε αύξουσα σειρά)

τότε τα συσσωρευμένα σφάλματα πολ/ζουν τους μικρότερους όρους
Εξαχνιώνουμε το σφάλμα που δημιουργείται

Ένα παρόμοιο πρόβλημα εμφανίζεται όταν αθροίζουμε εξαρτημένους αριθμούς.

Έστω ότι θέλουμε να υπολογίσουμε την τιμή e^{-x}, $x>0$

Για να βρούμε την τιμή e^{-x} θα βασιστούμε στο ανάπτυγμα Taylor της e^{-x},

$$e^{-x} = \underbrace{1 - x + \frac{x^2}{2} - \frac{x^3}{3!} + \frac{x^4}{4!} + \dots + \frac{(-1)^n x^n}{n!}}_{S_n(x)} + \dots$$

Συμβολίζουμε

$$S_n(x) = 1 + \sum_{k=1}^{n} (-1)^k \cdot \frac{x^k}{k!}, \qquad \lim_{n\to\infty} S_n(x) = e^{-x}$$

Για να υπολογίσουμε (προσεγγίσουμε) την έκφραση e^{-x}

Θεωρούμε ένα μέγεθος N και υπολογίζουμε το $S_N(x)$

Εφαρμόζουμε τον ψευτοκώδικα.

$S = 1$

Για $k = 1, \ldots, N$

$$S = S + \frac{(-1)^k x^k}{k!}$$

Παραγωγούμε (για Η/Υ) ότι δημιουργούμε μεγάλα σφάλματα.

Επίσης θα πρέπει $\widehat{S}_N \to 0$, $N \to \infty$, δεν γίνεται αυτό για μεγάλα $\times$

Γιατί γίνεται αυτό;

Τα ενδιάμεσα αθροίσματα, οι όροι δηλαδή $S_1, S_2, S_3, S_4, \ldots$

είναι μεγαλα κατά απόλυτο τιμή από το πραγματικό σύνολο S_N

Η γενική εξήγηση που μπορούμε να δώσουμε είναι τα μικρά θραύσματα που δημιουργούνται με την στρογγύλευση σε αριθμούς μηχανής πολ/ζονται με μεγάλους κατ' απόλυτο τιμή και η συσώρευση τους οδηγεί στην μη-αγνεία στο 0 του $\tilde{S}_N$

Τι μπορούμε να κάνουμε για να το διορθώσουμε;

Γνωρίζουμε ότι $e^{-x} = \frac{1}{e^x}$, $x > 0$

Οπότε υπολογίζουμε τον αριθμό $S_N = 1 + \sum_{k=1}^{N} \frac{x^k}{k!}$

(Προσθέτουμε μόνο θετικούς αριθμούς)

και έτσι έχουμε $e^{-x} \approx \frac{1}{S_N}$

Ο πρώτος αλγόριθμος για τον υπολογισμό του e^{-x} δείχνει μια "αστάθεια" (οι μικρές μεταβολές λόγω των σφαλμάτων στρογγύλευσης οδηγούν σε μεγάλες μεταβολές στο αποτέλεσμα του αλγορίθμου, αντί για μικρό αριθμό παίρνουμε έναν πολύ μεγάλο)

Ο δεύτερος αλγόριθμος για τον υπολογισμό του e^{-x} είναι "ευσταθής" (παρά οι μικρές μεταβολές, λόγω των σφαλμάτων στρογγύλευσης δεν οδηγούν σε μεγάλες μεταβολές στο τελικό αποτέλεσμα)

Ο πρώτος αλγόριθμος για τον υπολογισμό του e^{-x} είναι αστάθης

Ο δεύτερος αλγόριθμος θα είναι ευστάθης

Ένα ακόμα παράδειγμα ασταθείας - ευστάθειας αλγορίθμων.

Θέλουμε να υπολογίσουμε το ολοκλήρωμα

$$I_n = \int_0^1 x^n e^{x-1} dx \quad , \quad n = 1, 2, 3, \ldots$$

για αρκετά μεγάλο n.

Ιδιότητες του $I_n = \int_0^1 x^n e^{x-1} dx$

i) Προφανώς $I_n > 0$, $n \geq 1$

ii) $I_{n+1} < I_n$ γιατί $x^{n+1} \leq x^n$ στο $[0,1]$

iii) $I_n = \int_0^1 x^n e^{x-1} dx \leq \int_0^1 x^n dx = \frac{1}{n+1}$

Συνεπώς από γνωστό θεώρημα του Απειροστικού Λογισμού

επειδή $0 \leq I_n \leq \frac{1}{n+1}$, $\lim_{n \to \infty} I_n = 0$

Υπολογισμός του I_n

Μπορούμε να δούμε ότι $I_1 = \int_0^1 x e^{1-x} dx = 1/e$

Επίσης με επαγωγή μπορούμε να δείξουμε

$$I_n = \frac{n!}{(-1)^{n+1}} \left(\frac{1}{e} - \sum_{k=0}^{n} \frac{(-1)^k}{k!} \right), \quad n = 1, 2, 3, 4, 5, \ldots$$

Είναι αυτός ο τύπος καταλληλος για να υπολογίσουμε τους όρους I_n ;

Αν χρησιμοποιήσουμε τον προηγούμενο τύπο για τον υπολογισμό των I_n, τότε ο αλγόριθμος θα είναι ασταθής

Λόγοι αστάθειας

(i) Πιθανά σφάλματα από την άθροιση όρων με εναλλαγή προσήμου

(ii) Για μεγάλα n, $\frac{1}{e} \simeq \sum_{k=0}^{n} \frac{(-1)^k}{k!}$

όπου αφαιρούμε κοντινούς αριθμούς

Κατασκευή ενός αλγορίθμου για τον υπολογισμό του I_n.

Από ολοκλήρωση κατά μέρη έχουμε

$$I_n = \int_0^1 x^n e^{x-1} dx = x^n e^{x-1} \Big|_{x=0}^{1} - \int_0^1 e^{x-1} (n x^{n-1}) dx$$

$$= (1 e^0 - 0) - n \int_0^1 x^{n-1} e^{x-1} dx = 1 - n I_{n-1}$$

Βρίσκουμε $I_1 = \frac{1}{e}$ και για τον υπολογισμό του I_n έχουμε τον ψευτοκώδικα

$I = 1/e$

Για $k = 2, \ldots, N$

$I = 1 - kI$

Αν τον εφαρμόσουμε στον Η/Υ προκύπτουν αποτελέσματα

Μετά από κάποιους όρους παίρνουμε αρνητικό αριθμό!!

Ξέχνημα του φαινομένου.

Οι όροι που παράγαμε στον Η/Υ είναι $\tilde{I}_n$

Ο αλγόριθμος παράγει τη μορφή

$$\tilde{I}_1 = fl(1/e), \qquad \tilde{I}_n = 1 - n\tilde{I}_{n-1}, \quad n \geq 2.$$

Έστω $\tilde{I}_1 = I_1 + \varepsilon_1$, $\tilde{I}_n = I_n + \varepsilon_n$, $n \geq 2$.

Τότε $\varepsilon_n = \tilde{I}_n - I_n = 1 - n\tilde{I}_{n-1} - (1 - nI_{n-1})$

$$= -n(\tilde{I}_{n-1} - I_{n-1}) = -n\varepsilon_{n-1}$$

$$= -n(-(n-1))\varepsilon_{n-2} = \cdots = (-1)^{n-1} n!\, \varepsilon_1$$

Ξέχνημα του φαινομένου.

Οι όροι που παράγουμε στον Η/Υ είναι $\tilde{I}_n$

Ο αλγόριθμος παίρνει τη μορφή

$$\tilde{I}_1 = fl(1/e), \qquad \tilde{I}_n = 1 - n\tilde{I}_{n-1}, \quad n \geq 2.$$

Έστω $$\tilde{I}_1 = I_1 + \varepsilon_1, \qquad \tilde{I}_n = I_n + \varepsilon_n, \quad n \geq 2.$$

Τότε $$\varepsilon_n = \tilde{I}_n - I_n = 1 - n\tilde{I}_{n-1} - (1 - nI_{n-1})$$

$$= -n(\tilde{I}_{n-1} - I_{n-1}) = -n\varepsilon_{n-1}$$

$$= -n(-(n-1)) = \cdots = (-1)^{n-1} n!\, \varepsilon_1$$

Για να καταλάβουμε το μέγεθος του σφάλματος

$n=18$, $\varepsilon_{18} = (-1)^{17}(18!)\varepsilon_1$

Έστω ότι χρησιμοποιούμε Η/Υ με ακρίβεια 16 ψηφίων ($k=16$) οπότε

$u = \frac{1}{2} \times 10^{-15}$, $|\varepsilon_1| \leq \frac{1}{2} \times 10^{-15}$

Επομένως $18! \cong 6.4023 \times 10^{15}$

Αυτός ο αλγόριθμος είναι <u>ασταθής</u>

Ένας άλλος αλγόριθμος

$$I_n = 1 - n I_{n-1} \Rightarrow I_{n-1} = \frac{1 - I_n}{n}$$

Επειδή $I_n \to 0$, $n \to \infty$

Θεωρούμε ότι για μεγάλο $m > n$, $I_m \approx 0$

και υπολογίζουμε τους όρους $I_{m-1}, I_{m-2}, \ldots, I_{n+1}, I_n$

In [1]:
```
import numpy as np
```

In [2]:
```
#### Υπολογισμός αθροισμάτων
def g(n):
    s=1
    for k in range(1,n+1):
        s=s+1/(k*(k+1))
    return s

### Τυπώνουμε το άθροισμα μέχρι έναν συγκεκριμμένο όρο
for j in range(1,8):
    n=10**j-1 # Το όρος μέχρι τον οποίο θα αθροίσουμε

    print('Το άθροισμα S_n με n=%d: %.15f Το ακριβές άθροισμα:%.15f'%(n,g(n),2-1/(n+1)))
```

```
Το άθροισμα S_n με n=9: 1.900000000000000 Το ακριβές άθροισμα:1.900000000000000
Το άθροισμα S_n με n=99: 1.990000000000000 Το ακριβές άθροισμα:1.990000000000000
Το άθροισμα S_n με n=999: 1.998999999999997 Το ακριβές άθροισμα:1.999000000000000
Το άθροισμα S_n με n=9999: 1.999899999999997 Το ακριβές άθροισμα:1.999900000000000
Το άθροισμα S_n με n=99999: 1.999989999999978 Το ακριβές άθροισμα:1.999990000000000
Το άθροισμα S_n με n=999999: 1.999998999999913 Το ακριβές άθροισμα:1.999999000000000
Το άθροισμα S_n με n=9999999: 1.999999899998901 Το ακριβές άθροισμα:1.999999900000000
```

In [3]:
```
#### Υπολογισμός αθροισμάτων (αλλαγή σειράς αθροίσεως)
def g(n):
    s=1/(n*(n+1))
    for k in range(0,n):
        s=s+1/((n-k)*(n-k+1))
    return s+1

### Τυπώνουμε το άθροισμα μέχρι έναν συγκεκριμμένο όρο
for j in range(1,8):
    n=10**j-1 # Το όρος μέχρι τον οποίο θα αθροίσουμε

    print('Το άθροισμα T_n με n=%d: %.15f Το ακριβές άθροισμα:%.15f'%(n,g(n),2-1/(n+1)))
```

```
Το άθροισμα T_n με n=9: 1.911111111111111 Το ακριβές άθροισμα:1.900000000000000
Το άθροισμα T_n με n=99: 1.990101010101010 Το ακριβές άθροισμα:1.990000000000000
Το άθροισμα T_n με n=999: 1.999001001001001 Το ακριβές άθροισμα:1.999000000000000
Το άθροισμα T_n με n=9999: 1.999900010001000 Το ακριβές άθροισμα:1.999900000000000
Το άθροισμα T_n με n=99999: 1.999990000100001 Το ακριβές άθροισμα:1.999990000000000
Το άθροισμα T_n με n=999999: 1.999999000001000 Το ακριβές άθροισμα:1.999999000000000
Το άθροισμα T_n με n=9999999: 1.999999900000010 Το ακριβές άθροισμα:1.999999900000000
```

In [ ]:
```

```

In [1]:
```
import numpy as np
```

In [2]:
```
#### Υπολογισμός αθροισμάτων exp(-x)

def sn(n,x):
  s=1
  print('Το μερικό άθροισμα του αναπτύγματος Taylor της exp(%d) μέχρι τον '%(x))
  for i in range(1,n+1):
    k=np.math.factorial(i) #Συνάρτηση για παραγοντικό
    s=s+(x)**i/k
    print('όρο %d: %.15f'%(i, s))
  return


x=100.
sn(30,-x)
```

```
Το μερικό άθροισμα του αναπτύγματος Taylor της exp(-100) μέχρι τον
όρο 1: -99.000000000000000
όρο 2: 4901.000000000000000
όρο 3: -161765.666666666656965
όρο 4: 4004901.000000000000000
όρο 5: -79328432.333333328366280
όρο 6: 1309560456.555555582046509
όρο 7: -18531709384.714282989501953
όρο 8: 229484163631.158721923828125
όρο 9: -2526247758767.430175781250000
όρο 10: 25031071465218.460937500000000
όρο 11: -225490012389198.718750000000000
όρο 12: 1862185686397611.000000000000000
όρο 13: -14196858150424004.000000000000000
όρο 14: 100510597826873248.000000000000000
όρο 15: -664205775355108480.000000000000000
όρο 16: 4115271557032276992.000000000000000
όρο 17: -23999300986422931456.000000000000000
όρο 18: 132192768699439333376.000000000000000
όρο 19: -689870755962993508352.000000000000000
όρο 20: 3420446867349171077120.000000000000000
όρο 21: -16152494196042089103360.000000000000000
όρο 22: 72815419728463652716544.000000000000000
όρο 23: -314001597334604686032896.000000000000000
όρο 24: 1297735973761513955524608.000000000000000
όρο 25: -5149214310622960610705408.000000000000000
όρο 26: 19646748321625010354520064.000000000000000
όρο 27: -72190150316330463434637312.000000000000000
όρο 28: 255798773390653321214164992.000000000000000
όρο 29: -875197515254118288378036224.000000000000000
όρο 30: 2894790113561786962396839936.000000000000000
```

In [3]:
```
#### Υπολογισμός αθροισμάτων exp(-x) Δεύτερος αλγόριθμος
def sn(n,x):
  s=1
  print('Η προσέγγιση exp(%d) αν χρησιμοποιήσουμε το μερικό άθροισμα του αναπτύγματος Taylor της exp(%d) μέχρι τον '%(-x,x))
  for i in range(1,n+1):
    k=np.math.factorial(i) #Συνάρτηση για παραγοντικό
    s=s+(x)**i/k
    print('όρο %d: %.15f'%(i, 1/s))
  return


x=100.
sn(20,x)
```

```
Η προσέγγιση exp(-100) αν χρησιμοποιήσουμε το μερικό άθροισμα του αναπτύγματος Taylor της exp(100) μέχρι τον
όρο 1: 0.009900990099010
όρο 2: 0.000196039992158
όρο 3: 0.000005821817455
όρο 4: 0.000000230497899
όρο 5: 0.000000011406180
όρο 6: 0.000000000677250
όρο 7: 0.000000000046909
όρο 8: 0.000000000003713
όρο 9: 0.000000000000331
όρο 10: 0.000000000000033
όρο 11: 0.000000000000004
όρο 12: 0.000000000000000
όρο 13: 0.000000000000000
όρο 14: 0.000000000000000
όρο 15: 0.000000000000000
όρο 16: 0.000000000000000
όρο 17: 0.000000000000000
όρο 18: 0.000000000000000
όρο 19: 0.000000000000000
όρο 20: 0.000000000000000
```

In [1]:
```
import numpy as np
```

In [2]:
```
#### Υπολογισμός του ολοκλήρωματος I_n
def integral(n):
  s=1/np.exp(1) # Ο πρώτος όρος, I_1
  print('Υπολογισμός των όρων I_n, μέχρι n=%d '%(n))
  print('Όρος I_%d:%.5f'%(1, s))
  for k in range(2,n+1):
     s=1-k*s
     print('Όρος I_%d:%.5f'%(k, s))
  return


integral(20)
```

```
Υπολογισμός των όρων I_n, μέχρι n=20
Όρος I_1:0.36788
Όρος I_2:0.26424
Όρος I_3:0.20728
Όρος I_4:0.17089
Όρος I_5:0.14553
Όρος I_6:0.12680
Όρος I_7:0.11238
Όρος I_8:0.10093
Όρος I_9:0.09161
Όρος I_10:0.08388
Όρος I_11:0.07735
Όρος I_12:0.07177
Όρος I_13:0.06695
Όρος I_14:0.06273
Όρος I_15:0.05903
Όρος I_16:0.05546
Όρος I_17:0.05719
Όρος I_18:-0.02945
Όρος I_19:1.55962
Όρος I_20:-30.19239
```

In [3]:
```
#### Υπολογισμός του ολοκλήρωματος I_n (Δεύτερος τρόπος υπολογισμού)
def integral2(n,m):
##### ο m > n ειναι ο δείκτης για τον οποίο θεωρώ οτι ο I_m είναι 0
  s=0 # Ο όρος I_m
  print('Υπολογισμός του όρου I_n=%d '%(n))
  print('Όρος I_%d:%.15f'%(m, s))
  for k in range(m,n,-1):
     s=(1-s)/k
     print('Όρος I_%d:%.15f'%(k-1, s))
  return

## θεωρούμε οτι I_50 ειναι περιπου 0
integral2(20,50)
```

```
Υπολογισμός του όρου I_n=20
Όρος I_50:0.000000000000000
Όρος I_49:0.020000000000000
Όρος I_48:0.020000000000000
Όρος I_47:0.020416666666667
Όρος I_46:0.020842198581560
Όρος I_45:0.021286039161270
Όρος I_44:0.021749199129750
Όρος I_43:0.022232972747051
Όρος I_42:0.022738768075650
Όρος I_41:0.023268124569627
Όρος I_40:0.023822728669033
Όρος I_39:0.024404431783274
Όρος I_38:0.025015270979916
Όρος I_37:0.025657492868950
Όρος I_36:0.026333581273812
Όρος I_35:0.027046289409061
Όρος I_34:0.027798677445455
Όρος I_33:0.028594156545722
Όρος I_32:0.029436540710736
Όρος I_31:0.030330108102790
Όρος I_30:0.031279673932168
Όρος I_29:0.032290677535594
Όρος I_28:0.033369286981531
Όρος I_27:0.034522525464945
Όρος I_26:0.035758424982780
Όρος I_25:0.037086214423739
Όρος I_24:0.038516551423050
Όρος I_23:0.040061810357373
Όρος I_22:0.041736443027940
Όρος I_21:0.043557434407821
Όρος I_20:0.045544884075818
```

In [ ]: