# ON THE SOLUTION OF THE GPS LOCALIZATION AND CIRCLE FITTING PROBLEMS

GPS (sistemul global de pozitionare) foloseste in prezent 32 de sateliti cu obiectivul de a furniza informatii de incredere despre locatie si timp pentru orice pozitie de pe suprafata planetei unde exista linii neobstructionate de comunicare catre 4 sau mai multi sateliti.

**Abstract.**

Consideram problema de a localiza un utilizator folosind un set de pseudointervale zgomotase catre un grup de sateliti. Consideram atat nonliniara “least squares” (LS) formulare a problemei care este nonconvexa si “nonsmooth”, cat si varianta nonliniara “squared least squares” (SLS) in care funcita obiectiv este “smooth”, dar tot nonconvexa. Aratam ca SLS poate fi reformulate ca o generalizare de subproblema “trust region” si astfel poate fi rezolvata eficient. LS arata ca are legaturi stranse cu binecunoscuta problema de regresie “circle fitting”. In final, o metoda a unui punct fix pentru LS este analizata si rezolvata pentru obtinerea solutiei optime.

**1. Introducere.**

Fiecare satelit isi transmite locatia (x, y, z) si timestamp-ul (t). Userul estimeaza distanta fata de cel putin 4 sateliti dupa formula:

di = c(T - ti)

c = viteza luminii

T = ora user-ului

Ceaul user-ului nu este precis (o eroare de o milisecunda corespunde unei erori de 300 m). Din acest motiv distantele masurate se numesc „pseudoranges” si avem nevoie de un „clock bias” necunoscut. Din punct de vedere matematic aceste „pseudodistante” sau „pseudointervale” sunt date de urmatoarea ecuatie:

di ≈ x − ai − r, i = 1, . . . , m,

ai – locatia satelitului

r – eroarea de timp necunoscuta cauzata de ceas

x – locatia user-ului necunoscuta

di – a i-a pseudodistanta (poate sa fie si negativa)

m >= n + 1

Ecuatia de mai sus este o aproximare in cazul in care ne raportam la r ca fiind o eroare comuna, dar modelul exact este:

di = x − ai − r + εi, i = 1, . . . , m,

εi – necunoscuta de zgomot corespunzatoare pentru a i-a pseudodistanta

Problema de a estima x si r din setul d1, ..., dm este *problema de localizare GPS*.

O abordare populara este sa luam un satelit de referinta j si sa scadem a j-a ecuatie din toate celelalte. In acest fel eliminam eroarea r. Insa aceasta abordare are urmatoarele dezavantaje: nu avem o metoda clara prin care sa alegem acel satelit j si astfel solutia obtinuta este posibil sa fie sensibila fata de j si mai mult, scaderea acelei ecuatii din toate celelalte introduce dependente intre componentele de eraore si sunt greu de gestionat. O alta abordare este sa luam in considerare diferentele dintre toate ecuatiile pare. In mod surprinzator problema de localizare GPS este in stransa legatura cu problema „circle fitting”.

Lucrarea considera 2 formulari posibile pentru problema de localizare GPS:

- SLS (squared least squares) – nonconvex si smooth

- LS (least squares) – nonconvex and nonsmooth

Ambele abordari sunt generalizari ale problemei de circle fitting.

**2. Formularea problemei: navigarea GPS**

In aceasta sectiune se incearca formularea problemei LS. Intai arata ca pentru di = 0 problema coincide cu problema geometrica de circle fit, dar rezultatele sunt nonconvexe si nonsmooth lucru greu de rezolvat. In special, metodele iterative concepute pentru a rezolva problema nu garanteaza convergerea catre un minim global. Acesta este motivul pentru care gasirea unei solutii aproximative bune este un lucru important, poate fi un bun punct de inceput pentru o metoda iterativa.

## Functii ajutatoare, explicitarea functiilor

### Functii pentru generarea distantelor di

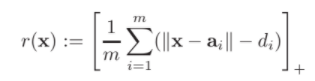
Folosim o distributie normala pentru a genera distantele di:exponentiala, norm.

Functia verifica daca toate elementele din di sunt pozitive: verificare\_vector\_di.

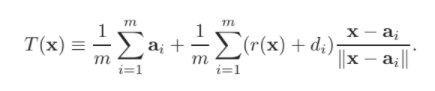
****

### Functii pentru GPS\_LS si CF\_LS

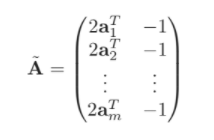
Calcularea eroarea de timp r cauzata de ceas: r()



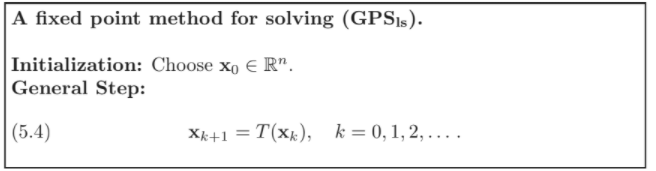
Functie pentru calculul lui xk+1: T()



Generarea pentru matrice A ̃: generare\_A\_caciula()



## GPS\_LS.py

In acest fisier avem functiile care se ocupa de rezolvarea problemei: 

Convergenta metodei GPS\_LS

def fixed\_point\_GPS\_LS\_afisare\_convergenta(xk, ai, di, pasi\_acuratete, x\_true):  
  
 nr\_sateliti = len(ai)  
 n\_spatiu = len(xk)  
  
 plt.title("Convergenta metodei GPS\_LS pentru o rulare la intamplare\n" + "Numarul satelitilor: " + str(nr\_sateliti) + "\nSpatiul n = "+ str(n\_spatiu))  
 lista\_pt\_plot = list()  
 for i in range(pasi\_acuratete):  
 xk = T(xk, ai, di)  
 lista\_pt\_plot.append(np.linalg.norm(xk - x\_true) )  
  
 plt.plot(range(pasi\_acuratete) , lista\_pt\_plot, linewidth = 3, label = "evolutia diferentei fata de x-ul real", color = 'magenta')  
 plt.legend()

Histograma erorilor

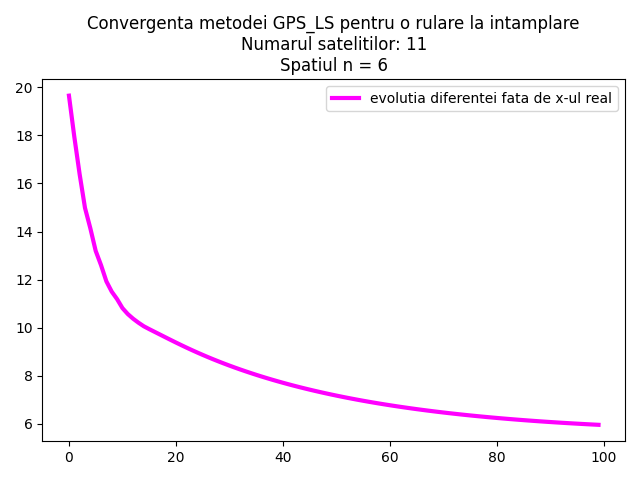
def fixed\_point\_GPS\_LS\_histograma\_erorilor(x, a, pasi\_acuratete, x\_true):  
 generari = 10\*\*3  
 B = list()  
  
 for i in range(generari):  
 #x\_true = [random.random() \* (10 - (-10)) + -10 , random.random() \* (10 - (-10)) + -10 ]  
 #x\_true = np.array(x\_true)  
  
 d = generare\_di(x\_true, a)  
 B.append(np.linalg.norm(fixed\_point\_GPS\_LS(x, a, d, pasi\_acuratete) - x\_true))  
  
 nr\_sateliti = len(a)  
 n\_spatiu = len(x)  
   
 plt.hist(B, bins = 13, ec='black', color='magenta')  
 plt.title("Histograma erorilor GPS\_LS\n" + "Numarul satelitilor: " + str(nr\_sateliti) + "\nSpatiul n = "+ str(n\_spatiu))

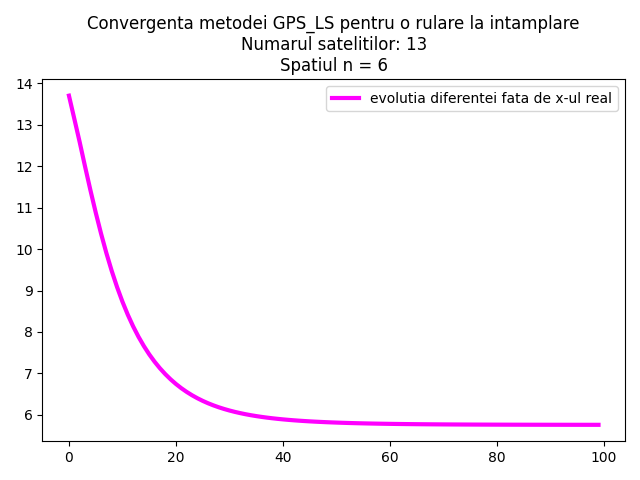
## CF\_LS.py

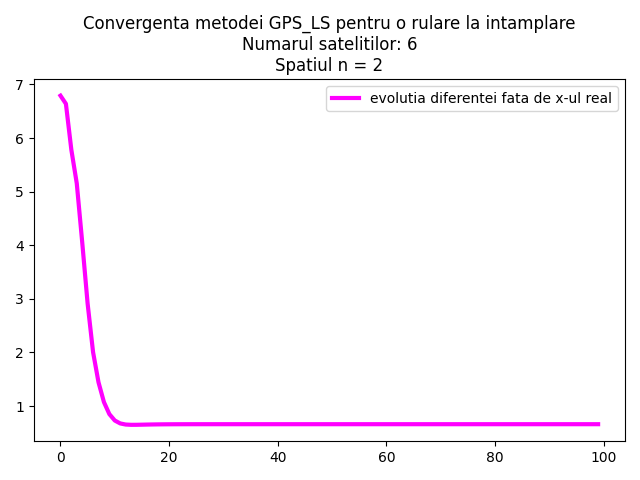
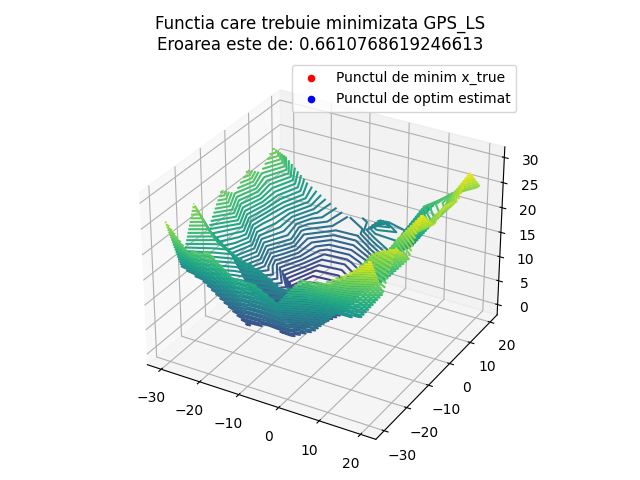
## 

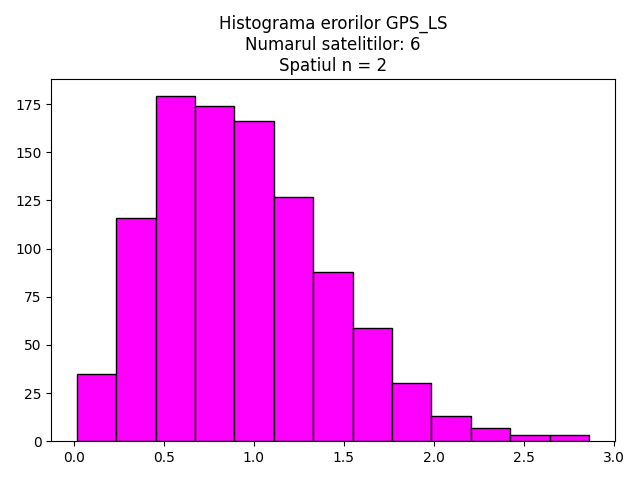
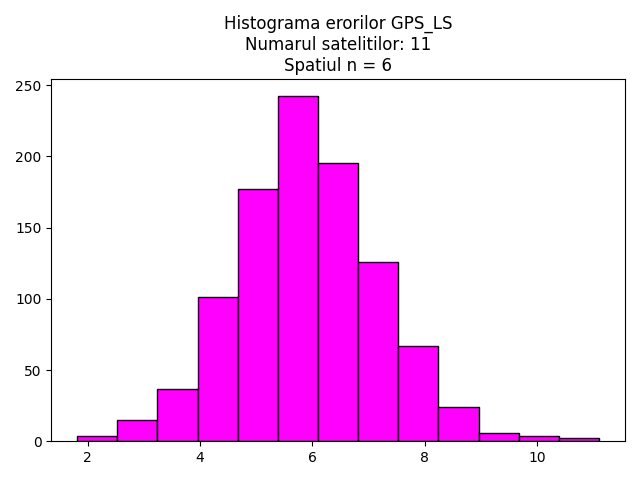
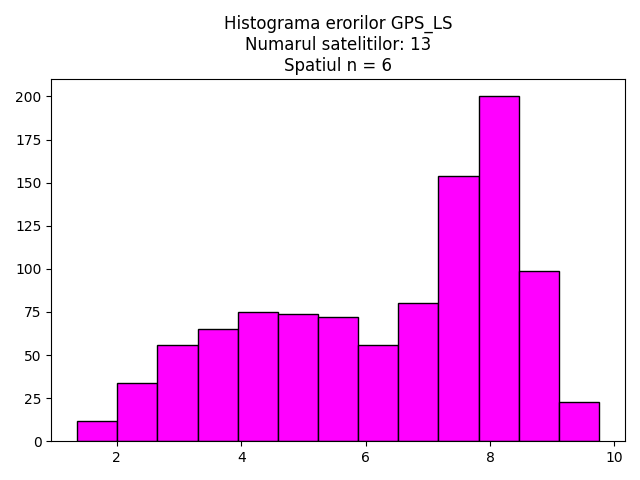
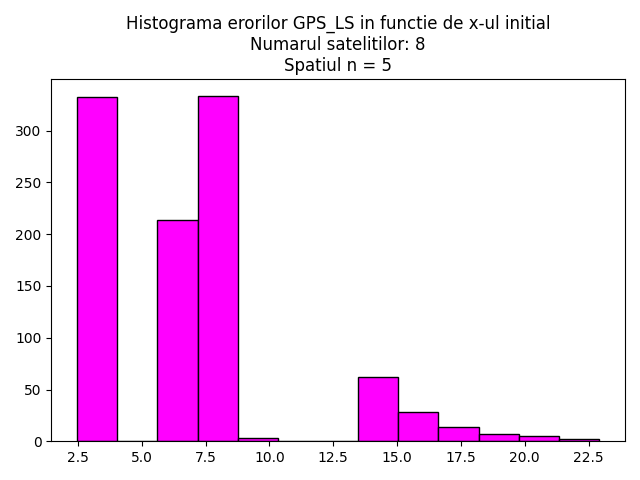
def CF\_LS\_step(xk, ai, di):  
 m = len(ai)  
 vec = [np.linalg.norm(xk - a) for a in ai]  
  
  
 vec = [elem / elemvec for elem, elemvec in zip(xk-ai, vec)]  
  
 vec = sum(vec)  
  
 vec = 1/m \* vec  
  
 vec = np.floor(vec)  
  
 vec = vec \* r(xk, ai, di)  
  
 vec = vec + sum(ai) \* 1/m  
   
 return vec  
  
def fixed\_point\_CF\_LS(xk, ai , di, pasi\_acuratete):  
   
 for i in range(pasi\_acuratete):  
  
 xk = CF\_LS\_step(xk, ai, di)  
   
 return xk

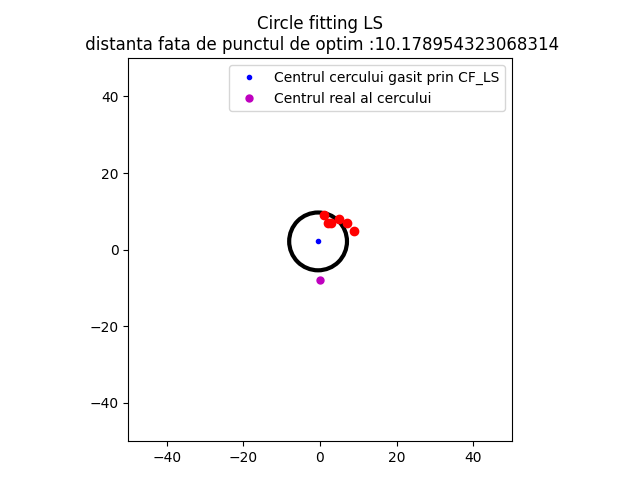
# Experimente numerice:

GPS\_LS:







CF\_LS: