Náhodná veličina X s normálnym rozdelením sa realizovala vektorom hodnôt

x = k*(rand(1,50)).^4 , kde k = 15

Odhadnite jej strednú hodnotu a rozptyl so spoľahlivosťou 90%.

Začneme strednou hodnotou. Jej bodový odhad je

```
>> xm = mean(x)
                        xm =    3.1141
```

Rozptyl je neznámy a tiež ho musíme odhadnúť predbežne bodovo:

```
>> S=sqrt(var(x))
                        S =    3.8991
```

Príkaz *var* paušálne koriguje hodnotu delením 1/(n-1), čo je v tomto prípade namieste. Na odhad strednej hodnoty musíme použiť v tomto prípade Studentovo rozdelenie s parametrom (stupňom voľnosti) 49. Konštantu c49 zistíme približne numericky, vyjde zhruba 1 / 2.5194489829 (ďalšie des. miesta sú nespoľahlivé).

```
>> g=inline('(1+x.^2/49).^(-25)/2.5194489829');
```

Pre spoľahlivosť 0,9 hľadáme príslušné *u*, aby g(u)=0,45. Orientačne: hľadané *u* leží medzi 1 a 2 (stačí skúsiť). Hľadajme presne bisekciou:

```
>> a=1; b=2; for k=1:40, s=(a+b)/2; if (quad(g,0,s, 1e-13)<0.45) a=s; else b=s; end, end, a,b

a =   1.676550892607338
b =   1.676550892608248
```

Takže zhruba u= 1.6765508926.

```
>> d = s*u/sqrt(50)
                        d =   0.3975103860233152
>> [xm-d, xm+d]
                        ans =   2.716572811203552    3.511593583250182
```

Hľadaná stredná hodnota je na 90% na intervale [2.7166   3.5116]

Rozptyl S je bodovým odhadom. Xí-kvadrát použijeme, vzhľadom na iba odhadnutú strednú hodnotu, s parametrom 49. Konštanta c bude približne 2.99e+030 . Túto konštantu musíme hľadať trpezlivo na viackrát. Najprv bude matlab protestovať:

```
>> fx=inline('x.^23.5.*exp(-x/2)');
>> quad(fx,0,490,1e-6)
Warning: Maximum function count exceeded; singularity likely.
> In quad at 92
ans =    3.950298428258852e+030
```

Po niekoľkých pokusoch sa ukazuje ako kandidát na hľadanú konštantu 3.95e+030.
Väčšiu hodnotu sme nedosiahli, pre dlhší interval integrovania sú už hodnoty nezmyselné. Varovanie Matlabu však netreba podceniť a výsledok si musíme overiť:

```
>> fx=inline('x.^23.5.*exp(-x/2)/3.95e+030');
>> quad(fx,0,1500)
         ans =   0.75679376700005
```

Matlab bez protestov integruje až po 1500 (ďalej už nie je presný), pričom miesto očakávanej jednotky máme len ca. 0.75. Nič sa nedeje, skorigujeme fx:

```
>> 3.95e+030*0.756793767
                                        ans =    2.989335379650000e+030
          >> fx=inline('x.^23.5.*exp(-x/2)/ 2.99e+030');
```

Teraz je to v poriadku, výsledky sú dobré a Matlab už nenarieka.

Späť k úlohe – potrebujeme nájsť 0.05 a 0.95 kvantil. Skusmo zistíme:

```
>> quad(fx,0,70)
                              ans =   0.97372586138726
>> quad(fx,0,60)
                              ans =   0.86494126544458
>> quad(fx,0,30)
                              ans =   0.01480284094728
>> quad(fx,0,40)
                              ans =   0.18318157598455
```

Potrebné kvantily sú teda na intervaloch [30 40], [60 70].

Spustíme bisekciu:

```
>> a=60;b=70; for k=1:60, s=(a+b)/2; if (quad(fx,0,s,1e-13)<0.95) a=s; else b=s; end,end, a, b

    a =  66.36385269696351
    b =  66.36385269696352

>> a=30;b=40; for k=1:60, s=(a+b)/2; if (quad(fx,0,s,1e-13)<0.05) a=s; else b=s; end,end, a, b

    a =  33.93115128787104
    b =  33.93115128787105
```

Rozptyl bude v týchto medziach:

```
>> S*49./[66.3638526969635 33.931151287871]

ans =    2.87889557489941   5.63065485876569
```

Rýchlo uspokojíme svoj pohľad kontrolou, že bodový odhad S je v tomto intervale.