Linearna Regresija

Jure Aplinc

30.4.2009

Povzetek

Linearna zveza $y = kx$ je najpreprostejša in najpogostejša zveza med dvema fizikalnima količinama, zlasti še, ker lahko tudi druge funkcijske odvisnosti v ozkem intervalu aproksimiramo z linearno zvezo: $\delta y = k\delta x$. Vemo, da sorazmernostni koeficient k za majhne δx limitira k odvodu $\frac{dy}{dx}$. $\frac{\delta y}{\delta x}$ vselej iščemo kot naklonski koeficient najboljše premice, ki jo potegnemo skozi naše mertve. V nadaljevanju bomo premice prilagajali z testom χ^2, ki je pogosto v uporabi tugi v programih, ki premice prilagajajo avtomatsko.

1 Naloga

Za meritve v datoteki "HitrostTokaOdFrekvence.txt"(naloga 6.1) določi parametra najboljše premice. Ker so podane napake hitrosti, lahko določiš tudi χ^2.

1.1 potek reševanja

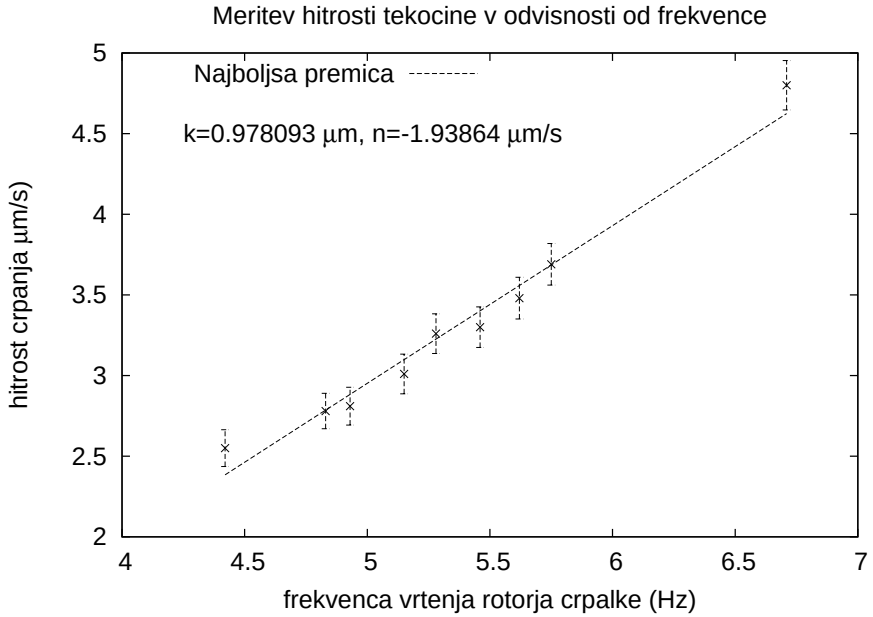
Za računanje naklona k, n in χ^2 sem napisal program v jeziku C.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
int main(void){
int i;
double e, x, y, sumi, sumx, sumy, sumxy, sumx2, sumy2, hi2, k, n;
FILE *fin;
i=0;
sumx=0;
sumy=0;
sumxy=0;
sumx2=0;
sumy2=0;
sumi=0;
fin=fopen("fluid.dat", "r");
while(fscanf(fin, "%lf %lf %lf", &x, &y, &e)==3){
i++;
sumi+=(1/(e*e));
sumx+=(x/(e*e));
sumy+=(y/(e*e));
sumxy+=(x*y/(e*e));
sumx2+=(x*x/(e*e));
sumy2+=(y*y/(e*e));
}
k=0;
```

```
n=0;
hi2=0;
k=(sumi*sumxy-sumx*sumy)/(sumi*sumx2-sumx*sumx);
n=(sumx2*sumy-sumx*sumxy)/(sumi*sumx2-sumx*sumx);
hi2=sumy2+k*k*sumx2-2*k*sumxy-2*n*sumy+2*k*n*sumx+n*n*sumi;
printf("y=k*x+n parametra k, n:\n k=%lf \n n=%lf \n hi2=%lf\n
 i+/-sqrt(2i)=%d+/-%g \n", k, n, hi2, i, (double) sqrt(2*i));
fclose(fin);
return 0;
}
```

1.2 rešitev

	k	n	χ^2	$n+/-\sqrt{2*n}$
C program	0.978093	-1.938642	5.437458	9+/-4.243
Gnuplot	0.978093+/-0.06346	-1.93864+/-0.3353	/	/



Slika 1: Preizkus mikrofluidične črpalke!

Komentar k rezultatom:

Vrednost χ^2 je znotraj meje $n+/-\sqrt{2*n}$, kar pomeni, da premica precej dobro opiše funkcijsko odvisnost izmerjenih točk.

Iz podobnosti rezultatov je očitno, da Gnuplot prilagaja krivulje z testom χ^2.

2 Naloga

Skozi oblak podatkov "Tintin.dat"potegni najboljšo premico. Uporabiš lahko kar korelacijske rezultate iz naloge 6.2.

2.1 potek reševanja

Uporabil sem program iz naloge 1. Ker napaka izmerkov ni navedena sem jo postavl na vrednost 1 (e=1), kar pomeni, da je program vse meritve obrvnaval enako. Nato sem vajo ponovil še z korelacijskimi faktorji in σ. Pri tem mi je bil v veliko pomoč program:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
int main(void){
FILE *fin;
int i;
double ypov, xpov, ysigma, xsigma, sum, sum2, sum3, yi, xi, a, n, skalab, r, R, e;
//sigmay (1 pass)
fin=fopen("Tintinb.dat", "r");
sum2=0;
sum=0;
n=0;
yi=0;
ypov=0;
ysigma=0;
while(fscanf(fin, "%lf %lf %lf", &a, &xi, &yi)==3){
yi=yi;
sum+=yi;
sum2+=yi*yi;
n+=1;
}
ypov=sum/n;
ysigma=sqrt(sum2/n-ypov*ypov);
printf("(one pass) ysigma=%g ypov=%g N=%g\n", ysigma, ypov, n);
fclose(fin);
//sigmax (1 pass)
fin=fopen("Tintinb.dat", "r");
sum2=0;
sum=0;
n=0;
xi=0;
xpov=0;
xsigma=0;
while(fscanf(fin, "%lf %lf %lf", &a, &xi, &yi)==3){
sum+=xi;
sum2+=xi*xi;
n+=1;
}
xpov=sum/n;
xsigma=sqrt(sum2/n-xpov*xpov);
printf("(one pass) xsigma=%g xpov=%g N=%g\n", xsigma, xpov, n);
fclose(fin);
//r(a, b)
fin=fopen("Tintinb.dat", "r");
skalab=0;
n=0;
r=0;
xi=0;
```

```
yi=0;
while(fscanf(fin, "%lf %lf %lf", &a, &xi, &yi)==3){
skalab+=xi*yi;
n++;
}
r=skalab/n;
printf("r=%lf, skalab=%lf\n", r, skalab);
R=(r-xpov*ypov)/(xsigma*ysigma);
double k;
k=R*ysigma/xsigma;
printf("R=%lf\n k=%lf\n", R, k);
fclose(fin);
return 0;

}
```

2.2 rešitev

(1) Vsi pacienti:

/	k	n	χ^2	$n+/-\sqrt{2n}$	R	x_{pov}	y_{pov}
C program	-0.124488	16.454574	419.574869	32+/-8	/	/	/
Korel. koef.	-0.124488	/	/	/	-0.394090	12.4	14.9
Gnuplot	-0.124488 +/-0.05301	16.4546+/-0.9336	/	/	/	/	/

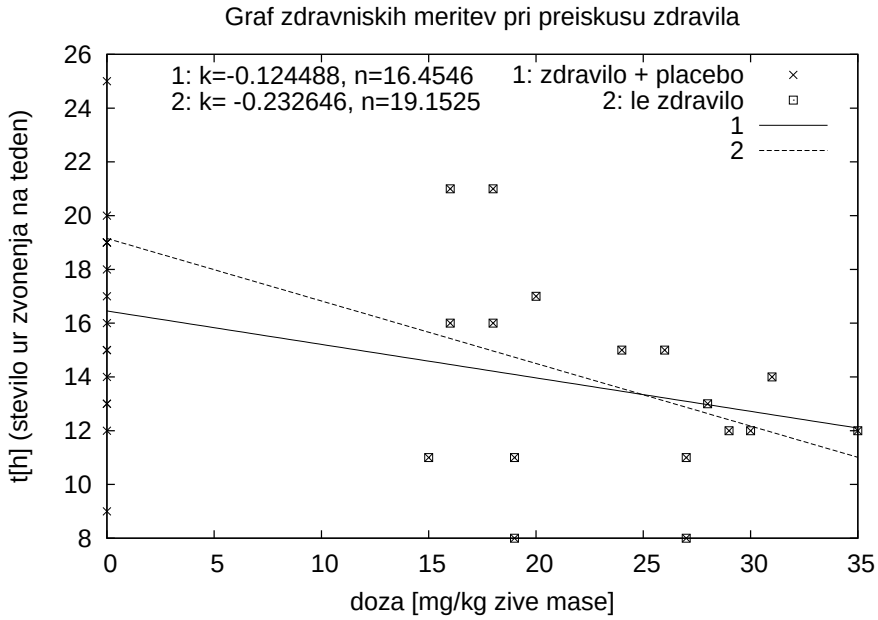
(2) Le pacienti, ki so dobili zdravilo:

/	k	n	χ^2	$n+/-\sqrt{2*n}$	R	x_{pov}	y_{pov}
C program	-0.232646	19.152526	194.507424	17+/-5.83095	/	/	/
Korel. koef.	-0.232646	/	/	/	-0.380963	23.4	13.7
Gnuplot	-0.232646+/-0.1458	19.1525+/-3.523	/	/	/	/	/

Komentar:

1: (vsi pacienti) χ^2 je 10.5-krat večji od $n+/-\sqrt{2*n}$. To pomeni, da je ujemanje meritev z linearno funkcijo slabo, kar napove že R, ki je le -0.394.

2: (le pacijenti, ki so prejeli zdravilo) Tudi pri tej prilaagoditvi premice meritvam dobimo vrednost χ^2, ki je 8.5-krat večja od $n+/-\sqrt{2*n}$, kar potrdi tudi korelacijsko koeficient, ki se la malo razlikuje od prejšnjega.



Slika 2: Preizkus novega zdravila proti zvonenju v ušesih!

3 Naloga

Skozi točke v histogramu podatkov "Interval.dat"poskusi potegniti najboljšo eksponentno funkcijo $w = Ae^{-\lambda * x}$, ki jo moramo najprej predelati v linearno zvezo. Z logaritmiranjem dobimo ln(w)=ln(A)-λx. V grafu y = ln(w) od x sta koeficienta premice k=-λ in n=ln(A). Po teoriji verjetnosti mora biti koeficient λ enak recipročni povprečni vrednosti histograma.

3.1 Potek reševanja

Iz podatkov sem narisal histograma pri čemer mi je bil v pomoč naslednji program:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
int main(void){
FILE *fin, *fout;
double x, xmax, xmin, s, s0, ds, all;
int n, i;
n=0;
fin=fopen("Interval.dat", "r");
xmax=1;
xmin=60;
all=0;
while (fscanf(fin, "%lf", &x)==1){
n++;
if(x>xmax) xmax=x;
if(x<xmin) xmin=x;
all+=x;
}
printf("n=%d\n xmax=%lf \n xmin=%lf\n all=%lf\n", n, xmax, xmin, all);
fclose(fin);
```

```
//histogram
double h[100];
for(i=0; i<99; i++){
h[i]=0;
}
s0=12.25;
ds=24.5;
fout=fopen("histd.out", "w");
for(i=0; i<=99; i++){
s=s0+ds*i;
fin=fopen("Interval.dat", "r");
while (fscanf(fin, "%lf", &x)==1){
if(x>(s-s0) && x<=(s+s0)){
h[i]++;
}
}
//if(h[i]!=0){
fprintf(fout, "%lf %lf\n", s, (h[i]));
//}
fclose(fin);
s=0;
}
fclose(fout);
return 0;
}
```

Podatke za število fotonov iz histogramov sem nato logaritmiral.

3.2 Rešitev

(i) 100 predalčkov:

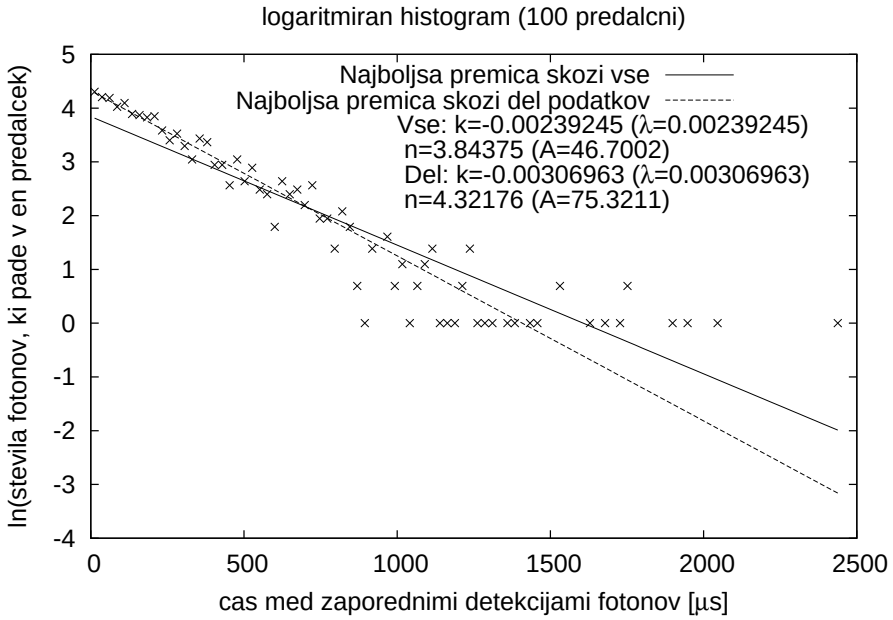
/	k	n	χ^2	$n+/-\sqrt{2n}$
C program	-0.002392	3.843749	25.016231	66+/-11.4891
Gnuplot	-0.00239245+/-0.0001377	3.84375+/-0.1417	/	/

/	A	λ	y_{pov}
C program	46.7002	0.002392	/
Gnuplot	46.7002	0.0023924	/
Povprečje	/	0.0016359	611.254

Najboljšo premico sem potegnil skozi vse meritve in nato samo skozi odsek za čase manjše od 857.5 μs.

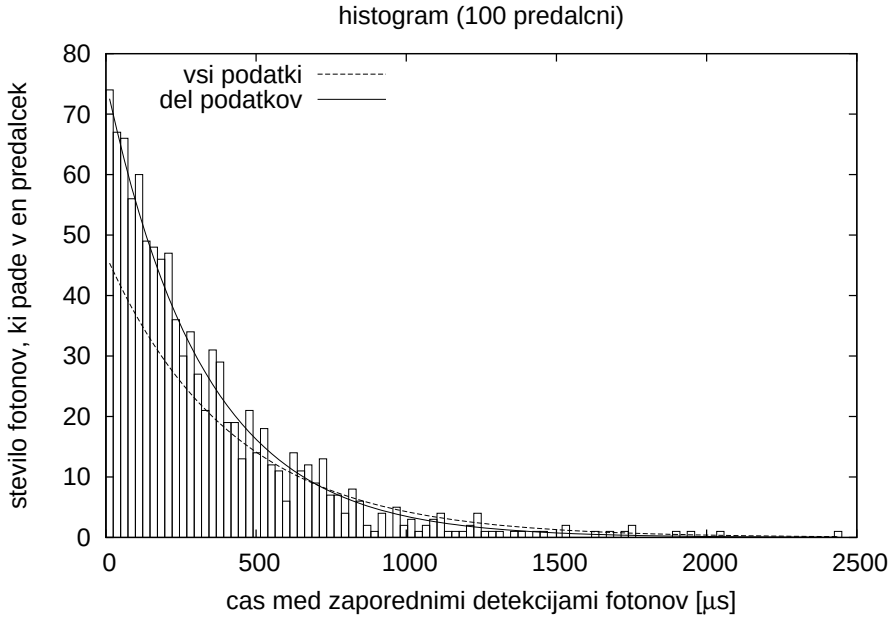
Rezultati, ki jih dobimo, če obravnavamo le točke od 0 do 771.75 μs so enaki bodisi če uporbljamo Gnuplot ali test χ^2:

/	A	λ
C program in gnuplot	75.3211	0.00306963



Slika 3: Najboljši premici skozi meritve, ki so logaritmirane po številu fotonov.

Poglejmo kako se koeficienti skladajo z histogramom:



Slika 4: Histogram (100 predalčni)!

Komentar k rezultatom:

Recipročna vrednost povprečja histograma je prav tako zelo podobna obema vrednostima za λ, ki smo ju izračunali že prej z C programom in Gnuplotom. Iz rezultatov (zadnjega histograma) lahko zaključimo, da smo boljše rezultate dobili, ko smo obravnavali eksponentno (linearno) funkcijo restringirano na intarvalu [0,771.75].

(ii) 50 predalčkov:

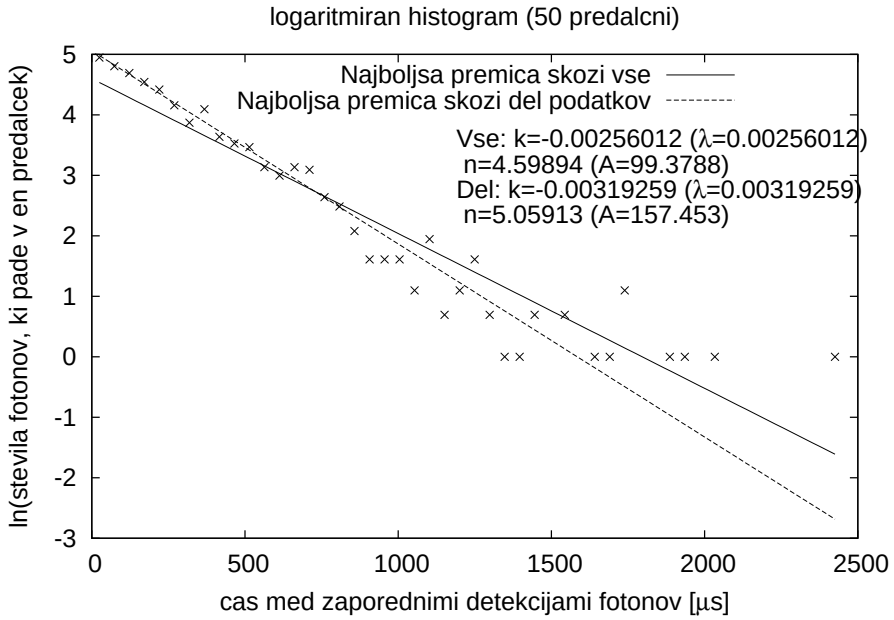
/	k	n	χ^2	$n+/-\sqrt{2n}$
C program	-0.002560	4.598939	11.129649	38+/-8.7178
Gnuplot	-0.00256012+/-0.0001488	4.59894+/-0.1705	/	/

/	A	λ	y_{pov}
C program	99.3788	0.002560	/
Gnuplot	99.3788	0.0025601	/
Povprečje	/	0.001533	652.167

Najboljšo premico sem potegnil skozi vse meritve in nato samo skozi odsek za čase manjše od 857.5 μs.

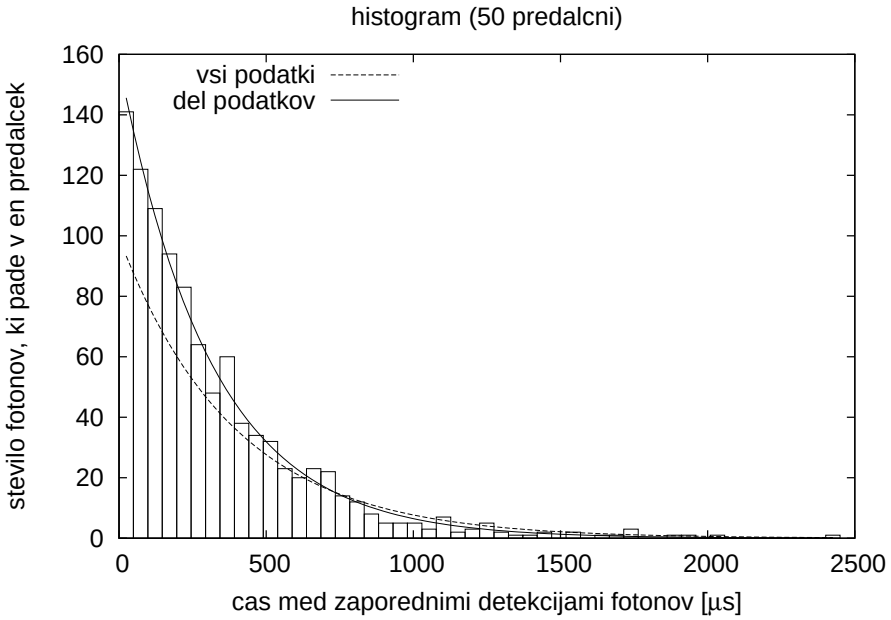
Rezultati, ki jih dobimo, če obravnavamo le točke od 0 do 857.5 μs so enaki bodisi če uporbljamo Gnuplot ali test χ^2:

/	A	λ
C program in gnuplot	157.453	0.00319259



Slika 5: Najboljši premici skozi meritve, ki so logaritmirane po številu fotonov.

Koeficienti, ki smo jih izračunali so v resnici parametri eksponentne funkcije, ki je prikazana na spodnjem grafu:



Slika 6: Histogram (50 predalčni)!

Komentar k rezultatom:

Program napisan v jeziku C izračuna k, n, λ, A skoraj tako kot Gnuplot, iz česar lahko sklepamo, da oba uporabljata isti način - (χ^2). Recipročna vrednost povprečja histograma je prav tako zelo podobna obema vrednostima za λ, ki smo ju izračunali že prej z C programom in Gnuplotom.

Iz rezultatov je razvidno, da dajo tisti, ki uporabljajo za test χ^2 le del definicijskega območja boljši rezultat.

(iii) 50 predalčkov (normirano):

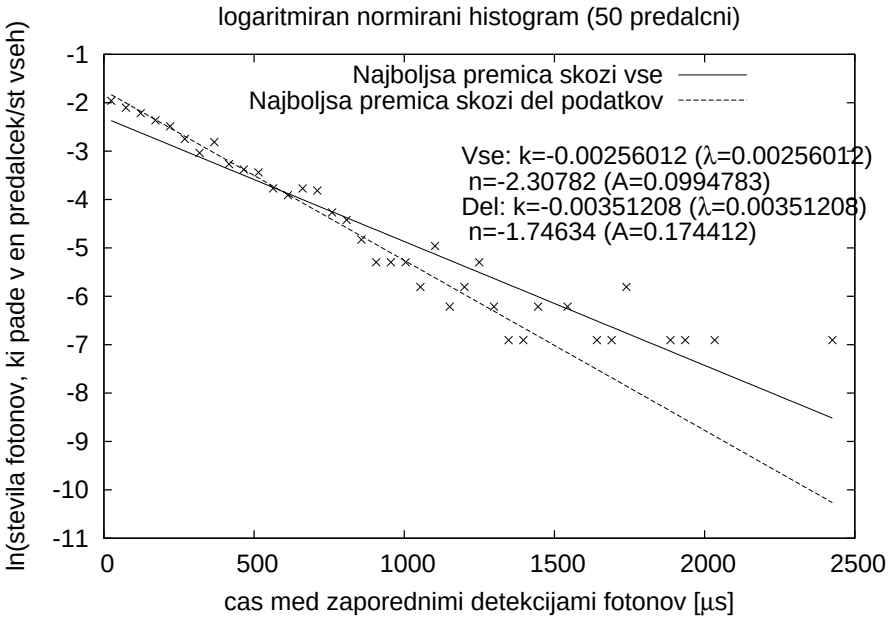
/	k	n	χ^2	$n+/-\sqrt{2n}$
C program	-0.002560	-2.307816	11.129649	38+/-8.7178
Gnuplot	-0.00256012+/-0.0001488	-2.30782+/-0.1705	/	/

/	A	λ	y_{pov}
C program	0.09947	0.002560	/
Gnuplot	0.0994783	0.00256012	/
Povprečje	/	0.001554	643.69

Najboljšo premico sem potegnil skozi vse meritve in nato samo skozi odsek za čase manjše od 1151.5 μs.

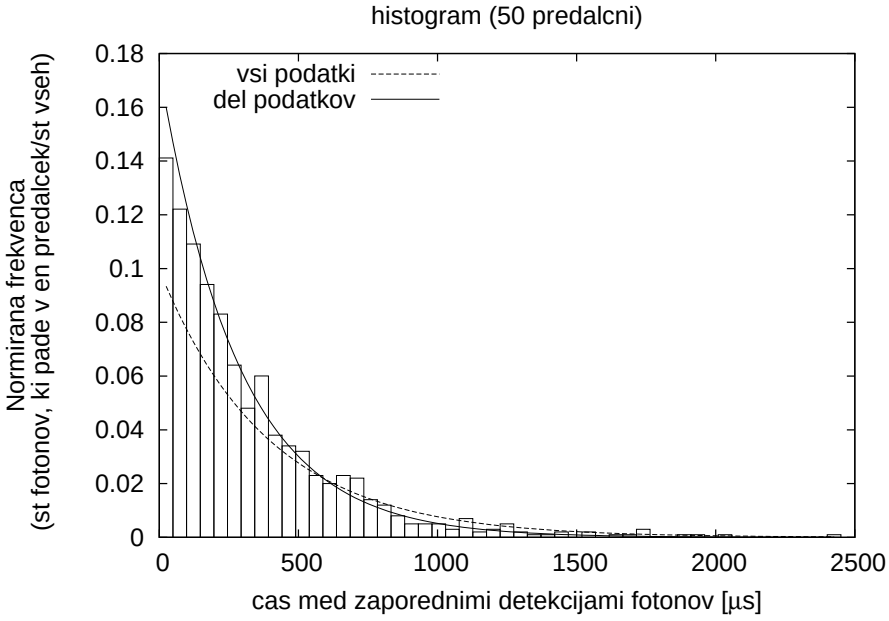
Rezultati, ki jih dobimo, če obravnavamo le točke od 0 do 1151.5 μs so enaki bodisi če uporbljamo Gnuplot ali test χ^2:

/	A	λ
C program in gnuplot	0.17441	0.00351208



Slika 7: Najboljši premici skozi meritve, ki so logaritmirane po normirani frekvenci.

Koeficienti, ki smo jih izračunali so v resnici parametri eksponentne funkcije, ki je prikazana na spodnjem grafu:



Slika 8: Histogram (50 predalčni) (normirani)!

Komentar k rezultatom:

λ je enaka bodisi jo izračunamo iz normiranega histograma bodisi iz nenormiranega!

Recipročna vrednost povprečja histograma je spet zelo podobna obema vrednostima za λ, ki smo ju izračunali že prej z C programom in Gnuplotom.

Iz rezultatov je razvidno, da dajo tisti, ki uporabljajo za test χ^2 le del definicijskega območja boljši rezultat.

4 Naloga

Teorija kemijske kinetike napove za sigmoidno krivuljo iz podatkov "Adrenalin.dat"(naloga 1.1) naslednjo odvisnost $\frac{F}{F_{max}} = \frac{C}{a+C}$, kjer pomeni a koncentracijo s polovičnim maksimalnim učinkom. Določi koeficienta F_{max} in a. Pretvori v linearno zvezo – ena pot je uvedba recipročnih spremenljivk $\frac{1}{F}$ in $\frac{1}{C}$, druga pa je uvedba spremenljivke $\frac{C}{F}$.

4.1 Potek reševanja

Uporabil sem metodo z recipročnimi spremenljivkami. Tako sem preoblikoval enačbo do oblike: $\frac{F_{max}}{F} = \frac{a}{C} + 1$ oziroma: $\frac{1}{F} = \frac{a}{C*F_{max}} + \frac{1}{F_{max}}$.

Nato sem spremenljivkam F in C priredil recipročne vrednosti-tako sem dobil linearno odvisnost.

Komentar: V datoteki "Adrenalin.dat"imamo podatke za razmerje $\frac{F}{F_{max}}$ (v procentih) v odvisnosti od C, torej je nemogoče izračunati kolikšna je sila (F_{max})! Lahko pa pokažemo, da je ta sila res maksimalna torej 1!

V nadaljevanju ne bom računal z procenti ampak z razmerjem med silama!

4.2 Rešitev

/	k	n	χ^2	$n +/- \sqrt{2n}$
C program (y=kx+n)	34.539110	0.648513	2.652376	6+/-3.4641
Gnuplot (y=kx+n)	34.5391+/-6.283	0.648513+/-0.467	/	/
Gnuplot (y=kx+1)	31.2177+/-4.275	/	/	/

/	a	F_{max}
C program	53.2589	1.54199
Gnuplot (y=kx+n)	53.2589+/-9.6883	1.54199+/-1.1104
Gnuplot (y=kx+1)	31.2177+/-4.275	1
gnuplot (direktno)	21.5337	1

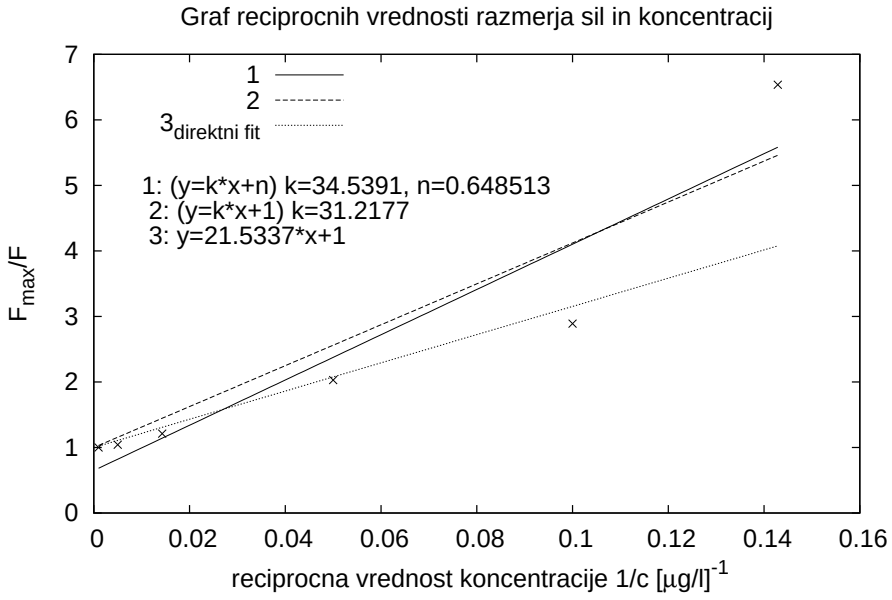
Komentar 1:

Vidimo, da je napaka, ki jo izračuna Gnuplot tolikšnja, da lahko n doseže tudi vrednost 1! Prav tako lahko k doseže tudi število 28.

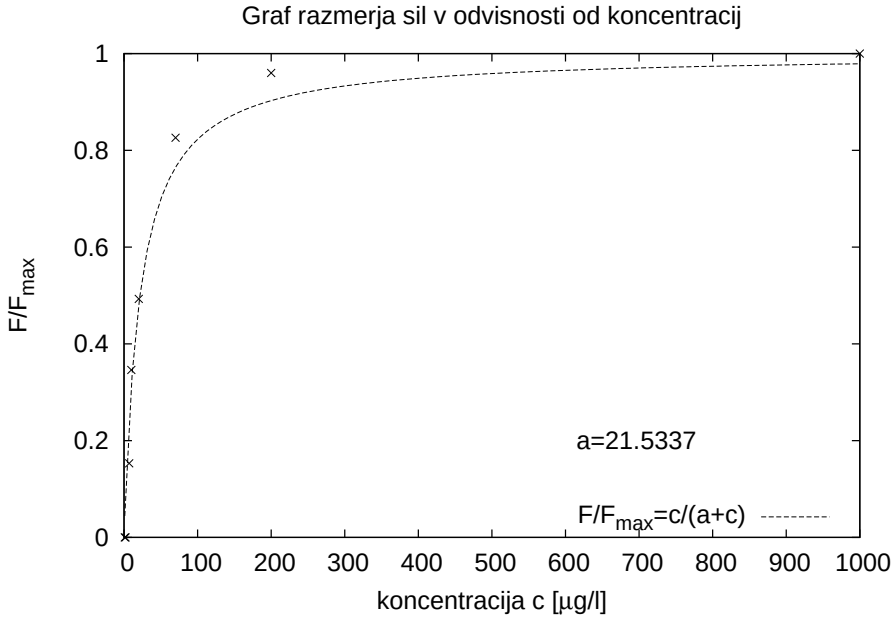
Posledično imata tudi a, in F_{max} precej veliko napako (celo tolikšnko, da pokrije tudi rešitev, ki jo da Gnuplot pri prilagajanju krivulje y=kx+1).

Komentar 2:

Tako veliko odstopanje koeficientov med sabo je posledica tega, da sta v datoteki dve od 8 vrednosti razmerja sil bili enaki 0. Ker recipročna vrednost števila 0 ne $\exists$ sem te podatke iz datoteke izbrisal in nato izračunal recipročne vrednosti-iz njih pa k, n, a, F_{max}. Seveda je očitno, da pri tako majhnem številu meritev to nikakor ne more prinesti enakega rezultata.



Slika 9: Graf recipročnih vrednosti in najboljše premice, ki sem jih dobil na različne načine!



Slika 10: Premico sem temu nelinearnemu grafu prilagodil direktno!