

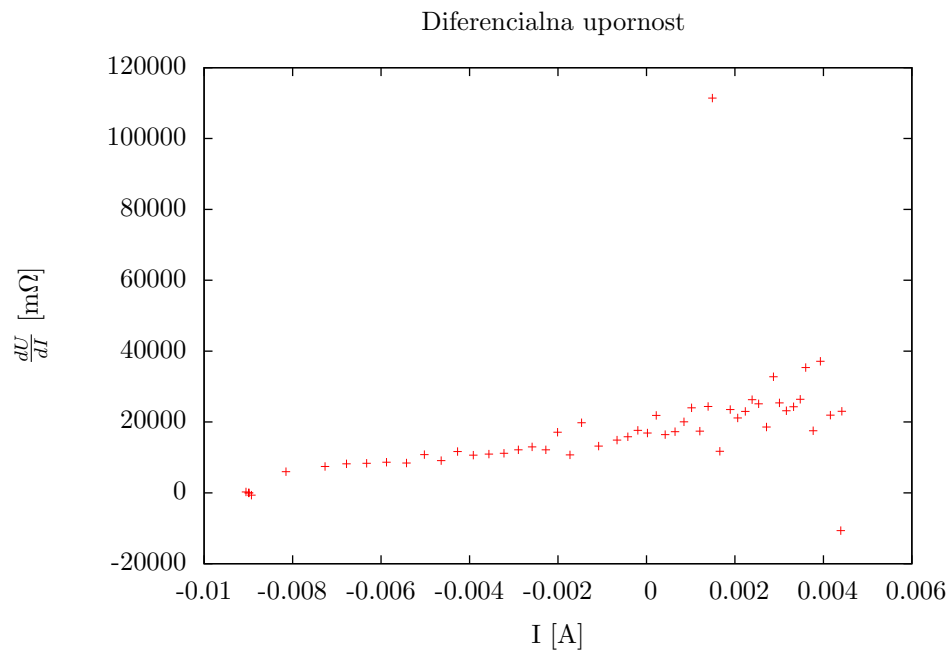
# Diferencialne operacije

Miha Čančula

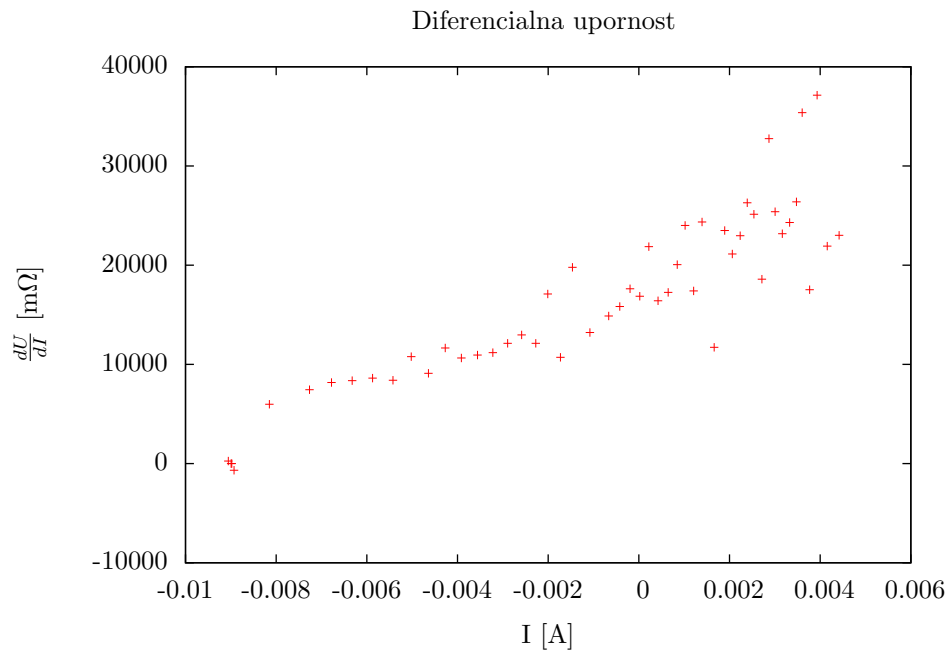
13. april 2009

## 1 Korozijska

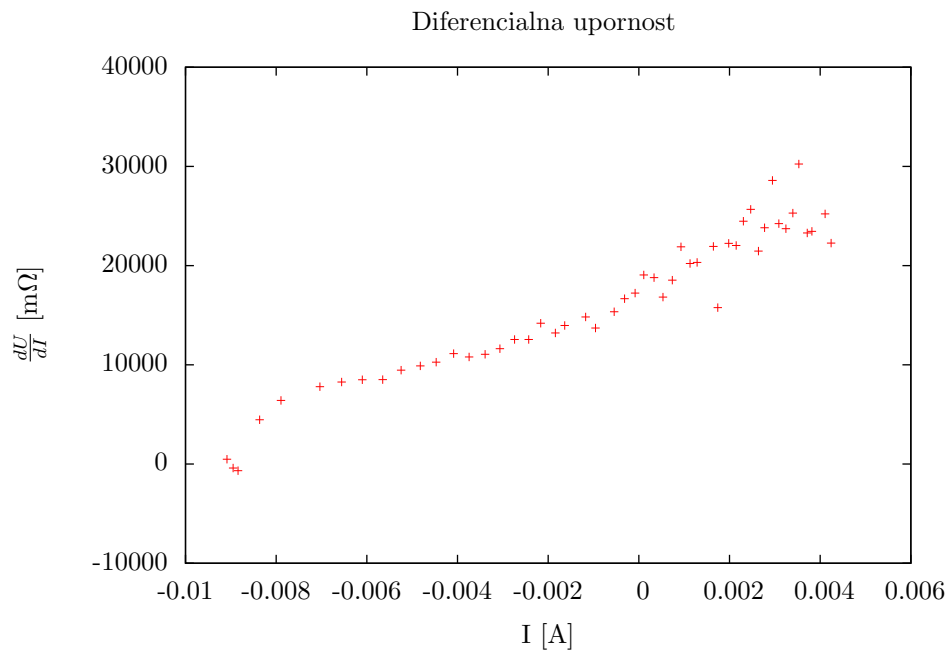
Najprej sem izračunal diferencialno upornost za vsaki dve zaporedni meritvi. Na tem grafu sta vidni dve večji odstopanji. To sta verjetno merski napaki, ki smo ju opazili že pri prvi vaji, ampak sta tu mnogo bolj vidni.



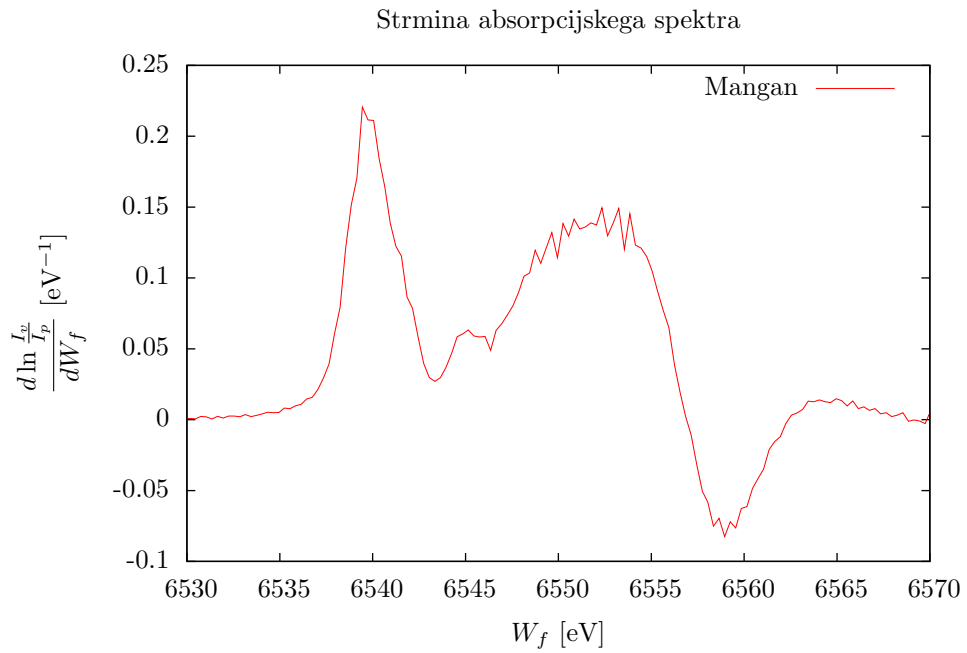
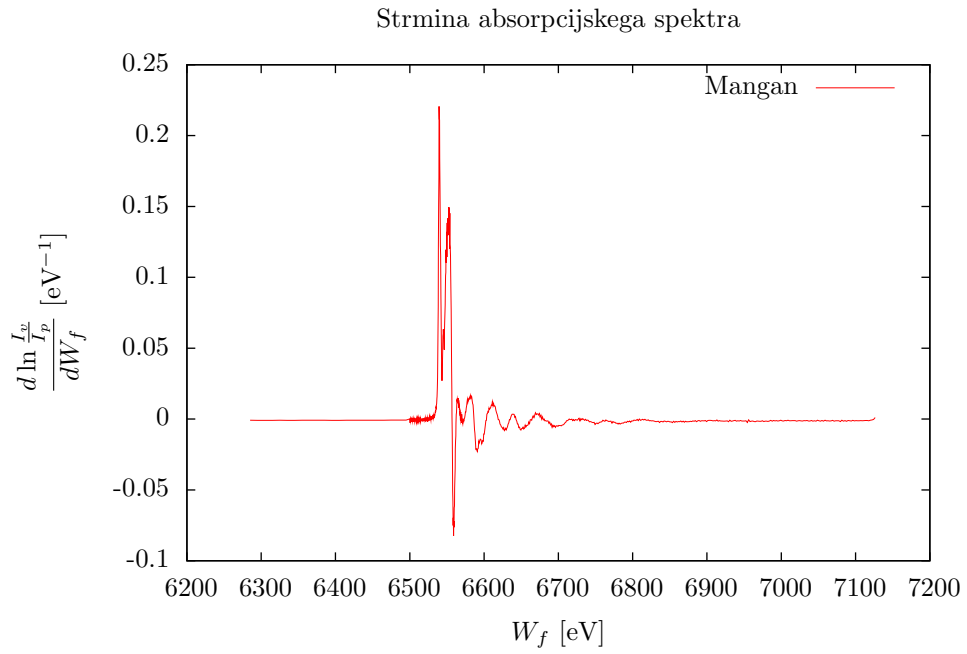
Za drugi graf sem odsranil ti dve meritvi in s tem zožal opazovani interval na osi y. V tem intervalu so odstopanja manjša in se bolje vidi tako spreminjanje diferencialne upornosti.



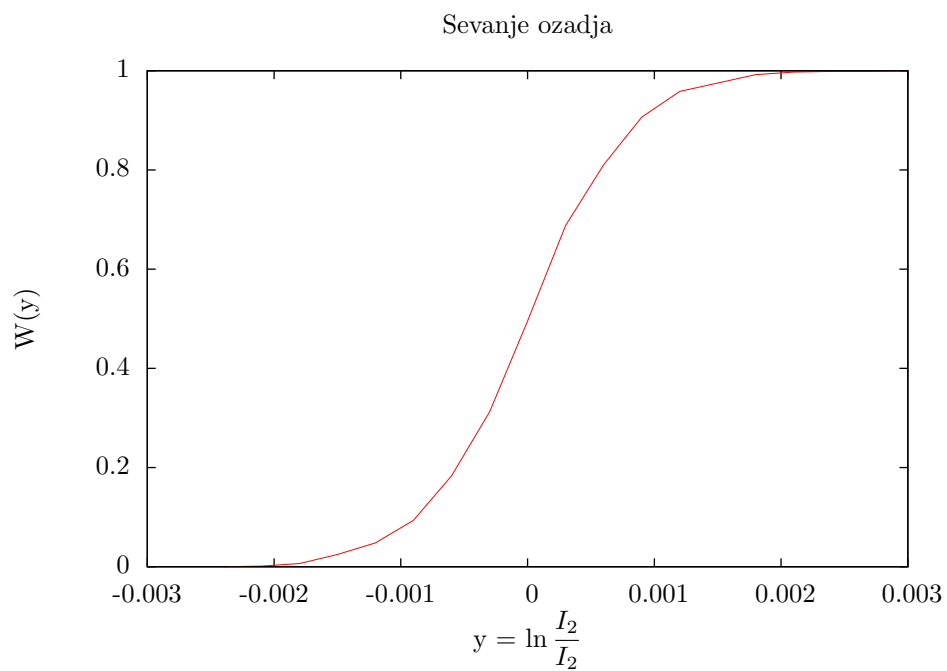
Sedaj sem izračunal diferencialno upornost na večjem intervalu, torej med meritvama  $y[i+1]$  in  $y[i-1]$ . Graf je bolj gladek, vendar na ta način izgubimo več podatkov.



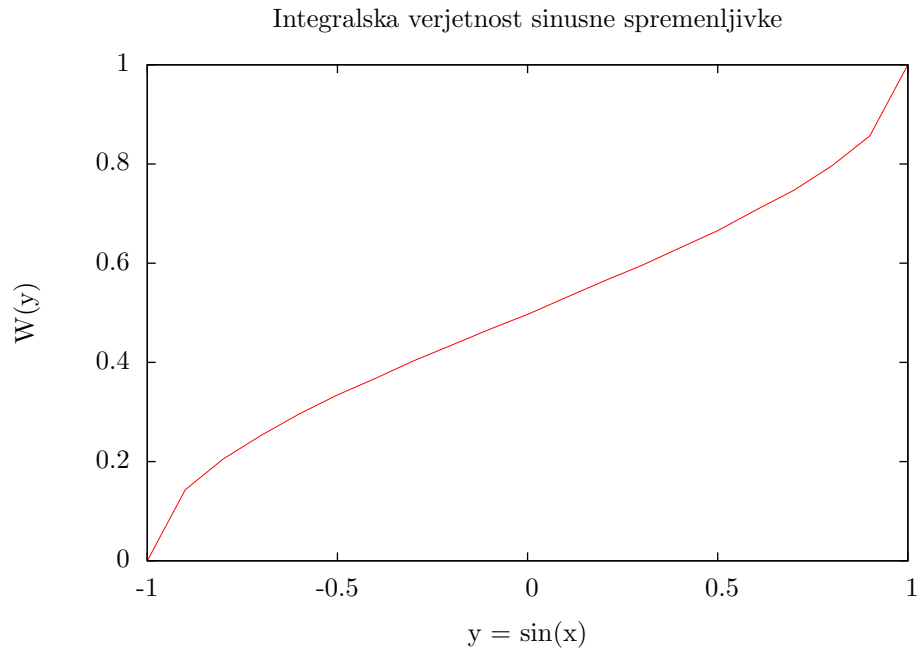
## 2 Strmina absorpcijskega spektra



### 3 Integralska verjetnost



Graf izgleda tak, kot bi ga pričakovali po Gaussovi porazdelitvi, z največjim naklonom blizu sredine. Ravno zrcalno izgleda graf sinusne porazdelitve, ki je najbolj strm na robovih.



Analično izračunana porazdelitev bi bila  $W(y) = \arcsin y$ . Oblika krivulje je podobna, le da je bolj strma na sredini in prepoločna na robovih. Dobljena krivulja je narejena iz histograma s 628 podatki in z 20 intervali, zato sklepam, da bi pri večjem številu meritev in intervalov dobili lepše ujemanje.

## 4 Izvorna koda

Prilagam izvorno kodo programov, s katerima sem izračunal odvode (prvi program) in integralski verjetnosti (drugi program). Napisana sta v jeziku C++.

### 4.1 Odvod

```
#include <fstream>
#include <iostream>
#include <math.h>

using namespace std;
int main()
{
    fstream kor("Korozija.dat", fstream::in);
    int n;
    kor >> n;
    double x[n], y[n];
```

```

for (int i = 0; i < n; i++) {
    kor >> y[i] >> x[i];
}
kor.close();
fstream kor_o("Korozija-odvod.dat", fstream::out);
for (int i = 0; i < n - 1; i++) {
    kor_o << (x[i] + x[i+1]) / 2 << " ";
    kor_o << (y[i+1] - y[i]) / (x[i+1] - x[i]) << endl;
}
kor_o.close();

fstream kor_o2("Korozija-odvod2.dat", fstream::out);
for (int i = 1; i < n - 1; i++) {
    kor_o2 << (x[i+1] + x[i-1]) / 2 << " ";
    kor_o2 << (y[i+1] - y[i-1]) / (x[i+1] - x[i-1]) << endl;
}
kor_o2.close();

fstream mn("Md29mn_00001.fio", fstream::in);
int st = 0;
double w[700], v[700], iz[700], bv;
while (!mn.eof()) {
    mn >> w[st] >> bv >> bv >> bv >> bv;
    mn >> v[st] >> iz[st] >> bv >> bv >> bv;
    st++;
}
mn.close();
fstream mn_o("Spekter-odvod.dat", fstream::out);
for (int i = 0; i < st - 1; i++) {
    mn_o << (w[i] + w[i+1]) / 2 << " ";
    mn_o << ((log(v[i+1] / iz[i+1]) - log(v[i] / iz[i])) / (w[i+1] - w[i]));
    mn_o << endl;
}
mn_o.close();
}

```

## 4.2 Integral

```

#include <fstream>
#include <iostream>
#include <math.h>

using namespace std;
int main()
{
    fstream oz("Ozadje.dat", fstream::in);

```

```

int n = 769;
double x[n], y[n];
for (int i = 0; i < n; i++) {
    oz >> x[i] >> y[i];
}
oz.close();

int p = 20;
int w[p];
for (int i = 0; i < p; i++) w[i] = 0;
for (int i = 0; i < n; i++) {
    for (int j = p - 1; j > 9 + y[i]*10.0 / 0.0003; j--) w[j]++;
}
fstream oz_i("Ozadje-int.dat", fstream::out);
for (int i = 1; i < p; i++)
    oz_i << (i - 9) *0.0003 << " " << w[i]*1.0 / n << endl;
oz_i.close();

int v[20];
for (int i = 0; i < 20; i++) v[i] = 0;
for (int j = 0; j < 628; j++) {
    for (int k = 19; k > 9 + 10*sin(0.1*j); k--) v[k]++;
}

fstream sin_i("Sinus-int.dat", fstream::out);
sin_i << -1.0 << " " << 0.0 << endl;
for (int i = 0; i < 20; i++) {
    sin_i << 0.1*(i - 9) << " " << v[i]*1.0 / 628 << endl;
}
sin_i.close();
}

```