

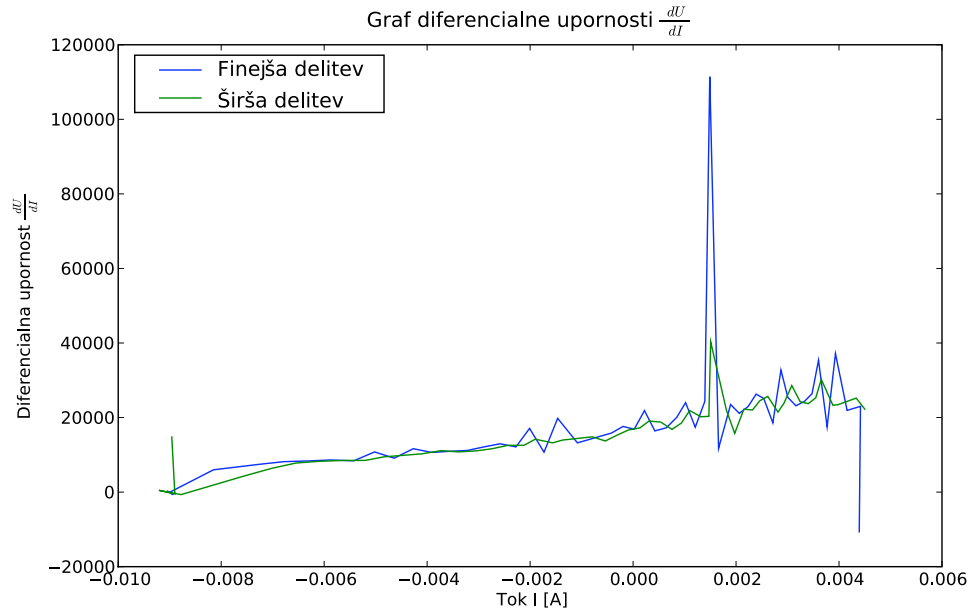
5. Domaća naloga - Diferencialne operacije

Urban Škudnik

2009-04-12

1 Korozija

Opis Pri prvi nalogi smo narisali graf diferencialne upornosti $\frac{dU}{dI}$ za podatke pri karoziji.



Tokovna odvisnost

Tehnični opis Na graf sem narisal isti niz podatkov, vendar enkrat z bolj širokim računskim intervalom, drugič pa z bolj ozkim intervalom. Pri ožjem intervalu uporabljam za izračun vrednosti odvoda v določeni točki dve sosednji točki (torej točko samo in njeno naslednico), za vrednost na abscisni osi pa se uporabi povprečje teh dveh točk, pri bolj širokem intervalu pa se za izračun odvoda v neki točki uporabita prejšnja in naslednja točka.

Ta graf in vsi ostali v tem poročilu so bili narisani z knjižico za Python *Matplotlib*.

Koda

```
korozija = csv.reader(open("Korozija.dat"), delimiter=" ")
data = list()
for row in korozija:
    c = list()
    for column in row:
        if column:
            c.append(float(column))
    data.append(c)

xx1 = list()
yy1 = list()

for i, ui in enumerate(data[0:-1]):
    n = data[i+1]

    y = (n[0]-ui[0])/(n[1]-ui[1])
    yy1.append(y)

    x = (n[1]+ui[1])/2
    xx1.append(x)

plot.plot(xx1, yy1, label=u"Finejša delitev")

xx2 = list()
yy2 = list()
for i, ui in enumerate(data[1:-1]):
    n = data[i+1]
    p = data[i-1]

    y = (n[0]-p[0])/(n[1]-p[1])
    yy2.append(y)

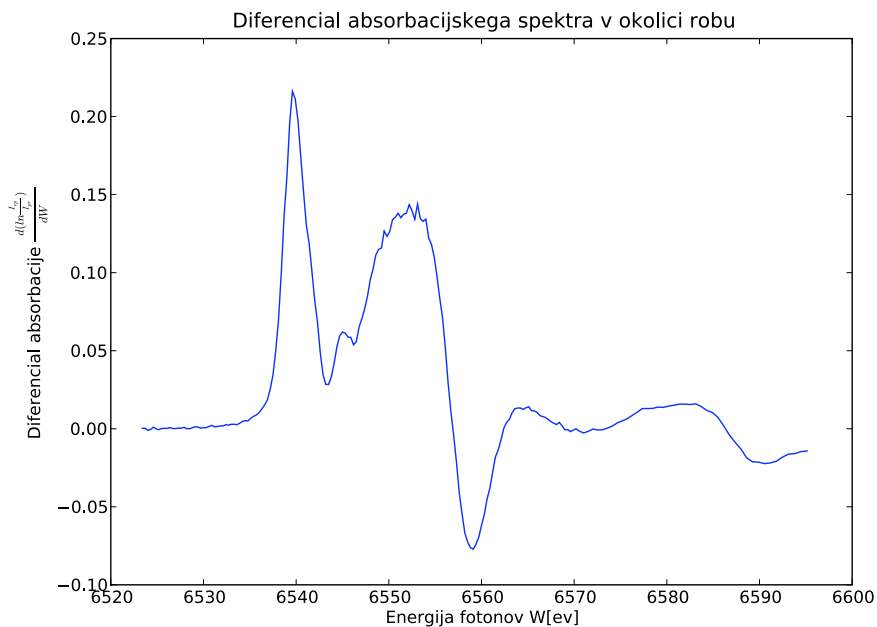
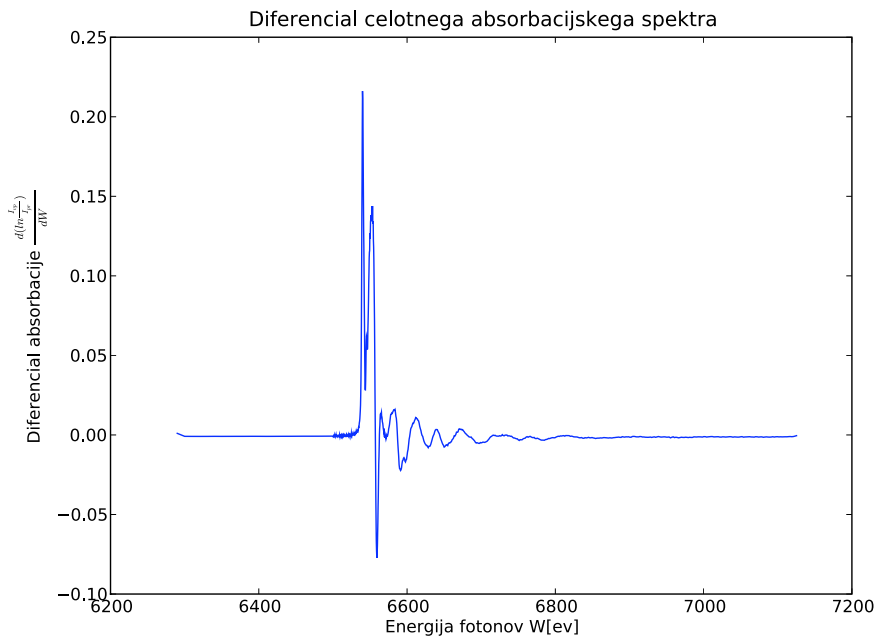
    xx2.append(ui[1])

plot.plot([i for u, i in data[1:-1]], yy2, label=u"Širša delitev")

plot.plot(xx, yy)
plot.xlabel("Tok I [A]")
plot.ylabel(r"Diferencialna upornost  $\frac{dU}{dI}$ ")
plot.title(r"Graf diferencialne upornosti  $\frac{dU}{dI}$ ")
plot.show()
```

2 Energija rentgenskega absorpcijskega robu kovinskega mangana

Opis Pri drugi nalogi smo se ponovno srečali z že videnimi podatki - tokrat smo iskali energijo rentgenskega absorpcijskega robu oziroma točko, v kateri ima absorpcijski spekter največjo strmino (je vrednost odvoda največja).



Rezultati Na zgornjem grafu lahko vidimo odvod po celotnem absorpcijskem spektru. Največjo strmino ima absorpcijski spekter pri energiji 6539.604 eV, ko je absorpcijski količnik 0.21613362226.

Ker nas zanima predvsem odvod v okolici robu, si lahko na drugem grafu ogledamo spreminjanje vrednosti odvoda samo v okolici le tega.

Koda (za celoten spekter, za omejitev na rob podobna)

```
absorb = csv.reader(open("Md29mn_00001.fio.txt"), delimiter=" ")
data = list()
for row in absorb:
    c = list()
    for column in row:
        if column:
            c.append(float(column))
    data.append(c)

xx, yy = [], []
emax, kmax = 0.0, 0.0

for i, ei in enumerate(data[1:-1]):
    n, p = data[i+1], data[i-1]

    y = (math.log(n[6]/n[7]) - math.log(p[6]/p[7])) / (n[0] - p[0])

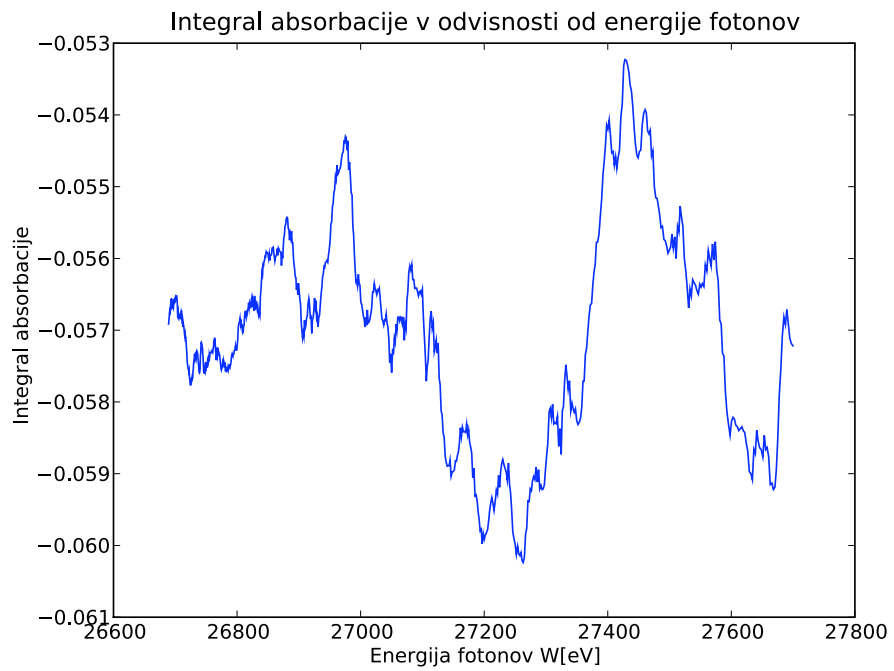
    if kmax < y:
        kmax = y
        emax = ei[0]
    yy.append(y)

print emax, kmax

plot.plot([x[0] for x in data[1:-1]], yy)
plot.xlabel("Energija fotonov [ev]")
plot.ylabel(r"Diferencial absorbacije $\frac{\ln \frac{I_{vp}}{I_{pr}}}{W}$")
plot.title(u"Diferencial absorbacijskega spektra")
plot.show()
```

3 Integralska porazdelitev ozadja

Opis Pri tretji nalogi sem za izračun integralske verjetnosti uporabil trapezno integracijo. Najprej sem uporabil za računanje podatke iz datoteke *Ozadje.dat*.



Integral absorbcije

Koda

```
ozd = csv.reader(open("Ozadje.dat"), delimiter=" ")
data = list()
for row in ozd:
    c = list()
    for column in row:
        if column:
            c.append(float(column))
    data.append(c)

xx = [ev for ev, i in data[1:]]
w = list()
w.append(0)

for i, ei in enumerate(data[1:]):
    p = data[i-1]
    h = ei[0]-p[0]
    wt = (w[-1] + 0.5*h*(p[1]+ei[1]))
    w.append(wt)

plot.plot(xx, w[1:])
plot.xlabel("Energija fotonov W[eV]")
plot.ylabel(r"Integral absorbacije")
plot.title(u"Integral absorbacije v odvisnosti od energije fotonov")
plot.show()
```