Modelska analiza I
2023/24

3. naloga – Numerična minimizacija

1. Thomsonov problem: na prevodno kroglo nanesemo N enakih (klasičnih) nabojev. Kako se razmestijo po površini?[1] Zahtevamo seveda minimum elektrostatične energije. Primerjaj učinkovitost in natančnost za različne minimizacijske metode, npr. Powellovo ali n-dimenzionalni simpleks (amebo oz. Nelder-Mead).

2. Problem optimalne vožnje skozi semafor, ki smo ga spoznali pri nalogi 1, lahko rešujemo tudi z numerično minimizacijo, če časovno skalo diskretiziramo.

   Lagrangianu $\int (\mathrm{d}v/\mathrm{d}t)^2\,\mathrm{d}t - \lambda \int v\,\mathrm{d}t$ lahko dodamo omejitev hitrosti v obliki členov $\exp(\beta(u - u_{\mathrm{lim}}))$, če hočemo (približno) zagotoviti $u \le u_{\mathrm{lim}}$. Izpolnitev pogoja je toliko ostrejša, kolikor večji β vzamemo. Poskusiš lahko tudi druge omejitvene funkcije, na primer kakšno funkcijo s polom.

   Za iskanje Lagrangeovega multiplikatorja lahko uporabiš bisekcijo ali kakšno drugo vgrajeno metodo za iskanje ničel na funkciji $l(\lambda) = \int v(\lambda, t)\,\mathrm{d}t$, kjer je $v(\lambda, t)$ rezultat minimizacije pri izbranem λ. V tem primeru je enakost izpolnjena eksaktno.

Nelinearna minimizacija: Iščemo minimum funkcije $f(\vec{r})$ več neodvisnih spremenljivk, ki jih združimo v vektor $\vec{r}$. Po najpreprostejši zamisli zaporedoma minimiziramo funkcijo po vsaki spremenljivki, vendar ta postopek praviloma zelo počasi konvergira. Rajši uporabimo bolj zahtevne, vendar hitrejše postopke. Zavedati se moramo, da so koordinate minimalne točke po katerikoli metodi določene samo z natančnostjo $\sqrt{\epsilon}$, če je ϵ relativna natančnost podajanja vrednosti funkcije f.

Kratka referenca metod iz *Numerical Recipes*:

- Podprogram `amoeba` implementira Nelder-Mead algoritem. Je preprost za uporabo, zahteva samo podprogram `func` za izračun funkcije f, je zelo robusten, vendar tudi počasen in približen. Načeloma ga lahko uporabimo tudi za negladke funkcije. Temelji na krčenju hipertetraedra (simpleksa), s katerim objamemo minimum, na področje z velikostjo tolerance.
- Podprogram `powell` predpostavlja zveznost minimizacijske funkcije in terja celo vrsto spremljevalnih procedur (po vrsti najprej minimizacijsko funkcijo `func`, nato `linmin` za 1D minimizacijo, ta pa `f1dim` in `brent`). Metoda določi konjugirane smeri v prostoru, nato pa vzdolž njih ujame minimum. Zato najde rezultat zelo natančno in hitro.
- Še nekoliko hitrejše metode, na primer metodo konjugiranih gradientov Fletcher-Reeves-Polak-Ribiere (`frprmn`) ali metodo s spremenljivo metriko Davidon-Fletcher-Powell (`dfpmin`), uporabimo, kadar lahko podamo tudi odvode funkcije, ki jo minimiziramo. Delujejo podobno Newtonovi metodi za iskanje ničel funkcije. Med podobne kvazi-Newtonske metode spada tudi metoda Broyden–Fletcher–Goldfarb–Shanno.

V zbirki *GSL* je podoben nabor minimizacijskih metod na voljo v modulu `gsl_multimin.h`, v Pythonu pa v modulu `scipy.optimize` s funkcijo `minimize(method='Nelder-Mead'|'Powell'|'BFGS'|...)`.

---

[1] Za prikaz se lahko poslužite algoritmov, ki določijo konveksno ovojnico množice točk (pod imeni "convex hull" v različnih orodjih).