3. DOMAĆA ZADAĆA – AK. GOD. 2024/25

Domaća zadaća

U okviru ove domaće zadaće potrebno je implementirati sljedeće metode optimiranja:

Postupak najbržeg spusta (gradijentni spust)

Algoritam je objašnjen na predavanjima i u skripti na stranici 4-58. Za funkcije koje će se koristiti u zadacima s ovim postupkom implementirajte analitičko računanje vektora gradijenta. Postupak treba podržavati dva načina rada: u prvom načinu točka se pomiče za čitav iznos dobivenog pomaka (iznos nenormiranog vektora gradijenta), dok u drugom načinu postupak korištenjem metode zlatnog reza pronalazi optimalan iznos pomaka na pravcu. Postupak je potrebno zaustaviti kada euklidska norma gradijenta postane manja od neke zadane mjere preciznosti. Potrebno je omogućiti da se bez prevođenja programa mogu definirati mjera preciznosti *e*, početna točka pretraživanja te hoće li postupak koristiti metodu zlatnog reza za određivanje optimalnog iznosa pomaka ili ne. Pretpostavljena vrijednost preciznosti je 10^{-6}.

Newton-Raphsonov postupak

Algoritam je objašnjen na predavanjima i u skripti na stranici 4-60. Za funkcije koje će se koristiti u zadacima s ovim postupkom implementirajte analitičko računanje gradijenta te Hesseove matrice. Postupak treba podržavati dva načina rada: u prvom načinu postupak se uvijek pomiče u novu točku za čitav iznos dobivenog pomaka, dok u drugom načinu postupak korištenjem metode zlatnog reza pronalazi optimalan iznos pomaka. Postupak je potrebno zaustaviti kada euklidska norma pomaka postane manja od neke zadane preciznosti. Potrebno je omogućiti da se bez prevođenja programa mogu definirati preciznost *e*, početna točka pretraživanja te hoće li postupak koristiti metodu zlatnog reza za određivanje optimalnog iznosa pomaka ili ne. Pretpostavljena vrijednost preciznosti je 10^{-6}. (**napomena**: *iskoristite LUP dekompoziciju iz prve domaće zadaće za rješavanje linearnog sustava koji se javlja u postupku*)

Gauss-Newtonov postupak

Algoritam je objašnjen na predavanjima i dostupan je na web stranici predmeta: https://www.fer.unizg.hr/_download/repository/Algoritam_Gauss_Newton.pdf. Za nelinearne sustave koje će se koristiti u zadacima s ovim postupkom implementirajte analitičko računanje Jacobijeve matrice. Postupak treba podržavati dva načina rada: u prvom načinu postupak se uvijek pomiče u novu točku za čitav iznos dobivenog pomaka, dok u drugom načinu postupak korištenjem metode zlatnog reza pronalazi optimalan iznos pomaka. Potrebno je omogućiti da se bez prevođenja programa mogu definirati preciznost *e*, početna točka pretraživanja te hoće li postupak koristiti metodu zlatnog reza za određivanje optimalnog iznosa pomaka ili ne. Pretpostavljena vrijednost preciznosti je 10^{-6}. (**napomena**: *iskoristite LUP dekompoziciju iz prve domaće zadaće za rješavanje linearnog sustava koji se javlja u postupku*)

Napomena: potrebno je osigurati da postupci ne zapnu u beskonačnoj petlji uslijed divergencije, već je u tom slučaju potrebno dojaviti prikladnu poruku o pogrešci. Primjerice, to možete napraviti na način da divergencijom smatrate slučaj kada u 10 uzastopnih iteracija nema poboljšanja u najboljoj dosegnutoj vrijednosti funkcije cilja (ili nadomjesne funkcije cilja za postupak Gauss-Newton) .

Svi algoritmi moraju podržavati proizvoljno veliku dimenzionalnost rješenja. *Funkciju cilja* potrebno je implementirati tako da vodi evidenciju o broju pozivanja, jer će algoritme biti potrebno usporediti na temelju broja evaluacija (preporučuje se ostvariti kao apstraktni razred). Za postupke koji koriste gradijente potrebno je voditi evidenciju o broju računanja gradijenta i Hesseove matrice.

Za svaki postupak na kraju ispišite točku minimuma koju je postupak pronašao te vrijednost funkcije cilja u toj točki. Također, za svaki postupak ispišite broj poziva funkcije cilja te broj poziva računanja gradijenta i Hesseove matrice, ako se koriste u postupku.

Prije dolaska na laboratorijsku vježbu pripremite svaki od zadataka u nastavku u obliku programa ili funkcije koju možete pokrenuti, tako da se svaki od zadataka može demonstrirati bez prevelikih promjena u programu. Alternativno, možete koristiti konfiguracijske datoteke kojima specificirate sve bitne parametre, tako da se program može izvršavati za različite postupke bez potrebe za ponovnih prevođenjem.

Funkcije cilja

- $f_1(\boldsymbol{x}) = 100 \cdot (x_2 - {x_1}^2)^2 + (1 - x_1)^2$ (*Rosenbrockova 'banana' funkcija*)
  Početna točka: $\mathbf{x}_0 = (-1.9, 2)$, minimum: $\mathbf{x}_{\min} = (1,1)$, $f_{\min} = 0$
- $f_2(\boldsymbol{x}) = (x_1 - 4)^2 + 4 \cdot (x_2 - 2)^2$
  Početna točka: $\mathbf{x}_0 = (0.1, 0.3)$, minimum: $\mathbf{x}_{\min} = (4, 2)$, $f_{\min} = 0$
- $f_3(\boldsymbol{x}) = (x_1 - 2)^2 + (x_2 + 3)^2$
  Početna točka: $\mathbf{x}_0 = \mathbf{0}$ (nul vektor), minimum: $\mathbf{x}_{\min} = (2, -3)$, $f_{\min} = 0$
- $f_4(\boldsymbol{x}) = \frac{1}{4}x_1^4 - x_1^2 + 2x_1 + (x_2 - 1)^2$
  minimum: $\boldsymbol{x}_{min} = (-1.76929, 1)$, $f_{min} = -4.219136$

Laboratorijska vježba

1. Primijenite postupak gradijentnog spusta na funkciju 3, uz i bez određivanja optimalnog iznosa koraka. Što možete zaključiti iz rezultata?
2. Primijenite postupak gradijentnog spusta i Newton-Raphsonov postupak na funkcije 1 i 2 s određivanjem optimalnog iznosa koraka. Kako se Newton-Raphsonov postupak ponaša na ovim funkcijama? Ispišite broj izračuna funkcije, gradijenta i Hesseove matrice.
3. Primijenite postupak Newton-Raphson bez traženja minimuma na pravcu na funkciju 4 uz početne točke (3,3) te (1,2). Što uočavate? Nakon toga primijenite istu postupak uz traženje minimuma na pravcu i komentirajte rezultate.
4. Primijenite postupak Gauss-Newton na funkciju 1 tako da problem definirate kao sustav nelinearnih jednadžbi (dvije jednadžbe za dvije komponente zadane funkcije cilja). Za početnu točku uzmite $\mathbf{x}_0 = (-1.9, 2)$.
5. Primijenite postupak Gauss-Newton na sljedeći sustav nelinearnih jednadžbi:
   $x_1^2 + x_2^2 - 1 = 0;\ (x_2 - x_1^2) = 0.$
   Postupak primijenite uz početne točke: (-2, 2), (2, 2) te (2, -2). Kakve rezultate daje postupak za sve početne točke? Uputa: prikažite sustav jednadžbi u 2D i nadomjesnu funkciju cilja u 3D! Sustav ima dva globalna minimuma u [+/-0.78615, 0.618] te jedan lokalni minimum u [0, -0.70711].
6. Potrebno je optimirati parametre zadanog modela koji opisuje niz mjerenja promatrane veličine u vremenu. Mjerenja su dana u obliku niza uređenih parova (t_i, y_i), gdje je t_i vremenska oznaka, a y_i promatrana vrijednost: (1,3), (2,4), (3,4), (5,5), (6,6), (7,8). Model kojim se opisuju podaci je $M(x,t) = (x_1 \exp(x_2 t) + x_3)$ i ima 3 parametra (x_1, x_2, x_3) koje treba optimirati. Primijenite postupak Gauss-Newton tako da definirate sustav nelinearnih jednadžbi sljedećeg oblika: $x_1 \exp(x_2 t_i) + x_3 - y_i = 0$, za svaki uređeni par i. Kao početno rješenje uzmite vrijednost 1 za svaki parametar modela. Uputa: prikažite (u nekom programu) zadane točke i rezultat modela u (t, y) ravnini.

Demonstracija funkcionalnosti u MATLAB-u

Ovaj dio vježbe izvodi se na predavanjima.

Potrebno je odabrati dvije optimizacijske funkcije, definirati ih kao funkcije u MATLABU i pronaći njihov minimum (bez ograničenja) uporabom ugrađenih MATLABovih funkcija. Neke od funkcija za optimiranje su sljedeće:

- `fminbnd`: minimum funkcije jedne varijable; koristi algoritam zlatnog reza i kvadratne interpolacije
- `fminsearch`: minimum funkcije više varijabli; koristi simplex postupak po Nelderu i Meadu
- `fminunc`: minimum funkcije više varijabli; ova funkcija može, ovisno o proslijeđenim parametrima, uporabiti nekoliko optimizacijskih algoritama, između ostaloga i postupak po Fletcheru i Powellu, poopćeni Newtonov postupak, metodu najbržeg spusta itd.
- `fmincon`: minimum funkcije više varijabli uz ograničenja

Funkcije se u MATLAB-u mogu definirati u posebnim .m datotekama, kao u sljedećem primjeru:

```
function f = apr_primjer(x)  % mora biti u prvom retku!
% Bezvezna funkcija dvije varijable
f = x(1) + x(2);
```

Datoteku je potrebno nazvati istim imenom kao i funkcija (apr_primjer.m). Iz MATLAB-a se funkciji tada može pristupiti s `apr_primjer([1,1])`.

Odabrana poglavlja iz MATLAB Helpa:

- DEMOS: Optimization: Minimization of the "banana function"
- MATLAB: Mathematics: Function Functions: Minimizing Functions and Finding Zeros
- Optimization Toolbox: Standard Algorithms: Unconstrained Optimization