

Datorövning 4 - Optimering

avsnitt 2.3–2.4 i 'Flervariabelanalys med MATLAB'

Information om genomförande och krav, vad det gäller datorövningarna, finns i PM för Datorövning 1 och på kurshemsidan.

Obligatoriska uppgifter

I uppgift 1–4 nedan är $f(x, y) = (x^2 + y)e^{-x^2 - y^2} + xy/10$ och $\Omega = \{(x, y) \in \mathbb{R}^2 : g(x, y) \leq 0\}$, där $g(x, y) = x^2 + y^2 - 4$.

Uppgift 1: (se avsnitt 1.2 & 2.2 i kompendiet 'Flervariabelanalys med MATLAB')

Beräkna för hand de partiella derivatorna $f_1(x, y)$ och $f_2(x, y)$ och plotta nivåkurvorna $f_1(x, y) = 0$, $f_2(x, y) = 0$ och $g(x, y) = 0$, med olika färg i samma figur. Avgör (ur figuren) hur många stationära punkter till $f(x, y)$ det finns i Ω och använd `fsolve` för att närmare bestämma dessa.

Uppgift 2: (se avsnitt 1.1, 2.3 & 2.4 i kompendiet 'Flervariabelanalys med MATLAB')

Plotta grafen till $f(x, y)$ på området Ω (se tips nedan). Ur figuren framgår det bl.a. att funktionen har ett lokalt maximum i närheten av punkten $(0.7, 0.5)$ och ett lokalt minimum i närheten av $(0, -0.7)$. Utför några steg i gradientmetoden, med $(0.7, 0.5)$ resp. $(0, -0.7)$ som startapproximation, för att bättre lokalisera dessa extrempunkter. Använd också `fminunc` för att bestämma dessa extrempunkter. Undersök detaljer kring varje steg i iterationen som `fminunc` utför. Jämför slutligen resultaten med varandra och med vad du fick i uppgift 1.

Tips: Om vi i plotten inte vill ha med den del av ytan där $g(x, y) > 0$ så kan vi byta ut motsvarande element i matrisen $Z=f(X,Y)$ mot "värdet" `NaN` (Not a Number) innan vi plottar med t.ex. `surf(X,Y,Z)`. Detta kan göras med kommandot `Z(find(g(X,Y)>0))=NaN`.

Eftersom området i detta fall är en cirkelskiva kan vi alternativt använda oss av polära koordinater för att skapa plotten, då blir kanten lite snyggare. Gör i så fall en mesh i de polära koordinaterna `R` och `THETA` och använd dem för att skapa `X` och `Y`. Beräkna sedan funktionsvärdena `Z` och plotta ytan med `surf` på vanligt sätt.

Kryssuppgifter

Uppgift 3: (se avsnitt 2.4 i kompendiet 'Flervariabelanalys med MATLAB')

Plotta skärningskurvan mellan funktionsytan $z = f(x, y)$ och cylindern $x^2 + y^2 = 4$ (betraktad som en ekvation i x, y, z). Försök avgöra ur figuren hur många lokala extrempunkter $f(x, y)$ har, under bivillkoret $g(x, y) = 0$, och använd `fmincon` för att bestämma dessa extrempunkter. Bestäm även extrempunkterna genom att parametrisera randen till området Ω och använd `fminbnd`.

Uppgift 4: (se avsnitt 2.3 & 2.4 i kompendiet 'Flervariabelanalys med MATLAB')

Plotta nivåkurvorna $f_1(x, y) = 0$ och $f_2(x, y) = 0$ med olika färg i samma figur. Som i uppgift 1 ger figuren information om var de stationära punkterna finns. Plotta därefter ca. 50 olika nivåkurvor till $f(x, y)$ i samma figur och försök avgöra de stationära punkternas karaktär (dvs. om de är max, min eller sadelpunkter). Undersök sedan egenvärdena till Hessianen i de stationära punkterna (använd approximationer från uppgift 1) och se om det bekräftar dina antaganden om punkternas karaktär.

Tips: Om du skapat anonyma funktioner för de partiella derivatorna $f_1(x, y)$ och $f_2(x, y)$ så kan du beräkna Hessianen approximativt m.h.a. differenskvoter. T.ex. är;

$$f_{12}(x, y) \approx \frac{f_1(x, y + h) - f_1(x, y - h)}{2h}$$

Egenvärdena till en matris kan beräknas med `eig`.

Uppgift 5: Skriv ett program `gradmet.m` med anropet `x=gradmet(f,x0,tol)` som använder gradientmetoden för att hitta maximum av en funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ då startapproximationen är $\mathbf{x}_0$. Programmet skall utföra iterationen med en `while`-snurra som avbryts när avståndet mellan två successiva punkter (approximationer) är mindre än `tol`. Programmet skall inte plotta något utan bara beräkna maximipunkter. Testa sedan ditt program på t.ex. funktionen $f(x, y)$ i uppgift 1–4 ovan. Prova lite olika startpunkter, både nära och långt ifrån något lokalt maximum. Om du vill kan du göra en liten snurra med `ginput` så går det lite smidigare att testa olika startpunkter.

Uppgift 6: Skriv ett program `hessian.m` med anropet `H=hessian(f,x)` som beräknar Hessianen till en funktionen $f : \mathbb{R}^n \rightarrow \mathbb{R}$ i en punkt $\mathbf{x}$ (använd differenskvoter). Testa sedan ditt program genom att beräkna Hessianen till någon funktion i dess kritiska punkter (välj t.ex. funktion från någon rekommenderad uppgift i kap 13).