Matlabövning 2
Flervariabelanalys för E2 Ht 2007

Matematiska Vetenskaper
Carl-Henrik Fant

Gradienter

I den första deluppgiften skall du beräkna gradienter och se att dessa är ortogonala mot nivåkurvorna.

Istället för att bestämma uttryck för de partiella derivatorna skall du låta Matlab beräkna närmevärden med hjälp av differenskvoter. Vi har ju att

$$f_1(a,b) = \lim_{h\to 0} \frac{f(a+h,b)-f(a,b)}{h} \approx \frac{f(a+h,b)-f(a,b)}{h}$$

om $|h|$ är tillräckligt litet. Vid datorberäkning kan man inte sätta in hur små värden på h som helst, avrundningsfelet vid beräkningen av $f(a+h,b)-f(a,b)$ blir då större än differensen och påverkar resultatet oerhört mycket. Med $|h| > 10^{-9}$ är det normalt inga problem, men för denna uppgift räcker det att låta $|h| = 10^{-6}$, vilket ger tillräcklig noggrannhet i derivataberäkningarna. Eftersom vi måste välja tecken på h får vi i princip antingen höger- eller vänsterderivata. Som kompromiss kan vi ta medelvärdet av de två och sätta

$$f_1(a,b) \approx \frac{1}{2}\left(\frac{f(a+h,b)-f(a,b)}{h} + \frac{f(a-h,b)-f(a,b)}{-h}\right) = \frac{f(a+h,b)-f(a-h,b)}{2h}$$

där alltså $h = 10^{-6}$.

Uppgift 1: Definiera en anonym funktion $f(x,y) = (7x+15y)e^{-(0.01x^2+0.02y^2)}$ och rita grafen till denna över en kvadrat $-10 \le x \le 10,\ -10 \le y \le 10$. Du bör se att funktionen har ett lokalt maximum och ett lokalt minimum i området.

Definiera också anonyma funktioner $f1(x,y)$ och $f2(x,y)$, som är de approximationer till partiella derivatorna $f_1(x,y)$ och $f_2(x,y)$ som beskrivs ovan. Enklast är att $f1$ och $f2$ anropar funktionen f så att du inte skriver in uttrycket för f gång på gång. Rita graferna till $f1$ och $f2$ över samma kvadrat som ovan. Med kommandot `quiver` kan du rita gradienter som vektorer i xy-planet. Här är det lämpligt att minska antalet punkter i rutnätet som ges av `meshgrid`. Rita nivåkurvor till f i samma figur, för dessa är det lämpligt med ganska många punkter i rutnätet. Kommandot `axis('square')` ger korrekta vinklar. □

Gradientmetoden för att hitta lokala min/max.

Funktionen $f(x,y) = (7x+15y)e^{-(0.01x^2+0.02y^2)}$ har som du sett en maxpunkt och en minpunkt i kvadraten ovan. Du har förhoppningsvis också sett att gradienten är en vektor som pekar i den riktning funktionsvärdena ökar mest. Kraftigare ökning betyder längre gradientvektor.

Detta motiverar att vi, då vi söker en funktions lokala maximum, kan använda
gradientmetoden som innebär:

- Välj en startpunkt (x_1, y_1). Troligen har vi inte råkat välja en punkt där $f(x,y)$ har lokalt maximum.
  Beräkna gradienten i denna punkt.
- Sätt $(x_2, y_2) = (x_1, y_1) + r \cdot \operatorname{grad} f(x_1, y_1)$ där r är ett lämpligt tal. Troligen kommer $f(x_2, y_2)$ att vara större än $f(x_1, y_1)$ om inte r är för stort. Faktorn r kan behövas eftersom funktionen kan växa väldigt brant. Med $r = 1$ kan man missa maxpunkten helt och i värsta fall hoppa fram och tillbaka. Tyvärr kan man inte veta i förväg vad r bör vara.

- Beräkna differensen $f(x_2, y_2) - f(x_1, y_1)$, som skall vara positiv annars är r för stort.
- Låt (x_2, y_2) vara ny startpunktoch upprepa tills ökningen i funktionsvärde är tillräckligt liten.

Söker vi minimum så skall vi istället gå i riktningen $-\operatorname{grad} f(x_1, y_1)$.

Uppgift 2: Din uppgift är nu producera en skriptfil där du valt en startpunkt (x_1, y_1) som ligger en bit ifrån maxpunkten till funktionen i uppgift 1. Då man kör skriptfilen skall först funktionsytan ritas. Sedan skall:

- en röd $*$ i ritas i punkten $(x_1, y_1, f(x_1, y_1))$. (Använd `plot3`-kommandot.)
- en ny punkt (x_2, y_2) beräknas med gradientmetoden för maximumberäkning.
- en röd $*$ ritas i den nya punkten och de två punkterna förbindas med en sträcka.
- de två föregående stegen upprepas till ökningen är mindre än ett valt tal, varefter extrempunkten anges på skärmen.

För att få ny plot efter varje ny punkt kan man sätta in `pause` efter `plot3`-kommandot. Då stegar man fram i iterationen genom att trycka på valfri tangent.

Då du producerar programmet kan det vara en god idé att pröva själva iterationssteget innan du lägger in det i en `while`-snurra. Du får antagligen pröva med olika värden på r innan det steget fungerar. Du kan låta programmet pröva sig fram till ett bra r-värde i varje iterationssteg.

När du ser att programmet fungerar skall du lägga in en `while`-snurra som stoppar då ökningen är mindre än ett givet tal, tex 10^{-9}. Pröva i början med större tal här.

Du kan om du vill lägga in funktionen f i en funktionsfil och göra en funktionsfil av programmet du skrivit. Den funktionsfilen bör ha funktionen (f), startpunkten, noggrannheten i funktionsvärde och eventuellt r som invariabler och maxpunkten $(x_2, y_2, f(x_2, y_2))$ som utvariabel.

På kursens webbsida finns en skriptfil som du kan titta på och modifiera om du tycker uppgiften är för svår.

Redovisning: Visa upp att ditt program fungerar. □

Newtons metod för ekvationssystem.

Grundidén i Newtons metod för numerisk lösning av ett ekvationssystem

$$\begin{cases} f(x, y) = 0 \\ g(x, y) = 0 \end{cases}$$

är att man kan ta en tämligen godtycklig punkt (x_1, y_1) som startpunkt. Man bestämmer ekvationer för tangentplanen till de två funktionsytorna $z = f(x, y)$ och $z = g(x, y)$. Dessa två plan skär varandra och xy-planet i en punkt (x_2, y_2). Denna punkt ligger närmare en punkt (x_0, y_0) som är lösning till ekvationssystemet. Man upprepar denna procedur tills båda funktionsvärdena är tillräckligt små.

I kursboken finns en formel för beräkning av de nya punkterna. Denna formel kommer från lösningen till det ekvationssystem som man får genom att sätta $z = 0$ i de två tangentplanens ekvationer:

$$\begin{cases} f(x_1, y_1) + f_1(x_1, y_1)(x - x_1) + f_2(x_1, y_1)(y - y_1) = 0 \\ g(x_1, y_1) + g_1(x_1, y_1)(x - x_1) + g_2(x_1, y_1)(y - y_1) = 0 \end{cases}$$

eller på matrisform

$$\begin{bmatrix} f_1(x_1, y_1) & f_2(x_1, y_1) \\ g_1(x_1, y_1) & g_2(x_1, y_1) \end{bmatrix} \begin{bmatrix} x - x_1 \\ y - y_1 \end{bmatrix} = \begin{bmatrix} -f(x_1, y_1) \\ -g(x_1, y_1) \end{bmatrix}$$

Naturligtvis kan man ge lösningen med Cramers regel (som i boken). Men man kan också bilda systemets utökade matris

$$C = \begin{bmatrix} f_1(x_1, y_1) & f_2(x_1, y_1) & -f(x_1, y_1) \\ g_1(x_1, y_1) & g_2(x_1, y_1) & -g(x_1, y_1) \end{bmatrix}$$

Kommandot `A=rref(C)` ger en matris $A = \begin{bmatrix} 1 & 0 & u \\ 0 & 1 & v \end{bmatrix}$ vars tredje kolonn är lösningen till

$$\begin{bmatrix} f_1(x_1, y_1) & f_2(x_1, y_1) \\ g_1(x_1, y_1) & g_2(x_1, y_1) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f(x_1, y_1) \\ -g(x_1, y_1) \end{bmatrix}$$

Kommandot `z = [x_1;y_1] + A(:,3)` ger slutligen den sökta punkten $z(1) = x_2$, $z(2) = y_2$.
Uppgift 3: Lös uppgift 13.6.5 i kursboken. Definiera anonyma funktioner som i gradientmetoden och välj ett lämpligt sätt att lösa matrisekvationen ovan. Skriv en skriptfil som itererar fram en lösning där både $|f(x, y)|$ och $|g(x, y)|$ är mindre än 10^{-6}. Redovisning: ett fungerande program. □