

Linjärisering och Jacobimatris

1 Inledning

Redan i envariabelkursen såg vi på linjärisering av en differentierbar (deriverbar) funktion i en variabel $f : \mathbb{R} \rightarrow \mathbb{R}$. Linjäriseringen av f runt punkten a ges av (Definition 8, Adams 4.9)

$$L(x) = f(a) + f'(a)(x - a).$$

Nära punkten a har vi $f(x) \approx L(x)$ och vi påminner oss om att den räta linjen $y = L(x)$ är tangenten till kurvan $y = f(x)$ vid a .

Newtons metod för lösning av ekvationer $f(x) = 0$ bygger på upprepad linjärisering. Vi skall se på linjärisering av funktioner i flera variabler, både reellvärda och vektorvärda. I nästa studioövning skall vi använda detta för att se hur vi kan generalisera Newtons metod så att vi kan lösa system av icke-linjära ekvationer $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. Men först en liten uppgift.

Uppgift 1. Linjärisera $f(x) = 0.5(x - 2)^2 - 2 \cos(2x) - 1.5$ runt $a = -1$, $a = 2.5$ och $a = 2.8$. Rita upp funktionskurvan tillsammans med tangenterna i $(a, f(a))$ för de olika a -värdena.

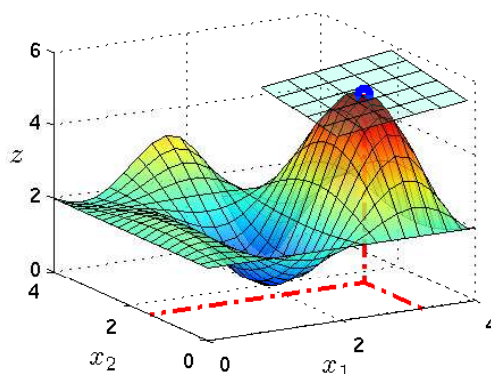
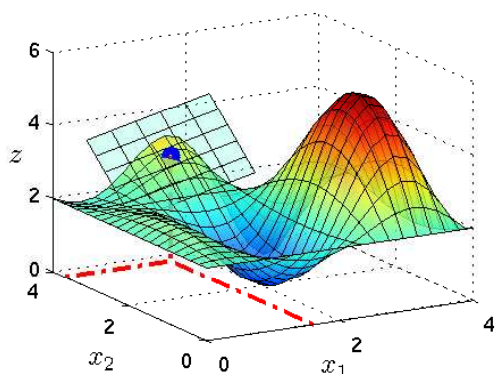
2 En funktion i två variabler, $f : \mathbb{R}^2 \rightarrow \mathbb{R}$

Låt $f(x_1, x_2)$ vara en differentierbar funktion i två variabler, $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. Linjäriseringen av f runt punkten (a_1, a_2) ges av (Adams 12.6)

$$L(x_1, x_2) = f(a_1, a_2) + \frac{\partial f}{\partial x_1}(a_1, a_2)(x_1 - a_1) + \frac{\partial f}{\partial x_2}(a_1, a_2)(x_2 - a_2)$$

och nära punkten (a_1, a_2) har vi $f(x_1, x_2) \approx L(x_1, x_2)$.

Det räta planet $z = L(x_1, x_2)$ är tangentplanet till ytan $z = f(x_1, x_2)$ ovanför punkten (a_1, a_2) .



Vi skall se hur man ritat upp tangentplan, men först skall vi skriva om linjäriseringen med vektorbeteckningar.

Vi låter $\mathbf{x} = (x_1, x_2)$, $f(\mathbf{x}) = f(x_1, x_2)$ och $\mathbf{a} = (a_1, a_2)$. Använder vi gradienten

$$\nabla f(\mathbf{x}) = \begin{bmatrix} f'_{x_1}(\mathbf{x}) \\ f'_{x_2}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} f'_{x_1}(x_1, x_2) \\ f'_{x_2}(x_1, x_2) \end{bmatrix}$$

kan vi skriva linjäriseringen

$$L(\mathbf{x}) = f(\mathbf{a}) + \nabla f(\mathbf{a})^T(\mathbf{x} - \mathbf{a}).$$

Lägg märke till att $\nabla f(\mathbf{a})$ och $\mathbf{x} - \mathbf{a}$ är kolonnvektorer så $\nabla f(\mathbf{a})^T(\mathbf{x} - \mathbf{a})$ är skalärprodukten $\nabla f(\mathbf{a}) \cdot (\mathbf{x} - \mathbf{a})$.

Låt oss som exempel ta $f(\mathbf{x}) = \frac{1}{2}x_1^2(1 + x_2^2) - x_1 + 1$. Då gäller

$$\frac{\partial f}{\partial x_1}(\mathbf{x}) = x_1(1 + x_2^2) - 1, \quad \frac{\partial f}{\partial x_2}(\mathbf{x}) = x_1^2 x_2,$$

och därmed $\nabla f(\mathbf{x})^T = [x_1(1 + x_2^2) - 1 \quad x_1^2 x_2]$. Linjäriseringen vid t.ex. $\mathbf{a} = (2, 1)$ blir

$$L(\mathbf{x}) = f(\mathbf{a}) + \nabla f(\mathbf{a})^T(\mathbf{x} - \mathbf{a}) = 3 + [3 \quad 4] \begin{bmatrix} x_1 - 2 \\ x_2 - 1 \end{bmatrix} = 3 + 3(x_1 - 2) + 4(x_2 - 1).$$

Nu skall vi beskriva linjäriseringen i MATLAB och rita upp tangentplan. Samtidigt passar vi på att rita normalen. Vi tar punkten $\mathbf{a} = (0.8, 0.1)$.

Vi beskriver funktionen och dess derivator med

```
>> f=@(x,y)0.5*x.^2.*(1+y.^2)-x+1; % Enklare skriva x,y istället för x1,x2
>> dfdx=@(x,y)x.*(1+y.^2)-1; dfdy=@(x,y)x.^2.*y;
```

Sedan bildar vi linjärisering och normalvektor med funktionerna

```
>> L=@(x,y,a,b)f(a,b)+dfdx(a,b)*(x-a)+dfdy(a,b)*(y-b);
>> n=@(x,y)[dfdx(x,y);dfdy(x,y);-1];
```

Vi anger tangeringspunkt och beräknar normalvektor samt anger området vi skall rita över.

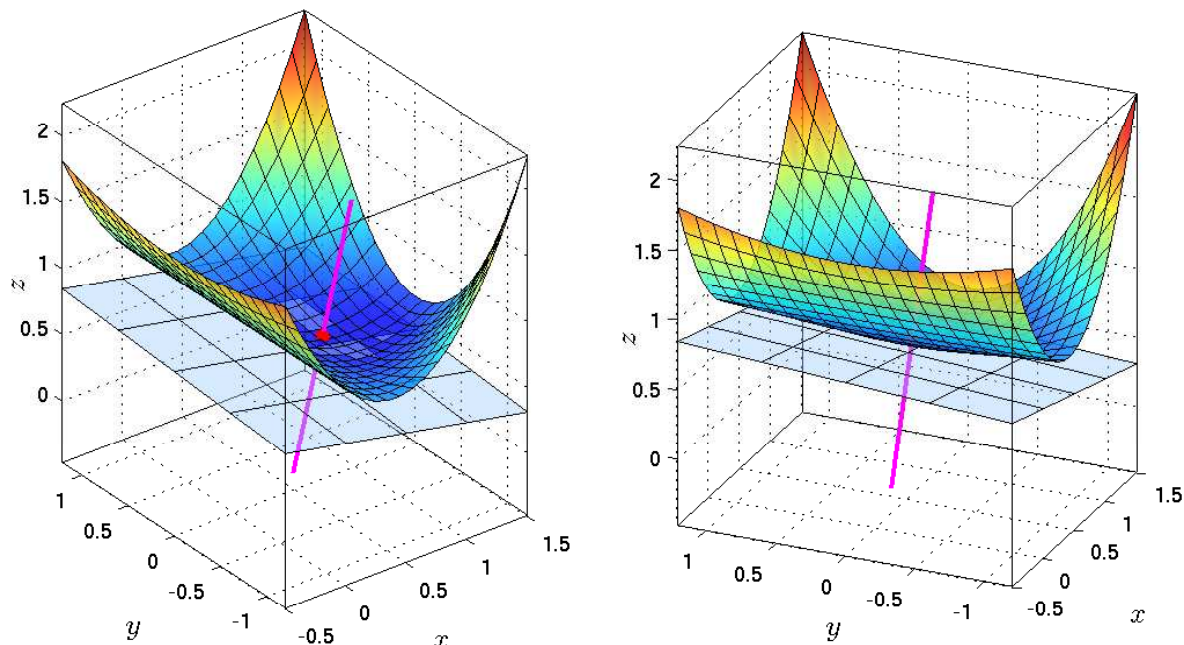
```
>> a=0.8; b=0.1; % Enklare skriva a,b istället för a1,a2
>> p0=[a;b;f(a,b)]; n0=n(a,b); % Tangeringspunkt och normalvektor
>> x=linspace(-0.5,1.5,20); y=linspace(-1.2,1.2,20);
>> [X,Y]=meshgrid(x,y); % Koordinatmatriser för området
```

Ritar funktionsyta, tangentplan samt normal med

```
>> Z=f(X,Y); T=L(X,Y,a,b);
>> surf(x,y,Z) % Funktionsytan
>> hold on
>> surf(x,y,T,'FaceColor','b','FaceAlpha',0.4) % Tangentplanet
>> s=[-1 1]; % För att rita normalen
>> plot3(p0(1)+s*n0(1),p0(2)+s*n0(2),p0(3)+s*n0(3),'m','linewidth',2)
>> hold off
>> xlabel('x'), ylabel('y'), zlabel('z'), box on
>> axis equal, axis vis3d, rotate3d on
```

Vi använder `axis vis3d` så att skalan inte förändras då vi vrider och vänder på grafen, med `rotate3d on` blir det möjligt att ta tag i grafen och vrida den.

Här ser vi ytan med tangentplanet och normalen. Vi tittar från två olika håll.



När det gäller att rita funktionsytor och tangentplan får vi kanske gå tillbaka till studioövning 1 och repetera. Lägg märke till att vi ritar upp tangentplanet som grafen till linjäriseringen, dvs. vi ritar tangentplanet på samma sätt som vi ritar funktionsytan.

När vi ritar normalen ritar vi den som en rät linje från en punkt i rummet till en annan. Vi använder `plot3` och då måste koordinater för start- och slutpunkt separeras i tre vektorer för x -, y - respektive z -koordinaterna. Eftersom $\mathbf{s}=[-1 \ 1]$ är en vektor (se koden på sidan 3) så ger $\mathbf{p0}+\mathbf{s}*\mathbf{n0}$ en vektor (i MATLAB) motsvarande en linje med start i $\mathbf{a} - \mathbf{n}$ och slut i $\mathbf{a} + \mathbf{n}$. Vi separerar koordinaterna

```
x:      p0(1)+s*n0(1)
y:      p0(2)+s*n0(2)
z:      p0(3)+s*n0(3)
```

och ritar linjen med

```
>> plot3(p0(1)+s*n0(1),p0(2)+s*n0(2),p0(3)+s*n0(3),'m','linewidth',2)
```

Vi ser en bit av normalen magentafärgad.

Uppgift 2. Linjärisera $f(\mathbf{x}) = x_1 \cos(2x_1) \sin(x_2)$ runt punkten $\mathbf{a} = (3, 1)$. Rita upp funktionsytan tillsammans med tangentplanet i \mathbf{a} . Även tangeringspunkten skall ritas ut.

3 Två funktioner i två variabler, $\mathbf{f}: \mathbb{R}^2 \rightarrow \mathbb{R}^2$

Låt $f_1(x_1, x_2)$ och $f_2(x_1, x_2)$ vara två differentierbara funktioner i två variabler, $f_1: \mathbb{R}^2 \rightarrow \mathbb{R}$ och $f_2: \mathbb{R}^2 \rightarrow \mathbb{R}$. Linjäriseringen av f_1 respektive f_2 runt punkten (a_1, a_2) ges av

$$\begin{aligned}L_1(x_1, x_2) &= f_1(a_1, a_2) + \frac{\partial f_1}{\partial x_1}(a_1, a_2)(x_1 - a_1) + \frac{\partial f_1}{\partial x_2}(a_1, a_2)(x_2 - a_2), \\L_2(x_1, x_2) &= f_2(a_1, a_2) + \frac{\partial f_2}{\partial x_1}(a_1, a_2)(x_1 - a_1) + \frac{\partial f_2}{\partial x_2}(a_1, a_2)(x_2 - a_2),\end{aligned}$$

Låter vi $\mathbf{x} = (x_1, x_2)$, $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$ och $\mathbf{a} = (a_1, a_2)$ kan vi skriva

$$\begin{aligned}L_1(\mathbf{x}) &= f_1(\mathbf{a}) + \nabla f_1(\mathbf{a})^T(\mathbf{x} - \mathbf{a}), \\L_2(\mathbf{x}) &= f_2(\mathbf{a}) + \nabla f_2(\mathbf{a})^T(\mathbf{x} - \mathbf{a}),\end{aligned}$$

eller med matrisbeteckningar

$$\mathbf{L}(\mathbf{x}) = \mathbf{f}(\mathbf{a}) + \mathbf{Df}(\mathbf{a})(\mathbf{x} - \mathbf{a}),$$

där $\mathbf{L}(\mathbf{x}) = (L_1(\mathbf{x}), L_2(\mathbf{x}))$ och

$$\mathbf{Df}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) \end{bmatrix},$$

är Jacobimatrisen.

Låt oss som exempel ta $\mathbf{f}(\mathbf{x}) = (x_1(1 + x_2^2) - 1, x_2(1 + x_1^2) - 2)$. Då gäller

$$\begin{aligned}\frac{\partial f_1}{\partial x_1}(\mathbf{x}) &= 1 + x_2^2, & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) &= 2x_1x_2, \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) &= 2x_1x_2, & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) &= 1 + x_1^2,\end{aligned}$$

och därmed

$$\mathbf{Df}(\mathbf{x}) = \begin{bmatrix} 1 + x_2^2 & 2x_1x_2 \\ 2x_1x_2 & 1 + x_1^2 \end{bmatrix}.$$

Linjäriseringen vid $\mathbf{a} = (2, 1)$ blir

$$\mathbf{L}(\mathbf{x}) = \mathbf{f}(\mathbf{a}) + \mathbf{Df}(\mathbf{a})(\mathbf{x} - \mathbf{a}) = \begin{bmatrix} 3 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 & 4 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} x_1 - 2 \\ x_2 - 1 \end{bmatrix}.$$

Uppgift 3. Linjärisera $\mathbf{f}(\mathbf{x}) = (x_1^3 + x_2^2 - 1, e^{x_1x_2} + x_1 + x_2 - 2)$ runt $\mathbf{a} = (2, 1)$. Nu är det ingen idé att försöka rita. Räkna ut för hand, skriv ned på papper och skriv MATLAB-kod. Använd matris-beteckningar.

4 Flera funktioner i flera variabler, $\mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^m$

Vi generaliserar nu till godtyckligt antal funktioner i godtyckligt många variabler. Låt f_i beteckna m funktioner i n variabler, dvs. $\mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^m$. Vi låter

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_m(x_1, \dots, x_n) \end{bmatrix},$$

och

$$D\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial f_m}{\partial x_n}(\mathbf{x}) \end{bmatrix}.$$

Här är $m \times n$ matrisen $D\mathbf{f}(\mathbf{x})$ Jacobimatrisen av \mathbf{f} i \mathbf{x} . Linjäriseringen av \mathbf{f} i punkten \mathbf{a} blir

$$L(\mathbf{x}) = \mathbf{f}(\mathbf{a}) + D\mathbf{f}(\mathbf{a})(\mathbf{x} - \mathbf{a}).$$

Som ni ser så är det väldigt likt då $n = m = 2$, det gäller bara att hålla ordning på antalet rader och kolonner.

5 Newtons metod

En viktig användning av linjärisering är vid lösning av system av icke-linjära ekvationer $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, där $\mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^n$, med Newtons metod. Metoden består i att utgående från en startapproximation av en lösning approximera funktionen med linjäriseringar för att successivt få allt bättre approximationer.

I nästa studioövning skall vi titta på den här typen av problem och bekanta oss med Newtons metod.