Noen hint for å gjøre Matlabøving i TMA4105

Brynjulf Owren

17. mars 2010

1 Innledning

Våren 2010 forsøker vi å introdusere en moderat bruk av Matlab i øvingene i TMA4105 Matematikk 2. Spesielt er dette med tanke på at mange studenter har lært Matlab i IT grunnkurs, vi ønsker å utnytte kunnskaper ervervet i dette grunnkurset samt skape kontinuitet i bruk av Matlab, som mange vil få bruke for i de senere faser av kursene. Fordi opplegget foreløpig er under utprøving, så vil vi gjerne gjøre ting enkelt og basere oss på tidligere erfaringer med bruk av Maple. Derfor er Matlab-øvingen nesten en kopi av tidligere års Maple-øving nr 2.

Vi skal gå gjennom bruk av en del sentrale matlabfunksjoner, og gi eksempler på hvordan disse brukes.

2 Anbefalt generell framgangsmåte

En forskjell mellom Maple og Matlab er at førstnevnte bruker såkalte "work sheetssom man kan lagre og hente opp igjen seinere, man kan også editere på disse. I Matlab vil vi anbefale deg å bruke *skript-filer.* En skript-fil er kun en fil med utvidelse `.m` (akkurat som funksjonsfiler), for eksempel kan man kalle en fil for `minfil.m`. Filen opprettes gjerne fra Matlabs kommandovindu med `File->New->Script`, slik at man får opp et blankt dokument i Matlab sin Editor. Med Editoren kan man skrive inn en sekvens av Matlab-kommandoer, akkurat på samme måte som om man skrev dem i kommando-vinduet. Så lagrer man filen ved å bruke menyen `File->Save as` i Editor-vinduet. I kommandovinduet skriver man navnet på filen (uten `.m`) og kommandoene vil bli utført.

```
%
%  Eksempelfilen minfil.m
%
disp('Dette er en skriptfil')
```

Om man lager filen ovenfor, så kan man i kommandovinduet skrive `minfil`, resultatet ses nedenfor.

```
>> minfil
```

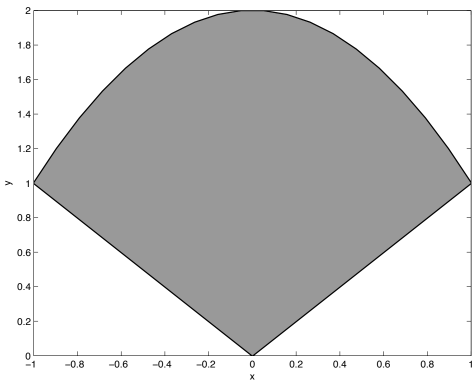
```
Dette er en skriptfil
>>
```

Vi foreslår at du lager en egen skriptfil for hvert delspørsmål der matlab skal brukes, dvs `Oppg1f.m`, `Oppg1g.m` osv.

3 Tegning av skraverte områder i planet `fill, plot`

Dette får du blant annet bruk for i Oppgave 1g). Om man skal tegne et lukket område i planet der randen er sammensatt av stykkevis glatte kurver, så fins det flere mulige framgangsmåter, vi skisserer et par av dem her. Som i Maple kan man også gjøre plotting av grafer til funksjoner med symboluttrykk, se for eksempel funksjonen `fplot`. Men vi skal gjøre det annerledes her. La oss som eksempel anta at randen er sammensatt av tre kurve-stykker C_1, C_2, og C_3.

Et eksempel er angitt i figuren, der vi har to rette linjerstykker og et krumt stykke som utgjør randen. I dette eksemplet er området avgrenset av $y = x$, $y = -x$ og $y = 2 - x^2$. Når man skal lage et plot i Matlab av en slik type, kan man gå fram ved å lage to lange vektorer X, Y slik at parene fra komponent i, $(X_i, Y_i)_{i=1}^n$ beskriver koordinatene til n punkter langs randen.



Vi kan sørge for at først og siste punkt er det samme, dvs $X_1 = X_n$ og $Y_1 = Y_n$ slik at kurven slutter der den starter. Man kan bruke kommandoen `plot(X,Y)` for å plotte randen rundt området. Man kan også bruke matlabfunksjonen `fill(X,Y,c)` for å skravere området med en farge `c`. En kan for eksempel bruke `fill(X,Y,'r')` for å få rød farge. Hvis man virkelig vil leke seg med farger, så kan man blandessine egne farger ved å la `c` være en vektor med tre tall mellom 0 og 1. For eksempel er figuren ovenfor laget med `fill(X,Y,[0.6,0.6,0.6]` (alle tall like gir gråtoner. Når randen består av mange mindre stykker, så kan det være lurt å generere delvektorer X_a, X_b, X_c, Y_a, Y_b, Y_c slik at f.eks. (X_a, Y_a) er koordinatene til del a av randen. Det må lages slik at siste element i (X_a, Y_a) beskriver nabopunktet til første koordinat av (X_b, Y_b) osv. Man skriver til slutt

```
X = [Xa, Xb, Xc];
Y = [Ya, Yb, Yc];
```

For å lage en vektor med n tall likt fordelt mellom grensene a og b kan man bruke funksjonen `linspace(a,b,n)`. For eksempel `linspace(0,1,6)` gir

```
>> linspace(0,1,6)
```

```
ans =

         0    0.2000    0.4000    0.6000    0.8000    1.0000
```

Her er skriptfilen brukt til å lage figuren ovenfor.

```
%
%
%
hold off
x = linspace(-1,1,20);   % Lag x-verdier mellom -1 og 1.
y = 2- x.^2;  % Lag tilhørende y-verdier
x = [0 x 0];  % Sett inn startpunkt 0 og sluttpunkt 0 i både x og y
y = [0 y 0];  % slik at både første og siste punkt blir origo.

plot (x,y,'k','LineWidth',2)
hold on
fill(x,y,[.6,.6,.6]);
xlabel('x'); ylabel('y');
```

4 Plotting av vektorfelt og romkurver

4.1 Vektorfelt og `quiver3`

Det fins to nyttige funksjoner `quiver` og `quiver3` i Matlab som plotter vektorfelt i henholdsvis planet og rommet som piler i et gitt sett av punkter. Vi beskriver `quiver3` her. Man kaller denne funksjonen med en av kommandoene

```
quiver3(X,Y,Z,U,V,W)
quiver3(X,Y,Z,U,V,W,S)
```

det siste kallet inkluderer en skaleringsfaktor for pilene. La oss ta et eksempel, vi forsøker å plotte vektorfeltet fra Eksempel 2 på side 861 i boka

$$\mathbf{F} = (y - x^2)\,\mathbf{i} + (z - y^2)\,\mathbf{j} + (x - z^2)\,\mathbf{k}$$

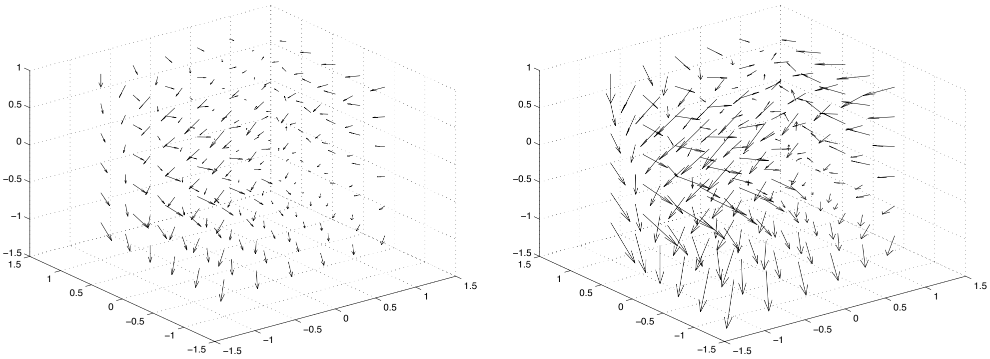
ved å bruke `quiver3`. Først bestemmer vi oss for et område å plotte på, vi ta kuben $-1 \le x \le 1$

gitter (rutenett) vi vil plotte piler på, la oss si vi bruker

```
%  Quiver test
%   Plotter vektorfeltet på side 861 (Eks 2) i boka

v1 = linspace(-1,1,6);
```

```
[X,Y,Z] = meshgrid(v1,v1,v1);

U = Y - X.^2;
V = Z - Y.^2;
W = X - Z.^2;

figure(1), hold off
quiver3(X,Y,Z,U,V,W,'k');
figure(2), hold off
quiver3(X,Y,Z,U,V,W,2,'k');
```

Resultatet med og uten skaleringsfaktor vises i figuren. Ikke det peneste plottet du har sett kanskje?

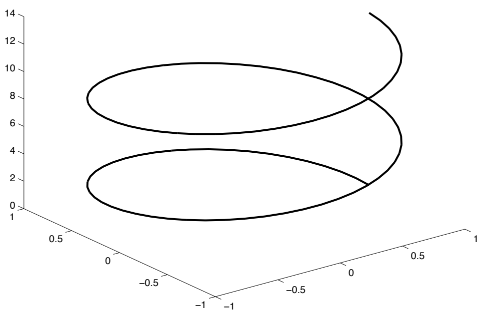


4.2 Romkurver og `plot3`

Plotting av romkurver er relativt enkelt, vi kan bruke funksjonen `plot3` og illustrerer den gjennom et eksempel, nemlig helixen

$$\mathbf{r}(t) = \cos t\,\mathbf{i} + \sin t\,\mathbf{j} + t\,\mathbf{k}$$

```
t=linspace(0,4*pi,100);  % bruk 100 parameterverdier mellom 0 og 4pi
plot3(cos(t),sin(t),t,'k','LineWidth',2)
```



5 Numerisk integrasjon i Matlab

Matlab har noen innebygde funksjoner for å finne integraler numerisk. Man kan noen ganger utføre slike integrasjoner enklere ved å bruke *Symbolic Math Toolbox* beskrevet i neste kapittel. Dette gjelder også numerisk beregnede intergraler. Først skal vi introdusere inline-funksjoner som er en nyttig snarvei for å lage enkle funksjoner.

5.1 Inline-funksjoner

I IT grunnkurs har du sikkert lært å lage en funksjon ved å opprette en fil, si `fun.m`, den kan for eksempel se slik ut

```
function c=pytagoras(a,b)
c=sqrt(a^2 + b^2)
```

Ofte kan funksjonene være lange og kompliserte, men hvis du trenger en funksjon av typen `pytagoras` ovenfor, så virker det tungvint å måtte lage en egen fil for dette. Da fins det flere muligheter, en er å bruke `inline`.

```
pytagoras = inline('sqrt(a^2+b^2)', 'a', 'b')
```

denne linjen kan man for eksempel legge inne i et skript og dermed unngå en ekstra funksjonsfil. Etter å ha utført dette, for eksempel i kommandovinduet, kan man skrive slikt som

```
>> pytagoras(3,4)

ans =

     5
```

En annen måte å definere samme funksjonen på er å skrive, f.eks. i skriptet

```
pytagoras = @(a,b) sqrt(a^2+b^2)
```

I begge tilfeller kan en tenke seg at `pytagoras` blir en peker (handle, håndtak) til en funksjon.

5.2 `quad`

Generelt skriver man `Q = quad(fun,a,b)` eller `Q = quad(fun,a,b,TOL)` for å integrere funksjonen definert i funksjonen `fun` mellom grenser `a` og `b`. Man kan også spesifisere en feiltoleranse `TOL`. Denne funksjonen er numerisk, og virker derfor ikke dersom for eksempel funksjonen man skal integrere avhenger av parametre. Vi tar et eksempel der vi kjenner svaret, nemlig

$$\int_0^1 \frac{4}{1+x^2}$$

Vi skriver da for eksempel

```
>> f = @(x) 4./(1+x.^2);
>> quad(f,0,1)

ans =

    3.1416
```

som ser ut til å stemme med eksakt svar π.

Det fins forøvrig mange flere matlabfunksjoner som beregner integraler av ulike typer med ulike metoder. Hvis du skriver for eksempel

```
>> help quad
```

så får du en liste nederst over slike funksjoner.

5.3 `quad2d`

Denne funksjonen kan brukes til å finne integraler av typen

$$\int_a^b \int_{c(x)}^{d(x)} f(x,y)\,\mathrm{d}y\,\mathrm{d}x$$

Man kaller funksjonen ved: `I=quad2d(fun,a,b,c,d)`. Her er

`fun`	Handle (peker, håndtak) til en funksjon som definerer $f(x,y)$
`a,b`	Nedre og øvre grense på ytterste integral
`c,d`	Pekere (handles) til funksjoner som definerer $c(x)$ og $d(x)$.

Eksempel. La oss beregne

$$\int_{-1}^{1} \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} (x^2+y^2)\,\mathrm{d}y\,\mathrm{d}x \tag{1}$$

Vi skriver følgende skript i Matlab

```
fun = @(x,y) x.^2 + y.^2;
a = -1;
b  = 1;
c=@(x) -sqrt(1-x.^2);
d=@(x)  sqrt(1-x.^2);

Q=quad2d(fun,a,b,c,d);
```

6 Symbolic Math Toolbox

Matlab er egentlig ment for numeriske beregninger, men det fins også en tilleggspakke (toolbox) som gir muligheter for å gjøre noen enkle symbolske utregninger. Denne toolboxen skal være tilgjengelig for studenter under NTNU sin lisensavtale. For å gjøre øvingen trenger du kun et lite utvalg av kommandoer. Hvis du skal bruke symbolske variable for eksempel i derivasjon eller integrasjon, må variablene først deklareres, og dette gjøres greit med `syms`. Hvis du for eksempel vil at x, y og z skal være symbolske variable, så kan du skrive

```
syms x y z
```

Etter dette er det mulig å gjøre en del enkle Maple-aktige kommandoer som for eksempel å derivere en funksjon `f` med hensyn på variablen `x` kan gjøres med `diff(f,x)`

```
>> f = exp(x)*sin(x)

f =

exp(x)*sin(x)

>> diff(f,x)

ans =

exp(x)*cos(x) + exp(x)*sin(x)
```

Integrere kan man gjøre med funksjonen `int`, skriver man `int(f,x)` betyr dette at man finner det ubestemte integralet av funksjonen `f` med hensyn på variablen `x`.

```
 >> f=log(x)

f =

log(x)

>> int(f,x)

ans =

x*(log(x) - 1)
```

Hvis man vil bruke grenser, så kan man spesifisere disse med ytterligere to argumenter til `int`, vi beregne det uegentlige integralet

$$\int_0^{\pi/2} \ln(\sin x)\,\mathrm{d}x,$$

ved å skrive

```
>> f = log(sin(x))

f =

log(sin(x))

>> Q=int(f,x,0,pi/2)

Q =

-(pi*log(2))/2
```

(Hadde du greid å løse dette for hånd?). Om vi vil finne ut hvordan dette svaret ser ut med desimaltall, kan vi bruke funksjonen `double`

```
>> double(Q)

ans =

   -1.0888
```

Ogsp hvis Matlab ikke klarer å beregne integralet eksakt, kan man gjøre det numerisk med `double`. Naturligvis er det mulig å bruke metoden med `quad`, `quad2d` etc beskrevet tidligere, men man kan også kombinere `int` med `double`. Som eksempel ser vi på integralet

$$\int_0^\pi \sqrt{1+\sin^2 x}\,\mathrm{d}x$$

Vi prøver med

```
 >> int( sqrt(1+sin(x)^2),0,pi)
Warning: Explicit integral could not be found.

ans =

int((sin(x)^2 + 1)^(1/2), x = 0..pi)

>> double(ans)

ans =

    3.8202
```

Vi ser at *int* rapporterer `Warning: Explicit integral could not be found`, og leverer problemet tilbake uløst. Men med `double(ans)` kobler vi inn en numerisk rutine som gir oss svaret. Denne måte å gjøre det på kan ofte være enklere for oss enn de numeriske rutinene.

Det er ingenting i veien for at vi kan nøste `int`-funksjonen og dermed regne ut doble og triple integraler. For eksempel integralet (1) kan bestemmes eksakt ved å skrive

```
>> int(int(x^2+y^2,y,-sqrt(1-x^2),sqrt(1-x^2)),x,-1,1)

ans =

pi/2
```

La oss også ta et trippelintegral, for eksempel løse et integral som dukker opp i en tidligere eksamensoppgave, nemlig volumberegningen

$$\int_{-1}^{1}\int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}}\int_{0}^{1+xy} \mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x$$

```
>> int(int(int(1,z,0,1+x*y),y,-sqrt(1-x^2),sqrt(1-x^2)),x,-1,1)

ans =

pi
```

så svaret er $V = \pi$, noe du sikkert greier å sjekke ved å regne for hånd.

A Plotting av flater med `surf`

Her skal vi i hovedsak studere et eksempel, nemlig flaten du regner på i Oppgave 2. Eksemplet er ganske omfattende, og kanskje ikke noen god modell for hvordan man skal gå fram for å få et kjapt innblikk i hvordan et legeme i rommet ser ut. Det fins helt sikkert mer hensiktsmessige måter å raskt få satt opp en slik figur, selv om den kanskje blir mindre presist tegnet enn det vi viser her. Det anbefales også å se på hvordan dette er gjort i eksempler i Maple tilgjengelig på mapleøvingenes websider.

En flate i rommet beskrives ofte ved hjelp av to parametre, si s og t. Man har da at punktene varierer over flaten når s og t varierer over et rektangel

$$x = x(s,t),\ y = y(s,t),\ z = z(s,t), \qquad a \le s \le b, \quad c \le t \le d.$$

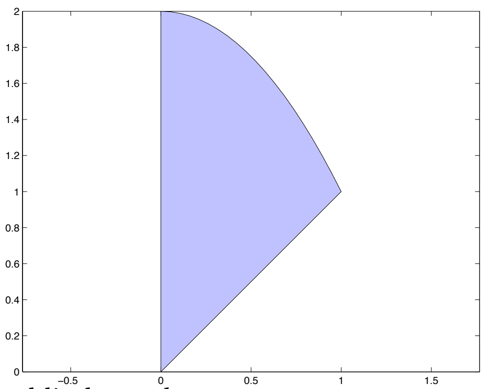
Matlab utnytter dette ved at man deler opp både s- og t-intervallene i delintervaller.

$$a = s_1 \le s_2 \le \cdots \le s_n = b, \qquad c = t_1 \le t_2 \le \cdots \le t_m = d$$

Nå lager man $n \times m$ matriser X, Y, Z slik at $X_{ij} = x(s_i, t_j)$. Det kan ofte være utfordrende å bestemme parametriseringer korrekt, og lage matrisene. Typisk må man lage hele flaten ved å sette sammen delflater. La oss ta et eksempel. Tenk deg at vi skal plotte flaten som er begrenset av flatene

$x = 0,$	$yz-$planet	Sidevegg
$y = x,$	vertikalt plan midt i mellom x- og y-aksen	Sidevegg
$2x - y + 4z = 0$	nesten horisontalt plan	Bunnflate
$x^2 + y = 2$	parabolsk sylinder	Krum Sidevegg
$2y^2 + z = 9$	parabolsk sylinder	Skrånende toppflate

Vi finner først projeksjonen (skyggen) på xy-planet, den er definert av linjene $x = 0$, $y = x$ og parabelen $y = 2 - x^2$. La oss nå se på sideveggen $x = 0$. Der vil y-verdien ligge mellom 0 og 2. z-verdien ligger mellom bunnflateverdien $z = y/4$ (siden $x = 0$) og toppverdien $z = 2 - 9y^2$.

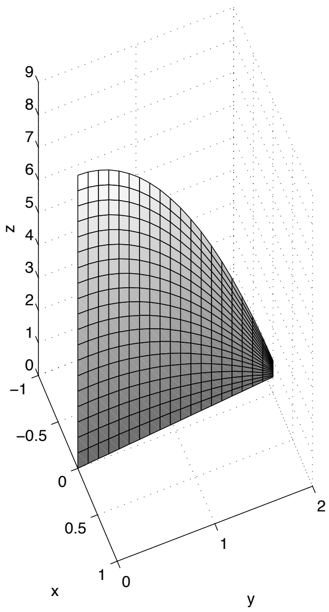


En parametrisering av sideveggen $x = 0$ blir dermed

$$x(s,t) = 0, \quad y(s,t) = 2s, \quad z(s,t) = (1-t)\,\frac{y}{4} + t\,(9 - 2y^2) = \frac{s(1-t)}{2} + t(9 - 8\,s^2)$$

Her har vi $0 \le s \le 1$ og $0 \le t \le 1$. Denne sideveggen er vi nå rede til å plotte. Forsøk følgende

```
% First we draw the plane x=0
figure(1), hold off
s=linspace(0,1,20)';
t=s';
X=zeros(length(s),length(t));
Y=2*s*ones(size(t));
Z = s*(1-t)/2 + (9-8*s.^2)*t;
surf(X,Y,Z);
axis equal
colormap(gray);
hold on
```



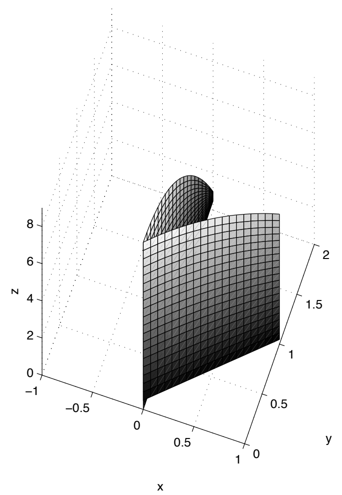
Vi har skrevet `hold on` for at ikke planet skal forsvinne fra plottet neste gang vi bruker `surf`. Kommandoen `colormap(gray)` gir gråtoner, uten denne ville vi fått flere farger i plottet, men ofte gir gråtoneplott et bedre visuelt inntrykk.

Vi kan nå fortsette med planet $y = x$. Horisontalt strekker dette planet seg langs den rette linjen fra (0,0 til (1,1). Vertikalt starter vi på $z = (2-2x+y)/4 = (2-x)/4$ (siden $y = x$). Toppunktet inntreffer ved $z = 9-2y^2 = 9-2x^2$. Dermed har vi parametriseringen

$$x(s,t) = s, \quad y(s,t) = s, \quad z(s,t) = \frac{2-s}{4}(1-t) + (9-2s^2)t, \quad (t,s) \in [0,1] \times [0,1]$$

La oss legge inn det nye planet i samme plott.

```
% Plane y=x
X=s*ones(size(t));
Y=X;
Z=(2-s)/4*(1-t)+(9-2*s.^2)*t;
surf(X,Y,Z);
```



Vi fortsetter nå med bunnplanet som vil ligne på projeksjonen ovenfor, men vil ikke være helt horisontalt. En mulighet for parametrisering er å bruke vinkelen θ med y-aksen som parameter i fra skyggen, slik at vi kan skrive

$$x(\theta,t) = r(\theta)\sin\theta\ t, \quad y(\theta,t) = r(\theta)\cos\theta\ t, \qquad 0 \le \theta \le \pi/4.$$

Siden vinkelen er med y-aksen blir det cosinus på y og sinus på x. Parameteren t ligger mellom 0 og 1. $r(\theta)$ er avstanden fra origo opp til den buede delen av skyggen. Vi finner at

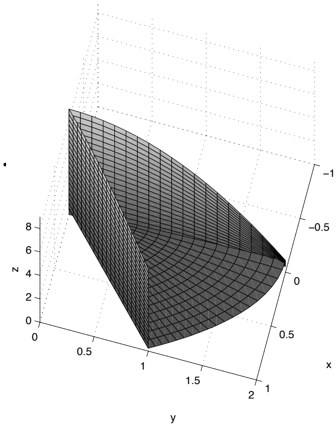
$$x^2 + y = 2 \quad \Rightarrow \quad r^2\sin^2\theta + r\cos\theta - 2 = 0,$$

og vi kan løse ut for den positive roten $r(\theta)$ som blir

$$r(\theta) = \frac{4}{\cos\theta + \sqrt{\cos^2(\theta) + 8\sin^2(\theta)}}, \quad 0 \le \theta \le \frac{\pi}{4}$$

Før vi er rede til å tegne, kan vi merke oss at $z(\theta,t) = (2-2x(\theta,t) + y(\theta,t))/4$.

```
% Planet 2x-y+4z=2
theta = linspace(0,pi/4,20)';
r= 4./(cos(theta)+sqrt(cos(theta).
t=linspace(0,1,20);
X = (r.*sin(theta)) * t ;
Y = (r.*cos(theta)) * t ;
Z = 1/4*(2-2*X+Y);
surf(X,Y,Z);
```

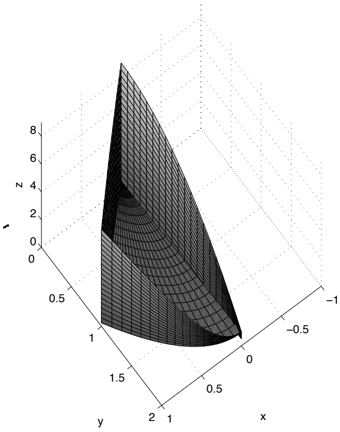


Da er det den siste sideveggen som står for tur. Dette er en vertikal krum vegg, horisontale snitt (skjæring med $z = c$) gir kurven $y = 2 - x^2$. Vi kan la $s = x$ være en parameter og får da umiddelbar

$$x(s,t) = s, \quad y(s,t) = 2 - s^2, \qquad 0 \le s \le 1.$$

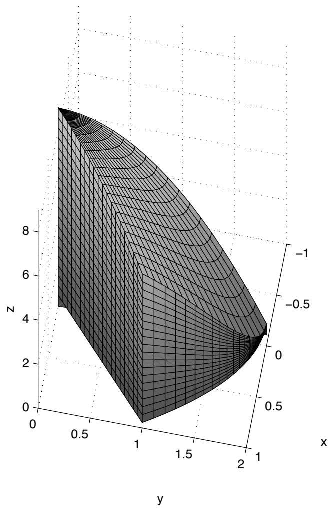
For en gitt (x, y) av denne formen ser vi på den vertikale linja fra bånn til topp. Bånnpunktet må være $z_{\min}(s) = (2-2x+y)/4 = 1 - s/2 - s^2/4$. Toppunktet blir istedet $z_{\max}(s) = 9 - 2y^2 = 9 - 2(2-s^2)^2$, så $z(s,t) = z_{\min}(s)(1-t) + z_{\max}(s)t$. Vi inkluderer veggen på følgende vis

```
X = s*ones(size(t));
Y = 2-X.^2;
Z=(1-s/2-s.^2/4)*(1-t)+(9-2*(2-s.
surf(X,Y,Z);
```



Da var det bare taket eller lokket igjen. Nå kan vi bruke samme $x(\theta, t)$ of $y(\theta, t)$ som for bunnplaten, vi må kun skifte ut z-koordinaten, $z(\theta, t) = 9 - 2y(\theta, t)^2$.

```
X = (r.*sin(theta)) * t ;
Y = (r.*cos(theta)) * t ;
Z = 9-2*Y.^2;
surf(X,Y,Z);
```



og vips så var figuren ferdig.