Flervariabelanalys och Matlab
Kapitel 4

Thomas Wernstål
Matematiska Vetenskaper

3 oktober 2012

4 Vektoranalys

4.1 Vektorfält

Vi kan illustrera vektorfält, såväl i planet som i rummet, med kommandona `quiver` resp. `quiver3`. Först måste vi dock skapa ett rutnät med de punkter i vilket vi vill sätta pilar (som illustrerar vektorfältets storlek och riktning i respektive punkt). Det plana vektorfältet $\mathbf{F} = xy\,\mathbf{i} + (x-y)\mathbf{j}$ kan vi t.ex. illustrera med följande kommandon;

```
>> F1=@(x,y) sin(x.*y); F2=@(x,y) cos(x-y);
>> x=linspace(-3,3,20);y=linspace(-3,3,21);
>> [X,Y]=meshgrid(x,y);
>> quiver(X,Y,F1(X,Y),F2(X,Y))
```

Det är viktigt att tänka på att man inte sätter pilarna för tätt (dvs. har för många element i vektorerna `x` och `y`). Å andra sidan vill man ha tillräckligt med pilar för att få en bra uppfattning om vektorfältets skiftningar. Det kan vara svårt att få det bra från början och man kan behöva pröva sig fram.

Låt oss även plotta ett vektorfält i rummet;

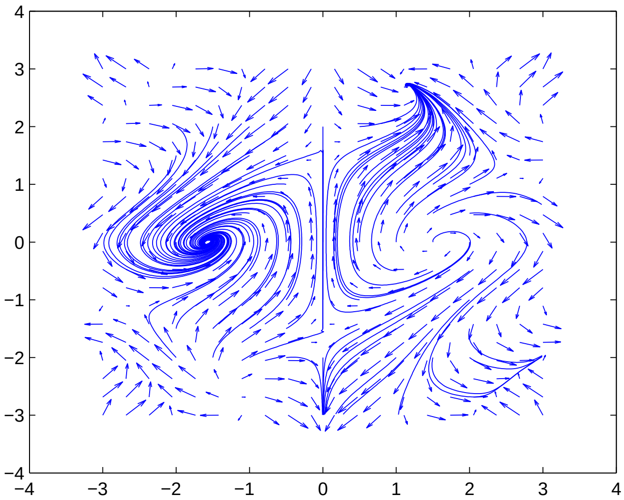
```
>> F1=@(x,y,z) x.^3.*y+z.^2; F2=@(x,y,z) x.*y.^2+z.^2; F3=@(x,y,z) x.^2+y.^2.*z;
>> x=linspace(-3,3,11);y=linspace(-3,3,10);z=linspace(-3,3,10);
>> [X,Y,Z]=meshgrid(x,y,z);
>> quiver3(X,Y,Z,F1(X,Y,Z),F2(X,Y,Z),F3(X,Y,Z))
```

Om man här väljer för många punkter finns också risken att det tar för lång tid för Matlab att plotta figuren (och att det hänger sig), så var försiktig när du skapar rutnätet med `meshgrid`. Börja hellre med ett lite grovare rutnät för att sedan förfina i den mån det behövs.

4.2 Fältlinjer och flöden

Vi kan även använda MATLAB för att plotta fältlinjer till vektorfält t.ex. med hjälp av kommandot `streamlines` (fältlinjerna kallas ju strömlinjer om vektorfältet representerar hastighetsvektorer i ett flöde). Man måste då också ange i vilka startpunkter som fältlinjerna skall börja i. Startpunkterna kan tex. skapas med `meshgrid`. Följande kommandon plottar några fältlinjer till vektorfältet $\mathbf{F} = xy\,\mathbf{i} + (x-y)\mathbf{j}$ i samma figur som vektorfältet självt;

```
>> x=linspace(-3,3,20);y=linspace(-3,3,21);
>> [X,Y]=meshgrid(x,y);
>> F1=sin(X.*Y); F2=cos(X-Y);
>> quiver(X,Y,F1,F2)
>> [Sx,Sy]=meshgrid(-2:0.5:2);
>> streamline(X,Y,F1,F2,Sx,Sy)
```



Startpunkterna anges här av matriserna `Sx` och `Sy`. Vi kan naturligtvis plotta fältlinjerna genom samtliga punkter i vilket vi ritat en pil, men då riskerar vi att figuren blir för plottrig. Pröva och avgör själv;

```
>> streamline(X,Y,F1,F2,X,Y)
```

Det går även bra att plotta fältlinjer till vektorfält i rummet;

```
>> x=linspace(-3,3,11); y=linspace(-3,3,10); z=linspace(-3,3,10);
>> [X,Y,Z]=meshgrid(x,y,z);
>> F1=X.^3.*Y+Z.^2; F2=X.*Y.^2+Z.^2; F3=X.^2+Y.^2.*Z;
>> quiver3(X,Y,Z,F1,F2,F3)
>> [Sx,Sy,Sz]=meshgrid(-2:1:2);
>> streamline(X,Y,Z,F1,F2,F3,Sx,Sy,Sz)
```

Om man vill kan man också välja startpunkterna för fältlinjerna så att de ligger på en och samma (parametriserad) yta. Pröva t.ex.

```
>> [S,T]=meshgrid(0:0.1:1);
>> Sx=S.^2-T; Sy=S+T.^2; Sz=S-T;
>> surf(Sx,Sy,Sz)
>> streamline(X,Y,Z,F1,F2,F3,Sx,Sy,Sz)
```

Startpunkterna kan också ligga på en nivåyta. Pröva t.ex.

```
>> [S,T,R]=meshgrid(-3:0.3:3);
>> G=S.^2+T.^2+R.^2;
>> clf, isosurface(S,T,R,G,1);
>> [p,q]=isosurface(S,T,R,G,1);
>> Sx=q(:,1); Sy=q(:,2); Sz=q(:,3);
>> streamline(X,Y,Z,F1,F2,F3,Sx,Sy,Sz)
```

Man kan också studera hur partiklar rör sig (utefter fältlinjerna) under inverkan av vektorfältet;

```
>> verts = stream3(X,Y,Z,F1,F2,F3,Sx,Sy,Sz);
>> iverts = interpstreamspeed(X,Y,Z,F1,F2,F3,verts,0.01);
>> streamparticles(iverts,30,'animate',10,'FrameRate',50)
```

Om man vill illustrera någon typ av flöde/strömning genom en yta så kan det räcka med att plotta hastighetsvektorer på själva ytan (och inte i det omgivande området). Låt oss titta på ett exempel.

Exempel. Låt S vara den yta som parametriseras av $x = s+t, y = s^2-t^2, z = st$, $0 \leq s,t \leq 1$ och låt $\mathbf{F} = (x-1)\mathbf{i} + 2z\,\mathbf{j} - y\,\mathbf{k}$ vara hastighetsvektorn i en vätskeströmning. Den vätskevolym som strömmar genom ytan S ges av normalytintegralen $\iint_S \mathbf{F}\bullet\hat{\mathbf{N}}\,dS$. För att illustrera hur strömningen varierar utefter ytan plottar vi ett antal hastighetsvektorer på ytan S.

```
>> [S,T]=meshgrid(0:0.05:1);
>> X=S+T; Y=S.^2-T.^2; Z=S.*T;
>> surf(X,Y,Z), hold on
>> F1=X-1;F2=2*Z;F3=-Y;
>> quiver3(X,Y,Z,F1,F2,F3), hold off
>> shading interp, axis equal
```

Om man är mer intresserad av hur mycket vätska som strömmar genom ytans olika delar och inte strömningens riktning så kan man plotta hastighetsvektorernas projektion i normalriktningen. Detta är lite mer komplicerat men kan t.ex. utföras med följande kommandorader.

```
>> [S,T]=meshgrid(0:0.05:1);
>> X=S+T; Y=S.^2-T.^2; Z=S.*T;
>> surf(X,Y,Z), hold on
>> [U,V,W]=surfnorm(X,Y,Z);
>> k=sqrt(U.^2+V.^2+W.^2); U=U./k; V=V./k; W=W./k;
>> [m,n]=size(U); r=m*n;
>> uq=U(1:r); vq=V(1:r); wq=W(1:r);
>> xq=X(1:r); yq=Y(1:r); zq=Z(1:r);
>> pq=dot([uq;vq;wq],[xq-1;2*zq;-yq]);
>> P=ones(size(U)); P(1:r)=pq;
>> quiver3(X,Y,Z,P.*U,P.*V,P.*W), hold off
>> shading interp, axis equal
```

Istället för att plotta normalvektorer till ytan som med sin längd anger strömningens storlek genom ytan så kan vi låta färgen på ytan variera med längden på vektorerna (om man inte anger något annat så bestäms färgen på ytan av höjden över xy-planet dvs. värdet på z i resp. punkt). Det kan då vara lättare att se åt vilket håll strömningen sker och hur stor den är på ytans olika delar. Istället för att använda `quiver3` (dvs. den näst sista kommandoraden ovan) så ger vi i så fall följande kommando;

```
>> surf(X,Y,Z,P), colorbar, shading interp
```

Det röda anger strömning genom ytan åt det ena hållet och blått åt det andra hållet. □

4.3 Divergens och rotation av vektorfält

Två viktiga begrepp inom vektoranalysen är *divergens* och *rotation*. För ett vektorfält $\mathbf{F}(x, y, z) = F_1(x, y, z)\mathbf{i} + F_2(x, y, z)\mathbf{j} + F_3(x, y, z)\mathbf{k}$ ger divergensen;

$$\mathrm{div}\mathbf{F} = \frac{\partial F_1}{\partial x} + \frac{\partial F_2}{\partial y} + \frac{\partial F_3}{\partial z}$$

ett mått på vektorfältets tendens att ”stråla” ut från (eller in mot) en viss punkt. Antag till exempel att vektorfältet beskriver hastigheten hos molekylerna i en gas som hettas upp eller kyls ner. I områden där gasen expanderar har divergensen ett positivt värde (sådana områden kallas källor) och i områden där gasen komprimeras är divergensen negativ (sådana områden kallas sänkor).

Divergensen kan i MATLAB beräknas med kommandot `divergence` och för att illustrera storleken divergensen hos ett plant vektorfält kan vi t.ex. använda `contourf`;

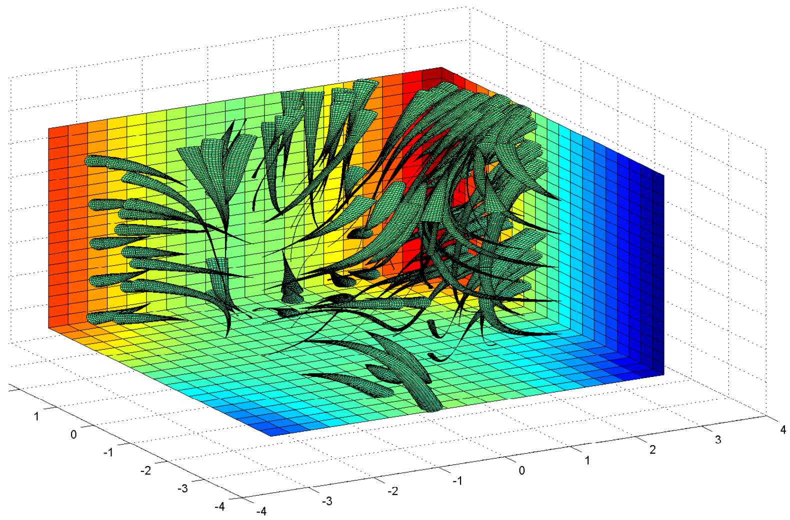
```
>> x=linspace(-3,3,30);y=linspace(-3,3,31);
>> [X,Y]=meshgrid(x,y);
>> F1=sin(X.*Y); F2=cos(X-Y);
>> DIV=divergence(X,Y,F1,F2);
>> contourf(X,Y,DIV), hold on
>> quiver(X,Y,F1,F2), hold off
```

och för vektorfält i rummet kan vi t.ex. använda `slice`;

```
>> x=linspace(-3,3,21); y=linspace(-3,3,20); z=linspace(-3,3,20);
>> [X,Y,Z]=meshgrid(x,y,z);
>> F1=X.^3.*Y+Z.^2; F2=X.*Y.^2+Z.^2; F3=X.^2+Y.^2.*Z;
>> DIV=divergence(X,Y,Z,F1,F2,F3);
>> slice(X,Y,Z,DIV,[3],[3],[-3])
```

Man kan också använda kommandot `streamtube` som plottar fältlinjer i form av tuber. Tjockleken på tuberna varierar proportionellt mot storleken på divergensen utefter resp. fältlinje;

```
>> [Sx,Sy,Sz]=meshgrid(-2:1:2);
>> streamtube(X,Y,Z,F1,F2,F3,Sx,Sy,Sz)
```



En annan typ av information om vektorfältet ger rotationen;

$$\mathrm{curl}\mathbf{F} = \left(\frac{\partial F_3}{\partial y} - \frac{\partial F_2}{\partial z}\right)\mathbf{i} + \left(\frac{\partial F_1}{\partial z} - \frac{\partial F_3}{\partial x}\right)\mathbf{j} + \left(\frac{\partial F_2}{\partial x} - \frac{\partial F_1}{\partial y}\right)\mathbf{i}$$

Till skillnad mot divergensen (som är en skalär) så ger rotationen en vektor i varje punkt. Längden på vektorn ger ett mått på vektorfältets tendens att virvla i resp. punkt och riktningen på vektorn beskriver den axel kring vilket det virvlar mest. För att illustrera storleken på divergensen hos ett plant vektorfält kan vi som ovan t.ex. använda `contourf`;

```
>> x=linspace(-3,3,30);y=linspace(-3,3,31);
>> [X,Y]=meshgrid(x,y);
>> F1=sin(X.*Y); F2=cos(X-Y);
>> Rz=curl(X,Y,F1,F2);
>> contourf(X,Y,abs(Rz)), hold on
>> quiver(X,Y,F1,F2), hold off
```

och för vektorfält i rummet kan vi t.ex. använda `slice`;

```
>> x=linspace(-3,3,21); y=linspace(-3,3,20); z=linspace(-3,3,20);
>> [X,Y,Z]=meshgrid(x,y,z);
>> F1=X.^3.*Y+Z.^2; F2=X.*Y.^2+Z.^2; F3=X.^2+Y.^2.*Z;
>> [Rx,Ry,Rz]=curl(X,Y,Z,F1,F2,F3);
>> slice(X,Y,Z,sqrt(Rx.^2+Ry.^2.^2+Rz.^2),[3],[3],[-3])
```

Man kan också använda kommandot `streamribbon` som plottar fältlinjer i form av band. Där bandet vrider sig mycket runt fältlinjen är rotationen stor;

```
>> [Sx,Sy,Sz]=meshgrid(-1:1:1);
>> streamribbon(X,Y,Z,F1,F2,F3,Sx,Sy,Sz)
```

Det kan ta lite tid för MATLAB att utföra beräkningarna och plotta resultatet så ha lite tålamod. Tänk på att vara försiktig med storleken på matriserna, börja hellre med en gles indelning och förfina gradvis.