

Linjära system av differentialekvationer

1 Inledning

Vi har i envariabelanalysen sett på allmänna system av differentialekvationer med begynnelsevillkor

$$\begin{cases} \mathbf{u}' = \mathbf{f}(t, \mathbf{u}), & a \leq t \leq b \\ \mathbf{u}(a) = \mathbf{u}_a \end{cases}$$

Nu skall vi studera lösningar till *linjära* system av differentialekvationer

$$\begin{cases} \mathbf{u}' = \mathbf{A}\mathbf{u}, & 0 \leq t \leq T \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases}$$

där \mathbf{A} är en konstant matris. Sambandet mellan \mathbf{f} och matrisen \mathbf{A} ges av:

$$\mathbf{f}(t, \mathbf{u}) = \mathbf{A}\mathbf{u}$$

Det är enkelt att lösa det linjära systemet i MATLAB med

```
>> f=@(t,u)A*u  
>> [t,U]=ode45(f,[0,T],u0)
```

Till skillnad från de flesta icke-linjära system så kan vi lösa de linjära systemen analytiskt (dvs. exakt) med den så kallade *egenvärdesmetoden*.

2 Riktningsfält och fasporträtt

För att få en lite bättre uppfattning av vilka egenskaper systemet $\mathbf{u}' = \mathbf{A}\mathbf{u}$ har så skall vi titta närmare på högerledet: $\mathbf{F}(\mathbf{u}) = \mathbf{A}\mathbf{u}$. Funktionen \mathbf{F} är exempel på en vektorvärd funktion och kallas *vektorfält*.

För varje initialvärde till $\mathbf{u}'(t) = \mathbf{A}\mathbf{u}(t)$ så har vi en (okänd) lösning $\mathbf{u}(t)$. Från differentialekvationen har vi

$$\mathbf{u}' = \mathbf{A}\mathbf{u} = \mathbf{F}(\mathbf{u})$$

Vektorfältet \mathbf{F} ger oss i varje punkt \mathbf{u} den riktning i vilken lösningen förändras och dess längd förändringstakten. Genom att i ett lämpligt antal punkter \mathbf{u} markera styrka och riktning för vektorn $\mathbf{F}(\mathbf{u})$ med en pil så får vi ett *riktningsfält*.

Genom att rita upp detta riktningsfält får vi ungefärlig information om hur lösningen $\mathbf{u}(t)$ till ekvationen beter sig för samtliga möjliga startvärden. Vi kan alltså skapa kurvan $\mathbf{u}(t)$ genom att sätta pennan på den startpunkt vi önskar och sedan följa pilarna.

För att rita vektorfält i planet använder vi kommandot `quiver`. Men först måste vi skapa ett gitter (grid) med de punkter i vilka vi vill sätta ut pilar (som beskriver vektorfältets storlek och riktning i respektive punkt).

Om vi som exempel tar $\mathbf{F}(\mathbf{u}) = (u_1 - u_2, u_1 u_2)$ kan vi rita riktningsfältet enligt

```
>> u1=linspace(-2,2,30); u2=linspace(-2,2,30);
>> [U1,U2]=meshgrid(u1,u2);
>> F1=@(u1,u2)u1-u2; F2=@(u1,u2)u1.*u2; s=1.2;
>> quiver(U1,U2,F1(U1,U2),F2(U1,U2),s) % s - skalfaktor som förlänger pilarna.
>> axis([-2 2 -2 2])
```

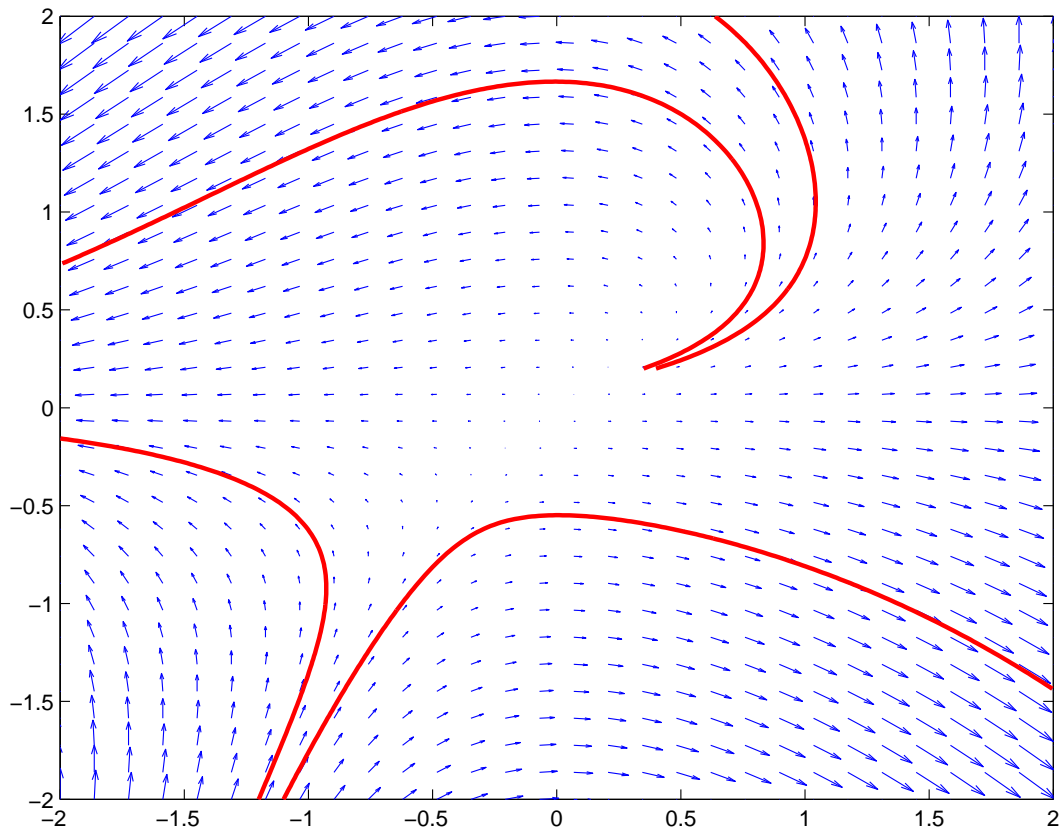
Vi kan också lägga till några *strömlinjer* (lösningsbanor) med funktionen `streamline`, som bygger på approximationen

$$\mathbf{u}(t+h) \approx \mathbf{u}(t) + h\mathbf{u}'(t) = \mathbf{u}(t) + h\mathbf{F}(\mathbf{u})$$

vilket är Eulers metod på systemet $\mathbf{u}' = \mathbf{F}(\mathbf{u})$.

```
>> s1=[-1.2 -1.1 0.35 0.4]; % Första-koordinater för fyra startpunkter
>> s2=[-2 -2 0.2 0.2]; % och deras andra-koordinater
>> h=streamline(U1,U2,F1(U1,U2),F2(U1,U2),s1,s2)
>> set(h,'Color','r','LineWidth',2)
```

och får en bild där pilarnas riktning visar fältets riktning.



Ytterligare ett exempel tar vi: Vi löser

$$\begin{cases} \mathbf{u}' = \mathbf{A}\mathbf{u}, & 0 \leq t \leq T \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases}$$

där

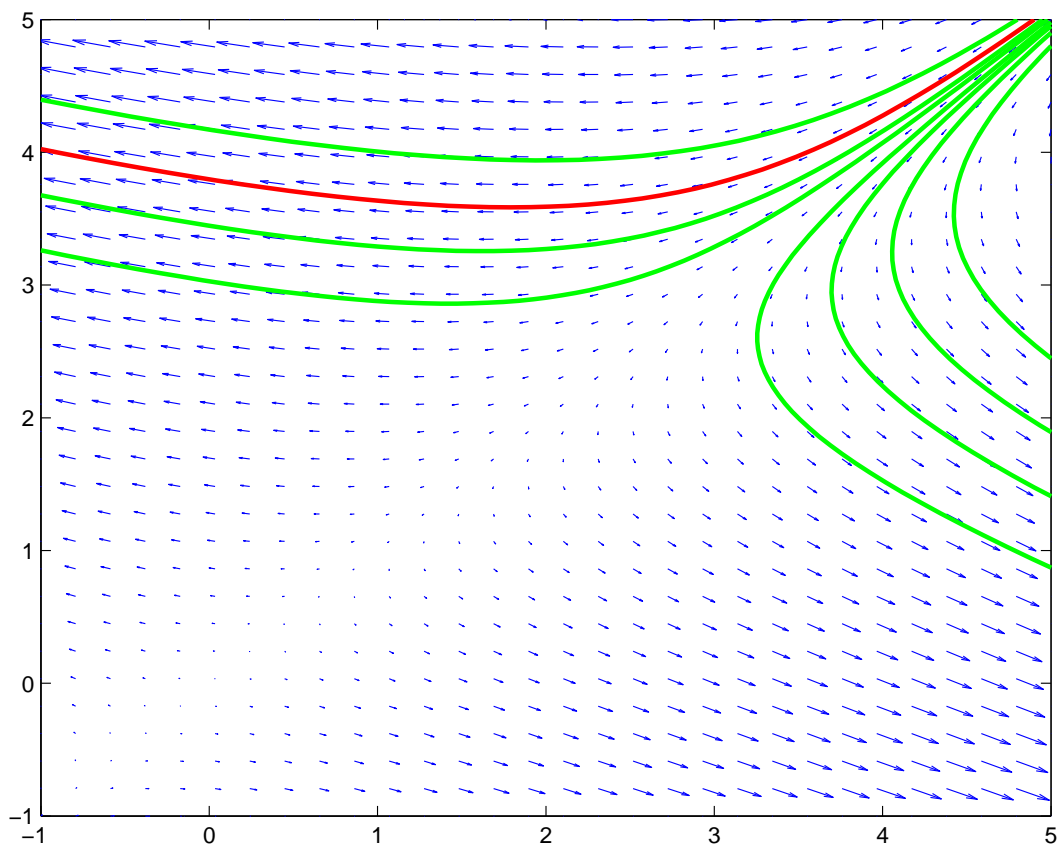
$$\mathbf{A} = \begin{bmatrix} 4 & -5 \\ -2 & 1 \end{bmatrix}, \quad \mathbf{u}_0 = \begin{bmatrix} 4.9 \\ 5.0 \end{bmatrix}$$

En (numerisk) lösning får vi enligt

```
>> A=[4 -5;-2 1]; u0=[4.9;5.0];  
>> F=@(t,u)A*u;  
>> [t,U]=ode45(F,[0 10],u0);
```

Tidigare har vi främst ritat ut graferna för koordinatfunktionerna $u_1(t)$ och $u_2(t)$, men det är också naturligt att visualisera lösningen i ett *fasporträtt* där vi ritar $u_1(t)$ mot $u_2(t)$. Vi ritar in några sådana kurvor (banor) i ett riktningsfält:

```
>> u1=linspace(-1,5,30); u2=linspace(-1,5,30);  
>> [U1,U2]=meshgrid(u1,u2);  
>> F1=A(1,1)*U1+A(1,2)*U2; F2=A(2,1)*U1+A(2,2)*U2;  
>> quiver(U1,U2,F1,F2,0.9)  
>> axis([-1 5 -1 5]), hold on  
>> plot(U(:,1),U(:,2),'r','LineWidth',2)
```



Vi ser hur lösningarna $\mathbf{u}(t)$ följer fältets riktning.

Uppgift 1. Betrakta begynnelsevärdesproblemet

$$\begin{cases} \mathbf{u}' = \mathbf{A}\mathbf{u}, & 0 \leq t \leq T \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases}$$

Rita riktningsfält och fasporträtt då

$$(a). \quad \mathbf{A} = \begin{bmatrix} 1 & -2 \\ 3 & -4 \end{bmatrix}, \quad \mathbf{u}_0 = \begin{bmatrix} -3 \\ 5 \end{bmatrix}, \quad T = 5 \quad (b). \quad \mathbf{A} = \begin{bmatrix} 0 & 3 \\ -12 & 1 \end{bmatrix}, \quad \mathbf{u}_0 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \quad T = 6$$

3 Egenvärdesmetoden

Antag att \mathbf{A} är en diagonaliserbar 2×2 -matris med en bas av egenvektorer \mathbf{v}_1 och \mathbf{v}_2 och med tillhörande egenvärden λ_1 respektive λ_2 . Då ges lösningen

$$\begin{cases} \mathbf{u}' = \mathbf{A}\mathbf{u}, & 0 \leq t \leq T \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases}$$

av formeln

$$\mathbf{u}(t) = c_1 \mathbf{v}_1 e^{\lambda_1 t} + c_2 \mathbf{v}_2 e^{\lambda_2 t},$$

där koefficienterna c_1 och c_2 bestäms av startvärdena. Vi sätter $t = 0$ och får

$$\mathbf{u}(0) = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 = [\mathbf{v}_1 \ \mathbf{v}_2] \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}.$$

Detta visar att

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \mathbf{V}^{-1} \mathbf{u}_0, \quad (\mathbf{u}(0) = \mathbf{u}_0),$$

där matrisen $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2]$ (kolonnerna är egenvektorerna till matrisen \mathbf{A}).

Uppgift 2. Undersök system $\mathbf{u}' = \mathbf{A}\mathbf{u}$ för matriserna \mathbf{A} nedan. Rita i samma graf egenvektorerna till \mathbf{A} och jämför riktningfältets egenskaper med egenvektorerna och motsvarande egenvärden.

Förklara sambandet mellan riktningfältet och matrisens egenvärden och egenvektorer.

När utgör origo en *källa* (alla lösningar strömmar från origo), en *sänka* (alla lösningar strömmar till origo), en *sadelpunkt* (lösningarna går mot origo men avviker sedan) eller en *spiralpunkt* (lösningarna går i spiral kring origo)?

Den som inte orkar beräkna alla egenvärden och egenvektorer för hand kan använda kommandot `eig` som gör detta. Med `[V,D]=eig(A)` hamnar egenvärdena för matrisen \mathbf{A} längs diagonalen i \mathbf{D} och egenvektorerna blir kolonner i \mathbf{V} .

Rita också lösningen till differentialekvationen för några startvärden som ni hittar på själva.

$$(a). \quad \mathbf{A} = \begin{bmatrix} -2 & -5 \\ 1 & 4 \end{bmatrix} \quad (b). \quad \mathbf{A} = \begin{bmatrix} 1 & -2 \\ 3 & -4 \end{bmatrix} \quad (c). \quad \mathbf{A} = \begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix}$$

$$(d). \quad \mathbf{A} = \begin{bmatrix} 1 & 4 \\ -9 & 1 \end{bmatrix} \quad (e). \quad \mathbf{A} = \begin{bmatrix} -2 & 1 \\ -8 & 2 \end{bmatrix}$$

1 Målsättning

Avsikten med laborationen är att åter se på system av differentialekvationer $\mathbf{u}' = \mathbf{f}(t, \mathbf{u})$ och då speciellt linjära sådana $\mathbf{u}' = \mathbf{A}\mathbf{u}$. Vi ritar riktningsfält och strömlinjer (vilket givetvis är intressant för både linjära och icke-linjära system). Vidare bestämmer vi egenvärden och egenvektorer med funktionen `eig` och ser hur vi kan använda dessa för att skriva upp formler för lösningar till linjära system av differentialekvationer.

2 Kommentarer och förklaringar

Förståelsen av detaljerna runt hur vi använder `meshgrid` och `quiver` får kanske vänta till kursen i flervariabelanalys. Då kommer vi rita grafer av funktioner av två variabler (funktionsytor) och lära oss allt om `meshgrid`. Med `quiver(x,y,u,v,s)` ritas en pil från punkten (x, y) i riktningen (u, v) och s är en skalfaktor för att få pilen lagom lång.

När vi använder `streamline` för att rita strömlinjer väljer vi att använda en pekare till det grafiska objektet, så att vi sedan lätt kan sätta lämplig färg och tjocklek på kurvan.

Vill vi rita fält i rummet använder vi `meshgrid` och `quiver3`. Strömlinjer ritar vi med `streamline`.

3 Lärandemål

Efter denna laboration skall du kunna

- rita riktningsfält och strömlinjer till ett system av linjära differentialekvationer $\mathbf{u}' = \mathbf{A}\mathbf{u}$ med `meshgrid` och `quiver` respektive `streamline` och `ode45`
- beräkna egenvärden och egenvektorer till en matris \mathbf{A} med `eig`
- lösa ett linjärt system $\mathbf{u}' = \mathbf{A}\mathbf{u}$ med egenvärdesmetoden