1 Inledning

Vi skall lösa egenvärdesproblem för matriser och egenvärdesproblem för differentialekvationer med randvärden. Sedan skall vi se lite på beräkningsmetoder. Avslutningsvis skall vi använda lösningar till egenvärdesproblem för att studera begynnelsevärdesproblem för linjära system av differentialekvationer $\boldsymbol{u}' = \boldsymbol{A}\boldsymbol{u}$.

2 Egenvärdesproblem för matriser

Vi skall se hur man i MATLAB löser egenvärdesproblemet $\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}$ för matrisen

$$\boldsymbol{A} = \begin{bmatrix} 5 & 1 & 7 \\ 1 & -2 & 9 \\ 8 & 3 & 1 \end{bmatrix}$$

Vi beskriver matrisen och beräknar egenvärden samt egenvektorer med

```
>> A=[5 1 7; 1 -2 9; 8 3 1];
>> [V,D]=eig(A)
V =
     -0.66688     -0.37769     -0.22509
     -0.43616      0.90902     -0.82741
     -0.60418      0.17615      0.51451
D =
       11.996            0            0
            0      -0.6715            0
            0            0     -7.32449
```

Vi kommer då finna egenvektorerna som kolonner i matrisen `V` med motsvarande egenvärden på diagonalen i diagonalmatrisen `D`. T.ex. tredje egenvektorn ges av `V(:,3)` och tillhörande egenvärde ges av `D(3,3)`.

Vi kontrollerar att $\boldsymbol{A}\boldsymbol{v}_3 = \lambda_3\boldsymbol{v}_3$ genom att bilda

```
>> r=A*V(:,3)-D(3,3)*V(:,3)
r =
  -6.6613e-16
            0
            0
```

som i praktiken är nollvektorn.

Uppgift 1. Beräkna egenvektorer och egenvärden till följande matriser. Kontrollera genom insättning att beräkningarna är korrekta.

(a). $\begin{bmatrix} 1 & 5 & 9 \\ 2 & 0 & 5 \\ 3 & 7 & 11 \end{bmatrix}$ **(b).** $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$ **(c).** $\begin{bmatrix} 3 & 2 & 1 \\ 0 & 8 & 5 \\ 7 & -4 & 1 \end{bmatrix}$

3 Egenvärdesproblem för differentialekvationer

Vi har tidigare sett på randvärdesproblem för ordinär differentialekvation av typen

$$\begin{cases} u''(x) = f(x, u, u'), \ x_a \le x \le x_b \\ u(x_a) = u_a, \ u(x_b) = u_b \end{cases}$$

och ett specialfall är *linjära* randvärdesproblem, dvs. sådana problem som kan skrivas

$$\begin{cases} u''(x) + a(x)u'(x) + b(x)u(x) = f(x), \ x_a \le x \le x_b \\ u(x_a) = u_a, \ u(x_b) = u_b \end{cases}$$

Vi skall nu se på linjära *egenvärdesproblem* för ordinär differentialekvation

$$\begin{cases} u''(x) + a(x)u'(x) + b(x)u(x) = \lambda u(x), \ x_a \le x \le x_b \\ u(x_a) = u(x_b) = 0 \end{cases}$$

De lösningar $u(x)$ vi söker är skilda från nollfunktionen (dvs. $u(x)$ är inte 0 för alla x) och kallas *egenfunktioner* och talen λ kallas *egenvärden* precis som för matrisproblemen.

Som exempel har

$$\begin{cases} u''(x) = \lambda u(x), \ 0 \le x \le 1 \\ u(0) = u(1) = 0 \end{cases}$$

följande egenfunktioner och motsvarande egenvärden

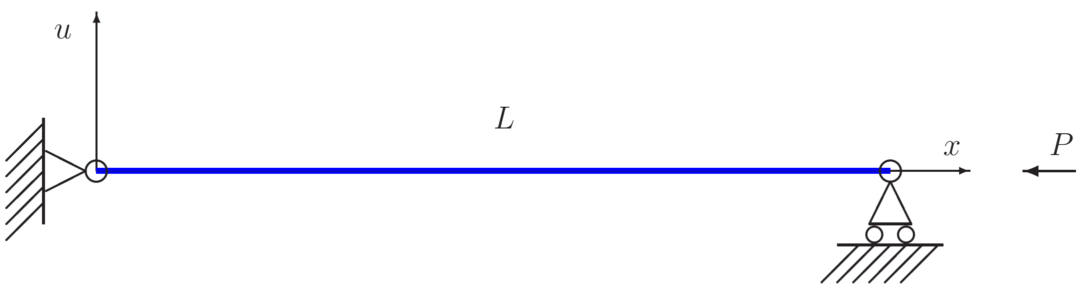
$$u_k(x) = B_k \sin(k\pi x), \quad \lambda_k = -k^2\pi^2, \quad k = 1, 2, \cdots$$

Lägg märke till att vi har oändligt många lösningar.

Som exempel på en tillämpning tar vi: *Euler knäckning.* Utböjningen u hos en ledlagrad stav av längd L som belastas med lasten P beskrivs av följande egenvärdesproblem för ordinär differentialekvation

$$\begin{cases} -EI\, u'' = Pu, \ 0 < x < L \\ u(0) = u(L) = 0 \end{cases}$$

där EI är böjstyvheten.



Givetvis kan vi skriva upp en formel för lösningen, men om böjstyvheten inte konstant skulle det vanligtvis inte gå. Därför skall vi se på differensmetoder som alltid ger en (numeriska) lösningar.

Vi diskretiserar problemet genom att införa en ekvidistant indelning av intervallet och använda en centraldifferensapproximation av derivatan.

Låter vi u_i beteckna approximationen av $u(x_i)$ får vi

$$\begin{cases} -u_{i+1} + 2u_i - u_{i-1} = h^2 \frac{P}{EI} u_i, \quad i = 1, 2, \cdots, n \\ u_0 = u_{n+1} = 0 \end{cases}$$

Med matriser kan detta skrivas

$$\boldsymbol{Au} = \lambda \boldsymbol{u}$$

där

$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}, \quad \lambda = h^2 \frac{P}{EI}$$

Egenvektorns olika komponenter u_i kommer beskriva vertikala förskjutningen hos staven, från neutralläget, för de olika x_i-värdena.

Uppgift 2. Lös egenvärdesproblemet för $L = 1$ och $EI = 1$ genom att lagra matrisen som en gles matris med `spdiags` och lösa med `eigs`, den glesa varianten av `eig`. Bestäm de tre minsta egenvärdena, motsvarande de tre lägsta kritiska lasterna, och rita ut egenvektorerna, motsvarande förskjutningarna av staven för respektive kritisk last. Skriv även ut de kritiska lasterna $P = \lambda EI/h^2$. Läs allra först hjälptexten för `eigs`. Tag $n = 50$ inre nodpunkter.

4 Beräkningsmetoder

För vanliga fyllda matriser finns numeriska metoder som beräknar alla egenvärden och egenvektorer på en gång. En av dessa bygger på upprepad QR-faktorisering och det är den som `eig` i MATLAB använder. När det gäller stora glesa matriser är det nästan alltid några få egenvärden och motsvarande egenvektorer man vill beräkna. Det finns ett antal olika metoder och gemensamt för dem är att de bygger på antingen den s.k. potensmetoden eller på den s.k. inversiteration. Läs i Lay kapitel 5.8.

4.1 Potensmetoden

Om $\boldsymbol{v}$ är en egenvektor till matrisen $\boldsymbol{A}$ så kan motsvarande egenvärde λ beräknas genom att man bildar den s.k. Rayleigh-kvoten

$$\lambda = \frac{\boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v}}{\boldsymbol{v}^T \boldsymbol{v}}$$

Potensmetoden går ut på att man bildar en följd av vektorer

$$\boldsymbol{u}_{k+1} = \boldsymbol{A}\boldsymbol{u}_k, \; k = 0, 1, \cdots$$

som förhoppningvis konvergerar mot en egenvektor till $\boldsymbol{A}$. Rayleigh-kvoterna

$$\sigma_k = \frac{\boldsymbol{u}_k^T \boldsymbol{A} \boldsymbol{u}_k}{\boldsymbol{u}_k^T \boldsymbol{u}_k}$$

ger då approximationer av motsvarande egenvärde.

Beräkningarna av steget i potensmetoden och Rayleigh-kvoten bryts lämpligen upp i följande delsteg:

(1). $\boldsymbol{w}_k = \boldsymbol{u}_k / \| \boldsymbol{u}_k \|$

(2). $\boldsymbol{u}_{k+1} = \boldsymbol{A}\boldsymbol{w}_k$

(3). $\sigma_k = \boldsymbol{w}_k^T \boldsymbol{u}_{k+1}$

Normaliseringen i första steget görs för att undvika överspill, normen på vektorerna kan annars växa enormt. Metoden startas med att vi tar $\boldsymbol{u}_0$ som en slumpvektor.

Antag att för egenvärdena λ_i till $\boldsymbol{A}$ gäller $|\lambda_1| \leq \cdots \leq |\lambda_{n-1}| < |\lambda_n|$ och att motsvarande egenvektorer $\boldsymbol{v}_i$, $i = 1, 2, \cdots, n$ är linjärt oberoende.

Då kan $\boldsymbol{u}_0$ skrivas som en linjär kombination av dessa egenvektorer

$$\boldsymbol{u}_0 = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i$$

Nu gäller att

$$\boldsymbol{u}_k = \boldsymbol{A}\boldsymbol{u}_{k-1} = \boldsymbol{A}^2 \boldsymbol{u}_{k-2} = \cdots = \boldsymbol{A}^k \boldsymbol{u}_0 = \sum_{i=1}^{n} \alpha_i \boldsymbol{A}^k \boldsymbol{v}_i = \sum_{i=1}^{n} \alpha_i \lambda_i^k \boldsymbol{v}_i$$

där vi har utnyttjat att $\boldsymbol{A}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$. En omskrivning ger

$$\boldsymbol{u}_k = \alpha_n \lambda_n^k \left(\boldsymbol{v}_n + \sum_{i=1}^{n-1} \frac{\alpha_i}{\alpha_n} \left(\frac{\lambda_i}{\lambda_n} \right)^k \boldsymbol{v}_i \right)$$

Eftersom λ_n är det till belopp största egenvärdet kommer $(\frac{\lambda_i}{\lambda_n})^k \to 0$ då $k \to \infty$, och därmed kommer $\boldsymbol{u}_k$ att gå mot $\alpha_n \lambda_n^k \boldsymbol{v}_n$, dvs. $\boldsymbol{u}_k$ kommer att konvergera mot egenvektorn $\boldsymbol{v}_n$ tillhörande det till belopp största egenvärdet λ_n. Vidare gäller att $\sigma_k \to \lambda_n$ då $k \to \infty$.

Uppgift 3. Använd potensmetoden för att bestämma egenvektorn sammanhörande med det till belopp största egenvärdet till matrisen

$$\boldsymbol{A} = \begin{bmatrix} 5 & 1 & 7 \\ 1 & -2 & 9 \\ 8 & 3 & 1 \end{bmatrix}$$

Kontrollera med `eig` att du får rätt resultat. En startvektor kan du bilda med `randn`, som ger normalfördelade slumptal.

4.2 Inversiteration

Med potensmetoden kan vi bestämma det till belopp största egenvärdet och motsvarande egenvektor. För att bestämma ett annat egenvärde och motsvarande egenvektor kan vi utnyttja att om $\boldsymbol{A}$ har egenvärdena λ_i, $i = 1, 2, \cdots, n$ så har matrisen $(\boldsymbol{A} - \mu \boldsymbol{I})^{-1}$ egenvärdena $(\lambda_i - \mu)^{-1}$. Så det egenvärde till $(\boldsymbol{A} - \mu \boldsymbol{I})^{-1}$ som är till belopp störst är det egenvärde till $\boldsymbol{A}$ som ligger närmast μ. (Se uppgift 3, räkneövning 4.) Vi kan därför om vi vill bestämma det egenvärde som ligger närmast μ genom att använda potensmetoden på matrisen $(\boldsymbol{A} - \mu \boldsymbol{I})^{-1}$.

Inversiterationsmetoden kan nu formuleras: Bestäm ett skift μ nära det sökta egenvärdet λ_j och låt $\boldsymbol{u}_0$ vara en startvektor, upprepa för $k = 0, 1, \cdots$

(1). $\boldsymbol{w}_k = \boldsymbol{u}_k / \parallel \boldsymbol{u}_k \parallel$

(2). $(\boldsymbol{A} - \mu \boldsymbol{I})\boldsymbol{u}_{k+1} = \boldsymbol{w}_k$

(3). $\sigma_k = \boldsymbol{w}_k^T \boldsymbol{u}_{k+1}$

Nu kommer σ_k att konvergera mot $(\lambda_j - \mu)^{-1}$, så λ_j approximeras av $\mu + 1/\sigma_k$, och $\boldsymbol{u}_k$ konvergerar mot $\boldsymbol{v}_j$. I andra steget skall man ju lösa ett linjärt ekvationssystem för $\boldsymbol{u}_{k+1}$, då gör man givetvis så att man LU- (eller Choleski-) faktoriserar matrisen $\boldsymbol{A} - \mu \boldsymbol{I}$ en gång och använder sedan dessa faktorer vid upprepad lösning av systemen.

Uppgift 4. Använd inversiteration för att bestämma egenvektorn sammanhörande med det till belopp minsta egenvärdet till matrisen i uppgift 3. Tag skiftet $\mu = 0$ och lös ekvationssystemet i steg (2) med backslash-kommandot (\). Kontrollera med `eig` att du får rätt resultat.

Uppgift 5. Vi ser åter på Euler knäckning från uppgift 2. Bestäm den lägsta kritiska lasten och motsvarande utböjning med inversiteration. Det räcker med $n = 30$ nodpunkter för att ge en hygglig approximation av egenfunktionen motsvarande lägsta egenvärdet. Använd skiftet $\mu = 0$, eftersom det är egenvärdet närmast noll vi vill bestämma.

5 Linjära system av ODE

I en tidigare laboration såg vi på allmäna system av differentialekvationer med begynnelsevillkor

$$\begin{cases} \boldsymbol{u}' = \boldsymbol{f}(t, \boldsymbol{u}), \ a \le t \le b \\ \boldsymbol{u}(a) = \boldsymbol{u}_a \end{cases}$$

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

$$\begin{cases} \boldsymbol{u}' = \boldsymbol{A}\boldsymbol{u}, \ 0 \le t \le T \\ \boldsymbol{u}(0) = \boldsymbol{u}_0 \end{cases}$$

där $\boldsymbol{A}$ är en konstant matris och $\boldsymbol{u}_0$ är en konstant vektor.

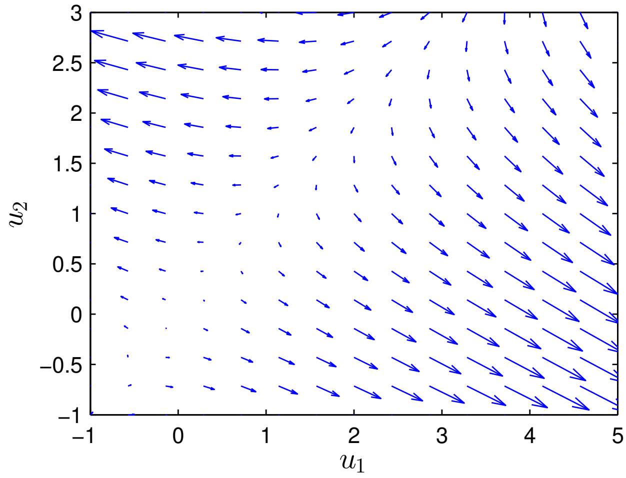
Vi beräknar lätt numeriska lösningar med t.ex. Eulers framåtmetod eller `ode45` i MATLAB men till skillnad från de flesta icke-linjära system så kan vi lösa de linjära systemen analytiskt (dvs. exakt) med *egenvärdesmetoden.*

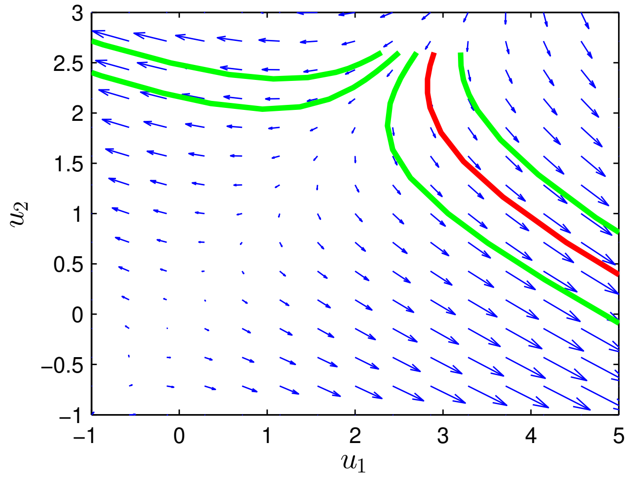
Som exempel löser vi

$$\begin{cases} \boldsymbol{u}' = \boldsymbol{A}\boldsymbol{u}, \ 0 \le t \le T \\ \boldsymbol{u}(0) = \boldsymbol{u}_0 \end{cases} \quad \text{där} \ \boldsymbol{A} = \begin{bmatrix} 4 & -5 \\ -2 & 1 \end{bmatrix}, \ \boldsymbol{u}_0 = \begin{bmatrix} 2.9 \\ 2.6 \end{bmatrix}$$

Vi ritar riktningsfältet

```
>> A=[4 -5;-2 1];
>> u1=linspace(-1,5,30); u2=linspace(-1,3,30);
>> [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,F2,1.5)
>> axis([-1 5 -1 3]), hold on
```





En (numerisk) lösning får vi enligt följande:

```
>> F=@(t,u)A*u;
>> u0=[2.9;2.6];
>> [t,U]=ode45(F,[0 5],u0);
>> plot(U(:,1),U(:,2),'r','LineWidth',2)
```

Vi ser hur lösningen $\boldsymbol{u}(t)$ följer fältets riktning i figuren ovan till höger. Där har vi även ritat ut lösningar för några andra begynnelsevillkor.

Vi kan ta fram en formel för lösningen till

$$\begin{cases} \boldsymbol{u}' = \boldsymbol{A}\boldsymbol{u}, \ 0 \le t \le T \\ \boldsymbol{u}(0) = \boldsymbol{u}_0 \end{cases}$$

Antag att $\boldsymbol{A}$ är diagonaliserbar så att $\boldsymbol{V}^{-1}\boldsymbol{A}\boldsymbol{V} = \boldsymbol{D}$, där $\boldsymbol{D}$ är en diagonalmatris med egenvärdena till $\boldsymbol{A}$ på diagonalen och kolonnerna i $\boldsymbol{V}$ är motsvarande egenvektorer.

Variabelbytet $\boldsymbol{w} = \boldsymbol{V}^{-1}\boldsymbol{u}$ överför systemet $\boldsymbol{u}' = \boldsymbol{A}\boldsymbol{u}$ till det okopplade systemet $\boldsymbol{w}' = \boldsymbol{D}\boldsymbol{w}$, dvs.

$$w_i' = \lambda_i w_i, \ i = 1, \cdots, n$$

Dessa ekvationer har lösningarna

$$w_i(t) = c_i \exp(\lambda_i t), \ i = 1, \cdots, n$$

där c_i är konstanter.

Om vi går tillbaka till ursprungsvariablerna får vi

$$\boldsymbol{u}(t) = \boldsymbol{V}\boldsymbol{w}(t) = \boldsymbol{v}_1 c_1 \exp(\lambda_1 t) + \cdots + \boldsymbol{v}_n c_n \exp(\lambda_n t)$$

och begynnelsetillståndet $\boldsymbol{u}(0) = \boldsymbol{u}_0$ ger $\boldsymbol{c} = \boldsymbol{V}^{-1}\boldsymbol{u}_0$.

Som ser åter på vårt exempel

$$\begin{cases} \boldsymbol{u}' = \boldsymbol{A}\boldsymbol{u}, \ 0 \le t \le T \\ \boldsymbol{u}(0) = \boldsymbol{u}_0 \end{cases} \quad \text{där} \ \boldsymbol{A} = \begin{bmatrix} 4 & -5 \\ -2 & 1 \end{bmatrix}, \ \boldsymbol{u}_0 = \begin{bmatrix} 2.9 \\ 2.6 \end{bmatrix}$$

Med MATLAB löser vi egenvärdesproblemet och ritar lösning enligt

```
>> A=[4 -5;-2 1]; u0=[2.9;2.6]; T=5;
>> [V,D]=eig(A)
>> t=linspace(0,T);
>> c=V\u0;
>> U=c(1)*V(:,1)*exp(D(1,1)*t)+c(2)*V(:,2)*exp(D(2,2)*t);
>> plot(U(1,:),U(2,:),'g')
```

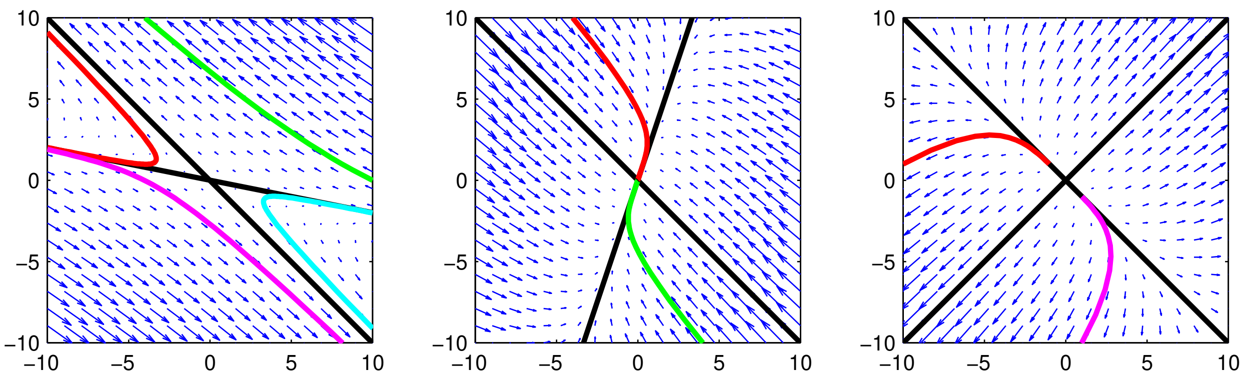
Lägg märke till hur vi bygger upp `U`, vektorn `t` är en radvektor och t.ex. `V(:,1)` är en kolonn så att `V(:,1)*exp(D(1,1)*t)` blir en matris. Längs första raden i `U`, dvs. `U(1,:)`, finner vi $u_1(t)$-värden för de olika t-värdena i vektorn `t` och på andra raden, dvs. `U(2,:)`, finner vi motsvarande $u_2(t)$-värden.

Uppgift 6. Undersök system $\boldsymbol{u}' = \boldsymbol{A}\boldsymbol{u}$ för matriserna $\boldsymbol{A}$ nedan. Rita riktningsfält och lösningar till differentialekvationen för några startvärden som ni hittar på själva. Använd ett lagom långt t-intervall. Matriserna har reella egenvärden. När utgör origo en *källa* (alla lösningar strömmar från origo), en *sänka* (alla lösningar strömmar till origo) eller en *sadelpunkt* (lösningarna går mot origo men avviker sedan)?

Rita i samma graf egenvektorerna till $\boldsymbol{A}$ och jämför riktningfältets egenskaper med egenvektorerna och motsvarande egenvärden. Förklara sambandet mellan riktningsfältet och matrisens egenvärden och egenvektorer.

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

När egenvärdena är reella har egenvektorerna en speciell betydelse (förutom att de bygger upp lösningarna). Om vi ritar linjer genom origo i egenvektorernas riktning får vi en uppdelning av fasplanet i sektorer ut från origo. En lösningskurva kan aldrig korsa en uppdelningslinje, den blir alltid kvar i samma sektor.



Här ovan har vi ritat dessa egenvektorslinjer (svarta), som delar upp planet i sektorer, för matriserna från uppgiften ovan.

Uppgift 7. Undersök systemet $\boldsymbol{u}' = \boldsymbol{A}\boldsymbol{u}$ för matriserna $\boldsymbol{A}$ nedan. Nu har vi komplexa egenvärden så origo kommer vara en *spiralpunkt* (lösningarna går i spiral kring origo).

Rita riktningfältet tillsammans med lösningar till differentialekvationen för några startvärden som ni hittar på själva.

(a). $\quad \boldsymbol{A} = \begin{bmatrix} 1 & 4 \\ -9 & 1 \end{bmatrix}$ $\quad$ **(b).** $\quad \boldsymbol{A} = \begin{bmatrix} -2 & 1 \\ -8 & 2 \end{bmatrix}$