TMV166 Linjär algebra för M

Datorlaboration 5: Egenvärden och egenvektorer

Allmänt

I den här laborationen skall vi beräkna egenvärden och egenvektorer numeriskt och se några tillämpningar.

Mål

- Praktiskt beräkna egenvärden och egenvektorer i Matlab
- Använda diagonalisering för att beräkna spänningar i mekanik
- Använda diagonalisering för att lösa system av ordinära differentialekvationer

Inledning

I föreläsning 13 såg vi hur man kan använda potensiteration för att hitta en egenvektor, och därmed även tillhörande egenvärde. Matlab använder istället en metod som bestämmer alla egenvärden och egenvektorer samtidigt. Den är baserad på en generalised form av *QR-iteration.* (Väldigt) enkelt uttryckt så sätter man först $A_1 = A$ och för $k = 1, 2, \ldots,$ faktoriserar man sedan $A_k = Q_k R_k$ och sätter $A_{k+1} = R_k Q_k$ tills man erhåller egenvärdena på diagonalen av R_k. Varje steg görs implicit för att hålla nere beräkningskostnaden, dvs. Q_k och R_k bildas aldrig explicit.

Via kommandot `eig` kan man få både egenvärden och egenvektorer, på olika sätt:

- `E = eig(A)` ger en kolonnmatris E, vars element är egenvärdena till matrisen A.
- `[V, D] = eig(A)` ger två matriser V och D. Kolonnerna i V är egenvektorerna till A, och D är en diagonalmatris med motsvarande egenvärden till A på diagonalen, ordnade i samma ordning. Vidare är kolonnerna i V normerade: `norm(V(:,k))= 1`. Om A är symmetrisk är kolonnerna dessutom parvis ortogonala, så att $V^{-1} = V^T$.
- `[V, D] = eig(A, 'vector')` ger samma sak som ovan, förutom att D istället blir en kolonnmatris.

Observera att `[V, D] = eig(A)` ger resultat även om A **inte** är diagonaliserbar. Detta inträffar om A har ett egenvärde med större (algebraisk) multiplicitet än motsvarande egenrums dimension; i så fall spänner inte egenrummen upp hela $\mathbb{R}^n$. Egenvärdena räknas upp i D enligt multipliciteten, men motsvarande kolonner i V är inte linjärt oberoende och V är inte

Ursprunglig datorlaboration författad av Carl-Henrik Fant, uppdaterad av Tony Stillfjord inverterbar. Produkten `inv(V)*A*V` ger också ett resultat, som ofta även blir D, men man får förhoppningsvis en varning av typen *Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 1.110223e-16.* `RCOND` står här för (en uppskattning av) *reciprocal condition number*, dvs. $1/\kappa(V)$ (se Lab 3!). Varningen säger alltså att MATLAB tror att konditionstalet är mycket stort. Egentligen är $\kappa(V) = \infty$ eftersom V inte är inverterbar, så det stämmer.

Uppgift 1. Bestäm egenvärden och egenvektorer till följande matriser. Undersök om matriserna är diagonaliserbara med reella matriser eller med komplexa matriser eller inte alls. Undersök också om den diagonaliserade matrisen är ortogonal eller ej.

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 3 & -1 & 0 \\ 5 & 4 & 0 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 5 & 4 & -2 \\ 4 & 5 & 2 \\ -2 & 2 & 8 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad A_4 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

□

Tillämpningar

Spänningsmatrisen

Spänningsmatrisen $\mathcal{S} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix}$ beskriver normalspänningar σ och skjuvspänningar τ i plan parallella med koordinatplanen, genom en kropp, ett kontinuum, som påverkas av inre och yttre krafter. Egenvektorer till matrisen $\mathcal{S}$ är huvudspänningsriktningarna och motsvarande egenvärden är normalspänningen i plan vinkelräta mot egenvektorn. I dessa plan är skjuvspänningen noll. Matrisen $\mathcal{S}$ är alltid symmetrisk och därmed alltid diagonaliserbar.

Spänningsvektorn s på en viss snittyta med enhetsnormalvektor n ges av $s = \mathcal{S}n$.

Normalspänningen på snittytan ges av längden av projektionen av spänningsvektorn på planets enhetsnormal, $\sigma = n^T\mathcal{S}n$. Skjuvspänningen på snittytan är längden av spänningsvektorns projektion på snittytan. Denna beräknas enkelt med Pythagoras sats: $\tau^2 = \|s\|^2 - \sigma^2$.

Om n är en normerad egenvektor till $\mathcal{S}$ så är $s = \mathcal{S}n = \lambda n$, där λ är motsvarande egenvärde. I det fallet är $\sigma = \|s\| = \lambda$ och $\tau = 0$. Därav begreppet huvudspänning och huvudspänningsriktning.

Uppgift 2. Spänningstillståndet i en punkt Q i en kropp har beräknats med finita-elementmetoden och uttrycks i ett kartesiskt koordinatsystem (x, y, z) med spänningsmatrisen

$$\mathcal{S} = \begin{bmatrix} 30 & 0 & 10 \\ 0 & 30 & 10 \\ 10 & 10 & 30 \end{bmatrix} \text{MPa.}$$

a Beräkna normal- och skjuvspänning på en snittyta med normalvektor $n = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 2 & 5 \end{bmatrix}^T$.

b Beräkna huvudspänningar och huvudspänningsriktningar.

□

System av differentialekvationer och matrisexponentialen

Ett enkelt sätt att lösa det vektorvärda systemet $x'(t) = Ax(t)$ är med hjälp av matrisexponentialfunktionen. För att något förstå dess definition utgår vi från diagonaliseringsmetoden. (Följande är repetition och utökning av resonemanget från föreläsning 12.)

Antag att A är en diagonaliserbar matris. Det innebär att det finns en matris P så att $P^{-1}AP = D$, där D är en diagonalmatris. Om vi nu gör variabelbytet $x = P\rho$ så erhåller vi $P\rho' = AP\rho$, dvs.

$$\rho' = P^{-1}AP\rho = D\rho.$$

Utskrivet blir det

$$\begin{cases} \rho_1'(t) = \lambda_1\rho_1(t) \\ \qquad \cdots \\ \rho_n'(t) = \lambda_n\rho_n(t) \end{cases}$$

Löser vi dessa ekvationer så får vi $\rho_i = c_i\mathrm{e}^{\lambda_i t}$, $i = 1, \ldots, n$, vilket kan skrivas $\rho = Bc$, med

$$c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \qquad \text{och diagonalmatrisen} \qquad B = \begin{bmatrix} \mathrm{e}^{\lambda_1 t} & & 0 \\ & \ddots & \\ 0 & & \mathrm{e}^{\lambda_n t} \end{bmatrix}.$$

Transformerar vi tillbaka så får vi $x = P\rho = PBc = P\begin{bmatrix} c_1\mathrm{e}^{\lambda_1 t} \\ \vdots \\ c_n\mathrm{e}^{\lambda_n t} \end{bmatrix} = c_1\mathrm{e}^{\lambda_1 t}v_1 + \cdots + c_n\mathrm{e}^{\lambda_n t}v_n$,

där v_i, $i = 1, \ldots, n$ är kolonnerna i P.

Antag nu att vi har begynnelsevillkoren $x_1(0) = x_{01}, \ldots, x_n(0) = x_{0n}$, eller kortare: $x(0) = x_0$. Då blir $x_0 = Pc$ (ty $B = I$ för $t = 0$), och därmed är $c = P^{-1}x_0$. Vi kan därför skriva $x(t) = PBP^{-1}x_0$.

Genom att skriva $\mathrm{e}^{\lambda_i t}$ som en Maclaurinserie, $\mathrm{e}^{\lambda_i t} = \sum_{k=0}^{\infty} \frac{1}{k!}(\lambda_i t)^k$ så inser man efter en smula eftertanke att diagonalmatrisen B kan skrivas som

$$B = \sum_{k=0}^{\infty} \frac{1}{k!}(tD)^k,$$

och eftersom $A^k = PD^kP^{-1}$ så får vi

$$PBP^{-1} = \sum_{k=0}^{\infty} \frac{1}{k!}(tA)^k.$$

Det är nu naturligt att göra följande definition:

Definition. Matrisexponentialen av matrisen tA ges av

$$\mathrm{e}^{tA} = \sum_{k=0}^{\infty} \frac{1}{k!}(tA)^k.$$

Matrisexponentialen är definierad även om A *ej* är diagonaliserbar, och det visar sig att det går att derivera som “vanligt”, dvs.

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{e}^{tA} = A\mathrm{e}^{tA} = \mathrm{e}^{tA}A.$$

Sammanfattningsvis: Differentialekvationen

$$x'(t) = Ax(t), \quad x(0) = x_0,$$

har lösningen

$$x(t) = \mathrm{e}^{tA} x_0,$$

där

$$\mathrm{e}^{tA} = \sum_{k=0}^{\infty} \frac{1}{k!} (tA)^k.$$

Om A är diagonaliserbar så är

$$\mathrm{e}^{tA} = PBP^{-1},$$

där

$$P = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \qquad \text{och} \qquad B = \begin{bmatrix} \mathrm{e}^{\lambda_1 t} & & 0 \\ & \ddots & \\ 0 & & \mathrm{e}^{\lambda_n t} \end{bmatrix},$$

och lösningen kan då skrivas

$$x = PBc = c_1 \mathrm{e}^{\lambda_1 t} v_1 + \cdots + c_n \mathrm{e}^{\lambda_n t} v_n,$$

där $Pc = x_0$.

Exempel 1: Vi löser

$$\begin{cases} x' = -4x + 6y, & x(0) = 7 \\ y' = -3x + 5y, & y(0) = 4 \end{cases}.$$

Här är $A = \begin{bmatrix} -4 & 6 \\ -3 & 5 \end{bmatrix}$, och vi får egenvärdena genom

$$0 = \det(A - \lambda I) = \begin{vmatrix} -4 - \lambda & 6 \\ -3 & 5 - \lambda \end{vmatrix} = \lambda^2 - \lambda - 2 = (\lambda + 1)(\lambda - 2),$$

dvs. $\lambda_1 = -1$, $\lambda_2 = 2$. Motsvarande egenvektorer blir $v_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, $v_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Alltså är

$$x = c_1 \mathrm{e}^{-t} \begin{bmatrix} 2 \\ 1 \end{bmatrix} + c_2 \, \mathrm{e}^{2t} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Från begynnelsevärdet $x(0) = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$ får vi

$$\begin{bmatrix} 7 \\ 4 \end{bmatrix} = Pc = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix},$$

så

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Lösningen blir således

$$\begin{cases} x(t) = 6\mathrm{e}^{-t} + \mathrm{e}^{2t} \\ y(t) = 3\mathrm{e}^{-t} + \mathrm{e}^{2t} \end{cases}.$$

Anm. Här är

$$\mathrm{e}^{tA} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{e}^{-t} & 0 \\ 0 & \mathrm{e}^{2t} \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 2\mathrm{e}^{-t} - \mathrm{e}^{2t} & -2\mathrm{e}^{-t} + 2\mathrm{e}^{2t} \\ \mathrm{e}^{-t} - \mathrm{e}^{2t} & -\mathrm{e}^{-t} + 2\mathrm{e}^{2t} \end{bmatrix}$$

och

$$x(t) = \mathrm{e}^{tA} \begin{bmatrix} 7 \\ 4 \end{bmatrix} = \begin{bmatrix} 6\mathrm{e}^{-t} + \mathrm{e}^{2t} \\ 3\mathrm{e}^{-t} + \mathrm{e}^{2t} \end{bmatrix}.$$

Vi skall nu låta MATLAB använda denna metod för att lösa system av differentialekvationer. I MATLAB skrivs e^{tA} som `expm(t*A)`. (Glöm inte m:et då `exp(t*A)` ger något helt annat!) Då `expm` inte kan räkna med objekt större än matriser, så måste vi räkna ut e^{tA} för ett t-värde i taget.

Exempel 2: Vi löser systemet

$$\begin{cases} x_1' = x_1 + x_2, & x_1(0) = 1 \\ x_2' = x_2 - x_1, & x_2(0) = -2 \end{cases}.$$

i intervallet $t \in [0, 10]$ på följande sätt:

```
A = [1, 1; -1, 1];
x0 = [1; -2];
% 400 equidistant points between (inclusive) 0 and 10:
t = linspace(0, 10, 400);
x = zeros(2, length(t));  % a 2-vector for each t-value
for i = 1:length(t)
  x(:,i) = expm(t(i)*A)*x0;
end
plot(t, x(1,:), t, x(2,:)) % plot x1 and x2 as two curves
```

Skriv in koden ovan som en skript-fil (kommentarer kan såklart utelämnas) och testa att den fungerar! Att kopiera in från PDF-filen går förmodligen, men ibland blir vissa tecken feltolkade, såsom t.ex. minustecknen.

Uppgift 3.

a Tag i tur och ordning $A = A_1, A_2, A_3, A_4$ från Uppgift 1 och lös systemen av differentialekvationer $x'(t) = Ax(t)$, $x(0) = \begin{bmatrix} 1 & 2 & -1 \end{bmatrix}^T$ med hjälp av `expm(t*A)` enligt ovan. Rita grafen till lösningskurvorna. Lös också systemen med diagonaliseringsmetoden och jämför resultaten. Tänk lite extra på A_4.

b Gör om ekvationen $y'' + 2y' + 2y = 0$, $y(0) = 1$, $y'(0) = -2$ till ett system av första ordningens differentialekvationer med hjälp av substitutionen $y_1 = y$, $y_2 = y'$. Lös detta med matrisexponentialfunktionen. Rita lösningskurvan.

$\square$