Institutionen för datavetenskap
Umeå Universitet
Martin Berggren

6 november 2008
Numeriska metoder för civilingenjörer
Laboration

Linjära system för nätverk

Redovisning

Rapporten förväntas innehålla följande

- Försättsblad med namn, datum, sökväg till källkoden, samt användarnamn
- Väl förklarade läsningar på alla ingående deluppgifter
- Kommenterade resultat med egna reflektioner och slutsatser
- Välformatterad källkod
- Kommentar om labben: rolig/tråkig, svår/lätt, oklarheter.

Vid komplettering (betyget O) lämnas originalrapporten in tillsammans med modifierade bitar av rapporten/koden. Ett nytt försättsblad måste skrivas ut. Rapporten lämnas i facket, märkt med kursens namn och kurskod, utanför institutionen.

Uppgiften löses **enskilt** och inlämnas senast 13 november 2008, klockan 13:15.

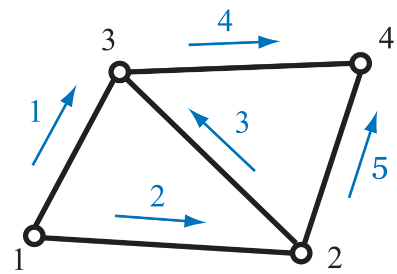
Inledning

Olika typer av *nätverksproblem* återkommer gång på gång i många olika tillämpningar och leder till linjära ekvationssystem med särskilda egenskaper. Figur 1 visar ett exempel på ett litet nätverk. Nätverk kan beskrivas matematiskt med hjälp av *grafteori*[1]. Ett *graf* består av *noder* (kallas även *hörn*) förbundna med *kanter* (kallas även *bågar*). Nätverket i figur 1 är en *riktad graf*[2] som skulle kunna representera ett elektriskt nätverk där det flyter ström genom kanter innehållande elektriska komponenter typ motstånd, kondensatorer, induktorer eller elektriska källor. Eller så skulle grafen kunna representera transportvägar för varor mellan olika städer. Exempel på andra system som kan beskrivas med nätverk är bärverk av stänger, sammankopplade datorer samt sammanlänkade websidor.

I denna lab ska vi behandla nätverk av *vattenledningar*. Vi tänker oss att vart och ett av de fem numrerade kanterna i figur 1 representerar rörledningar genom vilka det kan strömma vatten. Vidare kan man tappa av eller fylla på vatten i var och en av de 4 noderna. Antag nu att v vet vilka vattenflöden som man kommer att tappa av och fylla på i de olika noderna och att vi vill bestämma vilka vattentryck som detta kräver vid de olika noderna. Det är en relevant fråga t ex för att dimensionera pumpar och ledningar. Under vissa förenklade men inte helt verklighetsfrämmande fysikaliska antaganden ska vi beskriva en generell metodik för att lösa detta problem.

Pilarna i figur 1 indikerar teckenkonventionen för positivt flöde genom ledningarna: $u_1 > 0$ betyder att flödet u_1 (i m^3/s) strömmar genom ledning 1 från nod 1 till nod 3, medan $u_1 < 0$ indikerar flöde från nod 3 till nod 1. Vidare vet vi att flödet s_i (i m^3/s) tappas av eller tillföras i nod i: $s_i > 0$ betyder att vatten tillförs och $s_i < 0$ att vatten tappas av. Vi antar att det inte

[1] Se t ex Wikipedias artikel http://en.wikipedia.org/wiki/Graph_(mathematics)
[2] `http://en.wikipedia.org/wiki/Directed_graph`



Figur 1: Ett exempel på ett nätverk i form av en riktad graf med 4 noder och 5 kanter

finns några läckor i systemet. Det innebär att det flöde $-s_i$ som tappas *av* i nod i måste vara like med summan det som strömmar *in* via kanterna, vilket ger ekvationerna

$$\begin{aligned} -u_1 - u_2 &= -s_1 && \text{(vid nod 1)} \\ u_2 - u_3 - u_5 &= -s_2 && \text{(vid nod 2)} \\ u_1 + u_3 - u_4 &= -s_3 && \text{(vid nod 3)} \\ u_4 + u_5 &= -s_4 && \text{(vid nod 4).} \end{aligned} \tag{1}$$

Observera att om det inte finns några läckor måste det också gälla att allt som går in i ledningssystemet också kommer ut:

$$s_1 + s_2 + s_3 + s_4 = 0. \tag{2}$$

Ekvation (1) kan skrivas i matrisformen

$$Bu = -s, \tag{3}$$

där

$$B = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix}, \quad s = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} \tag{4}$$

Matrisen B kallas för den *riktade grafens incidensmatris*[3] och ger en fullständig beskrivning om hur nätverket är kopplat, dess "topologi" (däremot säger den inte någonting om de absoluta positionerna för noderna. Sådan information behövs inte för detta problem). En incidensmatris för en riktad graf har lika många rader som noder i grafen och lika många kolonner som kanter (ledningar) i grafen. Varje kolonn representerar således en kant och har värdena $+1$ och -1 på de rader som motsvarar de noder där det flödar in respektive ut från aktuell kant.

Vi betecknar trycket i nod i med p_i (i Pa). Flödet genom varje kant beror på tryckskillnaden mellan de noder som kanten förbinder. Vi antar ett enkelt linjärt samband, så att flödet är direkt proportionell mot tryckskillnaden:

$$\begin{aligned} u_1 &= \epsilon_1(p_1 - p_3) && \text{(ledning 1)} \\ u_2 &= \epsilon_2(p_1 - p_2) && \text{(ledning 2)} \\ u_3 &= \epsilon_3(p_2 - p_3) && \text{(ledning 3)} \\ u_4 &= \epsilon_4(p_3 - p_4) && \text{(ledning 4)} \\ u_5 &= \epsilon_5(p_2 - p_4) && \text{(ledning 5),} \end{aligned} \tag{5}$$

[3] http://en.wikipedia.org/wiki/Incidence_matrix

där koefficienterna $\epsilon_i \geq 0$ anger ledningens "konduktivitet"; en egenskap som till exempel kan fastställas genom mätningar. Om ekvation (5) (som är en motsvarighet till kretsteorins Ohms lag) skrivs i matrisform,

$$u = -EB^{\mathsf{T}}p, \tag{6}$$

där

$$E = \begin{pmatrix} \epsilon_1 & 0 & 0 & 0 & 0 \\ 0 & \epsilon_2 & 0 & 0 & 0 \\ 0 & 0 & \epsilon_3 & 0 & 0 \\ 0 & 0 & 0 & \epsilon_4 & 0 \\ 0 & 0 & 0 & 0 & \epsilon_5 \end{pmatrix}, \quad p = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix}, \tag{7}$$

ser vi att (transponatet till) incidensmatrisen också kommer in i tryck–flödesrelationen.

Sammanfattningsvis ger således modelleringen av flödesnätverket ekvationerna

$$Bu = -s, \quad u = -EB^{\mathsf{T}}p, \qquad s_1 + s_2 + s_3 + s_4 = 0. \tag{8}$$

För att förenkla modelleringen ytterligare skall vi betrakta en situation där trycket i nod 4 hålles konstant $p_4 = 0$. De obekanta trycken blir då p_1, p_2, p_3 och dessa betecknar då övertryck m a p trycket i nod 4, i analogi med hur man i elektrisk kretsteori "jordar"en punkt till nollpotential. Vidare så betraktas s_1, s_2 och s_3 som givna och $s_4 = -s_1 - s_2 - s_3$ i enlighet med ekvation (8). Denna situation modelleras genom att den fjärde raden i matrisen B stryks (vi vet ju redan vad som händer vid nod 4!) och vi tar bort s_4 och p_4 från problemet (dessa är kända). Då blir det en ekvation mindre (i $Bu = -s$) och en obekant mindre (i $u = -EB^{\mathsf{T}}p$), och vi får

$$\bar{B}u = -\bar{s} \tag{9a}$$

$$u = -E\bar{B}^T\bar{p}, \tag{9b}$$

där

$$\bar{B} = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 1 & -1 & 0 \end{pmatrix}, \quad \bar{s} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} \quad \bar{p} = \begin{pmatrix} p_1 \\ p_2 \\ p_3. \end{pmatrix} \tag{10}$$

Om man multiplicerar ekvation (9b):s bägge sidor från vänster med $-\bar{B}$ och adderar den resulterande ekvationen till ekvation (9a) erhålls följande ekvation endast i $\bar{p}$:

$$\bar{B}E\bar{B}^T\bar{p} = \bar{s} \tag{11}$$

Uppgifter

1. Specificera explicit ekvationssystemet (11) för nätverket i figur 1.

2. Skriv en Matlabfunktion som beräknar trycken i noderna av ett givet godtyckligt nätverk genom att sätta upp och lösa ekvationssystemet (11). Funktionshuvudet skall ha utseendet

   ```
   function p = computepressure(B, e, s, k0)
   ```

   Inparametrar är en incidensmatris B för en godtycklig riktad graf, en vektor $e = (\epsilon_1, \ldots, \epsilon_m)$ med konduktivitetsparametrar för alla kanter i grafen (längden på e skall vara lika med antalet kolonner i B), en vektor s (längden på s skall vara lika med antalet rader i B) med volymflöden vid varje nod samt ett heltal k_0 som talar om vilket nod som skall "jordas", d v s det skall gälla att $p_{k_0} = 0$.

Tips: Matlabfunktionen `diag` (läs dokumentationen!) är bra att ha!

Testa ditt program på nätverket i figur 1. Ett test kan t ex vara att välja några lätta värden på ϵ_k-parametrarna, räkna ut matrisen $\bar{B}E\bar{B}^T$ för hand och jämföra med matrisen som `computepressure` räknar fram. Redovisa hur du verifierat att programmet är korrekt.

3. Hitta på och rita ett valfritt nätverk med minst 10 noder, bilda incidensmatrisen för den riktade grafen och lös problemet (11) för ditt nätverk med värdet 1 på alla konduktiviteter och värdet 1 på alla källor förutom den vid det "jordade" noden (som kan vara vilken nod som helst).

4. Med ledning av utseendet de matriser som du erhållit i problem 1 och 3, slut dig till en regel för hur många nollskilda element det finns på en viss rad i matrisen $\bar{B}E\bar{B}^T$. (Antalet beror på utseendet på grafen vid aktuell nod.) Extra utmaning: bevisa påståendet!

5. Modifiera din kod så att den löser problemet (11) med B istället för $\bar{B}$ och s istället för s, d v s utan att "jorda" någon nod. Vad händer när du försöker läsa problemet? Sätt in vektorn $p = (1, \ldots, 1)^T$ i ekvationssystem och kolla residualen med högerledet $s = (0, \ldots, 0)^T$. Förklara vad som är problemet!

6. Antag nu att vi vill studera mycket större nätverk; sådana som t ex kan representera vattenledningssystemet i ett bostadsområde. På kurshemsidan finns filerna `Bc0.mat` och `Bc1.mat` som innehåller incidensmatriser av dimensionerna 399×1146 respektive 1545×4536. Matlabkommandot `load Bc0` respektive `load Bc1` läser in dessa matriser. Använd din kod för att lösa problem (med valfria parametrar i övrigt) med dessa två incidensmatriser. Ta tid, med hjälp av kommandona `tic` och `toc`, hur lång tid lösningen av ekvationssystemet med Matlabs backslash tar för incidensmatrisen given i `Bc0.mat`. Med hjälp av komplexitetsuppskattningen för gausselimination, uppskatta hur lång tid t_f varje flyttalsoperation tar. Med hjälp av t_f, uppskatta hur lång tid gausselimination när incidensmatrisen given i `Bc1.mat` används. Jämför med den tid det verkligen tar!

7. Nätverksproblem ger upphov till stora, glesa ekvationssystem. Det finns specialiserade algoritmer för sådana system som är mycket effektivare och minnessnålare än de metoder vi studerar i denna kurs. (Dessutom är det egentligen inte så klokt att explicit arbeta med incidensmatriser för mycket stora grafer: dessa innehåller ju mest nollor!) Algoritmer för glesa matriser finns tillgängliga i Matlab. Då skall matriserna lagras i ett speciellt *glest format*, då endast nollskillda element lagras. Du kan läsa mer om detta i dokumentationen för kommandot `sparse`. Om man redan har en matris `K` i "vanligt" format kan man omvandla den till glest format med kommandot `K=sparse(K)`. Matlabs backslashoperator känner igen om den ingående matrisen är definierad som en gles matris och väljer i så fall den specialiserade algoritmen. För bägge matriserna i uppgift 6, jämför tidsåtgången i lösningen av det linjära ekvationssystemet när `sparse(K)` används istället för `K`, där `K` är matrisen $\bar{B}E\bar{B}^T$. Tjänar man på att använda dessa speciella metoder?