

Esercitazione 6

Istruzioni trattate: `size`, `find`, `diag`, `list`.

Nella prima parte di questa esercitazione vedremo una realizzazione delle procedure SI, *sostituzione all'indietro* ed SA, *sostituzione in avanti*. Nella seconda parte descriveremo un'applicazione del procedimento di ricerca della soluzione di un sistema di equazioni lineari che utilizza la procedura EGP: *la realizzazione di un simulatore di circuiti lineari di soli resistori e generatori indipendenti di corrente*.

Prima parte

La definizione che segue è una realizzazione della procedura SI.

```
function x = SI(T,c)
//
// Per T matrice N x N triangolare superiore invertibile e c colonna di N
// componenti, calcola un'approssimazione x della colonna soluzione del sistema
// di equazioni lineari T z = c.
//
N = size(T,'r');
//
// *** Verifica se T invertibile.
//
if find(diag(T)==0) ~= [] then
    error("elementi nulli sulla diagonale");
else
    //
    // *** Calcolo soluzione
    //
    x = zeros(N,1);
    //
    x(N) = c(N) / T(N,N);
    for k = N-1:-1:1,
        x(k) = (c(k) - T(k,k+1:N)*x(k+1:N)) / T(k,k);
    end;
end;
endfunction
```

In questa realizzazione si è scelto di verificare i dati che l'utilizzatore fornisce in ingresso alla procedura *solo* riguardo all'invertibilità della matrice. In particolare, è responsabilità dell'utilizzatore che il primo argomento di ingresso sia una matrice *quadrata* e *triangolare superiore* e il secondo sia una colonna della *dimensione corretta*. Se il primo argomento è una matrice con qualche elemento nullo sulla diagonale la *funzione predefinita* `error` arresta l'esecuzione della procedura.

Nella definizione compaiono tre nuove istruzioni:

- `size`

Questa *funzione predefinita* restituisce il numero di righe e di colonne della matrice argomento. Precisamente, se A è una matrice ad elementi in M :

`size(A)`

restituisce una riga di due numeri reali $[r,c]$: il primo è il numero di righe, il secondo il numero di colonne di A ,

`size(A,'r')`

restituisce un numero reale: il numero di righe di A ,

`size(A,'c')`

restituisce un numero reale: il numero di colonne di A .

- **find**

Questa *funzione predefinita* ha lo scopo di individuare le componenti di valore %T in una matrice a componenti di *tipo booleano*.¹ Precisamente, se v e M sono, rispettivamente, un vettore ed una matrice a componenti di *tipo booleano*:

$$r = \text{find}(v)$$

individua gli indici i_1, \dots, i_k delle componenti di v di valore %T ed assegna ad r la riga $[i_1, \dots, i_k]$, oppure la matrice vuota [] se tutte le componenti di v hanno valore %F,

$$[r, c] = \text{find}(M)$$

individua gli indici $(r_1, c_1) \dots, (r_k, c_k)$ delle componenti di M di valore %T ed assegna ad r la riga $[r_1, \dots, r_k]$ ed a c la riga $[c_1, \dots, c_k]$, oppure la matrice vuota [] se tutte le componenti di M hanno valore %F. Ad esempio:

```
-->v = [1,0,2,3,0]; r = find(v > 1)
r =
    3.    4.

-->M = [1,0,2;3,0,-1]; [r,c] = find(M <= 0)
c =
    2.    2.    3.
r =
    1.    2.    2.
```

- **diag**

Questa *funzione predefinita*, applicata ad una matrice a componenti in M , restituisce il vettore colonna di componenti gli elementi sulla diagonale della matrice. Ad esempio:

```
-->A = [1,2,3;4,5,6]
A =
    1.    2.    3.
    4.    5.    6.

-->diag(A)
ans =
    1.
    5.
```

Una realizzazione della procedura SA si ottiene variando opportunamente la definizione precedente (Esercizio 2).

Seconda parte

Si consideri un circuito lineare costituito da NR resistori e NG generatori indipendenti di corrente. Numerati i nodi del circuito da 0 (*nodo di riferimento*) a N , indichiamo con E_k la tensione del nodo k rispetto al *nodo di riferimento* (nodo 0) e, numerati i componenti del circuito da 1 a $R = NR + NG$ e fissato per ciascuno di essi un *orientamento*, indichiamo con V_k la tensione ai capi del k -esimo componente e con I_k la corrente che scorre nel k -esimo componente, con la polarità ed il verso concordi all'orientamento fissato.

¹Ovvero a valori in $\{\%T, \%F\}$.

Applicando la *Legge di Kirchhoff delle correnti* ai nodi dal numero 1 al numero N ed utilizzando poi per le correnti nei resistori la *Legge di Ohm* nella forma:

$$\text{corrente nel resistore} = \text{conduttanza} \cdot \text{tensione ai capi del resistore}$$

si ottiene un sistema di N equazioni lineari nelle N incognite E_1, \dots, E_N (sistema delle *equazioni nodali*). Risolto questo sistema di equazioni, determinate quindi le tensioni dei nodi rispetto al *nodo di riferimento*, le rimanenti grandezze elettriche del circuito, le tensioni ai capi di ciascun componente V_1, \dots, V_R e le correnti in ciascun componente I_1, \dots, I_R , sono determinate in modo immediato.

Vogliamo utilizzare *Scilab* per ottenere approssimazioni dei valori di E_1, \dots, E_N , V_1, \dots, V_R e I_1, \dots, I_R .

Il primo problema da affrontare è la *descrizione del circuito*. Per descrivere (elettricamente) il circuito è necessario e sufficiente specificare, per ciascuno dei componenti, *il tipo* (resistore o generatore), *il valore*, *i due nodi* a cui è connesso e, nel caso di un generatore, *l'orientazione*.

Una descrizione equivalente, facilmente gestibile in *Scilab*, si ottiene definendo le matrici **RES**, di dimensione $NR \times 3$ e contenente le informazioni relative ai resistori, e **GEN**, di dimensione $NG \times 3$ e contenente le informazioni relative ai generatori, con righe della forma:

$$\langle \text{valore} \rangle , \langle \text{nodo 1} \rangle , \langle \text{nodo 2} \rangle$$

I componenti del circuito sono numerati dall'ordine delle corrispondenti righe nelle matrici, considerando prima **RES** e poi **GEN** e l'orientazione di ciascun componente è *dal nodo 1 al nodo 2*. Ad esempio, il circuito mostrato in Figura 1 è descritto dalle matrici:

$$\text{RES} = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 3 & 2 & 0 \\ 4 & 0 & 4 \end{bmatrix} , \quad \text{GEN} = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 3 & 4 \end{bmatrix}$$

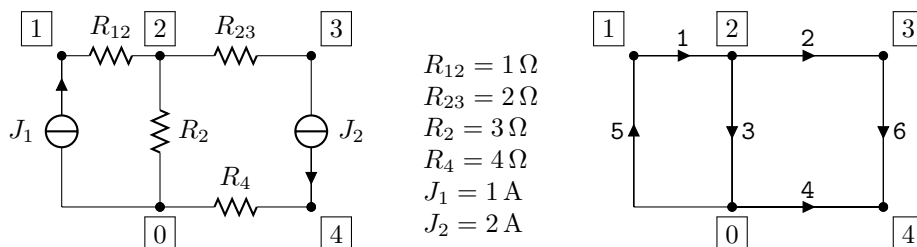


Figura 1: Esempio di circuito

Stabilito come descrivere il circuito, il secondo problema da affrontare è la *scrittura delle equazioni nodali*. Per il circuito dell'esempio, dette G_{ij} la *conduttanza* del resistore connesso tra il nodo i ed il nodo j e G_k quella del resistore connesso tra il nodo k ed il *nodo di riferimento*, il sistema risulta:

$$\begin{bmatrix} G_{12} & -G_{12} & 0 & 0 \\ -G_{12} & G_{12} + G_{23} + G_2 & -G_{23} & 0 \\ 0 & -G_{23} & G_{23} & 0 \\ 0 & 0 & 0 & G_4 \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{bmatrix} = \begin{bmatrix} J_1 \\ 0 \\ -J_2 \\ J_2 \end{bmatrix}$$

La k -esima equazione del sistema traduce la legge di Kirchhoff delle correnti applicata al nodo k nella forma:

$$\begin{array}{l} \text{somma delle correnti uscenti} \\ \text{dal nodo tramite resistori} \end{array} = \begin{array}{l} \text{somma delle correnti entranti} \\ \text{nel nodo tramite generatori} \end{array}$$

Un'osservazione attenta del sistema ottenuto e del procedimento utilizzato per ottenerlo mostra che la matrice dei coefficienti e la colonna dei termini noti possono essere ottenuti con un *procedimento sistematico* che analizza una per una le componenti del circuito. Precisamente, la matrice dei coefficienti **G** e la colonna dei termini noti **J** del sistema si costruiscono con la procedura seguente:

G = matrice nulla di dimensione $N \times N$;

J = colonna nulla di N componenti;

per $k = 1, \dots, NR + NG$ ripeti:

se il componente k è un resistore di conduttanza g connesso tra i nodi i e j allora

se $i \neq 0$ e $j \neq 0$ allora: $G_{ii} = G_{ii} + g$; $G_{jj} = G_{jj} + g$;

$G_{ij} = G_{ij} - g$; $G_{ji} = G_{ji} - g$;

se $j = 0$ allora: $G_{ii} = G_{ii} + g$;

se $i = 0$ allora: $G_{jj} = G_{jj} + g$;

se il componente k è un generatore di valore s connesso tra i nodi i e j allora

se $i \neq 0$ e $j \neq 0$ allora: $J_i = J_i - s$; $J_j = J_j + s$;

se $j = 0$ allora: $J_i = J_i - s$;

se $i = 0$ allora: $J_j = J_j + s$;

Una realizzazione di questa procedura è data nella `function` di intestazione:

```
function [G,J] = EquazioniNodali(Rete)
```

contenuta nel file `Esercitazione_06.sce`.² L'argomento di ingresso, `Rete`, deve essere una *lista* contenente le due matrici che descrivono il circuito in esame, nell'ordine: `RES`, `GEN`. Per il circuito riportato in Figura 1 la lista è generata nel file `ReteEs.sce`:

```
// valore dei resistori (in Ohm):
R12 = 1;
R23 = 2;
R2 = 3;
R4 = 4;
// valore dei generatori di corrente (in A):
J1 = 1;
J2 = 2;
//      valore , nodo 1 , nodo 2
RES = [  R12 ,      1,      2;
        R23 ,      2,      3;
        R2  ,      2,      0;
        R4  ,      0,      4];
//
GEN = [  J1,      0,      1;
        J2,      3,      4];
//
//
ReteEs = list(RES, GEN);
```

Nel file, per rendere più chiara la costruzione delle matrici, si è scelto di definire una variabile per ciascun elemento. La *lista* contenente la descrizione del circuito è creata dal comando finale `list`.

- `list`

Questa *funzione predefinita* ha lo scopo di creare una *lista* di oggetti. Ad esempio:

```
-->L = list('1', '2', 3, [4,1;3,0], 5)
```

```
L =
```

```
    L(1)
```

```
    1
```

```
    L(2)
```

²Tutti i file citati possono essere scaricati dalla pagina web del corso nella sezione **altro materiale didattico**.

2

L(3)

3.

L(4)

4. 1.

3. 0.

L(5)

5.

Dopo l'assegnamento, il comando $L(i)$ restituisce l' i -esimo oggetto della lista:

```
-->L(4)
```

```
ans =
```

```
4. 1.
```

```
3. 0.
```

Per estrarre più di un oggetto dalla lista si utilizza il comando $L([i_1, \dots, i_k])$ in cui gli elementi da estrarre sono individuati dal *vettore* dei loro indici. Si osservi che il comando restituisce k oggetti che *devono* essere assegnati ad altrettante variabili:

```
-->[ogg1,ogg4] = L([1,4])
```

```
ogg4 =
```

```
4. 1.
```

```
3. 0.
```

```
ogg1 =
```

```
1
```

```
-->L([1,4])
```

```
!--error 41
```

L'argomento in uscita è incompatibile.

L'assegnamento $\text{ans} = L([1,4])$ implicito nell'ultimo comando è *illegale* perché, in questo caso, si pretende di assegnare due oggetti ad una sola variabile. Se si vogliono estrarre *tutti* gli elementi della lista si può utilizzare anche il comando $L(:)$.

Il terzo problema da affrontare è *risolvere il sistema delle equazioni nodali*. Questo problema viene risolto con il procedimento che utilizza le procedure EGP (per la fattorizzazione della matrice del sistema), SA (per la risoluzione del sistema con matrice triangolare inferiore) ed SI (per la risoluzione del sistema finale con matrice triangolare superiore). La realizzazione in *Scilab* che si è scelto di utilizzare è la seguente (G e J sono, rispettivamente, la matrice ed il termine noto delle equazioni nodali relative al circuito in esame):

```
// (1) fattorizzazione di G
[S,D,P] = lu(G);
// (2) soluzione del sistema S x = PJ
c = SA(S, P*J);
// (3) soluzione del sistema D x = c
E = SI(D,c);
```

La *funzione predefinita* `lu` realizza la procedura EGP.

L'ultimo problema da affrontare è *determinare le rimanenti grandezze elettriche per ciascun componente del circuito*. Una volta nota la tensione di ciascun nodo rispetto al *nodo di riferimento*, è immediato determinare la tensione ai capi di ciascun componente: se il componente k è orientato dal nodo i al nodo j , la tensione V_k ai suoi capi si ottiene come *differenza* tra la tensione del nodo i e la tensione del nodo j . Poi, la corrente che scorre in ciascuno dei generatori è pari al valore del generatore, quella che scorre in un resistore è data dal prodotto del valore della conduttanza del resistore per la tensione ai suoi capi.

La `function` di intestazione:³

```
function [E,V,I] = SoluzioneRete(Rete)
```

realizza quanto ci eravamo proposti: determina un'approssimazione delle grandezze elettriche relative al circuito descritto nella *lista Rete*: E è il vettore delle *tensioni dei nodi rispetto al nodo di riferimento*, V è il vettore delle *tensioni ai capi dei componenti* e I è il vettore delle *correnti che scorrono nei componenti*.

³Anch'essa contenuta nel file `Esercitazione_06.sce`.

Esercizi

1. Discutere il seguente dialogo:

```
-->A = ones(4,4); b = [1;0;0;0];

-->x = SI(A,b)
x =

    1.
    0.
    0.
    0.

-->A * x == b
ans =

    T
    F
    F
    F
```

2. Modificare opportunamente la procedura SI per realizzare una procedura di intestazione

```
function x = SA(T,c)
```

che restituisce, se T è una matrice triangolare inferiore invertibile e c una colonna della dimensione opportuna, una colonna x approssimazione della soluzione del sistema di equazioni lineari $Tx = c$.

3. Simulare il circuito di Figura 2 con i componenti numerati ed orientati come indicato.

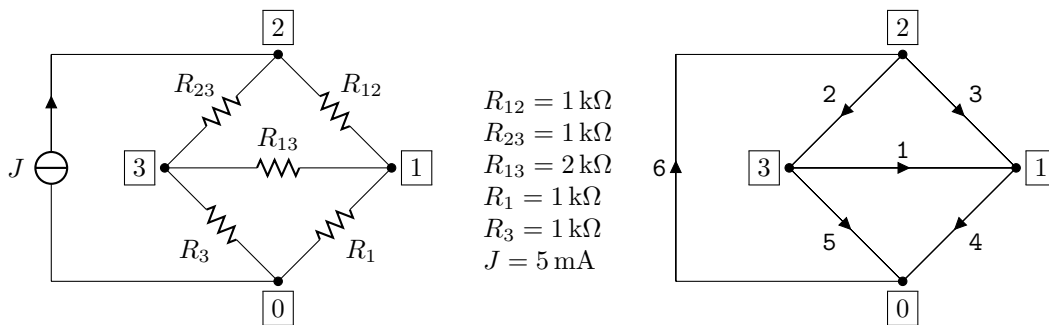


Figura 2: Circuito dell'Esercizio 3.

Constatare che la tensione ai capi del generatore di corrente è $V_6 = -5$ V, la corrente che scorre nel resistore R_{13} è 0 A e verificare che la *potenza* complessivamente dissipata dai resistori è 25 mW.

4. Il file `MisuraTensione.sci` contiene la definizione della `function` di intestazione:

```
function v = MisuraTensione(N1,N2,Rete,R)
```

che restituisce la misura v della tensione tra il nodo N1 ed il nodo N2 del circuito descritto nella lista `Rete` rilevata con un voltmetro schematizzato come un resistore di resistenza R.⁴

⁴Più correttamente: *un'approssimazione* di tale misura.

Si osservi che l'inserimento del voltmetro *modifica* il circuito. La procedura realizzata dalla funzione `MisuraTensione`: (a) modifica la *lista Rete* inserendo nella prima matrice la riga corrispondente al resistore che schematizza il voltmetro ottenendo così la descrizione del *circuito contenente il voltmetro*, (b) determina le tensioni di nodo del circuito contenente il voltmetro e (c) calcola la differenza tra la tensione del nodo `N1` e quella del nodo `N2`.

Determinare la misura della tensione tra il nodo 2 ed il nodo 0 del circuito dell'Esercizio 3, effettuata con un voltmetro schematizzato con un resistore di resistenza $1\text{ M}\Omega$ e confrontare il risultato con il valore della differenza tra la tensione del nodo 2 e quella del nodo 0 del circuito *senza voltmetro*. Ripetere l'esercizio effettuando la misura con un *voltmetro ideale*, ovvero schematizzato con un resistore di resistenza *infinito* (`%inf`).