

```
%{
    Soluzione di una rete quadratica (dipendenza della caduta di potenziale dal
    quadrato della corrente). In questo programma NON si utilizzano gli
    strumenti "Graph and Networks Algorithms" di Matlab (visualizzazione a
    parte)
%}
clear all
clc
close all

% Large number
bigg = 1/eps(1)/1e5;

% Tollerance
toll = 1e-8;

% Max nr of iterations
maxIter = 1000;

% Relaxation factor for the fixed-point iteration
rel = 0.5;

% Tabella connessione nodi-rami
T = [1 2;2 3;3 4;4 1;2 4];

% Numero nodi e rami
N = max(T(:));
M = size(T,1);

% Coefficiente di perdita nei rami:  $\Delta \Phi = \xi \cdot I^2$ 
xi = [0.1;0.2;0.12;0.3;0.5];

% Resistenze rami (definite come "anonimpous functions")
R = @(Phi,xi,C) 1./sqrt(xi.*abs(C*Phi));

% Potenziale imposto ai nodi
Phi0 = [0;nan;nan;nan];
nD = find(~isnan(Phi0));
nnD = find(isnan(Phi0));

% Corrente estratta dai nodi (non viene utilizzato per i nodi a potenziale imposto);
In = [0;0;1;1.3];

% Matrice di connettività
C = sparse(M,N);
C(sub2ind([M,N],[1:M]',T(:,1))) = 1;
C(sub2ind([M,N],[1:M]',T(:,2))) = -1;

% Soluzione iniziale
Phi = rand(N,1);

% Apro figura per plot del residuo
hr = figure;

% Fixed-point loop  $x(n+1) = f(x(n))$ , con  $f = \text{inv}(A(x(n))) * b$ 
```

```

err = toll + 1;
iter = 0;
while (err>toll)|| (iter>maxIter)

    % Update iteration number
    iter = iter + 1;

    % Matrice coefficienti
    A = (C'*spdiags(1./R(Phi,xi,C),0,M,M))*C;

    % Vettore termini noti
    b = -In;

    % Residuo (NO calcolato su nodi con potenziale imposto)
    err = norm(A(nnD,:) *Phi-b(nnD))/norm(b(nnD));

    % Imposto potenziale noto, ma prima salvo matrice non modificata
    As = A;
    A(sub2ind([N,N],nD,nD)) = bigg;
    b(nD) = bigg*Phi0(nD);

    % Soluzione aggiornata (attenzione a non sotto-rilassare i nodi a
    % potenziale imposto: possibili problemi se uno si dimentica di imporre
    % il carico imposto anche sulla soluzione di primo tentativo!)
    tmp = A\b;
    Phi(nD) = tmp(nD);
    Phi(nnD) = (1-rel)*Phi(nnD) + rel*tmp(nnD);

    % Corrente estratta dai nodi (non è necessaria nel corso
    % dell'iterazione: la calcolo qui per completezza, ma a tutti gli
    % effetti appesantisce il loop e basta)
    I = -As*Phi;

    % Plot del residuo
    figure(hr);
    h = loglog(iter,err,'ro'); if iter==1, hold on; grid on; end;

end

% Corrente nei rami
Ib = spdiags(1./R(Phi,xi,C),0,M,M)*C*Phi;

% Plot graph
figure
G=digraph(T(:,1),T(:,2),Ib);
h=plot(G,'EdgeLabel',G.Edges.Weight,'NodeLabel',Phi);
set(h,'linewidth',2.0);

```