《高等电力网络分析》 2020 年 第一次课程作业

姓名：蔡啸

学号：2020310612

班级：电博201

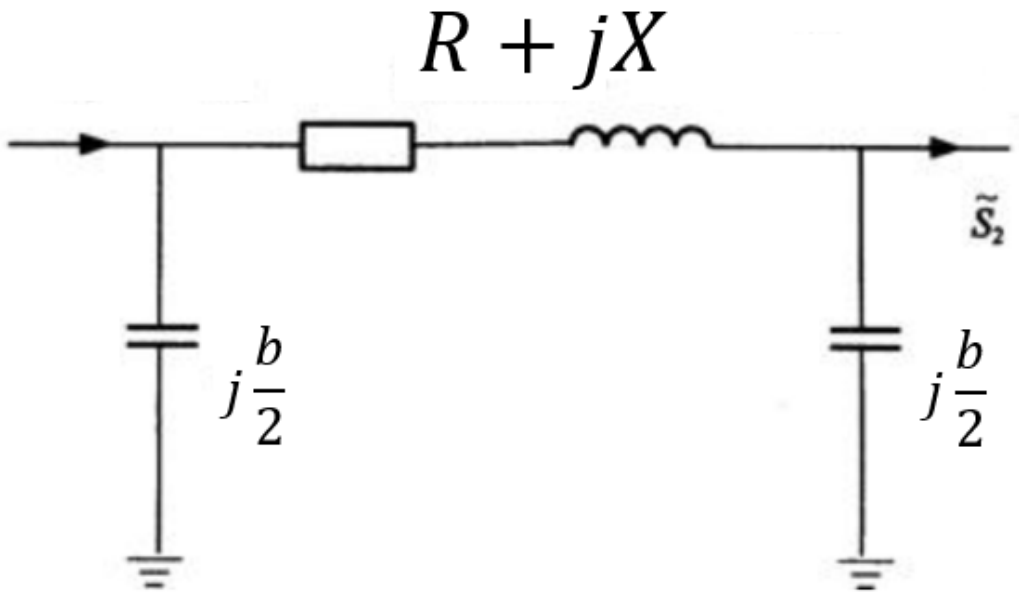
1、 本题基于 Matpower 7.1 中的 IEEE 14 节点系统（case14.m 文件），需要回答以下问题。

备注：注意在求解下列问题时，不可直接使用 matpower 的程序。

（1） 不考虑变压器变比， 生成 IEEE 14 节点系统的节点不定导纳矩阵Y_0 ，观察 case14.m 文件，思考“地节点”在何处体现了？基于Y_0 生成系统的导纳矩阵Y

1.1.1 生成节点不定导纳矩阵Y_0

首先考虑支路的接地电纳与节点接地导纳，将支路的电纳分别分到与支路连接的节点上，视为导纳为$\frac{b}{2}$的接地支路。



并将节点的短路导纳视为该节点的接地支路。

基于以下方式生成不定导纳矩阵Y_0：

$$Y_0 = \Sigma_{k=1}^{b} M_k y_k M_k$$

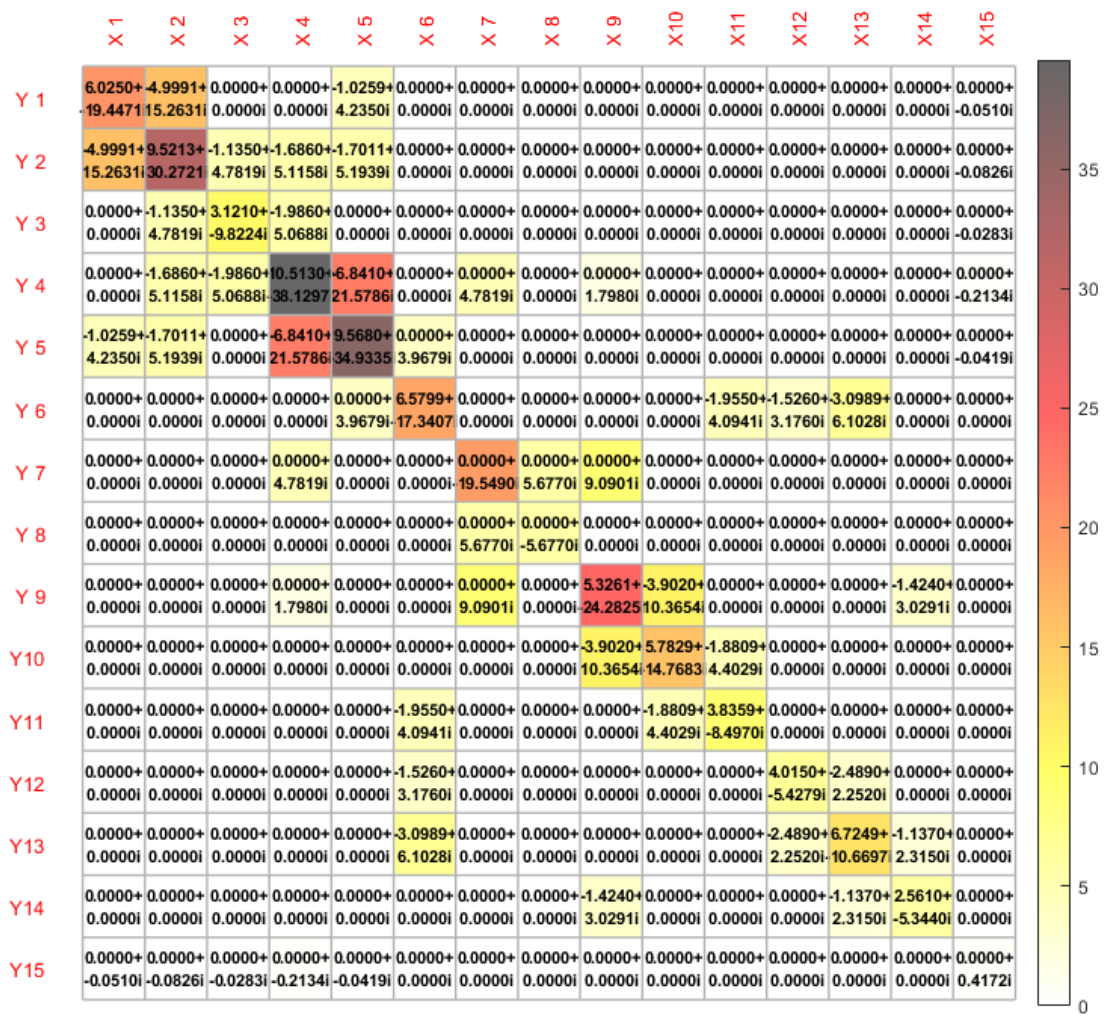
其中，M_k为列矢量

$$M_k = [0 \quad \dots \quad 1 \quad \dots \quad -1 \quad \dots \quad 0]^T$$

```
%% （1）不考虑变压器变比，生成 IEEE 14 节点系统的节点不定导纳矩阵Y_0
Y0=zeros(15,15);
for j=1:size(branch14,1)
    %     Y0(branch14(j,1),branch14(j,2))=branch14(j,3)+branch14(j,4)*1i;
```

```
    y=1/(branch14(j,3)+branch14(j,4)*1i);
    M=zeros(15,1);
    M(branch14(j,1))=1;
    M(branch14(j,2))=-1;
    Y0=Y0+M*y*M';% 支路的贡献
    if branch14(j,5)>0
        y2=(0.5*branch14(j,5)*1i);
        M1=zeros(15,1);
        M1(branch14(j,1))=1;
        M1(15)=-1;
        Y0=Y0+M1*y2*M1';% 线路接地导纳1
        M2=zeros(15,1);
        M2(branch14(j,2))=1;
        M2(15)=-1;
        Y0=Y0+M2*y2*M2';% 线路接地导纳2
    end
end
for j=1:size(bus14,1)
    y=bus14(j,6)*1i/baseMVA;
    M=zeros(15,1);
    M(j)=1;
    M(15)=-1;
    Y0=Y0+M*y*M';% 节点接地导纳
end
% 可视化部分
complexmatrixplot(Y0,'ColorBar','On');
colormap(flipud(hot)*0.6+[0.4 0.4 0.4])
```

生成矩阵如下所示：



1.1.2 观察 case14.m 文件，思考“地节点”在何处体现了?

地节点的体现包括：

- 在bus_data中，在Gs与Bs内体现了节点的接地阻抗，

```
%% bus data
%   bus_i   type    Pd  Qd  Gs  Bs  area    Vm  Va  baseKV  zone    Vmax    Vmin
mpc.bus = [
    1   3   0   0   0   0   1   1.06    0   0   1   1.06    0.94;
    2   2   21.7    12.7    0   0   1   1.045   -4.98   0   1   1.06    0.94;
    3   2   94.2    19  0   0   1   1.01    -12.72  0   1   1.06    0.94;
    4   1   47.8    -3.9    0   0   1   1.019   -10.33  0   1   1.06    0.94;
    5   1   7.6 1.6 0   0   1   1.02    -8.78   0   1   1.06    0.94;
    6   2   11.2    7.5 0   0   1   1.07    -14.22  0   1   1.06    0.94;
    7   1   0   0   0   0   1   1.062   -13.37  0   1   1.06    0.94;
    8   2   0   0   0   0   1   1.09    -13.36  0   1   1.06    0.94;
    9   1   29.5    16.6    0   19  1   1.056   -14.94  0   1   1.06    0.94;
    10  1   9   5.8 0   0   1   1.051   -15.1   0   1   1.06    0.94;
    11  1   3.5 1.8 0   0   1   1.057   -14.79  0   1   1.06    0.94;
    12  1   6.1 1.6 0   0   1   1.055   -15.07  0   1   1.06    0.94;
    13  1   13.5    5.8 0   0   1   1.05    -15.16  0   1   1.06    0.94;
    14  1   14.9    5   0   0   1   1.036   -16.04  0   1   1.06    0.94;
];
```

matpower的解释文件中对Gs与Bs的解释如下，表现为各个节点的短路导纳，也可以看做节点与地节点连接时

```
    5   Gs, shunt conductance (MW demanded at V = 1.0 p.u.)
    6   Bs, shunt susceptance (MVAr injected at V = 1.0 p.u.)
```

其中

- 在branch data内，b体现了线路与地的联系

```
%% branch data
%   fbus    tbus    r   x   b   rateA   rateB   rateC   ratio   angle   status
angmin  angmax
mpc.branch = [
    1   2   0.01938 0.05917 0.0528  0   0   0   0   0   1   -360    360;
    1   5   0.05403 0.22304 0.0492  0   0   0   0   0   1   -360    360;
    2   3   0.04699 0.19797 0.0438  0   0   0   0   0   1   -360    360;
    2   4   0.05811 0.17632 0.034   0   0   0   0   0   1   -360    360;
    2   5   0.05695 0.17388 0.0346  0   0   0   0   0   1   -360    360;
    3   4   0.06701 0.17103 0.0128  0   0   0   0   0   1   -360    360;
    4   5   0.01335 0.04211 0   0   0   0   0   0   1   -360    360;
    4   7   0   0.20912 0   0   0   0   0.978   0   1   -360    360;
    4   9   0   0.55618 0   0   0   0   0.969   0   1   -360    360;
    5   6   0   0.25202 0   0   0   0   0.932   0   1   -360    360;
    6   11  0.09498 0.1989  0   0   0   0   0   0   1   -360    360;
    6   12  0.12291 0.25581 0   0   0   0   0   0   1   -360    360;
    6   13  0.06615 0.13027 0   0   0   0   0   0   1   -360    360;
    7   8   0   0.17615 0   0   0   0   0   0   1   -360    360;
    7   9   0   0.11001 0   0   0   0   0   0   1   -360    360;
    9   10  0.03181 0.0845  0   0   0   0   0   0   1   -360    360;
    9   14  0.12711 0.27038 0   0   0   0   0   0   1   -360    360;
    10  11  0.08205 0.19207 0   0   0   0   0   0   1   -360    360;
    12  13  0.22092 0.19988 0   0   0   0   0   0   1   -360    360;
    13  14  0.17093 0.34802 0   0   0   0   0   0   1   -360    360;
];
```

在matlab帮助文件中可以看到，b代表了线路的电纳值

```
5   b, total line charging susceptance (p.u.)
```

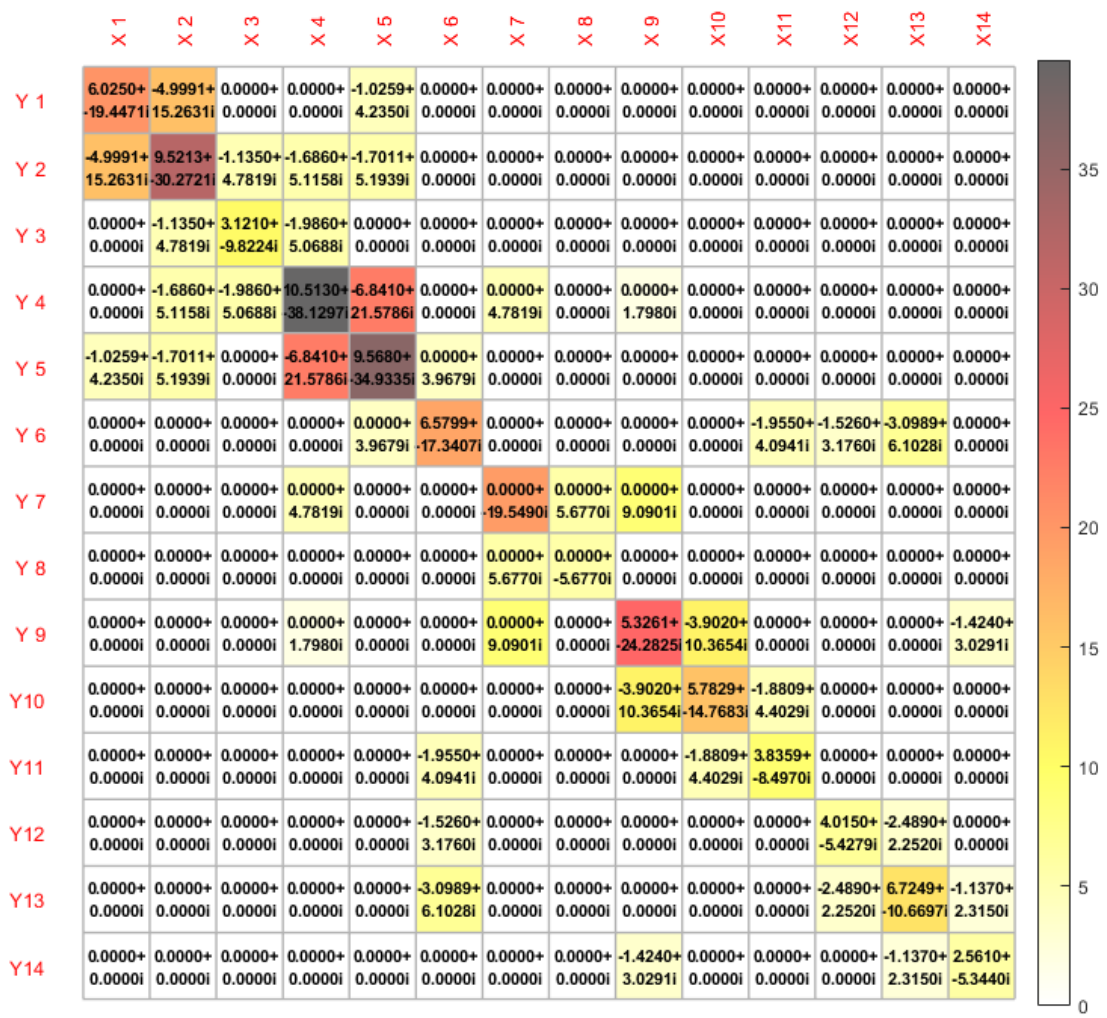
1.1.3 基于Y_0 生成系统的导纳矩阵Y

由于有公式：

$$\begin{bmatrix} Y & y_0 \\ {y_0}^T & y_{00} \end{bmatrix} = Y_0$$

```
Y=Y0(1:end-1,1:end-1);
% 可视化
complexmatrixplot(Y,'ColorBar','On');
colormap(flipud(hot)*0.6+[0.4 0.4 0.4])
```

取Y_0的前$n-1$行与$n-1$列，得到Y矩阵：



(2) 考虑变压器变比，生成 IEEE 14 节点系统对应导纳矩阵$\widetilde{Y}$

考虑变压器变比后，与上文区别在于：

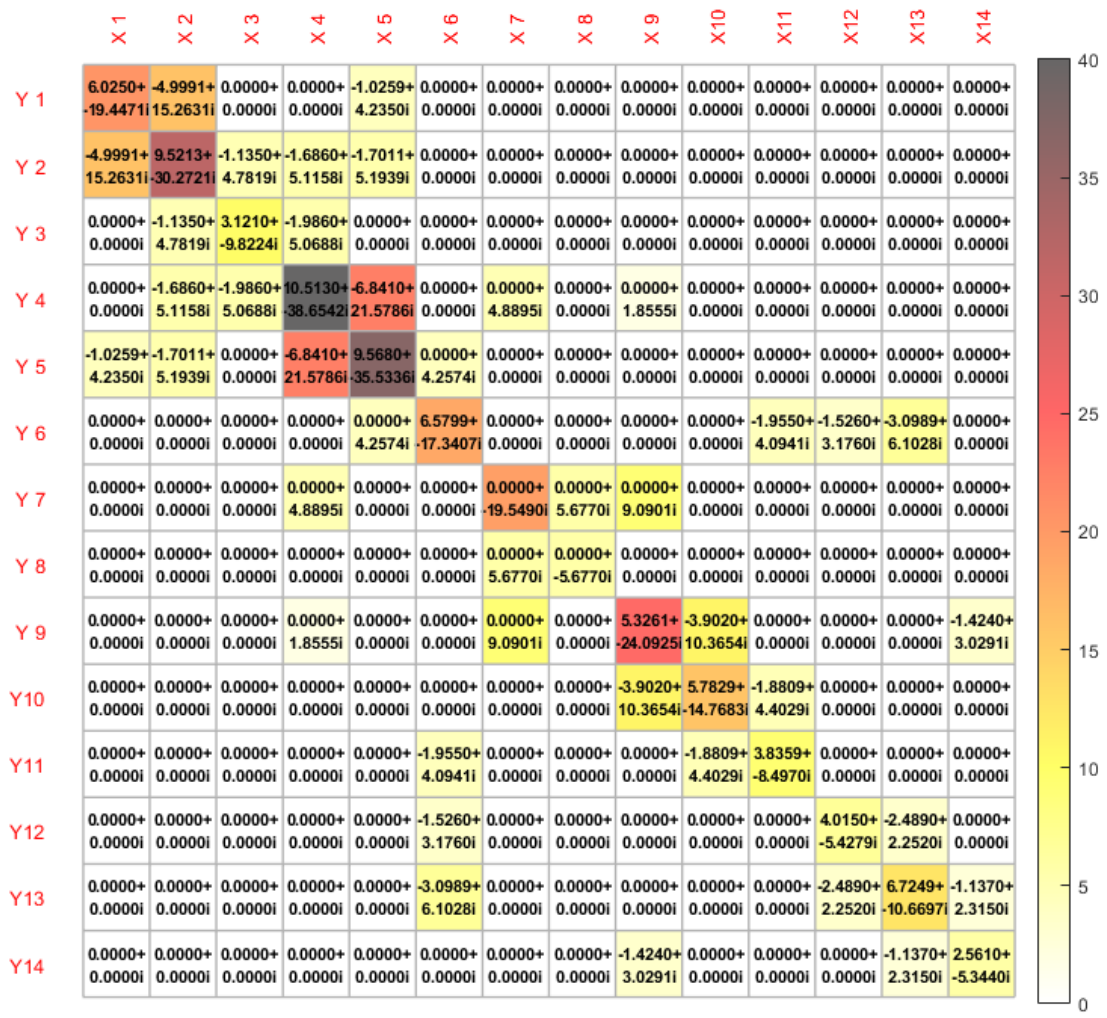
$$Y_0 = \Sigma_{k=1}^{b} M_k y_k M_k$$

如果线路中存在变压器，则将M_k替换为考虑变压器变比的形式。并注意到，matpower的变压器变比是“from”端电压：“to”端电压。

$$M_k = [0 \quad \dots \quad 1/t \quad \dots \quad -1 \quad \dots \quad 0]^T$$

```
%% （2）考虑变压器变比， 生成 IEEE 14 节点系统对应导纳矩阵Y
Y_widetilde=zeros(14,14);
for j=1:size(branch14,1)
%     Y0(branch14(j,1),branch14(j,2))=branch14(j,3)+branch14(j,4)*1i;
    y=1/(branch14(j,3)+branch14(j,4)*1i);
    M=zeros(14,1);
    if branch14(j,9)>0
        M(branch14(j,1))=1/branch14(j,9);
        M(branch14(j,2))=-1;
    else
        M(branch14(j,1))=1;
        M(branch14(j,2))=-1;
    end
    Y_widetilde=Y_widetilde+M*y*M';
```

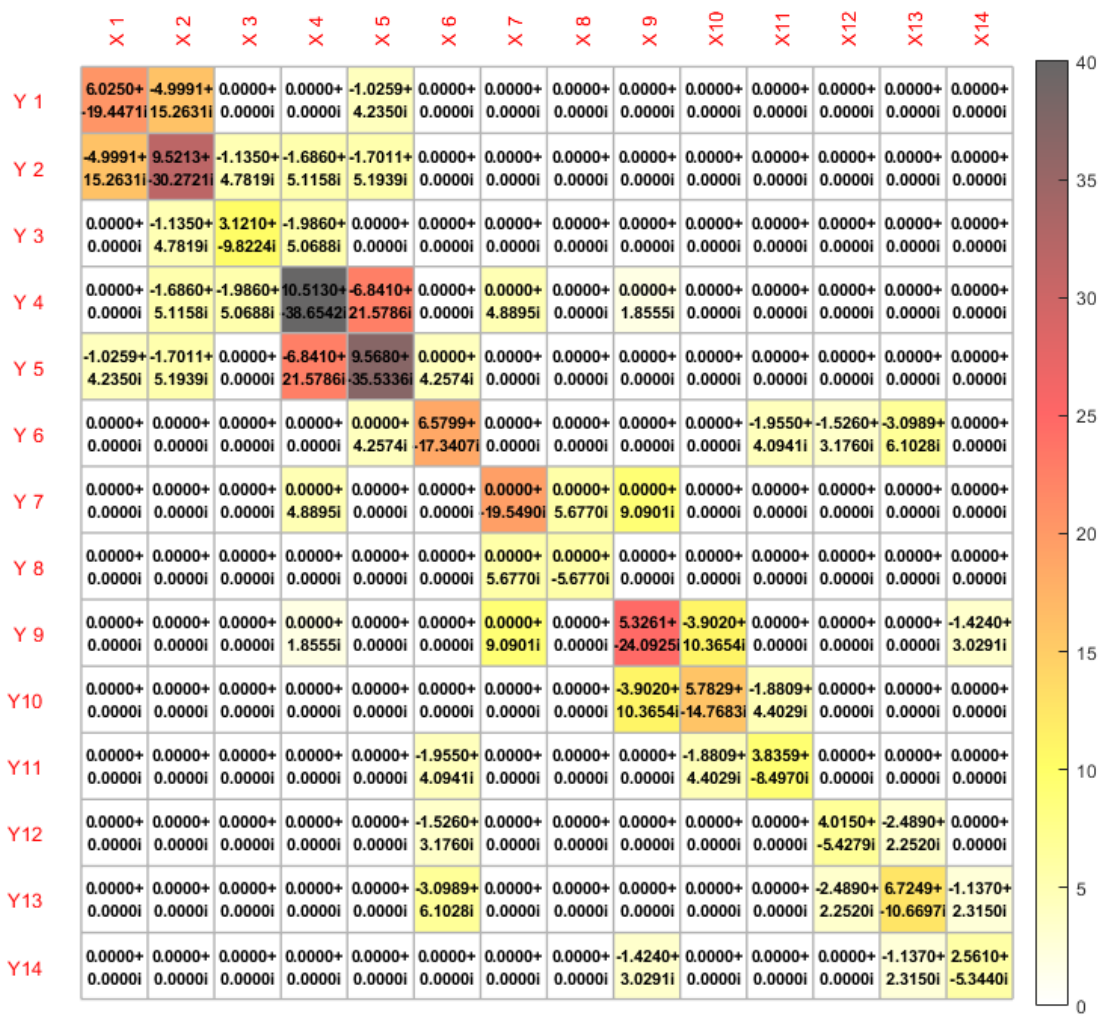
```
 Y_widetilde(branch14(j,1),branch14(j,1))=Y_widetilde(branch14(j,1),branch14(j,1
))+branch14(j,5)*1i/2;

 Y_widetilde(branch14(j,2),branch14(j,2))=Y_widetilde(branch14(j,2),branch14(j,2
))+branch14(j,5)*1i/2;
end
Y_widetilde=Y_widetilde+(diag(bus14(:,5))+diag(bus14(:,6))*1i)/baseMVA;
% 可视化
complexmatrixplot(Y_widetilde,'ColorBar','On');
colormap(flipud(hot)*0.6+[0.4 0.4 0.4])
```



（3）对比 matpower 中的 makeYbus 程序，验证导纳矩阵$\widetilde{Y}$的正确性， 思考并总结 makeYbus的编程技巧；

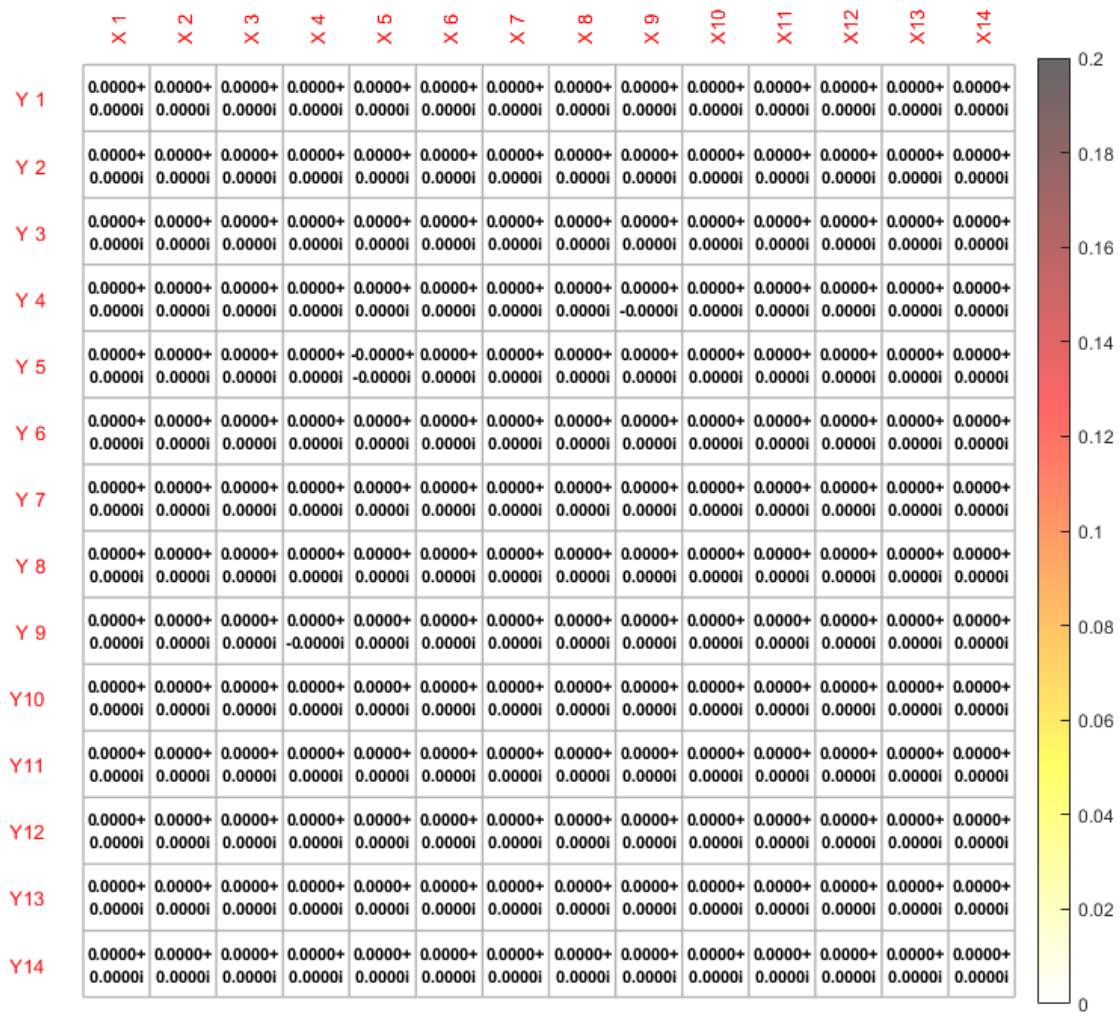
1.3.1 调用makeYbus程序

```
[YBUS, YF, YT] = makeYbus(mpc);% 调用matpower程序
Full_Ybus=full(YBUS);
```



并且将matpower生成的结果与自己编写的代码求解进行对比，发现结果

```
complexmatrixplot(Full_Ybus-Y_widetilde,'ColorBar','On');
colormap(flipud(hot)*0.6+[0.4 0.4 0.4])
```



1.3.2 思考并总结makeYbus编程技巧

- 函数接口部分

```
%% extract from MPC if necessary
if nargin < 3
    mpc      = baseMVA;
    baseMVA = mpc.baseMVA;
    bus      = mpc.bus;
    branch   = mpc.branch;
end
```

makeYbus使用nargin判断输入的方式（两种输入方式），增强函数的泛用性

```
    [YBUS, YF, YT] = makeYbus(MPC)
    [YBUS, YF, YT] = makeYbus(BASEMVA, BUS, BRANCH)
```

- 常数/定义部分

```
%% constants
nb = size(bus, 1);          %% number of buses
nl = size(branch, 1);       %% number of lines

%% define named indices into bus, branch matrices
[PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, ...
    VA, BASE_KV, ZONE, VMAX, VMIN, LAM_P, LAM_Q, MU_VMAX, MU_VMIN] = idx_bus;
[F_BUS, T_BUS, BR_R, BR_X, BR_B, RATE_A, RATE_B, RATE_C, ...
    TAP, SHIFT, BR_STATUS, PF, QF, PT, QT, MU_SF, MU_ST, ...
    ANGMIN, ANGMAX, MU_ANGMIN, MU_ANGMAX] = idx_brch;
```

由于matpower的各列矩阵的含义分别具有物理意义，所以将各列矩阵拆分并命名，从而增强程序可读性

```
%% check that bus numbers are equal to indices to bus (one set of bus numbers)
if any(bus(:, BUS_I) ~= (1:nb)')
    error('makeYbus: buses must be numbered consecutively in bus matrix; use
ext2int() to convert to internal ordering')
end
```

对输入数据进行格式检查，与上文类似，增强了函数的泛用性。

- 支路阻抗部分

```
%% for each branch, compute the elements of the branch admittance matrix where
%%
%%      | If |   | Yff  Yft |   | Vf |
%%      |    | = |          | * |    |
%%      | It |   | Ytf  Ytt |   | Vt |
%%
stat = branch(:, BR_STATUS);                    %% ones at in-service branches
Ys = stat ./ (branch(:, BR_R) + 1j * branch(:, BR_X));  %% series admittance
Bc = stat .* branch(:, BR_B);                           %% line charging
susceptance
tap = ones(nl, 1);                              %% default tap ratio = 1
i = find(branch(:, TAP));                       %% indices of non-zero tap
ratios
tap(i) = branch(i, TAP);                        %% assign non-zero tap ratios
tap = tap .* exp(1j*pi/180 * branch(:, SHIFT)); %% add phase shifters
Ytt = Ys + 1j*Bc/2;
Yff = Ytt ./ (tap .* conj(tap));
Yft = - Ys ./ conj(tap);
Ytf = - Ys ./ tap;

%% compute shunt admittance
%% if Psh is the real power consumed by the shunt at V = 1.0 p.u.
%% and Qsh is the reactive power injected by the shunt at V = 1.0 p.u.
%% then Psh - j Qsh = V * conj(Ysh * V) = conj(Ysh) = Gs - j Bs,
%% i.e. Ysh = Psh + j Qsh, so ...
Ysh = (bus(:, GS) + 1j * bus(:, BS)) / baseMVA; %% vector of shunt admittances

%% bus indices
f = branch(:, F_BUS);                           %% list of "from" buses
t = branch(:, T_BUS);                           %% list of "to" buses
```

1. 首先是非常直观的注释形式，代码可读性很高。
2. makeYbus的Y矩阵生成方式如下

分别计算$\begin{bmatrix} Y_{ff} & Y_{ft} \\ Y_{tf} & Y_{tt} \end{bmatrix}$，便于之后的稀疏矩阵计算，并且节省了程序的空间。

四个方程分开计算，并且考虑到最复杂的带相角变压器形式

$$Y_{tt} = Y_s + \frac{jB_c}{2}$$

$$Y_{tt} = Y_s + \frac{jB_c}{2}$$

$$Y_{ff} = \frac{Y_{tt}}{tap \times \hat{tap}}$$

$$Y_{ft} = \frac{Y_s}{\hat{tap}}$$

$$Y_{ft} = \frac{Y_s}{tap}$$

共占用$4 \times b$个数据的内存

- 小数据集模式下

```
if nb < 300 || have_feature('octave')   %% small case OR running on Octave
    %% build Yf and Yt such that Yf * V is the vector of complex branch currents
injected
    %% at each branch's "from" bus, and Yt is the same for the "to" bus end
    i = [1:nl 1:nl]';                           %% double set of row indices
    Yf = sparse(i, [f; t], [Yff; Yft], nl, nb);
    Yt = sparse(i, [f; t], [Ytf; Ytt], nl, nb);

    %% build Ybus
    Ybus = sparse([f;f;t;t], [f;t;f;t], [Yff;Yft;Ytf;Ytt], nb, nb) + ... %%
branch admittances
           sparse(1:nb, 1:nb, Ysh, nb, nb);        %% shunt admittance
```

Y_f与Y_t的生成：i占用$2 \times b$个整数内存，并用稀疏矩阵的形式录入

Y的生成：将$\begin{bmatrix} Y_{ff} & Y_{ft} \\ Y_{tf} & Y_{tt} \end{bmatrix}$录入稀疏矩阵

- 大数据集模式下

```
else                                %% large case running on MATLAB
    %% build connection matrices
    Cf = sparse(1:nl, f, ones(nl, 1), nl, nb);      %% connection matrix for
line & from buses
    Ct = sparse(1:nl, t, ones(nl, 1), nl, nb);      %% connection matrix for
line & to buses

    %% build Yf and Yt such that Yf * V is the vector of complex branch currents
injected
    %% at each branch's "from" bus, and Yt is the same for the "to" bus end
```

```
    Yf = sparse(1:nl, 1:nl, Yff, nl, nl) * Cf + sparse(1:nl, 1:nl, Yft, nl, nl)
* Ct;
    Yt = sparse(1:nl, 1:nl, Ytf, nl, nl) * Cf + sparse(1:nl, 1:nl, Ytt, nl, nl)
* Ct;

    %% build Ybus
    Ybus = Cf' * Yf + Ct' * Yt + ...            %% branch admittances
            sparse(1:nb, 1:nb, Ysh, nb, nb);    %% shunt admittance
end
```

Y_f与Y_t的生成：由于数据量较大，于是不采用i进行录入，C_f与C_t是节点支路关联矩阵A的正数部分与负数部分，在生产Y_f与Y_t时仅需要生成两个对角稀疏矩阵并与C_f和C_t相乘（分别是只有N个元素的$N \times N$的稀疏矩阵）

Y的生成：基于C_f与C_t的乘法运算，将$\begin{bmatrix} Y_{ff} & Y_{ft} \\ Y_{tf} & Y_{tt} \end{bmatrix}$录入稀疏矩阵

总结

- **makeYbus的程序通过用变量名称指代特定列，在注释中列写公式，添加help xxx命令等方法增强了程序的可读性与易用性。**
- **makeYbus的函数对输入进行校验，并给出错误反馈，便于使用者定位代码错误。**
- **makeYbus的程序在运算过程中通过矩阵的运算方式使得其占用内存小，makeYbus中仅使用四行矩阵进行导纳阵的计算，并尽可能使用稀疏的方式输出。**

（4）采用修正的形式求解变压器变比变化后的导纳矩阵。

备注：

基于导纳矩阵Y 的系统变压器变比发生变化（变为 mpc.branch(:,9)中所示），即修正Y 生成导纳矩阵$\widetilde{Y}$。

使用修正的方式求解，假设第k条支路中存在变压器，则视为移去一条普通支路并加上一条还有变压器的支路：

$$\widetilde{Y} = Y - M_k y_k M_k + M_k' y_k M_k'$$

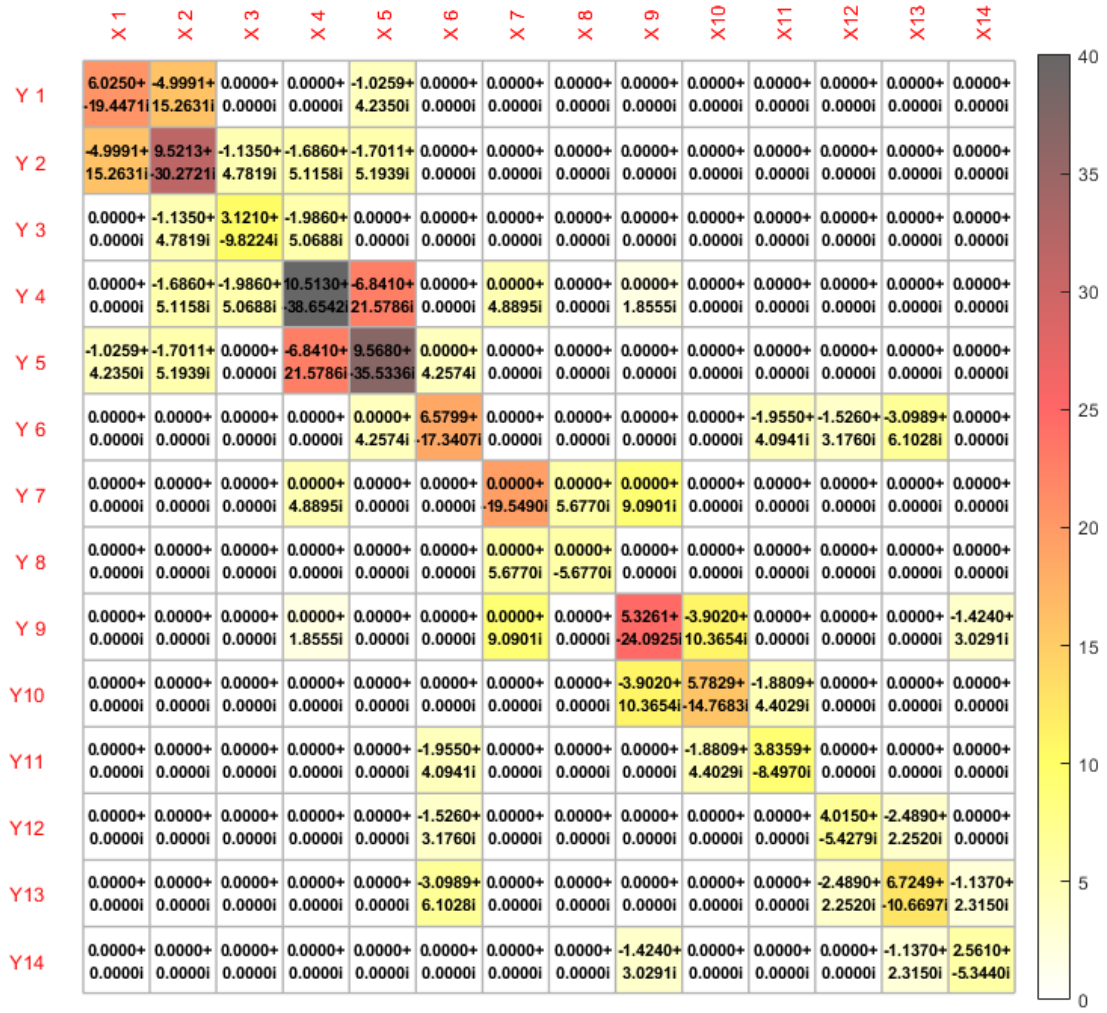
$$M_k = [0 \quad \dots \quad 1 \quad \dots \quad -1 \quad \dots \quad 0]^T$$

$$M_k\text{‘} = [0 \quad \dots \quad 1/t \quad \dots \quad -1 \quad \dots \quad 0]^T$$

```
%% （4）采用修正的形式求解变压器变比变化后的导纳矩阵
Y_widetilde2=Y;
for j=1:size(branch14,1)
%     Y0(branch14(j,1),branch14(j,2))=branch14(j,3)+branch14(j,4)*1i;
    if branch14(j,9)>0
        y=1/(branch14(j,3)+branch14(j,4)*1i);
        M=zeros(14,1);
        M2=zeros(14,1);

        M(branch14(j,1))=1/branch14(j,9);
        M(branch14(j,2))=-1;
        M2(branch14(j,1))=1;
        M2(branch14(j,2))=-1;
        Y_widetilde2=Y_widetilde2+M*y*M'-M2*y*M2';
    end
```

```
end
complexmatrixplot(Y_widetilde2,'ColorBar','On');
colormap(flipud(hot)*0.6+[0.4 0.4 0.4])
```

得到的结果与之前求得的$\widetilde{Y}$相同



2（10 月 24 日更新）、 本题基于 Matpower 7.1 中的 IEEE 14 节点系统（case14.m 文件），需要回答以下问题。

2.1 基于不考虑变压器变比的 IEEE 14 节点系统导纳矩阵Y， 实现 LDU 分解， 生成对应的因子表；

2.1.1 实现LDU分解

LDU分解流程与LU分解流程接近，但在分解流程中加入了将L矩阵转换为单位下三角矩阵的部分.

$$
\begin{aligned}
&Loop\ p = 1, \cdots, n-1 \\
&\qquad Loop\ j = p+1, \cdots, n \\
&\qquad\quad if\ a_{pj} \neq 0 \\
&\qquad\qquad a_{pj} = a_{pj}/a_{pp} \\
&\qquad\qquad\quad Loop\ i = p+1, \cdots, n \\
&\qquad\qquad\qquad if\ a_{ip} \neq 0 \\
&\qquad\qquad\qquad\quad a_{ij} = a_{ij} - a_{ip}a_{pj} \\
&\qquad\qquad\qquad End\ if \\
&\qquad\qquad\quad endLoop \\
&\qquad\quad End\ if \\
&\qquad endLoop \\
&\qquad Loop\ k = p+1, \cdots, n \\
&\qquad\qquad\quad a_{kp} = a_{kp}/a_{pp} \\
&endLoop
\end{aligned}
$$

LDU分解的因子表如图所示

	X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8	X 9	X10	X11	X12	X13	X14
Y 1	6.0250+ -19.4471i	-0.7888+ -0.0127i	0.0000+ 0.0000i	0.0000+ 0.0000i	-0.2136+ 0.0134i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i
Y 2	-0.7888+ -0.0127i	5.3845+ -18.2963i	-0.2573+ 0.0137i	-0.2823+ -0.0091i	-0.4666+ -0.0028i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i
Y 3	0.0000+ 0.0000i	-0.2573+ 0.0137i	2.8944+ -8.5763i	-0.7538+ -0.0196i	-0.2524+ 0.0219i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i
Y 4	0.0000+ 0.0000i	-0.2823+ -0.0091i	-0.7538+ -0.0196i	8.0938+ -32.1062i	-0.8103+ -0.0479i	0.0000+ 0.0000i	-0.1400+ 0.0353i	0.0000+ 0.0000i	-0.0527+ 0.0133i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i
Y 5	-0.2136+ 0.0134i	-0.4666+ -0.0028i	-0.2524+ 0.0219i	-0.8103+ -0.0479i	0.3106+ -9.0911i	-0.4360+ 0.0149i	-0.4266+ -0.0106i	0.0000+ 0.0000i	-0.1604+ -0.0040i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i
Y 6	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	-0.4360+ 0.0149i	6.6390+ -15.6109i	-0.0928+ 0.0368i	0.0000+ 0.0000i	-0.0349+ 0.0138i	0.0000+ 0.0000i	-0.2672+ -0.0116i	-0.2075+ -0.0095i	-0.4025+ -0.0273i	0.0000+ 0.0000i
Y 7	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	-0.1400+ 0.0353i	-0.4266+ -0.0106i	-0.0928+ 0.0368i	0.0885+ -17.0701i	-0.3326+ 0.0017i	-0.5871+ 0.0050i	0.0000+ 0.0000i	-0.0265+ -0.0017i	-0.0206+ -0.0013i	-0.0399+ -0.0035i	0.0000+ 0.0000i
Y 8	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	-0.3326+ 0.0017i	0.0098+ -3.7890i	-0.8796+ 0.0098i	0.0000+ 0.0000i	-0.0397+ -0.0024i	-0.0308+ -0.0019i	-0.0597+ -0.0051i	0.0000+ 0.0000i
Y 9	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	-0.0527+ 0.0133i	-0.1604+ -0.0040i	-0.0349+ 0.0138i	-0.5871+ 0.0050i	-0.8796+ 0.0098i	5.4655+ -14.9270i	-0.6967+ -0.0063i	-0.0343+ 0.0102i	-0.0266+ 0.0079i	-0.0521+ 0.0142i	-0.2097+ -0.0186i
Y10	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	-0.6967+ -0.0063i	2.9990+ -7.5712i	-0.6341+ -0.0009i	-0.0360+ 0.0114i	-0.0705+ 0.0205i	-0.2856+ -0.0204i
Y11	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	-0.2672+ -0.0116i	-0.0265+ -0.0017i	-0.0397+ -0.0024i	-0.0343+ 0.0102i	-0.6341+ -0.0009i	2.0540+ -4.3474i	-0.2390+ 0.0077i	-0.4644+ 0.0048i	-0.3314+ 0.0046i
Y12	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	-0.2075+ -0.0095i	-0.0206+ -0.0013i	-0.0308+ -0.0019i	-0.0266+ 0.0079i	-0.0360+ 0.0114i	-0.2390+ 0.0077i	3.5711+ -4.4939i	-0.9195+ -0.0239i	-0.0912+ 0.0339i
Y13	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	-0.4025+ -0.0273i	-0.0399+ -0.0035i	-0.0597+ -0.0051i	-0.0521+ 0.0142i	-0.0705+ 0.0205i	-0.4644+ 0.0048i	-0.9195+ -0.0239i	1.6822+ -3.5817i	-1.0535+ 0.0291i
Y14	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	0.0000+ 0.0000i	-0.2097+ -0.0186i	-0.2856+ -0.0204i	-0.3314+ 0.0046i	-0.0912+ 0.0339i	-1.0535+ 0.0291i	0.0177+ 0.4572i

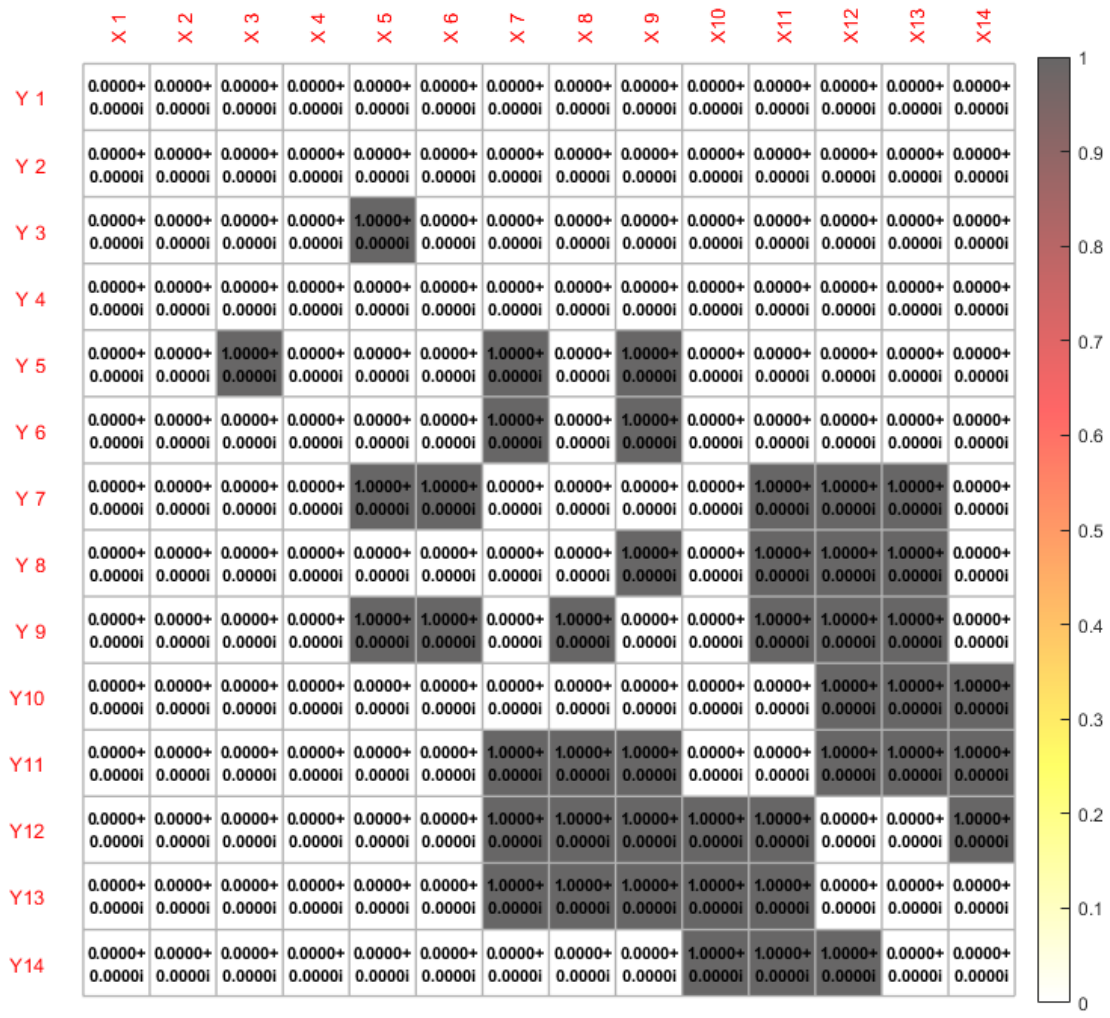
2.1.2 对Y矩阵的稀疏性进行分析：

判断$a_{pj} = 0$的语句共执行91次，其中有42次判断为非零

判断$a_{ij} = 0$的语句共执行295次，其中有156次判断为非零

（之后或许可以测试一下随着节点上升，稀疏度的变化）

2.1.3 分析该方程注入元



方程中标1的单元格为LDU分解中产生的注入元，Y矩阵原先有54个非零元，而后来增加了44个非零元，共98个非零元。

（2）基于 LDU 分解，采用稀疏矩阵技术，求解该导纳矩阵的逆（即阻抗矩阵Z ）

首先将LDU分解的因子表用稀疏矩阵的形式存储

考虑到Y为对称矩阵，LDU存储中，L矩阵与U矩阵可以只需要计算一次

选择三角检索格式进行存储

matlab代码如下

```
function [U,JU,IU,L,IL,JL,D]=cxLDU(A)
% A为矩阵的输入，要求为方阵
% [U,JU,IU,L,IL,JL,D]为稀疏格式的数据输出
[m,n]=size(A);
if m~=n
    disp('矩阵不为方阵！');
end

[row,col]=find(A);
```

```
U=[];
JU=[];
IU=[1];
L=zeros(1,1);
IL=zeros(1,1);
JL=zeros(1,1);
%% 计算下三角部分
for i=1:m
    JU_temp=sort(col(row==i));
    JU_temp=JU_temp(JU_temp>i);
    JU=[JU,JU_temp'];
    IU=[IU,size(JU,2)+1];
    Utemp=A(i,JU_temp);
    U=[U,Utemp];
end

%% 计算上三角部分
IL=JU;
JL=IU;
L=U;

%% 对角线部分
D=diag(A);
```

得到了以稀疏形式存储的LDU矩阵后，采用连续回代法的方式来获得阻抗矩阵Z，由于

$$Y^{-1}=Z$$

由此可以得到：

$$Z=(LDU)^{-1}=U^{-1}D^{-1}L^{-1}$$

令：

$$W=(LD)^{-1}$$

这是一个下三角矩阵，则有：

$$UZ=W$$

对于W上的对角元素有：

$$W_{ii}=\frac{1}{D_{ii}}$$

由于Y是一个对称阵，所以Z也是一个对称阵，在求取结果的时候只需要求取Z的上三角部分。因此，计算过程中只需要用到W的对角元素部分。

计算过程可以用以下的流程来实现：

$$\begin{aligned}
&Z_{NN}=1/D_{NN}\\
&Loop\ i=N-1,\cdots,1\\
&\qquad Loop\ j=N,\cdots,i+1\\
&\qquad\qquad Z_{ij}=-\sum_{k>i,U_{ik}\neq 0}U_{ik}Z_{kj}\\
&\qquad endLoop\\
&Z_{ii}=1/D_{ii}-\sum_{k>i,U_{ik}\neq 0}U_{ik}Z_{ik}\\
&endLoop
\end{aligned}$$

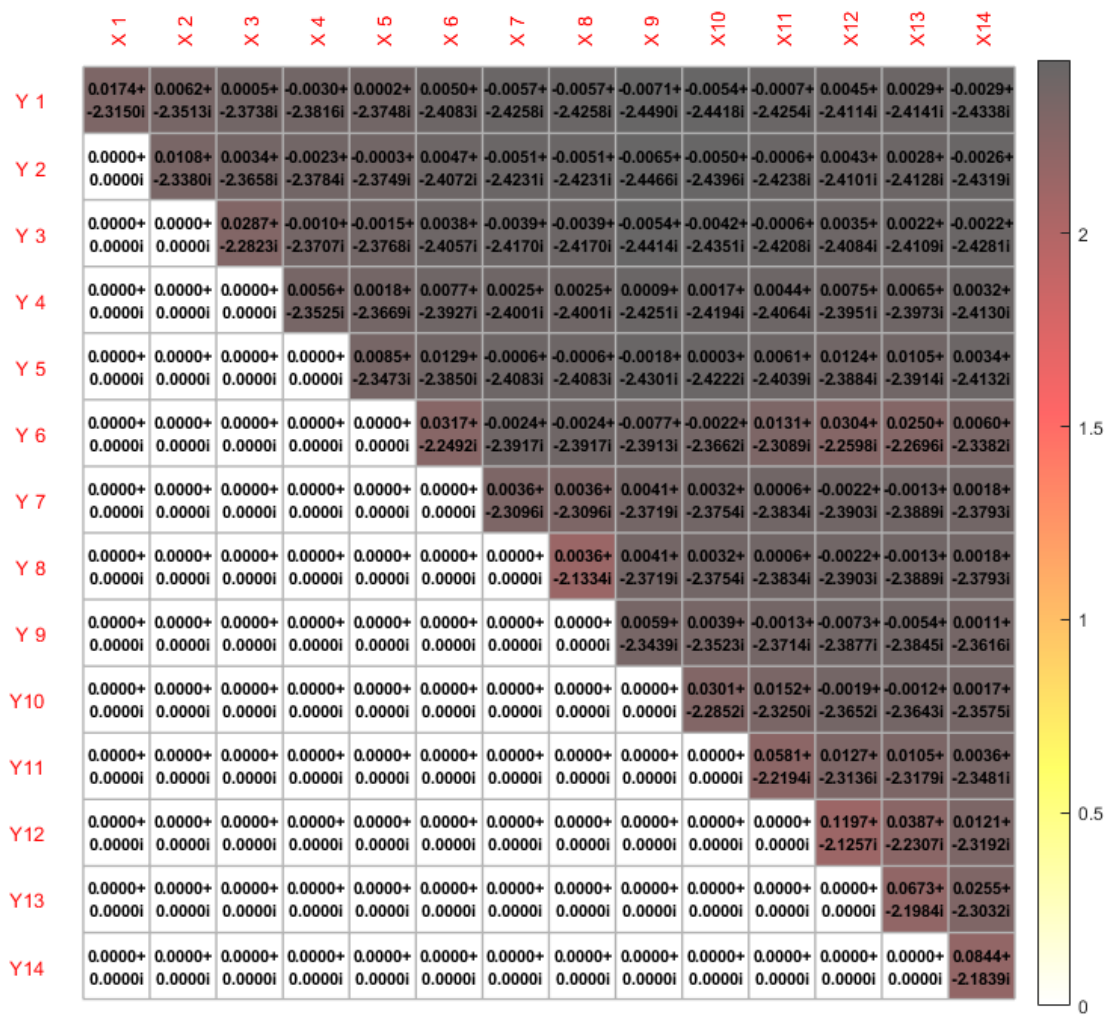
由于LDU分解中已经采用了稀疏矩阵的格式存储，所以可以使用以下形式来进行替代：

$$
\begin{aligned}
&Z_{NN}=1/D_{NN}\\
&Loop\ i=N-1,\cdots,1\\
&\qquad Loop\ j=N,\cdots,i+1\\
&\qquad\qquad Loop\ k=\mathbf{IU}(i),\cdots,\mathbf{IU}(i+1)-1\\
&\qquad\qquad\qquad Z_{ij}=Z_{ij}-\mathbf{U}(k)max\{Z_{\mathbf{JU}(k),j},Z_{j,\mathbf{JU}(k)}\}\\
&\qquad\qquad endLoop\\
&\qquad endLoop\\
&\qquad Z_{ii}=1/D_{ii}\\
&\qquad\qquad Loop\ k=\mathbf{IU}(i),\cdots,\mathbf{IU}(i+1)-1\\
&\qquad\qquad\qquad Z_{ij}=Z_{ij}-\mathbf{U}(k)Z_{i,\mathbf{JU}(k)}\\
&\qquad\qquad endLoop\\
&endLoop
\end{aligned}
$$

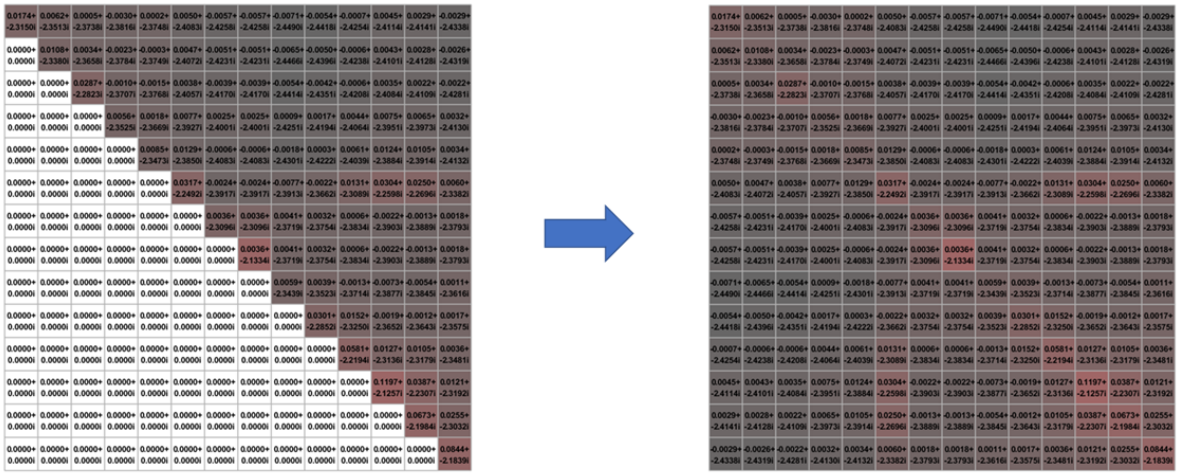
程序实现如下：

```
[U,JU,IU,L,IL,JL,D]=cxLDU(Y);
N=14;
Z=eye(N);
Z(N,N)=1/D(N);
for i=N-1:-1:1
    for j=N:-1:i+1
        for k=IU(i):IU(i+1)-1
            if JU(k)<j
                Z(i,j)=Z(i,j)-U(k)*Z(JU(k),j);
            else
                Z(i,j)=Z(i,j)-U(k)*Z(j,JU(k));
            end
        end
    end
    Z(i,i)=1/D(i);
    for k=IU(i):IU(i+1)-1
        Z(i,i)=Z(i,i)-U(k)*Z(i,JU(k));
    end
end
complexmatrixplot(Z,'ColorBar','On');
colormap(flipud(hot)*0.6+[0.4 0.4 0.4])

complexmatrixplot(inv(Y_copy),'ColorBar','On');
colormap(flipud(hot)*0.6+[0.4 0.4 0.4])
```

此外，由于L^{-1}和U^{-1}的稀疏特性并不强，故直接通过矩阵乘法进行计算，只需要计算矩阵乘法的上半部分，即可直接求解出整个Z矩阵。



经过对称后可以直接得到Z矩阵，与直接计算得到的Y^{-1}相同



（3） 在节点 5 与节点 8 之间添加支路 r、 x、 b 参数（标幺值）分别为 0.016、0.058、 0.162分别采用面向支路、面向节点修正的补偿法计算修正后的节点导纳矩阵，讨论两种方法区别。

支路增加引起的导纳矩阵的变化是

$$\Delta Y = \begin{bmatrix} \Delta y_l + y_c & -\Delta y_l \\ -\Delta y_l & \Delta y_l + y_c \end{bmatrix}$$

本题中有：

$$\Delta y_l = \frac{1}{0.016 + 0.058i}$$

$$y_c = 0.081i$$

2.3.1 考虑面向支路的修正方法

矩阵可以写为

$$\Delta Y = -M \delta y M^T$$

其中，M为列矩阵

$$M = \begin{bmatrix} 0 & \dots & 1 & \dots & -1 & \dots & 0 \end{bmatrix}^T$$

δy为标量（即1×1矩阵）

但本文中，由于考虑到还有对地阻抗，对于题中的ΔY，可以表示为如下的形式

$$\Delta Y = - \begin{bmatrix} & 1 & 1 & \\ & -1 & & 1 \\ & & & \end{bmatrix} \begin{bmatrix} \Delta y_l & & \\ & y_c & \\ & & y_c \end{bmatrix}_{3\times 3} \begin{bmatrix} 1 & -1 \\ 1 & \\ & 1 \end{bmatrix}$$

2.3.2 考虑面向节点的修正方法

$$\Delta Y = - \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} \begin{bmatrix} \Delta y_l + y_c & -\Delta y_l \\ -\Delta y_l & \Delta y_l + y_c \end{bmatrix}_{2\times 2} \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}$$

2.3.3 对比两种方法的差别

在本题中，由于不涉及求逆，所以两种方法并没有差别。

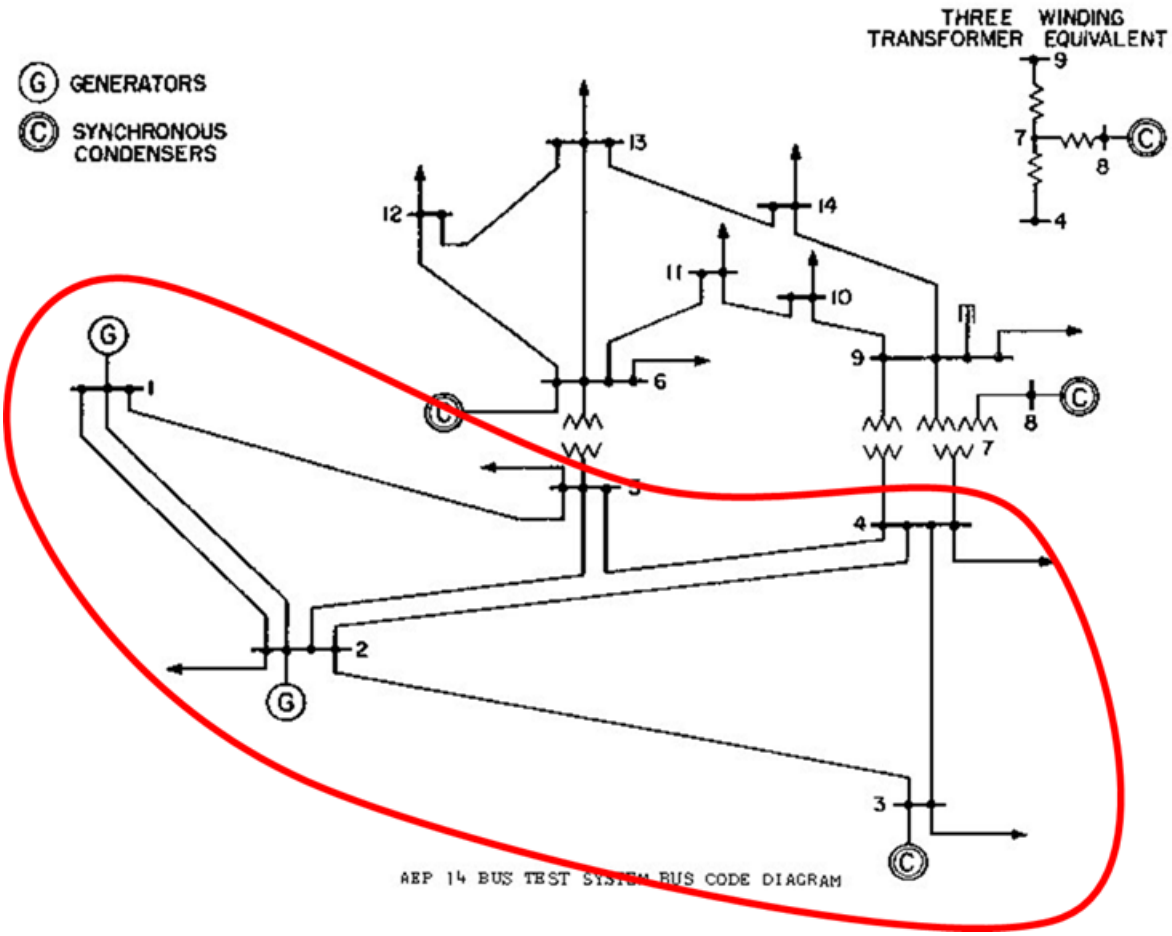
但是，在

3 本题基于 Matpower 7.1 中的 IEEE 14 节点系统（case14.m 文件），需要回答以下问题。

基于 IEEE 14 节点系统，试选节点 1,2,3,4,5 为内部节点， 6,7,8,9 为边界节点，10,11,12,13,14 为外部节点进行 WARD 等值，第 2,4,6,8,10,13 节点的注入电流分别为1.1,0.9,1.3,0.8,1.05,1.2，其余元素均为 0。

已知，节点注入电流

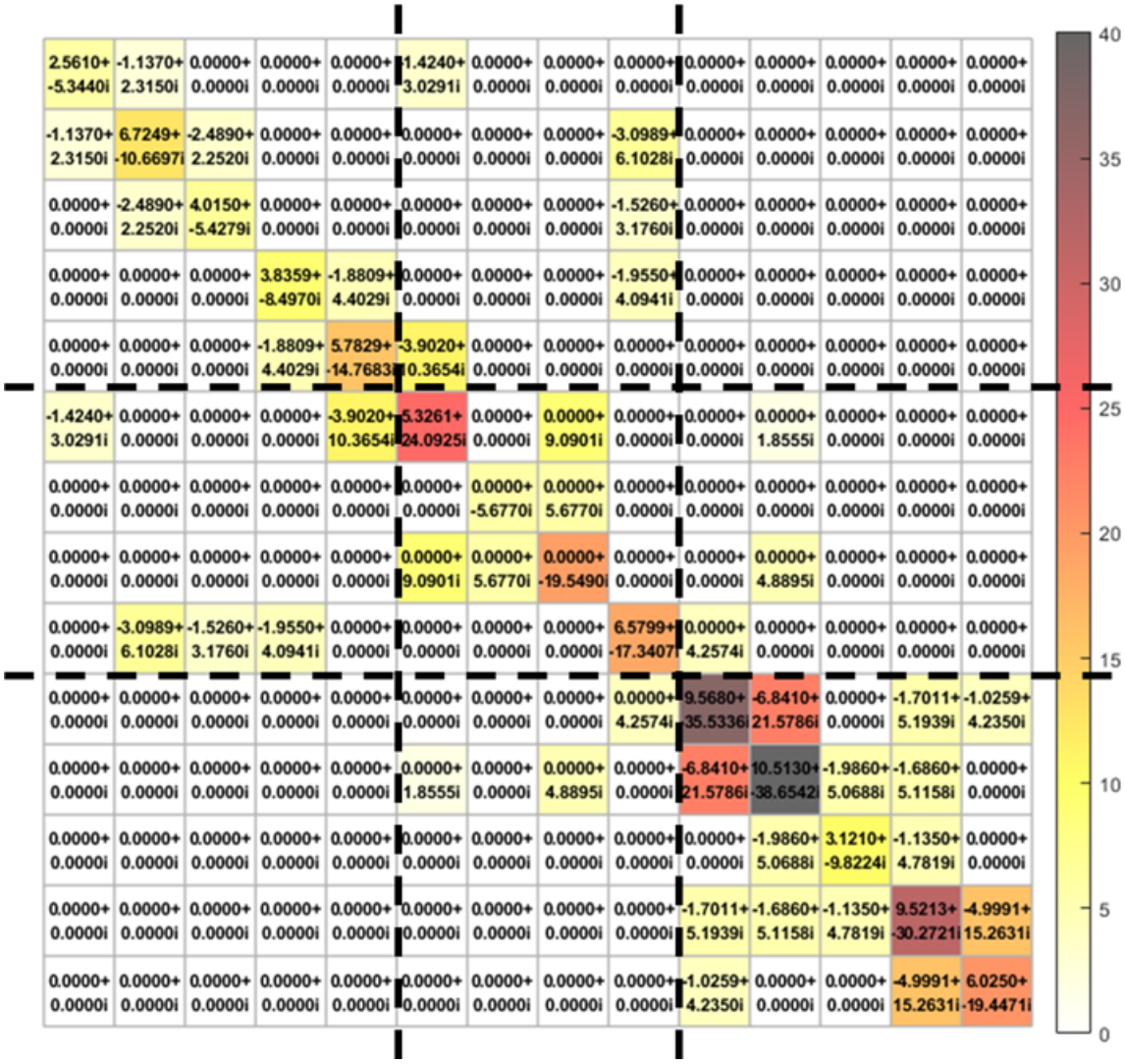
$$I_{input} = [0 \quad 1.1 \quad 0 \quad 0.9 \quad 0 \quad 1.3 \quad 0 \quad 0.8 \quad 0 \quad 1.05 \quad 0 \quad 0 \quad 1.2 \quad 0]^T$$



3.1 求出边界节点上的等值支路和等值注入电流，探究是否使用 WARD 等值的求解差异

首先将网络节点按照E、B、I的顺序排列，得到节点导纳矩阵如下：

$$\begin{bmatrix} Y_{EE} & Y_{EB} & \\ Y_{BE} & Y_{BB} & Y_{BI} \\ & Y_{IB} & Y_{II} \end{bmatrix} =$$



由此式可以得到Y_{EE}等参量

若不采用边界的处理方式，则有：

$$\begin{bmatrix} Y_{EE} & Y_{EB} & \\ Y_{BE} & Y_{BB} & Y_{BI} \\ & Y_{IB} & Y_{II} \end{bmatrix} \begin{bmatrix} \dot{V}_E \\ \dot{V}_B \\ \dot{V}_I \end{bmatrix} = \begin{bmatrix} \dot{I}_E \\ \dot{I}_B \\ \dot{I}_I \end{bmatrix}$$

```
V_1=Full_Ybus\I_input
```

采用边界的处理方式后有，消去外部节点的电压变量后有

$$\begin{bmatrix} \widetilde{Y}_{BB} & Y_{BI} \\ Y_{IB} & Y_{II} \end{bmatrix} \begin{bmatrix} \dot{V}_B \\ \dot{V}_I \end{bmatrix} = \begin{bmatrix} \dot{\tilde{I}}_B \\ \dot{I}_I \end{bmatrix}$$

边界等值导纳矩阵为

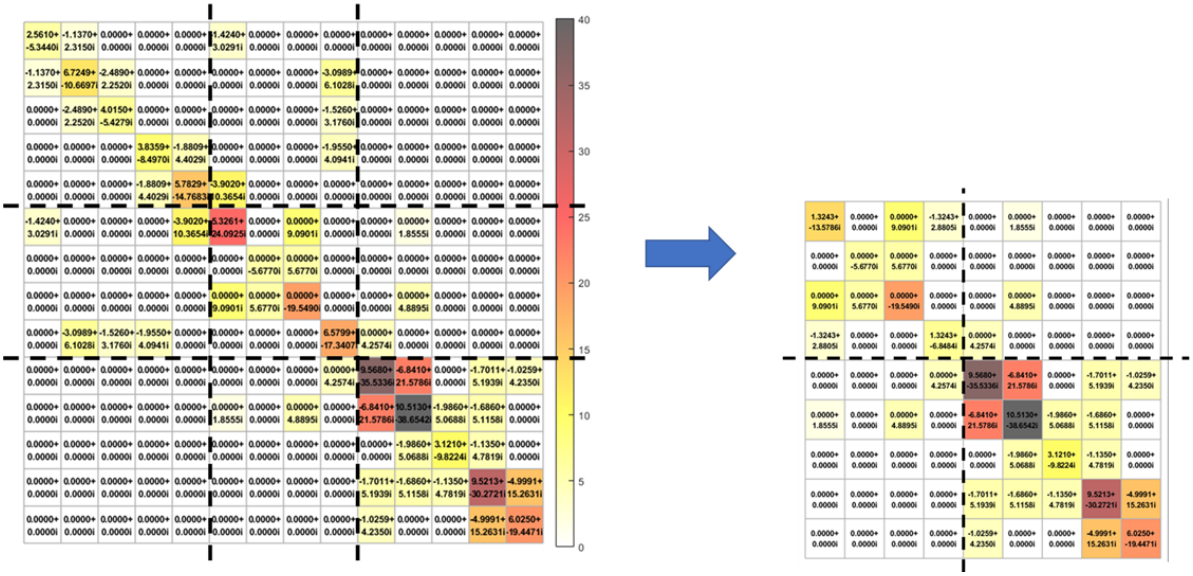
$$\widetilde{Y}_{BB} = Y_{BB} - Y_{BE} Y_{EE}^{-1} Y_{EB}$$

等值边界注入电流为

$$\dot{\tilde{I}}_B = \dot{I}_B - Y_{BE} Y_{EE}^{-1} \dot{I}_E$$

```
Y_EE=Full_Ybus(1:5,1:5);
```

```
Y_BB=Full_Ybus(6:9,6:9);
Y_EB=Full_Ybus(1:5,6:9);
Y_BE=Full_Ybus(6:9,1:5);
widetilde_Y_BB=Y_BB-Y_BE*inv(Y_EE)*Y_EB;
Y_2=
[widetilde_Y_BB,Full_Ybus(6:9,10:14);Full_Ybus(10:14,6:9),Full_Ybus(10:14,10:14)
];
complexmatrixplot(Y_2,'ColorBar','On');
colormap(flipud(hot)*0.6+[0.4 0.4 0.4])

I_E=I_input(1:5);
I_B=I_input(6:9);
widetilde_I_B=I_B-Y_BE*inv(Y_EE)*I_E;

I_2=[widetilde_I_B;I_input(10:14)];
V_2=Y_2\I_2;
```



直接计算结果为：

```
  0.0312 -15.7928i
  0.0775 -15.6853i
  0.0418 -15.7268i
  0.0273 -15.7810i
  0.0294 -15.7855i
 -0.0030 -15.8762i
 -0.0007 -15.5807i
 -0.0007 -15.7216i
  0.0252 -15.7762i
  0.0158 -15.2161i
  0.0037 -15.2521i
 -0.0156 -15.3090i
 -0.0074 -15.2887i
 -0.0129 -15.3101i
```

矩阵化简后计算结果为

```
-0.0030 -15.8762i
-0.0007 -15.5807i
-0.0007 -15.7216i
 0.0252 -15.7762i
 0.0158 -15.2161i
 0.0037 -15.2521i
-0.0156 -15.3090i
-0.0074 -15.2887i
-0.0129 -15.3101i
```

计算结果一致

探究差异:

3.2 求出以节点 13,14 对地为端口的戴维南等值参数，包括等值阻抗和等值戴维南电动势

由于节点13 14均不是参考节点，所以端口的节点-端口关联矢量为

$$M_\alpha = \begin{matrix}[\\ \\ \\ \end{matrix} \begin{matrix} 0 & \dots & 1 & \dots & -1 & \dots & 0 \\ & & \mathbf{13} & & \mathbf{14} & & \end{matrix} \begin{matrix}]^T \\ \\ \\ \end{matrix}$$

因此得到的多端口戴维南等值电路的等值阻抗矩阵为

$$Z_{eq} = M_L^T Z M_L$$

$$V_{eq} = M_L^T V$$

可以推导得到

$$Z_{eq} = Z_{13,13} + Z_{14,14} - 2 \times Z_{13,14}$$

$$V_{eq} = V_{13} - V_{14}$$

```
%% 求出以节点13,14对地为端口的戴维南等值参数，包括等值阻抗和等值戴维南电动势
Z_eq=Z(13,13)+Z(14,14)-2*Z(13,14);
V_eq=V_1(13)-V_1(14);
```

得到结果

$$Z_{eq} = 0.1010 + 0.2243i$$

$$V_{eq} = 0.0463 + 0.1074i$$

3.3 当支路（12,13）开断后，对（2）中求出的戴维南等值参数进行修正

支路12,13断开，等价于在支路（12,13）之间加上一条负阻抗支路，之后采用支路追加法进行求解。

$$\Delta Y = -M_\alpha \delta y M_\alpha^T$$

$$M_\alpha = \begin{matrix}[0 & \dots & 1 & \dots & -1 & \dots & 0]^T \\ & & \mathbf{12} & & \mathbf{13} & & \end{matrix}$$

根据矩阵求逆定理有：

$$Z' = Z - ZM_\alpha(-z_\alpha + M_\alpha^T ZM_\alpha)M_\alpha^T Z$$

实际上，由于只需要得到戴维南等值参数，所有不需要对整个网络进行计算

$$M_L = \begin{matrix}[0 & \dots & 1 & \dots & -1 & \dots & 0]^T \\ & & \mathbf{13} & & \mathbf{14} & & \end{matrix}$$

$$Z'_{eq} = Z_{eq} - Z_{L\alpha}Y_{\alpha\alpha}Z_{L\alpha}^T$$

$$V'_{eq} = V_{eq} - Z_{L\alpha}Y_{\alpha\alpha}V_\alpha$$

其中有：

$$Z_{L\alpha} = M_L^T ZM_\alpha$$

$$V_\alpha = (ZM_\alpha)^T I$$

$$Y_{\alpha\alpha} = (-y_\alpha + M_\alpha^T ZM_\alpha)^{-1}$$

```
%% 当支路（12,13）开断后，对（2）中求出的戴维南等值参数进行修正
M_alpha=zeros(14,1);
M_alpha(12)=1;
M_alpha(13)=-1;

M_L=zeros(14,1);
M_L(13)=1;
M_L(14)=-1;
y_alpha=1/(mpc.branch(19,3)+mpc.branch(19,4)*1i);
Y_alphaalpha=1/(-y_alpha+M_alpha'*Z*M_alpha);
Z_Lalpha=M_L'*Z*M_alpha;
V_alpha=(Z*M_alpha)'*I_input;

Z_eq_2=Z_eq-Z_Lalpha*Y_alphaalpha*Z_Lalpha';
V_eq_2=V_eq-Z_Lalpha*Y_alphaalpha*V_alpha;
```

得到结果

$$Z'_{eq} = 0.1011 + 0.2244i$$

$$V'_{eq} = 0.0466 + 0.1077i$$