Appendix A
DC Load Flow

A.1 The Load Flow Problem

Formulation of classic load flow problem requires considering four variables at each bus i of power system. These variables are

1. P_i (Net active power injection)
2. Q_i (Net reactive power injection)
3. V_i (Voltage magnitude)
4. θ_i (Voltage angle)

The active and reactive power injections are calculated as follows

$$P_i = P_{Gi} - P_{Di} \tag{A.1}$$

$$Q_i = Q_{Gi} - Q_{Di} \tag{A.2}$$

in which P_{Gi} and Q_{Gi} are active and reactive power generations at bus i, respectively, whereas P_{Di} and Q_{Di} are active and reactive power demands at this bus, respectively.

Based on the application of Kirchhoff's laws to each bus

$$\mathbf{I} = \mathbf{YV} \tag{A.3}$$

$$I_i = \frac{(P_i - jQ_i)}{|V_i|} e^{j\theta_i} \tag{A.4}$$

where

I_i Net injected current at bus i
V Vector of bus voltages
I Vector of injected currents at the buses
Y Bus admittance matrix of the system

I, **V** and **Y** are complex. $V_i = |V_i| e^{j\theta_i}$ is the ith element of vector **V**. The **Y** matrix is symmetrical. The diagonal element Y_{ii} (self admittance of bus i) contains the sum of admittances of all the branches connected to bus i. The off diagonal element Y_{ij} (mutual admittance) is equal to the negative sum of the admittances between buses i and j. $Y_{ij} = |Y_{ij}| e^{j\delta_{ij}} = G_{ij} + jB_{ij}$ lies in the ith row and the jth column of matrix **Y**. G and B are subsequently called conductance and susceptance, respectively..

Using (A.4) to replace **I** in (A.3) results in (A.5) and (A.6).

$$P_i = \sum_{j=1}^{N} |Y_{ij}||V_i||V_j| \cos(\theta_i - \theta_j - \delta_{ij}) \tag{A.5}$$

$$Q_i = \sum_{j=1}^{N} |Y_{ij}||V_i||V_j| \sin(\theta_i - \theta_j - \delta_{ij}) \tag{A.6}$$

where N is the number of system buses.

To solve full load flow equations, two of four variables must be known in advance at each bus. This formulation results in a non-linear system of equations which requires iterative solution methods. In this formulation, convergence is not guaranteed.

A.2 DC Load Flow Solution

Direct Current Load Flow (DCLF) gives estimations of lines power flows on AC power systems. DCLF looks only at active power flows and neglects reactive power flows. This method is non-iterative and absolutely convergent but less accurate than AC Load Flow (ACLF) solutions. DCLF is used wherever repetitive and fast load flow estimations are required.

In DCLF, nonlinear model of the AC system is simplified to a linear form through these assumptions

- Line resistances (active power losses) are negligible i.e. $R \ll X$.
- Voltage angle differences are assumed to be small i.e. $\sin(\theta) = \theta$ and $\cos(\theta) = 1$.
- Magnitudes of bus voltages are set to 1.0 per unit (flat voltage profile).
- Tap settings are ignored.

Based on the above assumptions, voltage angles and active power injections are the variables of DCLF. Active power injections are known in advance. Therefore for each bus i in the system, (A.5) is converted to

$$P_i = \sum_{j=1}^{N} B_{ij}(\theta_i - \theta_j) \tag{A.7}$$

in which B_{ij} is the reciprocal of the reactance between bus i and bus j. As mentioned earlier, B_{ij} is the imaginary part of Y_{ij}.

As a result, active power flow through transmission line i, between buses s and r, can be calculated from (A.8).

$$P_{Li} = \frac{1}{X_{Li}}(\theta_s - \theta_r) \tag{A.8}$$

where X_{Li} is the reactance of line i.

DC power flow equations in the matrix form and the corresponding matrix relation for flows through branches are represented in (A.9) and (A.10).

$$\theta = [\mathbf{B}]^{-1}\mathbf{P} \tag{A.9}$$

$$\mathbf{P_L} = (\mathbf{b} \times \mathbf{A})\theta \tag{A.10}$$

where

- **P** N × 1 vector of bus active power injections for buses 1, …, N
- **B** N × N admittance matrix with R = 0
- ***θ*** N × 1 vector of bus voltage angles for buses 1, …, N
- $\mathbf{P_L}$ M × 1 vector of branch flows (M is the number of branches)
- **b** M × M matrix (b_{kk} is equal to the susceptance of line k and non-diagonal elements are zero)
- **A** M × N bus-branch incidence matrix

Each diagonal element of **B** (i.e. B_{ii}) is the sum of the reciprocal of the lines reactances connected to bus i. The off-diagonal element (i.e. B_{ij}) is the negative sum of the reciprocal of the lines reactances between bus i and bus j.

A is a connection matrix in which a_{ij} is 1, if a line exists from bus i to bus j; otherwise zero. Moreover, for the starting and the ending buses, the elements are 1 and −1, respectively.

Example A.1 A simple example is used to illustrate the points discussed above. A three-bus system is considered. This system is shown in Fig. A.1, with the details given in Tables A.1 and A.2.

With base apparent power equal to 100 MVA, **B** and **P** are calculated as follows

$$\mathbf{B} = \begin{bmatrix} 23.2435 & -17.3611 & -5.8824 \\ -17.3611 & 28.2307 & -10.8696 \\ -5.8824 & -10.8696 & 16.7519 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \text{Unknown} \\ 0.53 \\ -0.9 \end{bmatrix}$$

As bus 1 is considered as slack,[1] the first row of **P** and the first row and column of **B** are disregarded. θ_2 and θ_3 are then calculated using (A.9) as follows.

[1] With angle = 0.

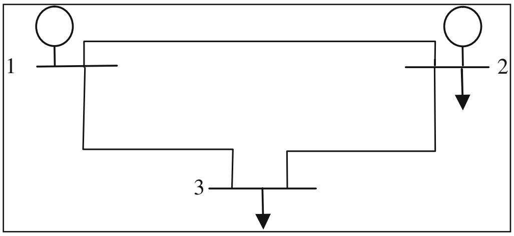

Fig. A.1 Three-bus system

Table A.1 Loads and generations

Bus number	Bus type	P_D (MW)	Q_D (MVAr)	P_G (MW)
1	Slack	0	0	Unknown
2	PV	10	5	63
3	PQ	90	30	0

Table A.2 Branches

Line number	From bus	To bus	X (p.u.)	Rating (MVA)
1	1	2	0.0576	250
2	2	3	0.092	250
3	1	3	0.17	150

$$\begin{bmatrix} \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} 28.2307 & -10.8696 \\ -10.8696 & 16.7519 \end{bmatrix}^{-1} \begin{bmatrix} 0.53 \\ -0.9 \end{bmatrix} = \begin{bmatrix} -0.0025 \\ -0.0554 \end{bmatrix} \text{Radian} = \begin{bmatrix} -0.1460^\circ \\ -3.1730^\circ \end{bmatrix}$$

A and **b** are then calculated as

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 17.3611 & 0 & 0 \\ 0 & 10.8696 & 0 \\ 0 & 0 & 5.8824 \end{bmatrix}$$

Therefore, the transmission flows are calculated using (A.10) as follows

$$\begin{bmatrix} P_{L1} \\ P_{L2} \\ P_{L3} \end{bmatrix} = \text{BaseMVA} \times \mathbf{b} \times \mathbf{A} \times \theta$$

$$= 100 \times \begin{bmatrix} 17.3611 & 0 & 0 \\ 0 & 10.8696 & 0 \\ 0 & 0 & 5.8824 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} 0 \\ -0.0025 \\ -0.0554 \end{bmatrix}$$

$$= \begin{bmatrix} 4.4243 \\ 57.4243 \\ 32.5757 \end{bmatrix} \text{MW}$$

Appendix B
A Simple Optimization Problem

In this appendix, a simple optimization problem is devised and solved by some optimization algorithms.

B.1 Problem Definition

The problem is *Economic Dispatch* (ED), briefly described in Chap. 1 in which the aim is to optimize the total generation cost, F_T, defined as

$$F_T = \sum_{i=1}^{N} F_i(P_i) \qquad i = 1, \ldots, N \tag{B.1}$$

where

P_i	The active power generation of generation unit i
N	The number of generation units
$F_i(P_i)$	Generation cost of unit i

$F_i(P_i)$ is defined as

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \tag{B.2}$$

where a_i, b_i and c_i are known in advance.

Two types of constraints are observed as follows

$$P_{i\,\min} \le P_i \le P_{i\,\max} \qquad i = 1, \ldots, N \tag{B.3}$$

$$\sum_{i=1}^{N} P_i - P_D = 0 \tag{B.4}$$

where (B.3) refers to satisfying the generation level of each unit to be within its respective minimum and maximum limits and (B.4) refers to the balance of total generation with the total demand (P_D).

B.2 Results

The following five algorithms are applied to solve this problem as bellow

- Interior Point (IP)
- Genetic Algorithm (GA)
- Simulated Annealing (SA)
- Particle Swarm (PS)
- Differential Evolution (DE)

IP is used as an analytical approach while the other four are used as meta-heuristic techniques. The first three are implemented using Matlab Toolbox. Codes are generated for PS and DE. PS is based on the approach detailed in [1], while DE is developed based on [2].

The system under study is *New England test system* with the details given in [3]. The population in GA, PS and DE is taken to be 100. The convergence criterion is taken to be the maximum number of iterations and set to 1,000 (although other criteria may also be employed).

There are three main files developed as

- IP_SA_GA
- DE
- PS

There are seven functions generated with the details given in Table B.1.

Except IP, the other approaches are tried 10 times, using various initial populations. The results are summarized in Table B.2.

Table B.1 Details of the generated functions

Function name	Function description	Called by
call_gendata	Generation units data	IP_SA_GA, DE, PS
costfun	Calculation of total cost	IP and GA in IP_SA_GA
costfunsa	Calculation of the sum of total cost and the penalty function	SA in IP_SA_GA
call_objective	Calculation of the sum of total cost and the penalty function	DE, PS
cut2lim	Applying the generation limits	DE, PS
select_individual	Select individual for mutation	DE
discrete_recombination	Recombination or crossover operator	DE

Table B.2 The results of the different approaches

Method	Best	Average	Worst	Time (second)
SA	39167.16	41400.54	43835.36	11.80
PS	37140.93	38250.73	39790.21	0.84
GA	36931.36	37016.33	37120.11	71.00
DE	36842.22	36842.23	36842.23	0.85
IP	36842.22	–	–	0.90

B.3 Matlab Codes

In the following pages, the Matlab codes are given. It should be mentioned that no specific reason is used in choosing the above methods and based on the type of the problem, alternative algorithms may be tried. The reader is encouraged to try other algorithms for which some details are given in the chapter body.

a) "IP_SA_GA" M-file code

```
clc,clear all,close all
format compact

%       -----------  Solving Method   -------------
% Method -->
% 1:Interior Point
% 2:Genetic Algorithm
% 3:Simulated Annealing
SM=3;

%       ------------  Parameter set up ------------
 demand=6254.23;        % Total load
 pcf=1e4;               % Penalty coefficient
 gen_data=call_gendata; % Returns generator's data
 % Dimension of problem (here;number of units)
 dimnsn=10;
 mni=1000;             % Maximum number of iterations
 npop=100;             % Population size
 lb = gen_data(:,2)'; % Set lower bounds (Pmin in generator)
 ub = gen_data(:,3)'; % Set upper bounds (Pmax in generator)
 Aeq=ones(1,10);       % Equality constraint  P1+P2+...+P10=PD
 beq=demand;
 % Make a starting guess ---> ( Random making )
    x0 = (gen_data(:,3)+ (gen_data(:,2)-gen_data(:,3)).*...
        rand(dimnsn,1))';

% ------------ Switching to solving method  -----------------
if SM==1                 %------  Interior Point  ---------
        options = optimset('Algorithm','interior-point',...
            'Display','iter');
```

```
         [x,fval] = fmincon(@costfun,x0,[],[],Aeq,beq,lb,...
             ub,[],options);
         cost=fval;

elseif SM==2             %------  Genetic Algorithm  ------
      options =
gaoptimset('Generations',mni,'InitialPenalty'...
          ,pcf,'PopulationSize',npop,...
       'TimeLimit',inf,'StallGenLimit',inf,'PlotFcns',...
       @gaplotbestf,'Display','iter');
      [x,fval] = ga(@costfun,dimnsn,[],[],Aeq,beq,lb,...
          ub,[],options);
      cost=fval;

else                     %------  Simulated Annealing  ----
      options = saoptimset('MaxFunEvals',mni,'PlotFcns',...
          @saplotbestf,...
          'StallIterLimit',inf,'TimeLimit',inf,...
          'Display','iter');
      [x,fval] = simulannealbnd(@costfunsa,x0,...
          lb,ub,options);
      cost=fval-pcf*(sum(x)-demand)^2;

end

fprintf('\n'),display('Final solution is:'),Fs=x'
fprintf('\n'),display('Load that not served:'),...
    load_Mismatch=sum(x)-demand
fprintf('\n'),display('Associated cost:'),cost
```

b) "DE" M-file code

```
clc,clear all,close all
format compact

%------------  Parameter set up -------------

npop=100;          % Population size
mni=1000;          % Maximum number of iteration
dimnsn=10;         % Dimension of problem
                   % here;number of units)
demand=6254.23;    % Total load
pcf=1e4;           % Penalty coefficient
F=0.5;             % Mutation factor (scaling factor)
RR=0.9;            % Recombination (crossover) rate

%------------  Generator data   -------------
gen_data=call_gendata;      % Returns generator's data

% Make matrices the same size as population
% from vector Pmin & Pmax
    Pminrep=repmat(gen_data(:,2),1,npop);
    Pmaxrep=repmat(gen_data(:,3),1,npop);
```

```
% Randomly initialize population
    population=Pmaxrep+ (Pminrep-Pmaxrep) .* rand...
        (dimnsn,npop);

% Total objective function includes total cost and
% penalty function
     objective=call_objective(population,gen_data,pcf,demand);

% Determine best solution
 [objmin index_individual]=min(objective);bestsolution=...
     population(:,index_individual);

% -----------  Main loop:'Scheme DE/rand/1/bin'   ---------
for iter=1:mni

   iter
  % Select three different individuals for making
  % each trial vector
    slind=select_individual(npop);

  % Make trial vectors based 'DE/rand/1' : Mutation Operator
    trial_vectors=population(:,slind(1,:))+F*(population...
        (:,slind(2,:))-population(:,slind(3,:)));

  % Implement discrete recombination: Binomial crossover
    unew=discrete_recombination(population,trial_vectors,RR);

  % Limits on decision variables
    unew=cut2lim(unew,Pminrep,Pmaxrep);

  % Total objective function includes total cost
  % and penalty function
    objectivenew=call_objective(unew,gen_data,pcf,demand);

  % Deterministic selection
     replace=objectivenew<objective;
     objective(replace)=objectivenew(replace);population...
         (:,replace)=unew(:,replace);

 % Best solution so far:
   [objmin index_individual]=min(objective);bestsolution=...
      population(:,index_individual);

     evolution(iter)=objmin;

end

fprintf('\n'),display('Final solution is:'),bestsolution
fprintf('\n'),display('Load that not served:'),...
    load_Mismatch=sum(bestsolution)-demand
fprintf('\n'),display('Associated cost:'),cost=objmin-pcf*...
    (sum(bestsolution)-demand)^2

plot(1:mni,evolution)
```

c) "PS" M-file code

```
clc,clear all,close all
format compact

%------------  Parameter set up -------------

sws=100;              % Swarm  size (Population size)
mni=1000;             % Maximum number of iterations
dimnsn=10;            % Dimension of problem
                      % (here; number of units)
demand=6254.23;       % Total load
pcf=1e4;              % Penalty coefficient

%----- Generator data -----
gen_data=call_gendata;      % Returns generator's data

 % Range of decision (control) variables
    rangd=gen_data(:,3)-gen_data(:,2);

 % Make matrices the same size as population
 % from vector Pmin, Pmax & rangd
    Pminrep=repmat(gen_data(:,2),1,sws);
    Pmaxrep=repmat(gen_data(:,3),1,sws);
    rangdrep=repmat(rangd,1,sws);

% Position initialization
     position=Pminrep+ rangdrep .* rand(dimnsn,sws);

% Velocity initialization: it is assumed that maximum
% velocity is limited to 0.1*(Pmax-Pmin)
     velocity=0.1*rangdrep .*(1- 2*rand(dimnsn,sws));

% Total objective function includes total cost
% and penalty function
     objective=call_objective(position,gen_data,pcf,demand);

% Pbest & Gbest (initial assignment)
  [objmin index_particle]=min(objective);
  gbest=position(:,index_particle);gbest_objective=objmin;
  pbest=position;pbest_objective=objective;

% ------------  Main loop   --------------------
for iter=1:mni
    iter
    socialcom=rand(dimnsn,sws).*...
        (repmat(gbest,1,sws)-position); % Social component
    cognitivcom=rand(dimnsn,sws).*...
        (pbest-position);             % Cognitive component

 % Update velocity based on constriction(Clerc's) coefficient
    velocity=0.73*(velocity+2.05*cognitivcom+2.05*socialcom);

 % Limit velocity (step)
```

```
        velocity=cut2lim(velocity,-0.1*rangdrep,0.1*rangdrep);

     % Update position
        position=position+velocity;

     % Limits on decision variables
        position=cut2lim(position,Pminrep,Pmaxrep);

     % Total objective function includes
     % total cost and penalty function
        objective=call_objective(position,gen_data,pcf,demand);

     % Update pbest
        replace=objective<pbest_objective;
        pbest_objective(replace)=objective(replace);
        pbest(:,replace)=position(:,replace);

     % Update gbest
        [objmin index_particle]=min(pbest_objective);
        gbest=pbest(:,index_particle);
        gbest_objective=objmin;

        swarming(iter)=gbest_objective;
    end

    fprintf('\n'),display('Final solution is:'),gbest
    fprintf('\n'),display('Load that not served:'),...
        load_Mismatch=sum(gbest)-demand
    fprintf('\n'),display('Associated cost:'),...
        cost=gbest_objective-pcf*(sum(gbest)-demand)^2
    plot(1:mni,swarming)
```

d) "call_gendata" M-file code

```
function gen_data=call_gendata

%  Function 'call_data' returns generator's data
%  Pmin<P<Pmax  &   Cost=a*P^2+b*P+c

%  Unit     Pmin     Pmax          a       b         c
gen_data=[
    1        0        350          0.01     0.3     0.2
    2        0        1145.55      0.01     0.3     0.2
    3        0        750          0.01     0.3     0.2
    4        0        732          0.01     0.3     0.2
    5        0        608          0.01     0.3     0.2
    6        0        750          0.01     0.3     0.2
    7        0        660          0.01     0.3     0.2
    8        0        640          0.01     0.3     0.2
    9        0        930          0.006    0.3     0.2
    10       0        1100         0.006    0.3     0.2  ];
```

e) "costfun" M-file code

```
function cost=costfun(x)
```

```
% Function 'costfun' calculates total cost for GA & IP methods

% Recall generator's data
gen_data=call_gendata;
x=x';

% Calculate cost
cost=sum(gen_data(:,4).*x.^2+gen_data(:,5).*x+gen_data(:,6));
```

f) "costfunsa" M-file code

```
function objective=costfunsa(x)
% Objective function includes total cost
% and penalty function for SA method

% Recall generator's data
gen_data=call_gendata;
x=x';

% Calculate cost
cost=sum(gen_data(:,4).*x.^2+gen_data(:,5).*x+gen_data(:,6));

objective=cost+1e4*(sum(x)-6254.23).^2;
```

g) "call_objective" M-file code

```
function  objective=...
    call_objective(population,gen_data,pcf,demand)
% Total objective function includes total cost
% and penalty function for DE & PSO method

npop=size(population,2);

% Calculate total cost
for i=1:npop
  tcost(i)=sum(gen_data(:,4).*population(:,i).^2+...
      gen_data(:,5).*population(:,i)+gen_data(:,6));
end

% Load violation penalized by penalty function
    penalty=pcf*(sum(population)-demand).^2;

% Total objective function includes
% total cost and penalty function
    objective=tcost+penalty;
```

h) "cut2lim" M-file code

```
function x=cut2lim(x,xminmat,xmaxmat)

% Limits on decision variables

rmin=x<xminmat;
rmax=x>xmaxmat;
```

```
x(rmin)=xminmat(rmin);
x(rmax)=xmaxmat(rmax);
```

i) "select_individual" M-file code

```
function slind=select_individual(npop)

%  Select three different individuals
%  for making each trial vector for DE method
    slind=zeros(3,npop);

% First individual in mutation operator term
    slind(1,:)=randperm(npop);

% in order to select three different individuals
% for each trial vector, shift elements of first
% row in 'slind'. This method guarantees that all
% individuals will participate in making trial vectors

slind(2,1:npop-1)=slind(1,2:npop);slind(2,npop)=slind(1,1);
slind(3,1:npop-1)=slind(2,2:npop);slind(3,npop)=slind(2,1);
```

j) "discrete_recombination" M-file code

```
function unew=discrete_recombination(population,...
    trial_vectors,RR)%#ok<FNDEF>

[dimnsn,npop]=size(population);

% Those genes that replaced by genes of trial vectors
    genslct=rand(dimnsn,npop)<RR;

% Check at least one gene is replaced
    checknonzero=sum(genslct);

    for i=1:npop
        if ~checknonzero(i)
            genslct(fix(1+dimnsn*rand),i)=1;
        end
    end

    unew=population;

% Discrete recombination
    unew(genslct)= trial_vectors(genslct);
```

References

1. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73
2. Storn R, Price Price K (1997) Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim11(4):341–59
3. Zimmerman RD, Murillo-Sanchez CE, Gan D. MATPOWER: A MATLAB power system simulation package 2006. www.pserc.cornell.edu/matpower

Appendix C
AutoRegressive Moving Average (ARMA) Modeling

ARMA models are mathematical models of autocorrelation, in a time series. ARMA models can be used to predict behavior of a time series from past values alone. Such a prediction can be used as a baseline to evaluate possible importance of other variables to the system. An AR model expresses a time series as a linear function of its past values. The order of the AR model tells how many lagged past values are included. The simplest AR model is the first order autoregressive as follows

$$y_t + a_t y_{t-1} = e_t \tag{C.1}$$

or

$$y_t = -a_t y_{t-1} + e_t \tag{C.2}$$

where y_t is the mean-adjusted series in year (or time) t, y_{t-1} is the series in previous year, a_t is the lag-1 autoregressive coefficient and e_t is the noise. We can see that the model has the form of a regression model in which y_t is regressed on its previous value. The name autoregressive refers to the regression on self (auto).

Higher order AR models may also be assumed. A second order case is as follows

$$y_t + a_1 y_{t-1} + a_2 y_{t-2} = e_t \tag{C.3}$$

The Moving Average (MA) model is a form of ARMA model in which time series is regarded as a moving average (unevenly weighted) of a random shock noise e_t. A first order moving average model is given by

$$y_t = e_t + c_1 e_{t-1} \tag{C.4}$$

If we include both AR and MA, we reach at the ARMA model. A first order ARMA model is given by

$$y_t + a_1 y_{t-1} = e_t + c_1 e_{t-1} \tag{C.5}$$

For more details on ARMA modeling, refer to the references at the end of Chap. 4 and vast literature available on the subject.

Appendix D
What is EViews

EViews provides sophisticated data analysis, regression, and forecasting tools on Windows based computers. With EViews you can quickly develop a statistical relation from your data and then use the relation to forecast future values of the data. Areas where EViews can be useful include: scientific data analysis and evaluation, financial analysis, macroeconomic forecasting, simulation, sales forecasting, and cost analysis.

EViews is a new version of a set of tools for manipulating time series data originally developed in the Time Series Processor software for large computers. The immediate predecessor of EViews was MicroTSP, first released in 1981. Though EViews was developed by economists and most of its uses are in economics, there is nothing in its design that limits its usefulness to economic time series. Even quite large cross-section projects can be handled in EViews.

EViews provides convenient visual ways to enter data series from the keyboard or from disk files, to create new series from existing ones, to display and print series, and to carry out statistical analysis of the relationships among series.

EViews takes advantage of the visual features of modern Windows software. You can use your mouse to guide the operation with standard Windows menus and dialogs. Results appear in windows and can be manipulated with standard Windows techniques.

Alternatively, you may use EViews powerful command and batch processing language. You can enter and edit commands in the command window. You can create and store the commands in programs that document your research project for later execution.

Appendix E
The Calculations of the Reliability Indices

The analytical approach in calculating the reliability indices of a generation system may be, briefly, described as follows

- **Generation model.** A Capacity Outage Probability Table (COPT) should be, initially, generated in which various generation capacities as well as their respective probabilities are described. If the generation units are identical, a simple procedure is adopted to generate COPT. If the units are not similar, a recursive approach should be followed.
- **Load model.** The load may be described as Daily Peak Load Variation Curve (DPLVC) or Load Duration Curve (LDC). DPLVC is a cumulative representation of loads; descending order generated from the daily peak loads. LDC is generated from the hourly loads; descending order generated. DPLVC is widely used due to its simplicity. However, LDC shows a more practical representation of the load behavior.
- **Risk model.** The Loss of Load Expectation (LOLE) can be determined from convolving the generation and the load models. If DPLVC (LDC) is used as the load model, LOLE represents the expected days (*hours*) during a specific period in which the daily peak (*hourly*) load exceeds the generation capacity. According to Fig. E.1, for a generation outage of O_k; more than the available reserve, the load is lost for a period of $t_{k.}$

  Mathematically speaking, *LOLE* is calculated as follows

$$LOLE = \sum_{i=1}^{N} p_k t_k = \sum_{i=1}^{N} P_k (t_k - t_{k-1}) \tag{E.1}$$

where

N The number of cases for which the generation outage is more than the reserve available

p_k The probability of the generation outage O_k

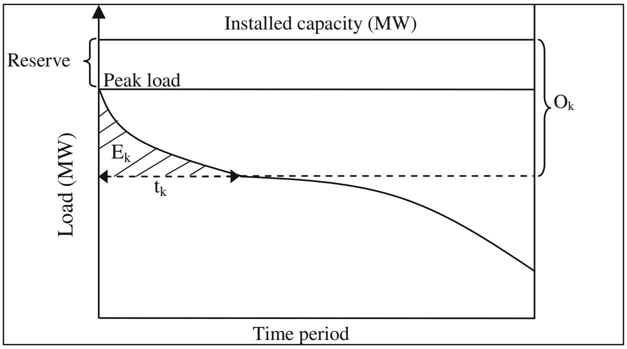

Fig. E.1 Relationship between capacity, load and reserve

t_k The period of lost load in generation outage O_k
P_k The cumulative probability of the generation outage O_k and more

If t_k is represented in per unit, the index calculated from (E.1) is called LOLP (Loss of Load Probability). The LOLP is expressed in terms of the average fraction of total time the system is expected to be in a state of failure. The area under an LDC shows the total energy demand. The Loss Of Expected Energy (LOEE) or the so called *Expected Energy Not Served* (EENS) or *Expected Unserved Energy* (EUE) may be calculated as

$$LOEE = \sum_{i=1}^{n} p_k E_k \tag{E.2}$$

where E_k is defined in Fig. E.1.

Example E.1 A generation system is composed of three units as follows

Unit 1: 10 MW, $FOR_1 = 1\%$
Unit 2: 20 MW, $FOR_2 = 2\%$
Unit 3: 60 MW, $FOR_3 = 3\%$

COPT is generated as shown in Table E.1. The probability of each capacity being *out* is FOR of its respective unit. Its probability being *in* is 1-FOR of its respective unit. For the LDC as shown in Fig. E.2, p_k, t_k and E_k (see (E.1)), should be determined for each row of Table E.1. Once done, (E.1) and (E.2) may be used to calculate LOLE and LOEE. As the reserve is 40 MW, the first four rows do not result in any lost load. Based on the results shown in Table E.2.

LOLE = 2.0298 (hours/100 hours)
LOLP = 0.020298
LOEE = 21.351 (MWh/100 hours)

Table E.1 The COPT of the generation system of the example

No.	Unit status (0:Out and 1:In)			Capacity (MW)		Probability	
	10 MW	20 MW	60 MW	In	Out	Individual	Cumulative
1	1	1	1	90	0	0.941094	1.000000
2	0	1	1	80	10	0.009506	0.058906
3	1	0	1	70	20	0.019206	0.049400
4	0	0	1	60	30	0.000194	0.030194
5	1	1	0	30	60	0.029106	0.030000
6	0	1	0	20	70	0.000294	0.000894
7	1	0	0	10	80	0.000594	0.000600
8	0	0	0	0	90	0.000006	0.000006

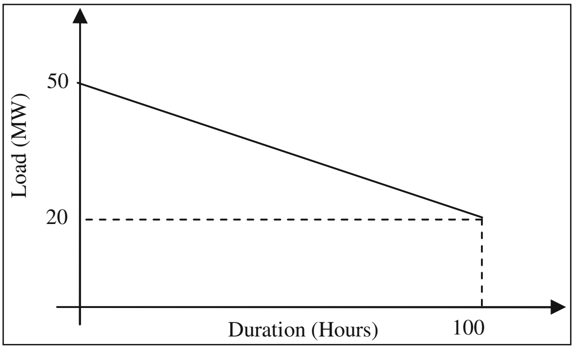

Fig. E.2 The load model (LDC) of the example

Table E.2 The required parameters for reliability indices calculation

No.	t_k	E_k	$p_k \times t_k$	$p_k \times E_k$	t_k/T	$p_k \times t_k/T$	t_k-t_{k-1}	$P_k \times (t_k$-$t_{k-1})$
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	66.67	666.67	1.9404	19.404	0.67	0.019440	66.67	2
6	100	1500	0.0294	0.441	1	0.000294	33.34	0.03
7	100	2500	0.0594	1.485	1	0.000594	0	0
8	100	3500	0.0006	0.021	1	0.000006	0	0

Note that the total energy demand is 3500 MWh; calculated from the area under LDC.

Appendix F
Garver Test System Data

In this book, Garver test system is used in Chaps. 6, 8 and 9 to describe generation and transmission network planning problems. The relevant data of this system are provided in current appendix. The base case, as used in Chaps. 6 and 8, is described in Sect. F.1. The modified case, as used in Chap. 9, is described in Sect. F.2.

F.1 The Base Case

The base Garver test system is shown in Fig. F.1, with the details given in Tables F.1 and F.2.

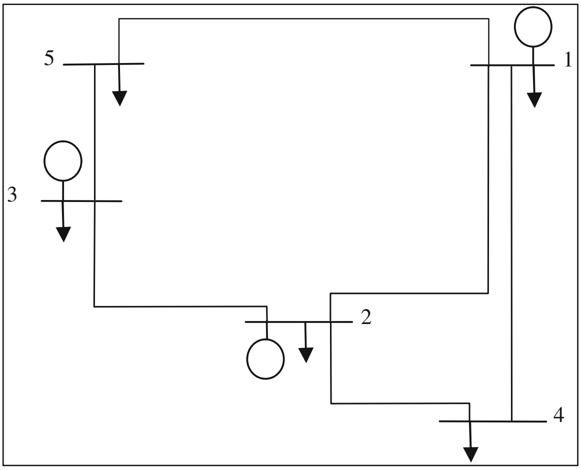

Fig. F.1 Garver test system

Table F.1 Network data[a]

Line no.	Bus		R (p.u.)	X (p.u.)	Capacity limit (p.u.)	Path length (km)
	From	To				
1	1	2	0.1000	0.40	1.0	400.0
2	1	4	0.1500	0.60	0.8	600.0
3	1	5	0.0500	0.20	1.0	200.0
4	2	3	0.0500	0.20	1.0	200.0
5	2	4	0.1000	0.40	1.0	400.0
6	3	5	0.0500	0.20	1.0	200.0
7	1	3	0.0950	0.38	1.0	380.0
8	2	5	0.0775	0.31	1.0	310.0
9	3	4	0.1475	0.59	0.8	590.0
10	4	5	0.1575	0.63	0.8	630.0

[a] It should be mentioned that some lines (7 through 10) are used as candidates in some places; while still some candidates may be considered in the some corridors of existing lines (1 through 6)

Table F.2 Generation and load data

Bus	Load		Generation
	P_D (p.u.)	Q_D (p.u.)	P_G (p.u.)
1	0.240	0.116	1.130
2	0.720	0.348	0.500
3	0.120	0.058	0.650
4	0.480	0.232	-
5	0.720	0.348	-

F.2 The Modified Case

A modified Garver test system is shown in Fig. F.2 in which two voltage levels are used to assess the algorithm proposed mainly in Chap. 9. The relevant data are provided in Tables F.3 and F.4.

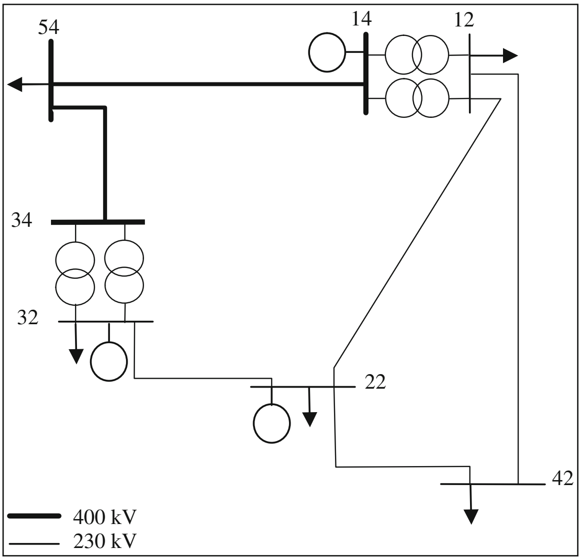

Fig. F.2 Modified Garver test system

Table F.3 Network data

Line no.	Bus		R (p.u.)	X (p.u.)	B (p.u.)	Capacity limit (p.u.)
	From	To				
1	12	22	0.10	0.40	0.8	1.0
2	12	42	0.15	0.60	1.2	0.8
3	14	54	0.05	0.20	0.4	1.0
4	22	32	0.05	0.20	0.4	1.0
5	22	42	0.10	0.40	0.8	1.0
6	34	54	0.05	0.20	0.4	1.0

Table F.4 Generation and load data

Bus	Load		Generation
	P_D (p.u.)	Q_D (p.u.)	P_G (p.u.)
12	0.240	0.116	-
22	0.720	0.348	0.500
32	0.120	0.058	0.650
42	0.480	0.232	-
14	-	-	1.13
54	0.720	0.348	-

Appendix G
Geographical Information System

A Geographical Information System (GIS) is a system of hardware, software and procedures to facilitate the management, manipulation, analysis, modeling, representation and display of *georeferenced* data to solve complex problems regarding planning and management of resources.

The georeferenced data or information is the geographic information identified according to locations (an alternative term is *spatial* data or information). In other words, the information; normally in digital form, is linked to specific places in the earth, using earth coordinates (such as latitude/longitude). In this way, a *layer* (also known as theme) may be formed, consisting of geographic data linked to descriptive, or tabular information. For various types of information, different layers may thus be created. The layers may then be combined as required to perform analyses.

GIS has found widespread use in many decision making activities in various desciplines. It may be used in both daily operation or long term planning of a system in which the decision making is, somehow, related to the geography. The issues referred to in this book, are mainly related to long term planning of a power system. As detailed in some chapters, the geographical information of load points, existing and candidate substations, transmission lines routes, etc. are used in some types of decision makings in GEP, SEP and NEP problems. So, if GIS is used, it can mathematically transform map features from one scale or projection to another, to allow map layers from different sources to be used together. If information is created through a GIS, it is quite simple to update the data on the computer to generate an updated product.

For data manipulation and storage in layers, two models, namely, *raster data model* and *vector data model* may be used. In the former, the region under study is divided into small regular blocks, with each block having a specific value attached to it. In the latter, all objects of interest are described in terms of geometric elements such as points, lines, polygons, etc.

While raster data are best used for representing continuous variables (such as elevations) and all satellite and aerial photograph data come in raster form, the vector data are very widely used in analysis of networks and municipal data bases (containing description of buildings, streets, etc.).

Briefly speaking, the GIS functions are as follows

- Capture
- Store
- Query
- Analyze
- Output

Capturing may be performed using hardcopy maps, Global Positioning Systems (GPS), digital data from some sources such as satellites, aerial photography, etc.

Storing can be carried out using one of the techniques already described (raster and vector).

Query may come in two forms. One is looking to identify or find features of interest of some points on the map. The other tries to identify the features based on some specific conditions (for instance, identifying the stream with the longest length and in the southern province).

Analysis of any type of data involves searching for patterns within one variable and relationships between two or more variables. For example, we can say that census tract A is next to census tract B, and both adjoin tract C; that city A is 100 km northwest of city B; that my house is on the same street as yours, etc.

Output may be in the form of paper/hardeopy files, map digital files, images, etc.

There are vast literatures on GIS. Instead of introducing some to the reader, we encourage him or her to search for the relevant materials in the form of books, tutorials, websites, etc.

Appendix H
84-Bus Test System Data

The relevant data of the 84-bus test system, as reported in Sect. 8.6.2, are provided in current appendix. This is a single voltage level network with detailed information as below

- Bus data are provided in Table H.1.
- Line data are provided in Table H.2.
- Candidate lines data are provided in Table H.3.
- Generation data are provided in Table H.4.

Table H.1 Bus data

No.	Bus name	X[a]	Y[a]	Area no.	P_D (p.u.)	No.	Bus name	X	Y	Area no.	P_D (p.u.)
1	B 1V4	47.65	37.19	1	0.36	19	B 19V4	50.90	35.42	4	0.00
2	B 2V4	46.17	38.08	1	0.00	20	B 20V4	51.38	35.75	1	5.66
3	B 3V4	54.90	36.93	1	6.28	21	B 21V4	51.57	35.75	1	4.50
4	B 4V4	51.20	36.50	1	2.08	22	B 22V4	51.57	35.75	1	4.50
5	B 5V4	52.63	36.35	1	5.40	23	B 23V4	51.65	35.33	1	2.05
6	B 6V4	53.25	36.82	1	6.26	24	B 24V4	51.13	35.75	1	7.86
7	B 7V4	53.43	35.60	2	3.25	25	B 25V4	50.47	36.10	1	6.56
8	B 8V4	54.87	36.42	2	2.23	26	B 26V4	57.40	37.05	2	1.76
9	B 9V4	51.87	35.43	1	0.00	27	B 27V4	59.40	36.42	2	0.00
10	B 10V4	51.30	35.62	4	6.90	28	B 28V4	58.68	36.28	2	1.08
11	B 11V4	51.30	35.62	4	6.83	29	B 29V4	59.02	33.75	2	2.63
12	B 12V4	51.85	35.42	1	0.00	30	B 30V4	58.77	36.20	2	3.71
13	B 13V4	51.28	35.77	1	6.53	31	B 31V4	57.75	36.25	2	0.27
14	B 14V4	50.90	35.42	4	0.00	32	B 32V4	59.08	35.18	2	3.66
15	B 15V4	51.83	35.75	1	2.60	33	B 33V4	57.93	37.40	2	2.27
16	B 16V4	50.32	36.15	1	0.00	34	B 34V4	60.65	35.23	2	2.53
17	B 17V4	50.32	36.15	1	0.00	35	B 35V4	59.40	36.42	2	1.42
18	B 18V4	51.58	35.52	1	8.10	36	B 36V4	54.38	31.81	2	3.47

(continued)

Table H.1 (continued)

No.	Bus name	X[a]	Y[a]	Area no.	P_D (p.u.)	No.	Bus name	X	Y	Area no.	P_D (p.u.)
37	B 37V4	54.17	31.90	2	3.39	61	B 61V4	50.87	32.24	2	3.19
38	B 38V4	48.28	30.45	4	5.22	62	B 62V4	51.22	32.49	2	0.00
39	B 39V4	49.60	32.05	4	0.00	63	B 63V4	51.31	32.41	2	1.05
40	B 40V4	48.82	31.30	4	6.64	64	B 64V4	52.71	27.45	3	1.34
41	B 41V4	48.67	31.45	4	2.77	65	B 65V4	52.61	27.45	3	0.00
42	B 42V4	48.35	32.47	4	3.04	66	B 66V4	50.92	28.83	3	3.36
43	B 43V4	49.37	32.02	4	1.86	67	B 67V4	51.02	28.98	4	0.00
44	B 44V4	48.08	30.37	4	2.20	68	B 68V4	51.02	28.98	4	2.54
45	B 45V4	48.12	32.50	4	0.00	69	B 69V4	53.67	29.08	3	1.90
46	B 46V4	49.98	31.93	3	0.00	70	B 70V4	52.05	27.83	3	2.15
47	B 47V4	49.25	30.58	4	4.08	71	B 71V4	51.72	29.52	3	0.00
48	B 48V4	49.70	30.80	3	8.06	72	B 72V4	54.32	29.20	2	1.59
49	B 49V4	49.68	30.85	4	0.00	73	B 73V4	52.45	29.58	3	2.58
50	B 50V4	48.75	32.15	4	4.85	74	B 74V4	52.83	31.00	3	2.27
51	B 51V4	51.37	30.60	3	1.11	75	B 75V4	45.54	34.74	4	1.38
52	B 52V4	49.83	34.00	4	3.47	76	B 76V4	46.60	34.12	4	2.67
53	B 53V4	48.22	33.43	4	4.35	77	B 77V4	47.35	34.35	4	0.68
54	B 54V4	48.87	35.13	1	0.00	78	B 78V4	56.11	27.15	2	0.00
55	B 55V4	51.47	32.25	2	4.56	79	B 79V4	56.00	28.32	2	0.26
56	B 56V4	51.49	32.80	2	2.29	80	B 80V4	54.30	27.02	2	2.62
57	B 57V4	50.32	33.41	2	2.48	81	B 81V4	48.58	36.65	1	3.97
58	B 58V4	51.47	32.25	2	0.00	82	B 82V4	56.78	30.23	2	2.33
59	B 59V4	51.33	32.59	2	2.97	83	B 83V4	55.75	29.43	2	4.03
60	B 60V4	51.42	32.25	2	8.51	84	B 84V4	49.63	37.18	1	0.00

[a] Geographical characteristics

Table H.2 Line data

No.	From bus	To bus	R (p.u.)	X (p.u.)	$\bar{P}_L$ (p.u.)	No.	From bus	To bus	R (p.u.)	X (p.u.)	$\bar{P}_L$ (p.u.)
1	B 1V4	B 81V4	0.0022	0.0258	15.0	15	B 9V4	B 12V4	0.0000	0.0002	13.9
2	B 2V4	B 16V4	0.0092	0.0927	11.7	16	B 9V4	B 15V4	0.0007	0.0089	18.2
3	B 3V4	B 6V4	0.0029	0.0326	15.0	17	B 9V4	B 15V4	0.0007	0.0089	18.2
4	B 3V4	B 6V4	0.0029	0.0326	15.0	18	B 9V4	B 21V4	0.0008	0.0113	12.5
5	B 3V4	B 26V4	0.0054	0.0567	14.7	19	B 9V4	B 22V4	0.0008	0.0113	12.5
6	B 4V4	B 5V4	0.0024	0.0278	15.0	20	B 10V4	B 14V4	0.0006	0.0083	15.3
7	B 4V4	B 25V4	0.0023	0.0264	11.7	21	B 11V4	B 19V4	0.0006	0.0085	18.4
8	B 5V4	B 6V4	0.0016	0.0181	15.0	22	B 12V4	B 18V4	0.0006	0.0057	15.1
9	B 6V4	B 7V4	0.0027	0.0315	15.0	23	B 12V4	B 19V4	0.0010	0.0164	22.2
10	B 6V4	B 12V4	0.0040	0.0550	9.1	24	B 12V4	B 19V4	0.0010	0.0164	22.2
11	B 7V4	B 8V4	0.0028	0.0319	15.0	25	B 12V4	B 23V4	0.0004	0.0045	15.0
12	B 7V4	B 12V4	0.0034	0.0317	9.0	26	B 12V4	B 57V4	0.0060	0.0633	10.7
13	B 8V4	B 31V4	0.0056	0.0588	14.7	27	B 13V4	B 20V4	0.0002	0.0021	16.6
14	B 9V4	B 12V4	0.0000	0.0002	9.9	28	B 13V4	B 25V4	0.0013	0.0180	11.7

(continued)

Table H.2 (continued)

No.	From bus	To bus	R (p.u.)	X (p.u.)	$\bar{P}_L$ (p.u.)	No.	From bus	To bus	R (p.u.)	X (p.u.)	$\bar{P}_L$ (p.u.)
29	B 14V4	B 19V4	0.0000	0.0002	16.6	73	B 40V4	B 49V4	0.0010	0.0177	27.1
30	B 15V4	B 24V4	0.0009	0.0126	4.8	74	B 41V4	B 50V4	0.0015	0.0167	15.0
31	B 15V4	B 24V4	0.0009	0.0126	4.8	75	B 42V4	B 43V4	0.0023	0.0265	15.0
32	B 16V4	B 17V4	0.0000	0.0002	16.1	76	B 42V4	B 53V4	0.0024	0.0275	15.0
33	B 16V4	B 19V4	0.0013	0.0185	16.8	77	B 43V4	B 46V4	0.0014	0.0165	15.0
34	B 16V4	B 24V4	0.0013	0.0177	18.2	78	B 43V4	B 50V4	0.0012	0.0134	22.0
35	B 16V4	B 25V4	0.0003	0.0044	24.5	79	B 43V4	B 57V4	0.0051	0.0583	15.0
36	B 16V4	B 25V4	0.0003	0.0044	24.5	80	B 45V4	B 50V4	0.0017	0.0196	15.0
37	B 16V4	B 84V4	0.0032	0.0340	10.7	81	B 45V4	B 53V4	0.0025	0.0291	15.0
38	B 17V4	B 19V4	0.0013	0.0185	10.7	82	B 46V4	B 48V4	0.0020	0.0306	22.5
39	B 17V4	B 24V4	0.0013	0.0177	18.2	83	B 46V4	B 55V4	0.0021	0.0319	22.5
40	B 18V4	B 23V4	0.0004	0.0045	15.0	84	B 46V4	B 62V4	0.0019	0.0292	22.5
41	B 19V4	B 52V4	0.0035	0.0395	7.6	85	B 48V4	B 49V4	0.0002	0.0019	14.8
42	B 19V4	B 52V4	0.0035	0.0395	7.6	86	B 49V4	B 67V4	0.0023	0.0402	27.1
43	B 20V4	B 24V4	0.0004	0.0050	16.6	87	B 49V4	B 68V4	0.0032	0.0477	22.5
44	B 24V4	B 25V4	0.0010	0.0147	16.6	88	B 51V4	B 67V4	0.0013	0.0198	22.5
45	B 26V4	B 30V4	0.0031	0.0325	14.7	89	B 52V4	B 57V4	0.0016	0.0181	15.0
46	B 26V4	B 31V4	0.0029	0.0304	14.7	90	B 53V4	B 77V4	0.0026	0.0299	15.0
47	B 26V4	B 33V4	0.0014	0.0151	14.7	91	B 54V4	B 77V4	0.0031	0.0350	15.0
48	B 27V4	B 35V4	0.0000	0.0001	22.7	92	B 54V4	B 81V4	0.0032	0.0371	15.0
49	B 28V4	B 30V4	0.0003	0.0026	14.7	93	B 55V4	B 58V4	0.0000	0.0002	15.0
50	B 29V4	B 32V4	0.0035	0.0370	14.7	94	B 55V4	B 58V4	0.0000	0.0002	15.0
51	B 29V4	B 34V4	0.0048	0.0507	14.7	95	B 55V4	B 60V4	0.0002	0.0023	15.0
52	B 30V4	B 32V4	0.0028	0.0290	14.7	96	B 55V4	B 62V4	0.0009	0.0107	15.0
53	B 30V4	B 35V4	0.0014	0.0144	14.7	97	B 56V4	B 57V4	0.0024	0.0276	15.0
54	B 34V4	B 35V4	0.0039	0.0410	14.7	98	B 56V4	B 62V4	0.0013	0.0148	15.0
55	B 36V4	B 37V4	0.0004	0.0071	27.1	99	B 57V4	B 62V4	0.0026	0.0295	15.0
56	B 36V4	B 37V4	0.0004	0.0071	27.1	100	B 58V4	B 74V4	0.0024	0.0360	22.5
57	B 36V4	B 82V4	0.0036	0.0540	22.0	101	B 59V4	B 62V4	0.0004	0.0049	15.0
58	B 36V4	B 83V4	0.0030	0.0531	27.1	102	B 59V4	B 62V4	0.0004	0.0041	15.0
59	B 37V4	B 56V4	0.0029	0.0517	27.1	103	B 60V4	B 62V4	0.0008	0.0087	15.0
60	B 37V4	B 58V4	0.0032	0.0486	22.5	104	B 61V4	B 62V4	0.0008	0.0093	15.0
61	B 38V4	B 41V4	0.0023	0.0268	15.0	105	B 62V4	B 63V4	0.0003	0.0031	15.0
62	B 38V4	B 44V4	0.0038	0.0436	15.0	106	B 62V4	B 63V4	0.0003	0.0031	15.0
63	B 38V4	B 47V4	0.0016	0.0185	15.0	107	B 64V4	B 65V4	0.0000	0.0002	15.0
64	B 39V4	B 40V4	0.0034	0.0284	12.8	108	B 64V4	B 70V4	0.0029	0.0237	16.0
65	B 39V4	B 46V4	0.0011	0.0122	15.0	109	B 65V4	B 69V4	0.0014	0.0216	22.5
66	B 39V4	B 49V4	0.0037	0.0304	12.8	110	B 65V4	B 80V4	0.0040	0.0328	16.0
67	B 39V4	B 52V4	0.0047	0.0544	15.0	111	B 66V4	B 68V4	0.0003	0.0045	22.5
68	B 39V4	B 57V4	0.0048	0.0550	15.0	112	B 66V4	B 68V4	0.0003	0.0045	22.5
69	B 39V4	B 61V4	0.0025	0.0291	15.0	113	B 67V4	B 68V4	0.0015	0.0225	22.5
70	B 40V4	B 41V4	0.0004	0.0051	18.0	114	B 67V4	B 71V4	0.0007	0.0108	22.5
71	B 40V4	B 43V4	0.0020	0.0227	18.0	115	B 67V4	B 73V4	0.0010	0.0177	27.1
72	B 40V4	B 47V4	0.0008	0.0112	18.0	116	B 68V4	B 70V4	0.0044	0.0364	16.0

(continued)

Table H.2 (continued)

No.	From bus	To bus	R (p.u.)	X (p.u.)	$\bar{P}_L$ (p.u.)	No.	From bus	To bus	R (p.u.)	X (p.u.)	$\bar{P}_L$ (p.u.)
117	B 69V4	B 72V4	0.0008	0.0143	27.1	123	B 78V4	B 79V4	0.0031	0.0340	14.4
118	B 69V4	B 73V4	0.0014	0.0253	27.1	124	B 78V4	B 80V4	0.0048	0.0414	16.0
119	B 71V4	B 73V4	0.0024	0.0200	16.0	125	B 78V4	B 83V4	0.0031	0.0552	27.1
120	B 72V4	B 83V4	0.0016	0.0283	27.1	126	B 79V4	B 83V4	0.0027	0.0302	14.4
121	B 75V4	B 76V4	0.0036	0.0412	15.0	127	B 81V4	B 84V4	0.0023	0.0268	15.0
122	B 76V4	B 77V4	0.0011	0.0171	22.5	128	B 82V4	B 83V4	0.0027	0.0309	15.0

Table H.3 Candidate lines data[a]

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
1	B 14V4	B 19V4	30	B 12V4	B 23V4	59	B 60V4	B 62V4	88	B 56V4	B 63V4
2	B 21V4	B 22V4	31	B 10V4	B 24V4	60	B 15V4	B 18V4	89	B 21V4	B 23V4
3	B 10V4	B 11V4	32	B 11V4	B 24V4	61	B 55V4	B 62V4	90	B 22V4	B 23V4
4	B 27V4	B 35V4	33	B 38V4	B 44V4	62	B 58V4	B 62V4	91	B 18V4	B 24V4
5	B 55V4	B 58V4	34	B 42V4	B 45V4	63	B 9V4	B 15V4	92	B 15V4	B 23V4
6	B 67V4	B 68V4	35	B 40V4	B 41V4	64	B 12V4	B 15V4	93	B 47V4	B 48V4
7	B 16V4	B 17V4	36	B 39V4	B 43V4	65	B 39V4	B 46V4	94	B 13V4	B 15V4
8	B 9V4	B 12V4	37	B 18V4	B 23V4	66	B 59V4	B 60V4	95	B 10V4	B 15V4
9	B 58V4	B 60V4	38	B 36V4	B 37V4	67	B 13V4	B 18V4	96	B 11V4	B 15V4
10	B 55V4	B 60V4	39	B 20V4	B 24V4	68	B 22V4	B 24V4	97	B 47V4	B 49V4
11	B 48V4	B 49V4	40	B 9V4	B 23V4	69	B 21V4	B 24V4	98	B 60V4	B 61V4
12	B 13V4	B 20V4	41	B 55V4	B 63V4	70	B 58V4	B 59V4	99	B 42V4	B 50V4
13	B 64V4	B 65V4	42	B 58V4	B 63V4	71	B 55V4	B 59V4	100	B 13V4	B 19V4
14	B 28V4	B 30V4	43	B 15V4	B 22V4	72	B 15V4	B 20V4	101	B 13V4	B 14V4
15	B 62V4	B 63V4	44	B 15V4	B 21V4	73	B 19V4	B 24V4	102	B 20V4	B 23V4
16	B 13V4	B 24V4	45	B 18V4	B 21V4	74	B 14V4	B 24V4	103	B 11V4	B 12V4
17	B 16V4	B 25V4	46	B 18V4	B 22V4	75	B 11V4	B 14V4	104	B 10V4	B 12V4
18	B 17V4	B 25V4	47	B 13V4	B 22V4	76	B 10V4	B 19V4	105	B 9V4	B 10V4
19	B 59V4	B 62V4	48	B 13V4	B 21V4	77	B 11V4	B 19V4	106	B 9V4	B 11V4
20	B 11V4	B 20V4	49	B 12V4	B 18V4	78	B 10V4	B 14V4	107	B 12V4	B 20V4
21	B 10V4	B 20V4	50	B 10V4	B 18V4	79	B 56V4	B 62V4	108	B 58V4	B 61V4
22	B 10V4	B 13V4	51	B 11V4	B 18V4	80	B 61V4	B 62V4	109	B 55V4	B 61V4
23	B 11V4	B 13V4	52	B 56V4	B 59V4	81	B 12V4	B 22V4	110	B 3V4	B 8V4
24	B 20V4	B 22V4	53	B 9V4	B 18V4	82	B 12V4	B 21V4	111	B 9V4	B 20V4
25	B 20V4	B 21V4	54	B 11V4	B 21V4	83	B 9V4	B 22V4	112	B 19V4	B 20V4
26	B 66V4	B 68V4	55	B 11V4	B 22V4	84	B 9V4	B 21V4	113	B 14V4	B 20V4
27	B 66V4	B 67V4	56	B 10V4	B 21V4	85	B 10V4	B 23V4	114	B 59V4	B 61V4
28	B 59V4	B 63V4	57	B 10V4	B 22V4	86	B 11V4	B 23V4	115	B 43V4	B 46V4
29	B 60V4	B 63V4	58	B 18V4	B 20V4	87	B 61V4	B 63V4	116	B 13V4	B 23V4

(continued)

Table H.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
117	B 43V4	B 50V4	161	B 16V4	B 24V4	205	B 21V4	B 25V4	249	B 43V4	B 45V4
118	B 26V4	B 33V4	162	B 12V4	B 14V4	206	B 22V4	B 25V4	250	B 17V4	B 84V4
119	B 55V4	B 56V4	163	B 12V4	B 19V4	207	B 42V4	B 53V4	251	B 16V4	B 84V4
120	B 56V4	B 58V4	164	B 9V4	B 19V4	208	B 38V4	B 40V4	252	B 53V4	B 77V4
121	B 56V4	B 60V4	165	B 9V4	B 14V4	209	B 42V4	B 43V4	253	B 40V4	B 46V4
122	B 27V4	B 30V4	166	B 4V4	B 17V4	210	B 66V4	B 71V4	254	B 78V4	B 79V4
123	B 30V4	B 35V4	167	B 4V4	B 16V4	211	B 5V4	B 7V4	255	B 69V4	B 73V4
124	B 14V4	B 18V4	168	B 4V4	B 21V4	212	B 39V4	B 41V4	256	B 5V4	B 20V4
125	B 18V4	B 19V4	169	B 4V4	B 22V4	213	B 81V4	B 84V4	257	B 57V4	B 59V4
126	B 15V4	B 24V4	170	B 40V4	B 47V4	214	B 39V4	B 40V4	258	B 5V4	B 18V4
127	B 69V4	B 72V4	171	B 68V4	B 71V4	215	B 41V4	B 47V4	259	B 46V4	B 62V4
128	B 12V4	B 13V4	172	B 67V4	B 71V4	216	B 4V4	B 18V4	260	B 57V4	B 62V4
129	B 9V4	B 13V4	173	B 46V4	B 61V4	217	B 44V4	B 47V4	261	B 41V4	B 44V4
130	B 23V4	B 24V4	174	B 20V4	B 25V4	218	B 5V4	B 21V4	262	B 82V4	B 83V4
131	B 27V4	B 28V4	175	B 41V4	B 43V4	219	B 5V4	B 22V4	263	B 43V4	B 49V4
132	B 28V4	B 35V4	176	B 30V4	B 31V4	220	B 30V4	B 32V4	264	B 4V4	B 9V4
133	B 14V4	B 23V4	177	B 14V4	B 15V4	221	B 41V4	B 49V4	265	B 16V4	B 18V4
134	B 19V4	B 23V4	178	B 15V4	B 19V4	222	B 38V4	B 41V4	266	B 17V4	B 18V4
135	B 65V4	B 70V4	179	B 10V4	B 25V4	223	B 41V4	B 42V4	267	B 4V4	B 12V4
136	B 14V4	B 21V4	180	B 11V4	B 25V4	224	B 46V4	B 50V4	268	B 39V4	B 49V4
137	B 14V4	B 22V4	181	B 38V4	B 47V4	225	B 18V4	B 25V4	269	B 41V4	B 46V4
138	B 19V4	B 21V4	182	B 26V4	B 31V4	226	B 75V4	B 76V4	270	B 46V4	B 63V4
139	B 19V4	B 22V4	183	B 40V4	B 50V4	227	B 17V4	B 21V4	271	B 4V4	B 23V4
140	B 45V4	B 50V4	184	B 40V4	B 43V4	228	B 17V4	B 22V4	272	B 6V4	B 7V4
141	B 71V4	B 73V4	185	B 40V4	B 49V4	229	B 16V4	B 21V4	273	B 23V4	B 25V4
142	B 24V4	B 25V4	186	B 13V4	B 17V4	230	B 16V4	B 22V4	274	B 5V4	B 13V4
143	B 76V4	B 77V4	187	B 13V4	B 16V4	231	B 39V4	B 61V4	275	B 40V4	B 42V4
144	B 12V4	B 24V4	188	B 16V4	B 19V4	232	B 41V4	B 48V4	276	B 43V4	B 48V4
145	B 9V4	B 24V4	189	B 17V4	B 19V4	233	B 4V4	B 19V4	277	B 39V4	B 48V4
146	B 5V4	B 6V4	190	B 14V4	B 16V4	234	B 4V4	B 14V4	278	B 57V4	B 61V4
147	B 64V4	B 70V4	191	B 14V4	B 17V4	235	B 5V4	B 9V4	279	B 46V4	B 60V4
148	B 41V4	B 50V4	192	B 5V4	B 15V4	236	B 46V4	B 49V4	280	B 27V4	B 32V4
149	B 4V4	B 25V4	193	B 4V4	B 11V4	237	B 51V4	B 71V4	281	B 32V4	B 35V4
150	B 52V4	B 57V4	194	B 4V4	B 10V4	238	B 5V4	B 12V4	282	B 72V4	B 83V4
151	B 39V4	B 50V4	195	B 40V4	B 48V4	239	B 40V4	B 44V4	283	B 38V4	B 49V4
152	B 4V4	B 13V4	196	B 4V4	B 15V4	240	B 79V4	B 83V4	284	B 28V4	B 33V4
153	B 13V4	B 25V4	197	B 1V4	B 81V4	241	B 39V4	B 42V4	285	B 38V4	B 48V4
154	B 28V4	B 31V4	198	B 45V4	B 53V4	242	B 28V4	B 32V4	286	B 25V4	B 84V4
155	B 4V4	B 24V4	199	B 16V4	B 20V4	243	B 41V4	B 45V4	287	B 7V4	B 9V4
156	B 4V4	B 20V4	200	B 17V4	B 20V4	244	B 56V4	B 57V4	288	B 26V4	B 28V4
157	B 19V4	B 25V4	201	B 11V4	B 17V4	245	B 46V4	B 48V4	289	B 32V4	B 34V4
158	B 14V4	B 25V4	202	B 10V4	B 16V4	246	B 15V4	B 25V4	290	B 15V4	B 16V4
159	B 56V4	B 61V4	203	B 10V4	B 17V4	247	B 31V4	B 33V4	291	B 15V4	B 17V4
160	B 17V4	B 24V4	204	B 11V4	B 16V4	248	B 4V4	B 5V4	292	B 43V4	B 61V4

(continued)

Table H.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
293	B 5V4	B 23V4	338	B 67V4	B 70V4	380	B 43V4	B 57V4	425	B 71V4	B 74V4
294	B 7V4	B 12V4	339	B 1V4	B 2V4	381	B 78V4	B 80V4	426	B 3V4	B 7V4
295	B 57V4	B 63V4	339	B 1V4	B 2V4	382	B 25V4	B 81V4	427	B 6V4	B 9V4
296	B 5V4	B 10V4	339	B 1V4	B 2V4	383	B 25V4	B 54V4	428	B 53V4	B 54V4
297	B 5V4	B 11V4	339	B 1V4	B 2V4	384	B 47V4	B 50V4	429	B 46V4	B 51V4
298	B 46V4	B 55V4	340	B 54V4	B 77V4	385	B 43V4	B 62V4	430	B 70V4	B 73V4
299	B 46V4	B 58V4	341	B 49V4	B 51V4	386	B 51V4	B 67V4	431	B 43V4	B 58V4
300	B 7V4	B 15V4	342	B 7V4	B 23V4	387	B 51V4	B 68V4	432	B 43V4	B 55V4
301	B 12V4	B 25V4	343	B 64V4	B 80V4	388	B 51V4	B 60V4	433	B 50V4	B 61V4
302	B 51V4	B 74V4	344	B 42V4	B 46V4	389	B 51V4	B 58V4	434	B 6V4	B 12V4
303	B 46V4	B 59V4	345	B 57V4	B 60V4	390	B 51V4	B 55V4	435	B 39V4	B 53V4
304	B 9V4	B 25V4	346	B 16V4	B 81V4	391	B 14V4	B 52V4	436	B 5V4	B 8V4
305	B 3V4	B 6V4	347	B 17V4	B 81V4	392	B 19V4	B 52V4	437	B 30V4	B 34V4
306	B 39V4	B 45V4	348	B 46V4	B 47V4	393	B 7V4	B 20V4	438	B 51V4	B 63V4
307	B 40V4	B 45V4	349	B 39V4	B 57V4	394	B 72V4	B 73V4	439	B 51V4	B 66V4
308	B 31V4	B 35V4	350	B 39V4	B 63V4	395	B 45V4	B 46V4	440	B 50V4	B 57V4
309	B 27V4	B 31V4	351	B 39V4	B 47V4	396	B 4V4	B 6V4	441	B 47V4	B 51V4
310	B 5V4	B 24V4	352	B 7V4	B 18V4	397	B 5V4	B 14V4	442	B 38V4	B 43V4
311	B 50V4	B 53V4	353	B 46V4	B 57V4	398	B 5V4	B 19V4	443	B 52V4	B 56V4
312	B 17V4	B 23V4	354	B 55V4	B 57V4	399	B 14V4	B 54V4	444	B 64V4	B 69V4
313	B 16V4	B 23V4	355	B 57V4	B 58V4	400	B 19V4	B 54V4	445	B 69V4	B 83V4
314	B 6V4	B 8V4	356	B 53V4	B 76V4	401	B 43V4	B 63V4	446	B 6V4	B 20V4
315	B 30V4	B 33V4	357	B 7V4	B 21V4	402	B 51V4	B 61V4	447	B 44V4	B 50V4
316	B 52V4	B 54V4	358	B 7V4	B 22V4	403	B 58V4	B 74V4	448	B 24V4	B 84V4
317	B 51V4	B 73V4	359	B 31V4	B 32V4	404	B 55V4	B 74V4	449	B 6V4	B 18V4
318	B 67V4	B 73V4	360	B 49V4	B 50V4	405	B 43V4	B 53V4	450	B 5V4	B 17V4
319	B 68V4	B 73V4	361	B 33V4	B 35V4	406	B 70V4	B 71V4	451	B 5V4	B 16V4
320	B 26V4	B 30V4	362	B 27V4	B 33V4	407	B 72V4	B 79V4	452	B 7V4	B 24V4
321	B 66V4	B 70V4	363	B 66V4	B 73V4	408	B 49V4	B 61V4	453	B 65V4	B 69V4
322	B 7V4	B 8V4	364	B 54V4	B 81V4	409	B 26V4	B 27V4	454	B 52V4	B 59V4
323	B 4V4	B 84V4	365	B 75V4	B 77V4	410	B 26V4	B 35V4	455	B 51V4	B 62V4
324	B 29V4	B 32V4	366	B 46V4	B 56V4	411	B 6V4	B 21V4	456	B 69V4	B 70V4
325	B 39V4	B 62V4	367	B 36V4	B 74V4	412	B 6V4	B 22V4	457	B 42V4	B 57V4
326	B 12V4	B 17V4	368	B 39V4	B 60V4	413	B 7V4	B 10V4	458	B 6V4	B 13V4
327	B 12V4	B 16V4	369	B 39V4	B 59V4	414	B 7V4	B 11V4	459	B 52V4	B 62V4
328	B 43V4	B 47V4	370	B 16V4	B 54V4	415	B 60V4	B 74V4	460	B 3V4	B 5V4
329	B 9V4	B 16V4	371	B 17V4	B 54V4	416	B 38V4	B 50V4	461	B 28V4	B 34V4
330	B 9V4	B 17V4	372	B 65V4	B 80V4	417	B 43V4	B 60V4	462	B 63V4	B 74V4
331	B 48V4	B 51V4	373	B 34V4	B 35V4	418	B 48V4	B 61V4	463	B 13V4	B 84V4
332	B 37V4	B 74V4	374	B 27V4	B 34V4	419	B 43V4	B 59V4	464	B 24V4	B 54V4
333	B 52V4	B 53V4	375	B 6V4	B 15V4	420	B 53V4	B 57V4	465	B 43V4	B 56V4
334	B 73V4	B 74V4	376	B 48V4	B 50V4	421	B 7V4	B 13V4	466	B 38V4	B 39V4
335	B 44V4	B 49V4	377	B 1V4	B 84V4	422	B 69V4	B 71V4	467	B 45V4	B 77V4
336	B 44V4	B 48V4	378	B 39V4	B 58V4	423	B 5V4	B 25V4	468	B 39V4	B 52V4
337	B 68V4	B 70V4	379	B 39V4	B 55V4	424	B 39V4	B 56V4	469	B 52V4	B 61V4

(continued)

Table H.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
470	B 42V4	B 52V4	513	B 13V4	B 54V4	556	B 42V4	B 61V4	599	B 21V4	B 52V4
471	B 6V4	B 23V4	514	B 19V4	B 57V4	557	B 64V4	B 73V4	600	B 22V4	B 52V4
472	B 42V4	B 49V4	515	B 14V4	B 57V4	558	B 45V4	B 47V4	601	B 50V4	B 60V4
473	B 6V4	B 10V4	516	B 7V4	B 14V4	559	B 48V4	B 67V4	602	B 48V4	B 59V4
474	B 6V4	B 11V4	517	B 7V4	B 19V4	560	B 48V4	B 68V4	603	B 47V4	B 66V4
475	B 40V4	B 61V4	518	B 49V4	B 58V4	561	B 70V4	B 80V4	604	B 15V4	B 84V4
476	B 43V4	B 44V4	519	B 49V4	B 55V4	562	B 47V4	B 61V4	605	B 23V4	B 54V4
477	B 79V4	B 80V4	520	B 45V4	B 52V4	563	B 20V4	B 52V4	606	B 1V4	B 54V4
478	B 51V4	B 59V4	521	B 46V4	B 52V4	564	B 25V4	B 52V4	607	B 18V4	B 84V4
479	B 29V4	B 34V4	522	B 65V4	B 68V4	565	B 48V4	B 71V4	608	B 21V4	B 54V4
480	B 3V4	B 26V4	523	B 65V4	B 67V4	566	B 45V4	B 48V4	609	B 22V4	B 54V4
481	B 23V4	B 52V4	524	B 38V4	B 46V4	567	B 50V4	B 63V4	610	B 49V4	B 66V4
482	B 20V4	B 84V4	525	B 61V4	B 74V4	568	B 72V4	B 80V4	611	B 71V4	B 72V4
483	B 4V4	B 7V4	526	B 52V4	B 77V4	569	B 69V4	B 79V4	612	B 40V4	B 51V4
484	B 52V4	B 63V4	527	B 49V4	B 63V4	570	B 12V4	B 52V4	613	B 78V4	B 83V4
485	B 41V4	B 53V4	528	B 18V4	B 52V4	571	B 17V4	B 52V4	614	B 50V4	B 58V4
486	B 6V4	B 24V4	529	B 48V4	B 55V4	572	B 16V4	B 52V4	615	B 50V4	B 55V4
487	B 43V4	B 52V4	530	B 48V4	B 58V4	573	B 40V4	B 53V4	616	B 65V4	B 72V4
488	B 10V4	B 52V4	531	B 39V4	B 51V4	574	B 52V4	B 60V4	617	B 26V4	B 32V4
489	B 11V4	B 52V4	532	B 54V4	B 57V4	575	B 49V4	B 68V4	618	B 37V4	B 58V4
490	B 38V4	B 42V4	533	B 64V4	B 66V4	576	B 49V4	B 67V4	619	B 37V4	B 55V4
491	B 62V4	B 74V4	534	B 49V4	B 62V4	577	B 51V4	B 56V4	620	B 67V4	B 69V4
492	B 42V4	B 48V4	535	B 46V4	B 53V4	578	B 9V4	B 52V4	621	B 68V4	B 69V4
493	B 79V4	B 82V4	536	B 4V4	B 81V4	579	B 42V4	B 76V4	622	B 8V4	B 31V4
494	B 41V4	B 61V4	537	B 42V4	B 44V4	580	B 49V4	B 71V4	623	B 4V4	B 54V4
495	B 65V4	B 66V4	538	B 48V4	B 63V4	581	B 23V4	B 57V4	624	B 45V4	B 61V4
496	B 19V4	B 84V4	539	B 50V4	B 62V4	582	B 65V4	B 71V4	625	B 6V4	B 25V4
497	B 14V4	B 84V4	540	B 22V4	B 84V4	583	B 72V4	B 74V4	626	B 13V4	B 81V4
498	B 59V4	B 74V4	541	B 21V4	B 84V4	584	B 47V4	B 68V4	627	B 18V4	B 57V4
499	B 49V4	B 60V4	542	B 45V4	B 49V4	585	B 47V4	B 67V4	628	B 10V4	B 57V4
500	B 42V4	B 47V4	543	B 8V4	B 26V4	586	B 43V4	B 51V4	629	B 11V4	B 57V4
501	B 11V4	B 54V4	544	B 54V4	B 76V4	587	B 50V4	B 59V4	630	B 37V4	B 60V4
502	B 10V4	B 54V4	545	B 39V4	B 44V4	588	B 52V4	B 55V4	631	B 6V4	B 19V4
503	B 24V4	B 52V4	546	B 48V4	B 62V4	589	B 52V4	B 58V4	632	B 6V4	B 14V4
504	B 69V4	B 74V4	547	B 56V4	B 74V4	590	B 49V4	B 59V4	633	B 40V4	B 62V4
505	B 38V4	B 45V4	548	B 44V4	B 45V4	591	B 48V4	B 66V4	634	B 72V4	B 82V4
506	B 42V4	B 77V4	549	B 13V4	B 52V4	592	B 19V4	B 81V4	635	B 12V4	B 57V4
507	B 45V4	B 57V4	550	B 69V4	B 80V4	593	B 14V4	B 81V4	636	B 1V4	B 16V4
508	B 45V4	B 76V4	551	B 65V4	B 73V4	594	B 18V4	B 54V4	637	B 1V4	B 17V4
509	B 11V4	B 84V4	552	B 64V4	B 68V4	595	B 64V4	B 71V4	638	B 47V4	B 71V4
510	B 10V4	B 84V4	553	B 64V4	B 67V4	596	B 24V4	B 81V4	639	B 3V4	B 31V4
511	B 50V4	B 52V4	554	B 20V4	B 54V4	597	B 64V4	B 72V4	640	B 40V4	B 63V4
512	B 48V4	B 60V4	555	B 54V4	B 84V4	598	B 44V4	B 46V4	641	B 2V4	B 81V4

(continued)

Table H.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
641	B 2V4	B 81V4	685	B 41V4	B 55V4	729	B 9V4	B 56V4	773	B 8V4	B 22V4
642	B 9V4	B 57V4	686	B 41V4	B 58V4	730	B 15V4	B 57V4	774	B 8V4	B 21V4
643	B 15V4	B 52V4	687	B 12V4	B 84V4	731	B 36V4	B 83V4	775	B 37V4	B 73V4
644	B 41V4	B 62V4	688	B 1V4	B 25V4	732	B 48V4	B 57V4	776	B 23V4	B 59V4
645	B 50V4	B 56V4	689	B 48V4	B 56V4	733	B 42V4	B 56V4	777	B 54V4	B 75V4
646	B 41V4	B 57V4	690	B 9V4	B 84V4	734	B 14V4	B 56V4	778	B 36V4	B 62V4
647	B 32V4	B 33V4	691	B 42V4	B 59V4	735	B 19V4	B 56V4	779	B 15V4	B 81V4
648	B 40V4	B 60V4	692	B 53V4	B 61V4	736	B 38V4	B 51V4	780	B 36V4	B 73V4
649	B 66V4	B 69V4	693	B 47V4	B 55V4	737	B 36V4	B 63V4	781	B 53V4	B 63V4
650	B 70V4	B 72V4	694	B 47V4	B 58V4	738	B 50V4	B 76V4	782	B 37V4	B 82V4
651	B 42V4	B 62V4	695	B 41V4	B 59V4	739	B 48V4	B 73V4	783	B 38V4	B 67V4
652	B 11V4	B 81V4	696	B 23V4	B 56V4	740	B 18V4	B 81V4	784	B 38V4	B 68V4
653	B 10V4	B 81V4	697	B 47V4	B 63V4	741	B 52V4	B 76V4	785	B 36V4	B 69V4
654	B 20V4	B 81V4	698	B 5V4	B 84V4	742	B 47V4	B 59V4	786	B 45V4	B 60V4
655	B 24V4	B 57V4	699	B 47V4	B 62V4	743	B 53V4	B 62V4	787	B 55V4	B 73V4
656	B 37V4	B 56V4	700	B 28V4	B 29V4	744	B 29V4	B 35V4	788	B 58V4	B 73V4
657	B 41V4	B 63V4	701	B 8V4	B 15V4	745	B 27V4	B 29V4	789	B 38V4	B 66V4
658	B 40V4	B 55V4	702	B 36V4	B 60V4	746	B 25V4	B 57V4	790	B 53V4	B 56V4
659	B 40V4	B 58V4	703	B 21V4	B 57V4	747	B 48V4	B 74V4	791	B 8V4	B 18V4
660	B 6V4	B 16V4	704	B 22V4	B 57V4	748	B 45V4	B 63V4	792	B 60V4	B 73V4
661	B 6V4	B 17V4	705	B 67V4	B 74V4	749	B 42V4	B 54V4	793	B 23V4	B 81V4
662	B 12V4	B 54V4	706	B 68V4	B 74V4	750	B 36V4	B 59V4	794	B 61V4	B 71V4
663	B 7V4	B 25V4	707	B 37V4	B 62V4	751	B 45V4	B 54V4	795	B 37V4	B 61V4
664	B 29V4	B 30V4	708	B 31V4	B 34V4	752	B 37V4	B 72V4	796	B 37V4	B 83V4
665	B 3V4	B 33V4	709	B 21V4	B 81V4	753	B 49V4	B 74V4	797	B 10V4	B 56V4
666	B 40V4	B 57V4	710	B 22V4	B 81V4	754	B 45V4	B 59V4	798	B 11V4	B 56V4
667	B 23V4	B 84V4	711	B 53V4	B 75V4	755	B 29V4	B 31V4	799	B 8V4	B 23V4
668	B 41V4	B 51V4	712	B 7V4	B 17V4	756	B 49V4	B 73V4	800	B 40V4	B 52V4
669	B 9V4	B 54V4	713	B 7V4	B 16V4	757	B 40V4	B 56V4	801	B 51V4	B 70V4
670	B 41V4	B 60V4	714	B 72V4	B 78V4	758	B 18V4	B 56V4	802	B 46V4	B 71V4
671	B 37V4	B 63V4	715	B 36V4	B 82V4	759	B 50V4	B 51V4	803	B 52V4	B 81V4
672	B 49V4	B 56V4	716	B 42V4	B 60V4	760	B 37V4	B 51V4	804	B 45V4	B 55V4
673	B 13V4	B 57V4	717	B 46V4	B 74V4	761	B 80V4	B 83V4	805	B 45V4	B 58V4
674	B 15V4	B 54V4	718	B 36V4	B 72V4	762	B 66V4	B 74V4	806	B 44V4	B 51V4
675	B 40V4	B 59V4	719	B 45V4	B 62V4	763	B 41V4	B 52V4	807	B 36V4	B 51V4
676	B 50V4	B 77V4	720	B 49V4	B 57V4	764	B 58V4	B 71V4	808	B 38V4	B 61V4
677	B 47V4	B 60V4	721	B 8V4	B 9V4	765	B 55V4	B 71V4	809	B 1V4	B 77V4
678	B 20V4	B 57V4	722	B 8V4	B 33V4	766	B 4V4	B 52V4	810	B 19V4	B 59V4
679	B 42V4	B 63V4	723	B 36V4	B 56V4	767	B 53V4	B 59V4	811	B 14V4	B 59V4
680	B 37V4	B 59V4	724	B 12V4	B 56V4	768	B 3V4	B 15V4	812	B 37V4	B 69V4
681	B 36V4	B 58V4	725	B 57V4	B 77V4	769	B 17V4	B 57V4	813	B 45V4	B 56V4
682	B 36V4	B 55V4	726	B 8V4	B 12V4	770	B 16V4	B 57V4	814	B 49V4	B 53V4
683	B 51V4	B 69V4	727	B 42V4	B 58V4	771	B 60V4	B 71V4	815	B 23V4	B 62V4
684	B 77V4	B 81V4	728	B 42V4	B 55V4	772	B 41V4	B 56V4	816	B 12V4	B 59V4

(continued)

Table H.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
817	B 3V4	B 9V4	863	B 12V4	B 62V4	909	B 8V4	B 24V4	955	B 26V4	B 34V4
818	B 9V4	B 59V4	864	B 38V4	B 53V4	910	B 19V4	B 77V4	956	B 9V4	B 55V4
819	B 73V4	B 83V4	865	B 47V4	B 53V4	911	B 14V4	B 77V4	957	B 9V4	B 58V4
820	B 43V4	B 77V4	866	B 50V4	B 54V4	912	B 3V4	B 28V4	958	B 41V4	B 68V4
821	B 2V4	B 84V4	867	B 61V4	B 73V4	913	B 45V4	B 75V4	959	B 41V4	B 67V4
822	B 3V4	B 12V4	868	B 1V4	B 75V4	914	B 47V4	B 74V4	960	B 19V4	B 60V4
823	B 69V4	B 78V4	869	B 9V4	B 62V4	915	B 41V4	B 77V4	961	B 14V4	B 60V4
824	B 67V4	B 72V4	870	B 63V4	B 73V4	916	B 39V4	B 71V4	962	B 9V4	B 60V4
825	B 68V4	B 72V4	871	B 8V4	B 10V4	917	B 18V4	B 63V4	963	B 46V4	B 66V4
826	B 8V4	B 20V4	872	B 8V4	B 11V4	918	B 65V4	B 79V4	964	B 14V4	B 58V4
827	B 44V4	B 68V4	873	B 66V4	B 72V4	919	B 43V4	B 74V4	965	B 14V4	B 55V4
828	B 44V4	B 67V4	874	B 3V4	B 4V4	920	B 75V4	B 81V4	966	B 19V4	B 58V4
829	B 44V4	B 66V4	875	B 40V4	B 68V4	921	B 38V4	B 71V4	967	B 19V4	B 55V4
830	B 63V4	B 71V4	876	B 40V4	B 67V4	922	B 65V4	B 78V4	968	B 10V4	B 63V4
831	B 6V4	B 84V4	877	B 62V4	B 71V4	923	B 43V4	B 76V4	969	B 11V4	B 63V4
832	B 48V4	B 53V4	878	B 76V4	B 81V4	924	B 11V4	B 62V4	970	B 43V4	B 71V4
833	B 51V4	B 72V4	879	B 36V4	B 61V4	925	B 10V4	B 62V4	971	B 17V4	B 53V4
834	B 12V4	B 81V4	880	B 16V4	B 77V4	926	B 78V4	B 82V4	972	B 16V4	B 53V4
835	B 1V4	B 4V4	881	B 17V4	B 77V4	927	B 3V4	B 13V4	973	B 38V4	B 60V4
836	B 47V4	B 56V4	882	B 3V4	B 18V4	928	B 43V4	B 54V4	974	B 38V4	B 62V4
837	B 9V4	B 81V4	883	B 19V4	B 63V4	929	B 39V4	B 54V4	975	B 53V4	B 81V4
838	B 3V4	B 22V4	884	B 14V4	B 63V4	930	B 1V4	B 24V4	976	B 36V4	B 71V4
839	B 3V4	B 21V4	885	B 44V4	B 61V4	931	B 8V4	B 30V4	977	B 70V4	B 74V4
840	B 23V4	B 63V4	886	B 11V4	B 59V4	932	B 23V4	B 61V4	978	B 38V4	B 63V4
841	B 53V4	B 60V4	887	B 10V4	B 59V4	933	B 59V4	B 73V4	979	B 1V4	B 13V4
842	B 69V4	B 82V4	888	B 73V4	B 80V4	934	B 20V4	B 59V4	980	B 25V4	B 53V4
843	B 18V4	B 59V4	889	B 64V4	B 79V4	935	B 1V4	B 14V4	981	B 39V4	B 76V4
844	B 14V4	B 62V4	890	B 64V4	B 78V4	936	B 1V4	B 19V4	982	B 3V4	B 24V4
845	B 19V4	B 62V4	891	B 12V4	B 63V4	937	B 24V4	B 59V4	983	B 42V4	B 75V4
846	B 47V4	B 73V4	892	B 18V4	B 62V4	938	B 21V4	B 59V4	984	B 41V4	B 71V4
847	B 39V4	B 74V4	893	B 9V4	B 63V4	939	B 22V4	B 59V4	985	B 24V4	B 62V4
848	B 51V4	B 57V4	894	B 44V4	B 53V4	940	B 46V4	B 73V4	986	B 61V4	B 68V4
849	B 22V4	B 56V4	895	B 40V4	B 66V4	941	B 37V4	B 71V4	987	B 61V4	B 67V4
850	B 21V4	B 56V4	896	B 40V4	B 71V4	942	B 19V4	B 61V4	988	B 20V4	B 62V4
851	B 20V4	B 56V4	897	B 3V4	B 20V4	943	B 14V4	B 61V4	989	B 41V4	B 66V4
852	B 4V4	B 8V4	898	B 8V4	B 28V4	944	B 13V4	B 59V4	990	B 38V4	B 55V4
853	B 15V4	B 56V4	899	B 3V4	B 23V4	945	B 42V4	B 51V4	991	B 38V4	B 58V4
854	B 24V4	B 56V4	900	B 46V4	B 67V4	946	B 3V4	B 10V4	992	B 44V4	B 71V4
855	B 74V4	B 83V4	901	B 46V4	B 68V4	947	B 3V4	B 11V4	993	B 5V4	B 81V4
856	B 39V4	B 77V4	902	B 23V4	B 55V4	948	B 12V4	B 55V4	994	B 18V4	B 55V4
857	B 8V4	B 13V4	903	B 23V4	B 58V4	949	B 12V4	B 58V4	995	B 18V4	B 58V4
858	B 19V4	B 53V4	904	B 33V4	B 34V4	950	B 54V4	B 56V4	996	B 46V4	B 77V4
859	B 14V4	B 53V4	905	B 23V4	B 60V4	951	B 1V4	B 76V4	997	B 18V4	B 60V4
860	B 53V4	B 55V4	906	B 59V4	B 71V4	952	B 41V4	B 76V4	998	B 22V4	B 62V4
861	B 53V4	B 58V4	907	B 25V4	B 77V4	953	B 12V4	B 60V4	999	B 21V4	B 62V4
862	B 13V4	B 56V4	908	B 62V4	B 73V4	954	B 3V4	B 30V4	1000	B 12V4	B 61V4

(continued)

Table H.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
1001	B 5V4	B 54V4	1045	B 24V4	B 77V4	1089	B 1V4	B 18V4	1133	B 36V4	B 79V4
1002	B 60V4	B 67V4	1046	B 65V4	B 83V4	1090	B 46V4	B 76V4	1134	B 3V4	B 16V4
1003	B 60V4	B 68V4	1047	B 10V4	B 61V4	1091	B 20V4	B 77V4	1135	B 3V4	B 17V4
1004	B 40V4	B 77V4	1048	B 11V4	B 61V4	1092	B 41V4	B 74V4	1136	B 6V4	B 33V4
1005	B 55V4	B 68V4	1049	B 23V4	B 53V4	1093	B 43V4	B 73V4	1137	B 4V4	B 77V4
1006	B 58V4	B 67V4	1050	B 39V4	B 66V4	1094	B 48V4	B 70V4	1138	B 3V4	B 32V4
1007	B 58V4	B 68V4	1051	B 44V4	B 60V4	1095	B 70V4	B 83V4	1139	B 73V4	B 82V4
1008	B 55V4	B 67V4	1052	B 13V4	B 53V4	1096	B 8V4	B 32V4	1140	B 50V4	B 66V4
1009	B 9V4	B 61V4	1053	B 7V4	B 84V4	1097	B 3V4	B 35V4	1141	B 1V4	B 12V4
1010	B 39V4	B 68V4	1054	B 40V4	B 74V4	1098	B 3V4	B 27V4	1142	B 12V4	B 46V4
1011	B 39V4	B 67V4	1055	B 60V4	B 66V4	1099	B 8V4	B 35V4	1143	B 1V4	B 9V4
1012	B 6V4	B 26V4	1056	B 55V4	B 66V4	1100	B 8V4	B 27V4	1144	B 18V4	B 46V4
1013	B 64V4	B 83V4	1057	B 58V4	B 66V4	1101	B 3V4	B 25V4	1145	B 40V4	B 54V4
1014	B 1V4	B 20V4	1058	B 44V4	B 55V4	1102	B 1V4	B 15V4	1146	B 2V4	B 16V4
1015	B 1V4	B 10V4	1059	B 44V4	B 58V4	1103	B 17V4	B 76V4	1147	B 2V4	B 17V4
1016	B 1V4	B 11V4	1060	B 43V4	B 66V4	1104	B 16V4	B 76V4	1148	B 10V4	B 39V4
1017	B 46V4	B 54V4	1061	B 1V4	B 21V4	1105	B 49V4	B 70V4	1149	B 11V4	B 39V4
1018	B 51V4	B 65V4	1062	B 1V4	B 22V4	1106	B 18V4	B 77V4	1150	B 18V4	B 39V4
1019	B 18V4	B 61V4	1063	B 18V4	B 53V4	1107	B 6V4	B 31V4	1151	B 74V4	B 79V4
1020	B 24V4	B 53V4	1064	B 10V4	B 77V4	1108	B 2V4	B 54V4	1152	B 12V4	B 77V4
1021	B 54V4	B 61V4	1065	B 11V4	B 77V4	1109	B 50V4	B 74V4	1153	B 48V4	B 69V4
1022	B 20V4	B 63V4	1066	B 39V4	B 73V4	1110	B 23V4	B 77V4	1154	B 2V4	B 77V4
1023	B 24V4	B 63V4	1067	B 20V4	B 53V4	1111	B 50V4	B 71V4	1155	B 12V4	B 39V4
1024	B 22V4	B 63V4	1068	B 74V4	B 82V4	1112	B 70V4	B 78V4	1156	B 9V4	B 77V4
1025	B 21V4	B 63V4	1069	B 66V4	B 80V4	1113	B 23V4	B 46V4	1157	B 44V4	B 73V4
1026	B 10V4	B 53V4	1070	B 68V4	B 80V4	1114	B 8V4	B 16V4	1158	B 7V4	B 36V4
1027	B 11V4	B 53V4	1071	B 67V4	B 80V4	1115	B 8V4	B 17V4	1159	B 80V4	B 82V4
1028	B 73V4	B 79V4	1072	B 71V4	B 83V4	1116	B 47V4	B 70V4	1160	B 49V4	B 69V4
1029	B 43V4	B 68V4	1073	B 7V4	B 26V4	1117	B 41V4	B 54V4	1161	B 5V4	B 26V4
1030	B 43V4	B 67V4	1074	B 13V4	B 77V4	1118	B 23V4	B 39V4	1162	B 37V4	B 79V4
1031	B 45V4	B 51V4	1075	B 70V4	B 79V4	1119	B 38V4	B 73V4	1163	B 15V4	B 77V4
1032	B 51V4	B 64V4	1076	B 19V4	B 39V4	1120	B 1V4	B 23V4	1164	B 76V4	B 84V4
1033	B 8V4	B 14V4	1077	B 14V4	B 39V4	1121	B 50V4	B 68V4	1165	B 71V4	B 79V4
1034	B 8V4	B 19V4	1078	B 64V4	B 74V4	1122	B 50V4	B 67V4	1166	B 38V4	B 76V4
1035	B 10V4	B 60V4	1079	B 26V4	B 29V4	1123	B 50V4	B 75V4	1167	B 11V4	B 43V4
1036	B 11V4	B 60V4	1080	B 65V4	B 74V4	1124	B 21V4	B 77V4	1168	B 10V4	B 43V4
1037	B 10V4	B 55V4	1081	B 37V4	B 46V4	1125	B 22V4	B 77V4	1169	B 38V4	B 74V4
1038	B 11V4	B 58V4	1082	B 7V4	B 31V4	1126	B 25V4	B 76V4	1170	B 44V4	B 76V4
1039	B 11V4	B 55V4	1083	B 8V4	B 25V4	1127	B 36V4	B 46V4	1171	B 2V4	B 25V4
1040	B 10V4	B 58V4	1084	B 3V4	B 19V4	1128	B 6V4	B 81V4	1172	B 51V4	B 83V4
1041	B 71V4	B 80V4	1085	B 3V4	B 14V4	1129	B 7V4	B 54V4	1173	B 37V4	B 48V4
1042	B 2V4	B 75V4	1086	B 40V4	B 73V4	1130	B 7V4	B 37V4	1174	B 37V4	B 49V4
1043	B 77V4	B 84V4	1087	B 14V4	B 46V4	1131	B 41V4	B 73V4	1175	B 2V4	B 76V4
1044	B 40V4	B 76V4	1088	B 19V4	B 46V4	1132	B 29V4	B 33V4	1176	B 12V4	B 43V4

(continued)

Table H.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
1177	B 38V4	B 77V4	1205	B 24V4	B 45V4	1233	B 48V4	B 65V4	1261	B 21V4	B 45V4
1178	B 37V4	B 67V4	1206	B 13V4	B 42V4	1234	B 16V4	B 50V4	1262	B 49V4	B 72V4
1179	B 37V4	B 68V4	1207	B 24V4	B 50V4	1235	B 17V4	B 50V4	1263	B 47V4	B 64V4
1180	B 9V4	B 43V4	1208	B 75V4	B 84V4	1236	B 41V4	B 75V4	1264	B 29V4	B 36V4
1181	B 29V4	B 82V4	1209	B 47V4	B 69V4	1237	B 38V4	B 70V4	1265	B 7V4	B 30V4
1182	B 24V4	B 43V4	1210	B 36V4	B 48V4	1238	B 20V4	B 50V4	1266	B 5V4	B 33V4
1183	B 24V4	B 42V4	1211	B 5V4	B 31V4	1239	B 3V4	B 84V4	1267	B 22V4	B 76V4
1184	B 49V4	B 77V4	1212	B 36V4	B 49V4	1240	B 36V4	B 66V4	1268	B 21V4	B 76V4
1185	B 9V4	B 37V4	1213	B 42V4	B 71V4	1241	B 20V4	B 45V4	1269	B 6V4	B 28V4
1186	B 17V4	B 42V4	1214	B 20V4	B 42V4	1242	B 48V4	B 76V4	1270	B 40V4	B 75V4
1187	B 16V4	B 42V4	1215	B 44V4	B 74V4	1243	B 25V4	B 75V4	1271	B 51V4	B 80V4
1188	B 44V4	B 77V4	1216	B 16V4	B 75V4	1244	B 20V4	B 76V4	1272	B 37V4	B 47V4
1189	B 25V4	B 42V4	1217	B 17V4	B 75V4	1245	B 49V4	B 65V4	1273	B 22V4	B 37V4
1190	B 73V4	B 78V4	1218	B 67V4	B 83V4	1246	B 48V4	B 64V4	1274	B 21V4	B 37V4
1191	B 7V4	B 33V4	1219	B 68V4	B 83V4	1247	B 66V4	B 83V4	1275	B 68V4	B 79V4
1192	B 36V4	B 67V4	1220	B 9V4	B 42V4	1248	B 8V4	B 84V4	1276	B 67V4	B 79V4
1193	B 36V4	B 68V4	1221	B 37V4	B 66V4	1249	B 45V4	B 74V4	1277	B 4V4	B 76V4
1194	B 24V4	B 76V4	1222	B 45V4	B 81V4	1250	B 45V4	B 71V4	1278	B 71V4	B 82V4
1195	B 7V4	B 81V4	1223	B 13V4	B 76V4	1251	B 47V4	B 65V4	1279	B 36V4	B 70V4
1196	B 48V4	B 77V4	1224	B 49V4	B 76V4	1252	B 44V4	B 70V4	1280	B 40V4	B 70V4
1197	B 13V4	B 43V4	1225	B 42V4	B 67V4	1253	B 2V4	B 4V4	1281	B 37V4	B 70V4
1198	B 37V4	B 43V4	1226	B 42V4	B 68V4	1254	B 7V4	B 28V4	1282	B 15V4	B 36V4
1199	B 1V4	B 5V4	1227	B 74V4	B 80V4	1255	B 49V4	B 64V4	1283	B 29V4	B 37V4
1200	B 17V4	B 45V4	1228	B 13V4	B 50V4	1256	B 48V4	B 72V4	1284	B 6V4	B 30V4
1201	B 16V4	B 45V4	1229	B 42V4	B 81V4	1257	B 15V4	B 37V4	1285	B 1V4	B 6V4
1202	B 42V4	B 74V4	1230	B 13V4	B 45V4	1258	B 45V4	B 67V4	1286	B 66V4	B 79V4
1203	B 50V4	B 73V4	1231	B 47V4	B 76V4	1259	B 45V4	B 68V4	1287	B 20V4	B 37V4
1204	B 25V4	B 45V4	1232	B 25V4	B 50V4	1260	B 8V4	B 29V4			

[a] The length of any candidate line may be readily calculated from geographical characteristics of the sending and receiving buses. For details, see problem 6 of Chap. 7

Table H.4 Generation data

No.	Bus name	P_G (p.u.)	No.	Bus name	P_G (p.u.)
1[a]	B 2V4	1.03	14	B 54V4	2.06
2	B 6V4	14.21	15	B 56V4	7.37
3	B 9V4	13.06	16	B 58V4	7.11
4	B 14V4	6.08	17	B 60V4	2.06
5	B 16V4	9.25	18	B 65V4	4.26
6	B 17V4	7.46	19	B 66V4	8.18
7	B 27V4	7.48	20	B 69V4	2.29
8	B 30V4	8.36	21	B 70V4	0.97
9	B 33V4	7.48	22	B 71V4	3.00
10	B 39V4	8.11	23	B 78V4	0.16
11	B 43V4	16.30	24	B 82V4	3.89
12	B 45V4	3.65	25	B 84V4	2.34
13	B 46V4	16.30			

[a] Slack bus

Appendix I
Numerical Details of the Basic Approach

The details of the proposed approach in Chap. 8 for transmission expansion planning, as discussed and tested on the 84-bus test system (see Chap. 8, Sect. 8.6.2) are given here (as Tables I.1, I.2 and I.3).

Table I.1 The detailed results of the backward stage

No.[a]	From bus	To bus	Length[b] (km)	Voltage level (kV)	No. of lines[c]	Capacity limit (p.u.)	Line flow (p.u.)	Maximum line flow in contingency conditions		
								Flow on line (p.u.)	Relevant contingency	
17	B 16V4	B 25V4	14.56	400	2	6.6	5.725	7.537	B 5V4	B 6V4
18	B 17V4	B 25V4	14.56	400	2	6.6	5.624	7.505	B 5V4	B 6V4

[a] The number shown is taken from the candidate line number given in Table H.3
[b] As X and Y are known for each bus, the line length can be readily calculated. For details, see problem 6 of Chap. 7
[c] Two lines are considered in each corridor

Table I.2 The detailed results of the forward stage

No.	From bus	To bus	Length (km)	Voltage level (kV)	No. of lines	Capacity limit (p.u.)	Line flow (p.u.)	Maximum line flow in contingency conditions		
								Flow on line (p.u.)	Relevant contingency	
2	B 21V4	B 22V4	1	400	2	6.6	0.265	4.5	B 9V4	B 22V4
3	B 10V4	B 11V4	1	400	2	6.6	1.895	6.197	B 11V4	B 19V4
17	B 16V4	B 25V4	14.56	400	2	6.6	4.845	5.782	B 17V4	B 25V4
18	B 17V4	B 25V4	14.56	400	2	6.6	4.868	5.785	B 16V4	B 25V4
21	B 10V4	B 20V4	16.15	400	2	6.6	1.576	4.344	B 20V4	B 24V4
33	B 38V4	B 44V4	21.13	400	2	6.6	2.074	2.2	B 38V4	B 44V4
54	B 11V4	B 21V4	28.33	400	2	6.6	0.533	2.833	B 9V4	B 21V4

(continued)

Table I.2 (continued)

No.	From bus	To bus	Length (km)	Voltage level (kV)	No. of lines	Capacity limit (p.u.)	Line flow (p.u.)	Maximum line flow in contingency conditions		
								Flow on line (p.u.)	Relevant contingency	
284	B 28V4	B 33V4	141.19	400	2	6.6	0.198	2.985	B 26V4	B 33V4
302	B 51V4	B 74V4	146.26	400	2	6.6	1.048	2.272	B 58V4	B 74V4
309	B 27V4	B 31V4	148.89	400	2	6.6	3.805	5.221	B 27V4	B 35V4
339	B 1V4	B 2V4	163.51	400	2	6.6	-3.304	6.357	B 43V4[a]	
374	B 27V4	B 34V4	173.68	400	2	6.6	2.91	5.769	B 30V4	B 32V4
473	B 6V4	B 10V4	219.84	400	2	6.6	4.736	5.721	B 6V4	B 7V4
699	B 47V4	B 62V4	282.55	400	2	6.6	-3.032	5.303	B 40V4	B 47V4
868	B 1V4	B 75V4	331.81	400	2	6.6	2.055	4.053	B 76V4	B 77V4
1113	B 23V4	B 46V4	408.13	400	2	6.6	-3.034	3.411	B 39V4	B 46V4
1161	B 5V4	B 26V4	431.95	400	2	6.6	-2.546	3.869	B 3V4	B 26V4
1197	B 13V4	B 43V4	452.34	400	2	6.6	-2.734	3.258	B 43V4	B 50V4
1253	B 2V4	B 4V4	477.96	400	2	6.6	0.822	3.319	B 6V4[b]	
1266	B 5V4	B 33V4	485.24	400	2	6.6	-3.414	4.841	B 26V4	B 33V4

[a, b] Contingency on generation which is located in this bus

Table I.3 The detailed results of the decrease stage

No.	From bus	To bus	Length (km)	Voltage level (kV)	No. of lines	Capacity limit (p.u.)	Line flow (p.u.)	Maximum line flow in contingency conditions		
								Flow on line (p.u.)	Relevant contingency	
2	B 21V4	B 22V4	1	400	2	6.6	0.312	4.5	B 9V4	B 22V4
3	B 10V4	B 11V4	1	400	2	6.6	1.975	6.501	B 11V4	B 19V4
17	B 16V4	B 25V4	14.56	400	2	6.6	4.874	5.816	B 17V4	B 25V4
18	B 17V4	B 25V4	14.56	400	2	6.6	4.894	5.816	B 16V4	B 25V4
21	B 10V4	B 20V4	16.15	400	2	6.6	1.487	4.108	B 20V4	B 24V4
33	B 38V4	B 44V4	21.13	400	1	3.3	1.962	2.2	B 38V4	B 44V4
54	B 11V4	B 21V4	28.33	400	1	3.3	0.628	2.569	B 9V4	B 21V4
284	B 28V4	B 33V4	141.19	400	2	6.6	-0.196	4.559	B 26V4	B 33V4
302	B 51V4	B 74V4	146.26	400	1	3.3	0.899	2.272	B 58V4	B 74V4
309	B 27V4	B 31V4	148.89	400	2	6.6	4.024	5.548	B 26V4	B 33V4
339	B 1V4	B 2V4	163.51	400	2	6.6	-3.213	6.295	B 43V4[a]	
374	B 27V4	B 34V4	173.68	400	2	6.6	2.895	5.768	B 30V4	B 32V4
473	B 6V4	B 10V4	219.84	400	2	6.6	4.713	5.656	B 6V4	B 7V4
699	B 47V4	B 62V4	282.55	400	2	6.6	-3.018	5.295	B 40V4	B 47V4
868	B 1V4	B 75V4	331.81	400	2	6.6	1.987	4.053	B 76V4	B 77V4
1113	B 23V4	B 46V4	408.13	400	1	3.3	-1.83	2.083	B 39V4	B 46V4
1161	B 5V4	B 26V4	431.95	400	2	6.6	-3.208	4.969	B 3V4	B 26V4
1197	B 13V4	B 43V4	452.34	400	2	6.6	-3.047	3.612	B 42V4	B 43V4
1253	B 2V4	B 4V4	477.96	400	2	6.6	0.898	3.398	B 6V4[b]	
1266	B 5V4	B 33V4	485.24	400	1	3.3	-2.114	3.268	B 26V4	B 33V4

[a, b] Contingency on generation which is located in this bus

Appendix J
77-Bus Test System Data

A 77-bus dual voltage level test system is used in Chap. 9 to assess the capability of the proposed hybrid approach for transmission expansion planning problem. Moreover, this test system is used in Chap. 10 for RPP analysis. The relevant data of this test system are provided as follows

- Bus data are provided in Table J.1.
- Line data are provided in Table J.2.
- Candidate lines data are provided in Table J.3.
- Generation data are provided in Table J.4.

Table J.1 Bus data

No.	Bus name	X	Y	P_D (p.u.)	Q_D (p.u.)	No.	Bus name	X	Y	P_D (p.u.)	Q_D (p.u.)
1	B 1V4	53.43	35.60	0.00	0.00	20	B 20V2	50.00	36.28	0.00	0.00
2	B 2V2	50.07	36.22	0.63	0.58	21	B 21V2	50.87	34.68	1.24	0.54
3	B 3V2	51.52	35.75	2.17	1.05	22	B 22V2	50.75	34.58	1.56	0.76
4	B 4V4	49.83	34.00	0.00	0.00	23	B 23V2	50.95	34.62	1.61	0.54
5	B 5V2	49.83	34.00	0.00	0.00	24	B 24V2	51.43	35.68	4.25	1.98
6	B 6V2	50.15	35.95	1.68	0.66	25	B 25V2	52.15	35.67	0.22	0.11
7	B 7V2	51.32	35.67	3.00	1.28	26	B 26V4	50.32	33.41	0.00	0.00
8	B 8V2	50.95	33.71	0.00	0.00	27	B 27V4	51.20	36.50	0.28	1.43
9	B 9V2	51.43	35.63	1.73	0.58	28	B 28V2	50.77	35.95	1.56	0.71
10	B 10V2	50.10	35.75	0.42	0.44	29	B 29V4	51.85	35.42	0.00	0.00
11	B 11V4	51.87	35.43	0.00	0.00	30	B 30V2	51.77	35.82	0.00	0.00
12	B 12V2	51.50	35.70	3.46	1.75	31	B 31V2	50.83	35.82	3.48	1.68
13	B 13V2	51.27	35.57	2.68	1.30	32	B 32V2	51.28	35.77	2.29	0.80
14	B 14V4	51.30	35.62	0.00	0.00	33	B 33V4	51.28	35.77	0.00	0.00
15	B 15V2	51.30	35.62	0.00	0.00	34	B 34V2	51.02	35.85	1.93	0.93
16	B 16V4	51.30	35.62	0.00	0.00	35	B 35V2	51.33	34.05	0.68	0.23
17	B 17V2	52.73	35.77	0.50	0.24	36	B 36V2	51.67	35.75	0.00	0.00
18	B 18V2	51.12	35.43	0.08	0.01	37	B 37V2	51.25	35.70	2.94	1.23
19	B 19V2	52.33	35.25	0.00	0.00	38	B 38V2	51.05	35.77	0.09	0.02

(continued)

Table J.1 (continued)

No.	Bus name	X	Y	P_D (p.u.)	Q_D (p.u.)	No.	Bus name	X	Y	P_D (p.u.)	Q_D (p.u.)
39	B 39V2	51.02	35.73	0.00	0.00	59	B 59V4	50.90	35.42	0.00	0.00
40	B 40V2	51.00	35.75	3.92	1.86	60	B 60V2	52.97	36.17	0.00	0.00
41	B 41V2	51.42	35.73	3.77	0.70	61	B 61V2	50.37	35.02	0.27	0.44
42	B 42V2	51.48	35.62	2.24	1.08	62	B 62V2	51.07	35.68	2.48	0.98
43	B 43V4	49.63	37.18	0.00	0.00	63	B 63V2	51.38	35.75	0.92	0.36
44	B 44V2	51.02	35.47	0.00	0.00	64	B 64V4	51.38	35.75	0.00	0.00
45	B 45V4	50.90	35.42	0.00	0.00	65	B 65V2	51.43	35.67	2.12	1.02
46	B 46V2	51.38	35.78	3.25	1.15	66	B 66V2	50.55	35.82	0.59	0.23
47	B 47V4	53.25	36.82	0.00	0.00	67	B 67V4	46.17	38.08	0.00	0.00
48	B 48V2	50.57	34.23	0.40	0.19	68	B 68V2	51.35	35.73	3.25	0.83
49	B 49V2	51.52	35.80	1.95	0.67	69	B 69V2	51.43	35.63	0.00	0.00
50	B 50V2	51.02	35.47	2.47	1.67	70	B 70V4	51.57	35.75	0.00	0.00
51	B 51V2	51.83	35.75	1.71	0.83	71	B 71V4	51.57	35.75	0.00	0.00
52	B 52V4	51.83	35.75	0.00	0.00	72	B 72V2	51.57	35.75	1.78	0.73
53	B 53V2	51.43	35.80	1.60	0.56	73	B 73V4	51.65	35.33	1.52	0.74
54	B 54V4	50.32	36.15	0.00	0.00	74	B 74V2	51.13	35.75	2.55	1.11
55	B 55V4	50.32	36.15	0.00	0.00	75	B 75V4	51.13	35.75	0.00	0.00
56	B 56V2	51.40	35.52	3.15	1.52	76	B 76V2	50.47	36.10	0.82	0.40
57	B 57V2	51.58	35.52	2.89	1.19	77	B 77V4	50.47	36.10	0.00	0.00
58	B 58V4	51.58	35.52	0.00	0.00						

Table J.2 Line data

No.	From bus	To bus	R (p.u.)	X (p.u.)	B (p.u.)	$\bar{P}_L$ (p.u.)
1	B 10V2	B 31V2	0.0148	0.0611	-0.3425	3.0
2	B 12V2	B 57V2	0.0027	0.0156	-0.0524	4.9
3	B 12V2	B 72V2	0.0008	0.0046	-0.0155	4.9
4	B 13V2	B 15V2	0.0016	0.0092	-0.0163	2.8
5	B 13V2	B 18V2	0.0031	0.0196	-0.0407	2.8
6	B 13V2	B 44V2	0.0059	0.0267	-0.0472	2.8
7	B 13V2	B 56V2	0.0029	0.0172	-0.0305	2.8
8	B 15V2	B 32V2	0.0024	0.0139	-0.0466	4.8
9	B 15V2	B 39V2	0.0050	0.0220	-0.0569	2.8
10	B 15V2	B 56V2	0.0020	0.0120	-0.0294	3.4
11	B 15V2	B 69V2	0.0023	0.0121	-0.0210	3.4
12	B 15V2	B 69V2	0.0023	0.0121	-0.0210	3.4
13	B 17V2	B 60V2	0.0003	0.0024	-0.0043	2.7
14	B 18V2	B 21V2	0.0150	0.0886	-0.1672	3.4
15	B 18V2	B 44V2	0.0021	0.0128	-0.0189	3.2
16	B 18V2	B 56V2	0.0050	0.0293	-0.0553	3.4
17	B 19V2	B 57V2	0.0090	0.0670	-0.1223	3.8
18	B 21V2	B 22V2	0.0020	0.0120	-0.0213	2.7

(continued)

Table J.2 (continued)

No.	From bus	To bus	R (p.u.)	X (p.u.)	B (p.u.)	$\bar{P}_L$ (p.u.)
19	B 21V2	B 22V2	0.0020	0.0120	-0.0213	2.7
20	B 21V2	B 23V2	0.0026	0.0156	-0.0285	3.2
21	B 22V2	B 23V2	0.0037	0.0218	-0.0401	3.2
22	B 22V2	B 23V2	0.0037	0.0218	-0.0401	6.5
23	B 22V2	B 44V2	0.0296	0.1314	-0.2323	2.8
24	B 23V2	B 35V2	0.0108	0.0665	-0.1202	4.0
25	B 24V2	B 41V2	0.0003	0.0027	-0.5850	5.7
26	B 24V2	B 65V2	0.0002	0.0017	-0.3656	6.4
27	B 17V2	B 25V2	0.0077	0.0556	-0.0687	2.7
28	B 25V2	B 51V2	0.0184	0.1326	-0.1638	2.7
29	B 27V4	B 77V4	0.0023	0.0264	-0.6973	11.7
30	B 28V2	B 31V2	0.0030	0.0178	-0.0329	3.3
31	B 28V2	B 76V2	0.0043	0.0257	-0.0375	3.3
32	B 1V4	B 29V4	0.0034	0.0317	-0.8608	9.0
33	B 11V4	B 29V4	0.0000	0.0002	-0.0062	9.9
34	B 11V4	B 29V4	0.0000	0.0002	-0.0060	9.9
35	B 26V4	B 29V4	0.0060	0.0633	-1.7161	10.7
36	B 29V4	B 47V4	0.0040	0.0550	-1.3729	9.1
37	B 29V4	B 58V4	0.0006	0.0057	-0.0823	12.1
38	B 29V4	B 59V4	0.0010	0.0164	-0.6666	22.2
39	B 29V4	B 59V4	0.0010	0.0164	-0.6733	22.2
40	B 29V4	B 73V4	0.0004	0.0045	-0.1207	15.0
41	B 2V2	B 76V2	0.0064	0.0452	-0.0815	3.2
42	B 31V2	B 40V2	0.0031	0.0188	-0.0259	3.3
43	B 31V2	B 40V2	0.0069	0.0388	-0.1568	2.3
44	B 31V2	B 74V2	0.0040	0.0223	-0.0846	4.9
45	B 32V2	B 37V2	0.0011	0.0063	-0.0216	4.9
46	B 32V2	B 46V2	0.0010	0.0058	-0.0194	4.9
47	B 32V2	B 74V2	0.0016	0.0091	-0.0333	4.9
48	B 33V4	B 77V4	0.0013	0.0180	-0.6305	11.7
49	B 31V2	B 34V2	0.0023	0.0128	-0.0487	4.9
50	B 34V2	B 74V2	0.0010	0.0046	-0.0333	4.9
51	B 30V2	B 36V2	0.0004	0.0058	-0.2063	2.3
52	B 9V2	B 36V2	0.0072	0.0364	-0.3079	2.3
53	B 37V2	B 39V2	0.0039	0.0246	-0.0844	4.9
54	B 38V2	B 39V2	0.0008	0.0046	-0.0089	1.4
55	B 38V2	B 39V2	0.0008	0.0046	-0.0089	1.4
56	B 39V2	B 40V2	0.0006	0.0025	-0.0059	4.7
57	B 39V2	B 40V2	0.0006	0.0025	-0.0059	4.7
58	B 39V2	B 49V2	0.0059	0.0331	-0.1072	4.8
59	B 39V2	B 74V2	0.0038	0.0108	-0.0352	4.8
60	B 3V2	B 49V2	0.0015	0.0086	-0.0287	4.8
61	B 3V2	B 72V2	0.0010	0.0051	-0.0172	4.9
62	B 41V2	B 63V2	0.0003	0.0025	-0.0059	3.5
63	B 41V2	B 63V2	0.0002	0.0021	-0.0076	3.5

(continued)

Table J.2 (continued)

No.	From bus	To bus	R (p.u.)	X (p.u.)	B (p.u.)	$\bar{P}_L$ (p.u.)
64	B 42V2	B 57V2	0.0020	0.0112	-0.0392	2.9
65	B 43V4	B 54V4	0.0032	0.0340	-0.8453	10.7
66	B 44V2	B 61V2	0.0162	0.0733	-0.1294	2.7
67	B 14V4	B 45V4	0.0006	0.0083	-0.2686	8.3
68	B 46V2	B 49V2	0.0017	0.0098	-0.0330	4.8
69	B 1V4	B 47V4	0.0027	0.0315	-0.8335	15.0
70	B 21V2	B 48V2	0.0070	0.0401	-0.0764	3.3
71	B 49V2	B 53V2	0.0011	0.0059	-0.0214	4.8
72	B 4V4	B 26V4	0.0016	0.0181	-0.4794	15.0
73	B 4V4	B 59V4	0.0035	0.0395	-1.0460	7.6
74	B 44V2	B 50V2	0.0010	0.0008	-0.0016	3.2
75	B 44V2	B 50V2	0.0010	0.0008	-0.0016	3.2
76	B 30V2	B 51V2	0.0014	0.0066	-0.0131	2.7
77	B 11V4	B 52V4	0.0007	0.0089	-0.3122	18.2
78	B 11V4	B 52V4	0.0007	0.0089	-0.3122	18.2
79	B 52V4	B 75V4	0.0009	0.0126	-0.4433	4.8
80	B 53V2	B 72V2	0.0023	0.0130	-0.0417	4.9
81	B 54V4	B 55V4	0.0000	0.0002	-0.0124	12.1
82	B 54V4	B 59V4	0.0013	0.0185	-0.6867	16.8
83	B 54V4	B 77V4	0.0003	0.0044	-0.1561	13.5
84	B 54V4	B 77V4	0.0003	0.0044	-0.1561	13.5
85	B 55V4	B 59V4	0.0013	0.0185	-0.6805	10.7
86	B 56V2	B 57V2	0.0023	0.0136	-0.0253	3.4
87	B 56V2	B 57V2	0.0023	0.0136	-0.0253	3.4
88	B 57V2	B 65V2	0.0039	0.0246	-0.0844	4.9
89	B 58V4	B 73V4	0.0004	0.0045	-0.1207	15.0
90	B 16V4	B 59V4	0.0006	0.0085	-0.2846	18.4
91	B 4V4	B 59V4	0.0035	0.0395	-1.0460	7.6
92	B 45V4	B 59V4	0.0000	0.0002	0.0000	16.6
93	B 5V2	B 22V2	0.0260	0.1323	-0.2399	2.7
94	B 40V2	B 62V2	0.0025	0.0140	-0.0529	4.9
95	B 33V4	B 64V4	0.0002	0.0021	-0.0448	16.6
96	B 65V2	B 69V2	0.0001	0.0010	-0.2194	5.5
97	B 15V2	B 66V2	0.0028	0.0016	-0.0544	4.9
98	B 62V2	B 66V2	0.0008	0.0045	-0.0169	4.9
99	B 54V4	B 67V4	0.0092	0.0927	-2.5224	11.7
100	B 32V2	B 68V2	0.0008	0.0046	-0.0157	4.8
101	B 32V2	B 68V2	0.0008	0.0046	-0.0157	4.8
102	B 6V2	B 10V2	0.0033	0.0191	-0.0371	4.0
103	B 11V4	B 70V4	0.0008	0.0113	-0.3871	12.5
104	B 11V4	B 71V4	0.0008	0.0113	-0.3640	12.5
105	B 57V2	B 72V2	0.0027	0.0156	-0.0617	4.9
106	B 52V4	B 75V4	0.0009	0.0126	-0.4433	4.8
107	B 54V4	B 75V4	0.0013	0.0177	-0.6400	18.2
108	B 64V4	B 75V4	0.0004	0.0050	-0.1792	16.6

(continued)

Table J.2 (continued)

No.	From bus	To bus	R (p.u.)	X (p.u.)	B (p.u.)	$\bar{P}_L$ (p.u.)
109	B 75V4	B 77V4	0.0010	0.0147	-0.5073	16.6
110	B 20V2	B 76V2	0.0099	0.0518	-0.0895	3.0
111	B 6V2	B 76V2	0.0054	0.0306	-0.0593	2.8
112	B 55V4	B 77V4	0.0013	0.0177	-0.6400	18.2
113	B 7V2	B 15V2	0.0002	0.0034	-0.0118	9.6
114	B 7V2	B 15V2	0.0002	0.0034	-0.0118	9.6
115	B 7V2	B 24V2	0.0004	0.0039	-0.8555	6.4
116	B 8V2	B 35V2	0.0089	0.0509	-0.0970	3.7
117	B 8V2	B 48V2	0.0097	0.0555	-0.1058	3.3
118	B 9V2	B 42V2	0.0006	0.0040	-0.0137	4.8
119	B 9V2	B 56V2	0.0035	0.0154	-0.0269	2.8
120	B 14V4	B 15V2	0.0013	0.0257	1.0000	5.0
121	B 14V4	B 15V2	0.0013	0.0257	1.0000	5.0
122	B 16V4	B 15V2	0.0013	0.0257	1.0000	5.0
123	B 16V4	B 15V2	0.0013	0.0257	1.0000	5.0
124	B 33V4	B 32V2	0.0012	0.0242	1.0000	5.0
125	B 33V4	B 32V2	0.0012	0.0242	1.0000	5.0
126	B 52V4	B 51V2	0.0013	0.0257	1.0000	5.0
127	B 52V4	B 51V2	0.0013	0.0257	1.0000	5.0
128	B 58V4	B 57V2	0.0012	0.0240	1.0000	5.0
129	B 58V4	B 57V2	0.0012	0.0240	1.0000	5.0
130	B 64V4	B 63V2	0.0004	0.0257	1.0000	5.0
131	B 64V4	B 63V2	0.0004	0.0257	1.0000	5.0
132	B 70V4	B 72V2	0.0012	0.0229	1.0000	5.0
133	B 71V4	B 72V2	0.0012	0.0229	1.0000	5.0
134	B 75V4	B 74V2	0.0012	0.0241	1.0000	5.0
135	B 75V4	B 74V2	0.0012	0.0241	1.0000	5.0
136	B 77V4	B 76V2	0.0013	0.0269	1.0000	5.0
137	B 77V4	B 76V2	0.0013	0.0269	1.0000	5.0

Table J.3 Candidate lines data[a]

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
1	B 44V2	B 50V2	12	B 71V4	B 72V2	38	B 11V4	B 29V4	49	B 41V2	B 64V4
2	B 44V2	B 50V2	13	B 70V4	B 72V2	39	B 39V2	B 40V2	50	B 41V2	B 64V4
3	B 74V2	B 75V4	14	B 9V2	B 69V2	40	B 39V2	B 40V2	51	B 41V2	B 63V2
4	B 45V4	B 59V4	15	B 9V2	B 69V2	41	B 46V2	B 64V4	52	B 41V2	B 63V2
5	B 45V4	B 59V4	16	B 54V4	B 55V4	42	B 46V2	B 64V4	53	B 65V2	B 69V2
6	B 76V2	B 77V4	17	B 54V4	B 55V4	43	B 46V2	B 63V2	54	B 65V2	B 69V2
7	B 57V2	B 58V4	18	B 32V2	B 33V4	44	B 46V2	B 63V2	55	B 9V2	B 65V2
8	B 63V2	B 64V4	19	B 4V4	B 5V2	45	B 64V4	B 68V2	56	B 9V2	B 65V2
9	B 14V4	B 15V2	35	B 24V2	B 65V2	46	B 64V4	B 68V2	57	B 3V2	B 71V4
10	B 15V2	B 16V4	36	B 24V2	B 65V2	47	B 63V2	B 68V2	58	B 3V2	B 71V4
11	B 51V2	B 52V4	37	B 11V4	B 29V4	48	B 63V2	B 68V2	59	B 3V2	B 72V2

(continued)

Table J.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
60	B 3V2	B 72V2	104	B 41V2	B 65V2	148	B 12V2	B 70V4	192	B 37V2	B 68V2
61	B 3V2	B 70V4	105	B 12V2	B 65V2	149	B 12V2	B 71V4	193	B 65V2	B 68V2
62	B 3V2	B 70V4	106	B 12V2	B 65V2	150	B 12V2	B 71V4	194	B 65V2	B 68V2
63	B 42V2	B 69V2	107	B 7V2	B 37V2	151	B 24V2	B 64V4	195	B 3V2	B 53V2
64	B 42V2	B 69V2	108	B 7V2	B 37V2	152	B 24V2	B 64V4	196	B 3V2	B 53V2
65	B 9V2	B 42V2	109	B 49V2	B 70V4	153	B 24V2	B 63V2	197	B 21V2	B 23V2
66	B 9V2	B 42V2	110	B 49V2	B 70V4	154	B 24V2	B 63V2	198	B 21V2	B 23V2
67	B 46V2	B 53V2	111	B 49V2	B 71V4	155	B 36V2	B 72V2	199	B 7V2	B 65V2
68	B 46V2	B 53V2	112	B 49V2	B 71V4	156	B 36V2	B 72V2	200	B 7V2	B 65V2
69	B 38V2	B 40V2	113	B 49V2	B 72V2	157	B 36V2	B 70V4	201	B 63V2	B 65V2
70	B 38V2	B 40V2	114	B 49V2	B 72V2	158	B 36V2	B 70V4	202	B 63V2	B 65V2
71	B 38V2	B 39V2	115	B 53V2	B 64V4	159	B 36V2	B 71V4	203	B 64V4	B 65V2
72	B 38V2	B 39V2	116	B 53V2	B 64V4	160	B 36V2	B 71V4	204	B 64V4	B 65V2
73	B 3V2	B 49V2	117	B 53V2	B 63V2	161	B 12V2	B 42V2	205	B 14V4	B 37V2
74	B 3V2	B 49V2	118	B 53V2	B 63V2	162	B 12V2	B 42V2	206	B 14V4	B 37V2
75	B 9V2	B 24V2	119	B 39V2	B 62V2	163	B 32V2	B 46V2	207	B 15V2	B 37V2
76	B 9V2	B 24V2	120	B 39V2	B 62V2	164	B 32V2	B 46V2	208	B 15V2	B 37V2
77	B 24V2	B 69V2	121	B 42V2	B 65V2	165	B 33V4	B 46V2	209	B 16V4	B 37V2
78	B 24V2	B 69V2	122	B 42V2	B 65V2	166	B 33V4	B 46V2	210	B 16V4	B 37V2
79	B 24V2	B 41V2	123	B 7V2	B 68V2	167	B 24V2	B 68V2	211	B 7V2	B 24V2
80	B 24V2	B 41V2	124	B 7V2	B 68V2	168	B 24V2	B 68V2	212	B 7V2	B 24V2
81	B 3V2	B 12V2	125	B 38V2	B 74V2	169	B 2V2	B 20V2	213	B 40V2	B 62V2
82	B 3V2	B 12V2	126	B 38V2	B 74V2	170	B 2V2	B 20V2	214	B 40V2	B 62V2
83	B 7V2	B 16V4	127	B 38V2	B 75V4	171	B 33V4	B 63V2	215	B 9V2	B 12V2
84	B 7V2	B 16V4	128	B 38V2	B 75V4	172	B 33V4	B 63V2	216	B 9V2	B 12V2
85	B 7V2	B 14V4	129	B 32V2	B 68V2	173	B 33V4	B 64V4	217	B 12V2	B 69V2
86	B 7V2	B 14V4	130	B 32V2	B 68V2	174	B 33V4	B 64V4	218	B 12V2	B 69V2
87	B 7V2	B 15V2	131	B 33V4	B 68V2	175	B 32V2	B 64V4	219	B 18V2	B 44V2
88	B 7V2	B 15V2	132	B 33V4	B 68V2	176	B 32V2	B 64V4	220	B 18V2	B 44V2
89	B 46V2	B 68V2	133	B 41V2	B 53V2	177	B 32V2	B 63V2	221	B 18V2	B 50V2
90	B 46V2	B 68V2	134	B 41V2	B 53V2	178	B 32V2	B 63V2	222	B 18V2	B 50V2
91	B 13V2	B 15V2	135	B 12V2	B 41V2	179	B 3V2	B 41V2	223	B 38V2	B 62V2
92	B 13V2	B 15V2	136	B 12V2	B 41V2	180	B 3V2	B 41V2	224	B 38V2	B 62V2
93	B 13V2	B 16V4	137	B 24V2	B 42V2	181	B 34V2	B 38V2	225	B 39V2	B 74V2
94	B 13V2	B 16V4	138	B 24V2	B 42V2	182	B 34V2	B 38V2	226	B 39V2	B 74V2
95	B 13V2	B 14V4	139	B 49V2	B 53V2	183	B 30V2	B 52V4	227	B 39V2	B 75V4
96	B 13V2	B 14V4	140	B 49V2	B 53V2	184	B 30V2	B 52V4	228	B 39V2	B 75V4
97	B 41V2	B 68V2	141	B 33V4	B 37V2	185	B 30V2	B 51V2	229	B 7V2	B 63V2
98	B 41V2	B 68V2	142	B 33V4	B 37V2	186	B 30V2	B 51V2	230	B 7V2	B 63V2
99	B 41V2	B 46V2	143	B 32V2	B 37V2	187	B 62V2	B 75V4	231	B 7V2	B 64V4
100	B 41V2	B 46V2	144	B 32V2	B 37V2	188	B 62V2	B 75V4	232	B 7V2	B 64V4
101	B 12V2	B 24V2	145	B 12V2	B 72V2	189	B 62V2	B 74V2	233	B 53V2	B 68V2
102	B 12V2	B 24V2	146	B 12V2	B 72V2	190	B 62V2	B 74V2	234	B 53V2	B 68V2
103	B 41V2	B 65V2	147	B 12V2	B 70V4	191	B 37V2	B 68V2	235	B 7V2	B 69V2

(continued)

Table J.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
236	B 7V2	B 69V2	279	B 3V2	B 65V2	322	B 46V2	B 65V2	365	B 41V2	B 72V2
237	B 7V2	B 9V2	280	B 3V2	B 65V2	323	B 16V4	B 68V2	366	B 41V2	B 72V2
238	B 7V2	B 9V2	281	B 12V2	B 63V2	324	B 16V4	B 68V2	367	B 41V2	B 70V4
239	B 41V2	B 69V2	282	B 12V2	B 63V2	325	B 15V2	B 68V2	368	B 41V2	B 70V4
240	B 41V2	B 69V2	283	B 12V2	B 64V4	326	B 15V2	B 68V2	369	B 41V2	B 71V4
241	B 9V2	B 41V2	284	B 12V2	B 64V4	327	B 14V4	B 68V2	370	B 41V2	B 71V4
242	B 9V2	B 41V2	285	B 37V2	B 74V2	328	B 14V4	B 68V2	371	B 53V2	B 70V4
243	B 7V2	B 41V2	286	B 37V2	B 74V2	329	B 3V2	B 46V2	372	B 53V2	B 70V4
244	B 7V2	B 41V2	287	B 37V2	B 75V4	330	B 3V2	B 46V2	373	B 53V2	B 71V4
245	B 3V2	B 24V2	288	B 37V2	B 75V4	331	B 68V2	B 69V2	374	B 53V2	B 71V4
246	B 3V2	B 24V2	289	B 45V4	B 50V2	332	B 68V2	B 69V2	375	B 53V2	B 72V2
247	B 34V2	B 40V2	290	B 45V4	B 50V2	333	B 9V2	B 68V2	376	B 53V2	B 72V2
248	B 34V2	B 40V2	291	B 50V2	B 59V4	334	B 9V2	B 68V2	377	B 49V2	B 64V4
249	B 12V2	B 49V2	292	B 50V2	B 59V4	335	B 42V2	B 56V2	378	B 49V2	B 64V4
250	B 12V2	B 49V2	293	B 44V2	B 45V4	336	B 42V2	B 56V2	379	B 49V2	B 63V2
251	B 7V2	B 32V2	294	B 44V2	B 45V4	337	B 24V2	B 53V2	380	B 49V2	B 63V2
252	B 7V2	B 32V2	295	B 44V2	B 59V4	338	B 24V2	B 53V2	381	B 32V2	B 53V2
253	B 7V2	B 33V4	296	B 44V2	B 59V4	339	B 34V2	B 39V2	382	B 32V2	B 53V2
254	B 7V2	B 33V4	297	B 9V2	B 56V2	340	B 34V2	B 39V2	383	B 33V4	B 53V2
255	B 40V2	B 75V4	298	B 9V2	B 56V2	341	B 7V2	B 46V2	384	B 33V4	B 53V2
256	B 40V2	B 75V4	299	B 56V2	B 69V2	342	B 7V2	B 46V2	385	B 12V2	B 68V2
257	B 40V2	B 74V2	300	B 56V2	B 69V2	343	B 41V2	B 42V2	386	B 12V2	B 68V2
258	B 40V2	B 74V2	301	B 3V2	B 63V2	344	B 41V2	B 42V2	387	B 12V2	B 46V2
259	B 9V2	B 16V4	302	B 3V2	B 63V2	345	B 33V4	B 41V2	388	B 12V2	B 46V2
260	B 9V2	B 16V4	303	B 3V2	B 64V4	346	B 33V4	B 41V2	389	B 9V2	B 64V4
261	B 9V2	B 14V4	304	B 3V2	B 64V4	347	B 32V2	B 41V2	390	B 9V2	B 64V4
262	B 9V2	B 14V4	305	B 12V2	B 53V2	348	B 32V2	B 41V2	391	B 9V2	B 63V2
263	B 9V2	B 15V2	306	B 12V2	B 53V2	349	B 14V4	B 24V2	392	B 9V2	B 63V2
264	B 9V2	B 15V2	307	B 46V2	B 49V2	350	B 14V4	B 24V2	393	B 63V2	B 69V2
265	B 16V4	B 69V2	308	B 46V2	B 49V2	351	B 15V2	B 24V2	394	B 63V2	B 69V2
266	B 16V4	B 69V2	309	B 37V2	B 64V4	352	B 15V2	B 24V2	395	B 64V4	B 69V2
267	B 15V2	B 69V2	310	B 37V2	B 64V4	353	B 16V4	B 24V2	396	B 64V4	B 69V2
268	B 15V2	B 69V2	311	B 37V2	B 63V2	354	B 16V4	B 24V2	397	B 42V2	B 58V4
269	B 14V4	B 69V2	312	B 37V2	B 63V2	355	B 3V2	B 36V2	398	B 42V2	B 58V4
270	B 14V4	B 69V2	313	B 15V2	B 65V2	356	B 3V2	B 36V2	399	B 42V2	B 57V2
271	B 30V2	B 36V2	314	B 15V2	B 65V2	357	B 32V2	B 75V4	400	B 42V2	B 57V2
272	B 30V2	B 36V2	315	B 16V4	B 65V2	358	B 32V2	B 75V4	401	B 16V4	B 56V2
273	B 41V2	B 49V2	316	B 16V4	B 65V2	359	B 33V4	B 74V2	402	B 16V4	B 56V2
274	B 41V2	B 49V2	317	B 14V4	B 65V2	360	B 33V4	B 74V2	403	B 15V2	B 56V2
275	B 24V2	B 46V2	318	B 14V4	B 65V2	361	B 33V4	B 75V4	404	B 15V2	B 56V2
276	B 24V2	B 46V2	319	B 13V2	B 56V2	362	B 33V4	B 75V4	405	B 14V4	B 56V2
277	B 7V2	B 13V2	320	B 13V2	B 56V2	363	B 32V2	B 74V2	406	B 14V4	B 56V2
278	B 7V2	B 13V2	321	B 46V2	B 65V2	364	B 32V2	B 74V2	407	B 36V2	B 51V2

(continued)

Table J.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
408	B 36V2	B 51V2	452	B 24V2	B 49V2	496	B 42V2	B 70V4	540	B 9V2	B 37V2
409	B 36V2	B 52V4	453	B 3V2	B 69V2	497	B 42V2	B 71V4	541	B 7V2	B 56V2
410	B 36V2	B 52V4	454	B 3V2	B 69V2	498	B 42V2	B 71V4	542	B 7V2	B 56V2
411	B 53V2	B 65V2	455	B 3V2	B 9V2	499	B 7V2	B 12V2	543	B 9V2	B 58V4
412	B 53V2	B 65V2	456	B 3V2	B 9V2	500	B 7V2	B 12V2	544	B 9V2	B 58V4
413	B 13V2	B 37V2	457	B 21V2	B 22V2	501	B 14V4	B 33V4	545	B 9V2	B 57V2
414	B 13V2	B 37V2	458	B 21V2	B 22V2	502	B 14V4	B 33V4	546	B 9V2	B 57V2
415	B 55V4	B 77V4	459	B 37V2	B 41V2	503	B 14V4	B 32V2	547	B 57V2	B 69V2
416	B 55V4	B 77V4	460	B 37V2	B 41V2	504	B 14V4	B 32V2	548	B 57V2	B 69V2
417	B 54V4	B 76V2	461	B 13V2	B 69V2	505	B 16V4	B 32V2	549	B 58V4	B 69V2
418	B 54V4	B 76V2	462	B 13V2	B 69V2	506	B 16V4	B 32V2	550	B 58V4	B 69V2
419	B 54V4	B 77V4	463	B 9V2	B 13V2	507	B 16V4	B 33V4	551	B 22V2	B 23V2
420	B 54V4	B 77V4	464	B 9V2	B 13V2	508	B 16V4	B 33V4	552	B 22V2	B 23V2
421	B 55V4	B 76V2	465	B 16V4	B 64V4	509	B 15V2	B 32V2	553	B 16V4	B 46V2
422	B 55V4	B 76V2	466	B 16V4	B 64V4	510	B 15V2	B 32V2	554	B 16V4	B 46V2
423	B 36V2	B 49V2	467	B 15V2	B 63V2	511	B 15V2	B 33V4	555	B 34V2	B 62V2
424	B 36V2	B 49V2	468	B 15V2	B 63V2	512	B 15V2	B 33V4	556	B 34V2	B 62V2
425	B 37V2	B 46V2	469	B 15V2	B 64V4	513	B 42V2	B 68V2	557	B 30V2	B 71V4
426	B 37V2	B 46V2	470	B 15V2	B 64V4	514	B 42V2	B 68V2	558	B 30V2	B 71V4
427	B 24V2	B 72V2	471	B 14V4	B 64V4	515	B 42V2	B 64V4	559	B 30V2	B 72V2
428	B 24V2	B 72V2	472	B 14V4	B 64V4	516	B 42V2	B 64V4	560	B 30V2	B 72V2
429	B 24V2	B 70V4	473	B 14V4	B 63V2	517	B 42V2	B 63V2	561	B 30V2	B 70V4
430	B 24V2	B 70V4	474	B 14V4	B 63V2	518	B 42V2	B 63V2	562	B 30V2	B 70V4
431	B 24V2	B 71V4	475	B 16V4	B 63V2	519	B 31V2	B 40V2	563	B 37V2	B 38V2
432	B 24V2	B 71V4	476	B 16V4	B 63V2	520	B 31V2	B 40V2	564	B 37V2	B 38V2
433	B 3V2	B 42V2	477	B 16V4	B 42V2	521	B 49V2	B 68V2	565	B 37V2	B 53V2
434	B 3V2	B 42V2	478	B 16V4	B 42V2	522	B 49V2	B 68V2	566	B 37V2	B 53V2
435	B 34V2	B 75V4	479	B 14V4	B 42V2	523	B 9V2	B 46V2	567	B 31V2	B 39V2
436	B 34V2	B 75V4	480	B 14V4	B 42V2	524	B 9V2	B 46V2	568	B 31V2	B 39V2
437	B 34V2	B 74V2	481	B 15V2	B 42V2	525	B 46V2	B 69V2	569	B 18V2	B 59V4
438	B 34V2	B 74V2	482	B 15V2	B 42V2	526	B 46V2	B 69V2	570	B 18V2	B 59V4
439	B 28V2	B 31V2	483	B 56V2	B 57V2	527	B 31V2	B 34V2	571	B 18V2	B 45V4
440	B 28V2	B 31V2	484	B 56V2	B 57V2	528	B 31V2	B 34V2	572	B 18V2	B 45V4
441	B 65V2	B 70V4	485	B 56V2	B 58V4	529	B 46V2	B 72V2	573	B 12V2	B 14V4
442	B 65V2	B 70V4	486	B 56V2	B 58V4	530	B 46V2	B 72V2	574	B 12V2	B 14V4
443	B 65V2	B 71V4	487	B 12V2	B 36V2	531	B 46V2	B 70V4	575	B 12V2	B 15V2
444	B 65V2	B 71V4	488	B 12V2	B 36V2	532	B 46V2	B 70V4	576	B 12V2	B 15V2
445	B 65V2	B 72V2	489	B 37V2	B 62V2	533	B 46V2	B 71V4	577	B 31V2	B 38V2
446	B 65V2	B 72V2	490	B 37V2	B 62V2	534	B 46V2	B 71V4	578	B 31V2	B 38V2
447	B 7V2	B 42V2	491	B 49V2	B 65V2	535	B 7V2	B 53V2	579	B 13V2	B 18V2
448	B 7V2	B 42V2	492	B 49V2	B 65V2	536	B 7V2	B 53V2	580	B 13V2	B 18V2
449	B 3V2	B 68V2	493	B 42V2	B 72V2	537	B 37V2	B 69V2	581	B 29V4	B 73V4
450	B 3V2	B 68V2	494	B 42V2	B 72V2	538	B 37V2	B 69V2	582	B 29V4	B 73V4
451	B 24V2	B 49V2	495	B 42V2	B 70V4	539	B 9V2	B 37V2	583	B 32V2	B 38V2

(continued)

Table J.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
584	B 32V2	B 38V2	628	B 13V2	B 33V4	672	B 13V2	B 44V2	716	B 31V2	B 74V2
585	B 33V4	B 38V2	629	B 36V2	B 53V2	673	B 13V2	B 50V2	717	B 18V2	B 62V2
586	B 33V4	B 38V2	630	B 36V2	B 53V2	674	B 13V2	B 50V2	718	B 18V2	B 62V2
587	B 37V2	B 39V2	631	B 30V2	B 49V2	675	B 31V2	B 66V2	719	B 49V2	B 51V2
588	B 37V2	B 39V2	632	B 30V2	B 49V2	676	B 31V2	B 66V2	720	B 49V2	B 51V2
589	B 14V4	B 75V4	633	B 6V2	B 10V2	677	B 58V4	B 70V4	721	B 49V2	B 52V4
590	B 14V4	B 75V4	634	B 6V2	B 10V2	678	B 58V4	B 70V4	722	B 49V2	B 52V4
591	B 15V2	B 74V2	635	B 11V4	B 73V4	679	B 58V4	B 71V4	723	B 13V2	B 57V2
592	B 15V2	B 74V2	636	B 11V4	B 73V4	680	B 58V4	B 71V4	724	B 13V2	B 57V2
593	B 15V2	B 75V4	637	B 51V2	B 72V2	681	B 58V4	B 72V2	725	B 13V2	B 58V4
594	B 15V2	B 75V4	638	B 51V2	B 72V2	682	B 58V4	B 72V2	726	B 13V2	B 58V4
595	B 14V4	B 74V2	639	B 52V4	B 70V4	683	B 57V2	B 70V4	727	B 13V2	B 39V2
596	B 14V4	B 74V2	640	B 52V4	B 70V4	684	B 57V2	B 70V4	728	B 13V2	B 39V2
597	B 16V4	B 74V2	641	B 52V4	B 71V4	685	B 57V2	B 71V4	729	B 39V2	B 50V2
598	B 16V4	B 74V2	642	B 52V4	B 71V4	686	B 57V2	B 71V4	730	B 39V2	B 50V2
599	B 16V4	B 75V4	643	B 52V4	B 72V2	687	B 57V2	B 72V2	731	B 39V2	B 44V2
600	B 16V4	B 75V4	644	B 52V4	B 72V2	688	B 57V2	B 72V2	732	B 39V2	B 44V2
601	B 32V2	B 62V2	645	B 51V2	B 70V4	689	B 31V2	B 62V2	733	B 13V2	B 38V2
602	B 32V2	B 62V2	646	B 51V2	B 70V4	690	B 31V2	B 62V2	734	B 13V2	B 38V2
603	B 33V4	B 62V2	647	B 51V2	B 71V4	691	B 36V2	B 58V4	735	B 56V2	B 72V2
604	B 33V4	B 62V2	648	B 51V2	B 71V4	692	B 36V2	B 58V4	736	B 56V2	B 72V2
605	B 3V2	B 32V2	649	B 13V2	B 75V4	693	B 36V2	B 57V2	737	B 56V2	B 70V4
606	B 3V2	B 32V2	650	B 13V2	B 75V4	694	B 36V2	B 57V2	738	B 56V2	B 70V4
607	B 3V2	B 33V4	651	B 13V2	B 74V2	695	B 29V4	B 57V2	739	B 56V2	B 71V4
608	B 3V2	B 33V4	652	B 13V2	B 74V2	696	B 29V4	B 57V2	740	B 56V2	B 71V4
609	B 13V2	B 62V2	653	B 2V2	B 55V4	697	B 29V4	B 58V4	741	B 25V2	B 52V4
610	B 13V2	B 62V2	654	B 2V2	B 55V4	698	B 29V4	B 58V4	742	B 25V2	B 52V4
611	B 16V4	B 62V2	655	B 2V2	B 54V4	699	B 6V2	B 54V4	743	B 25V2	B 51V2
612	B 16V4	B 62V2	656	B 2V2	B 54V4	700	B 6V2	B 54V4	744	B 25V2	B 51V2
613	B 15V2	B 62V2	657	B 44V2	B 62V2	701	B 6V2	B 55V4	745	B 28V2	B 40V2
614	B 15V2	B 62V2	658	B 44V2	B 62V2	702	B 6V2	B 55V4	746	B 28V2	B 40V2
615	B 14V4	B 62V2	659	B 50V2	B 62V2	703	B 18V2	B 56V2	747	B 2V2	B 6V2
616	B 14V4	B 62V2	660	B 50V2	B 62V2	704	B 18V2	B 56V2	748	B 2V2	B 6V2
617	B 32V2	B 49V2	661	B 3V2	B 30V2	705	B 53V2	B 74V2	749	B 56V2	B 73V4
618	B 32V2	B 49V2	662	B 3V2	B 30V2	706	B 53V2	B 74V2	750	B 56V2	B 73V4
619	B 33V4	B 49V2	663	B 33V4	B 39V2	707	B 53V2	B 75V4	751	B 40V2	B 44V2
620	B 33V4	B 49V2	664	B 33V4	B 39V2	708	B 53V2	B 75V4	752	B 40V2	B 44V2
621	B 57V2	B 73V4	665	B 32V2	B 39V2	709	B 11V4	B 57V2	753	B 40V2	B 50V2
622	B 57V2	B 73V4	666	B 32V2	B 39V2	710	B 11V4	B 57V2	754	B 40V2	B 50V2
623	B 58V4	B 73V4	667	B 28V2	B 66V2	711	B 11V4	B 58V4	755	B 49V2	B 57V2
624	B 58V4	B 73V4	668	B 28V2	B 66V2	712	B 11V4	B 58V4	756	B 49V2	B 57V2
625	B 13V2	B 32V2	669	B 28V2	B 34V2	713	B 31V2	B 75V4	757	B 28V2	B 76V2
626	B 13V2	B 32V2	670	B 28V2	B 34V2	714	B 31V2	B 75V4	758	B 28V2	B 76V2
627	B 13V2	B 33V4	671	B 13V2	B 44V2	715	B 31V2	B 74V2	759	B 28V2	B 77V4

(continued)

Table J.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
760	B 28V2	B 77V4	803	B 56V2	B 62V2	846	B 2V2	B 77V4	889	B 18V2	B 57V2
761	B 66V2	B 76V2	804	B 56V2	B 62V2	847	B 18V2	B 38V2	890	B 18V2	B 57V2
762	B 66V2	B 76V2	805	B 44V2	B 56V2	848	B 18V2	B 38V2	891	B 18V2	B 58V4
763	B 66V2	B 77V4	806	B 44V2	B 56V2	849	B 6V2	B 66V2	892	B 18V2	B 58V4
764	B 66V2	B 77V4	807	B 50V2	B 56V2	850	B 6V2	B 66V2	893	B 39V2	B 66V2
765	B 20V2	B 54V4	808	B 50V2	B 56V2	851	B 25V2	B 29V4	894	B 39V2	B 66V2
766	B 20V2	B 54V4	809	B 36V2	B 56V2	852	B 25V2	B 29V4	895	B 25V2	B 36V2
767	B 20V2	B 55V4	810	B 36V2	B 56V2	853	B 6V2	B 20V2	896	B 25V2	B 36V2
768	B 20V2	B 55V4	811	B 56V2	B 75V4	854	B 6V2	B 20V2	897	B 11V4	B 30V2
769	B 28V2	B 38V2	812	B 56V2	B 75V4	855	B 28V2	B 75V4	898	B 11V4	B 30V2
770	B 28V2	B 38V2	813	B 18V2	B 74V2	856	B 28V2	B 75V4	899	B 29V4	B 72V2
771	B 44V2	B 74V2	814	B 18V2	B 74V2	857	B 28V2	B 74V2	900	B 29V4	B 72V2
772	B 44V2	B 74V2	815	B 18V2	B 75V4	858	B 28V2	B 74V2	901	B 29V4	B 70V4
773	B 50V2	B 74V2	816	B 18V2	B 75V4	859	B 71V4	B 74V2	902	B 29V4	B 70V4
774	B 50V2	B 74V2	817	B 11V4	B 52V4	860	B 71V4	B 74V2	903	B 29V4	B 71V4
775	B 50V2	B 75V4	818	B 11V4	B 52V4	861	B 11V4	B 36V2	904	B 29V4	B 71V4
776	B 50V2	B 75V4	819	B 11V4	B 51V2	862	B 11V4	B 36V2	905	B 11V4	B 70V4
777	B 44V2	B 75V4	820	B 11V4	B 51V2	863	B 29V4	B 36V2	906	B 11V4	B 70V4
778	B 44V2	B 75V4	821	B 39V2	B 45V4	864	B 29V4	B 36V2	907	B 11V4	B 72V2
779	B 45V4	B 62V2	822	B 39V2	B 45V4	865	B 38V2	B 45V4	908	B 11V4	B 72V2
780	B 45V4	B 62V2	823	B 39V2	B 59V4	866	B 38V2	B 45V4	909	B 31V2	B 59V4
781	B 59V4	B 62V2	824	B 39V2	B 59V4	867	B 38V2	B 59V4	910	B 31V2	B 59V4
782	B 59V4	B 62V2	825	B 29V4	B 51V2	868	B 38V2	B 59V4	911	B 31V2	B 45V4
783	B 28V2	B 39V2	826	B 29V4	B 51V2	869	B 40V2	B 66V2	912	B 31V2	B 45V4
784	B 28V2	B 39V2	827	B 29V4	B 52V4	870	B 40V2	B 66V2	913	B 31V2	B 77V4
785	B 6V2	B 77V4	828	B 29V4	B 52V4	871	B 10V2	B 66V2	914	B 31V2	B 77V4
786	B 6V2	B 77V4	829	B 11V4	B 25V2	872	B 10V2	B 66V2	915	B 31V2	B 76V2
787	B 6V2	B 76V2	830	B 11V4	B 25V2	873	B 54V4	B 66V2	916	B 31V2	B 76V2
788	B 6V2	B 76V2	831	B 18V2	B 40V2	874	B 54V4	B 66V2	917	B 29V4	B 30V2
789	B 38V2	B 44V2	832	B 18V2	B 40V2	875	B 55V4	B 66V2	918	B 29V4	B 30V2
790	B 38V2	B 44V2	833	B 30V2	B 58V4	876	B 55V4	B 66V2	919	B 38V2	B 66V2
791	B 38V2	B 50V2	834	B 30V2	B 58V4	877	B 34V2	B 50V2	920	B 38V2	B 66V2
792	B 38V2	B 50V2	835	B 30V2	B 57V2	878	B 34V2	B 50V2	921	B 28V2	B 54V4
793	B 51V2	B 57V2	836	B 30V2	B 57V2	879	B 34V2	B 44V2	922	B 28V2	B 54V4
794	B 51V2	B 57V2	837	B 40V2	B 45V4	880	B 34V2	B 44V2	923	B 28V2	B 55V4
795	B 52V4	B 57V2	838	B 40V2	B 45V4	881	B 22V2	B 48V2	924	B 28V2	B 55V4
796	B 52V4	B 57V2	839	B 40V2	B 59V4	882	B 22V2	B 48V2	925	B 11V4	B 19V2
797	B 52V4	B 58V4	840	B 40V2	B 59V4	883	B 34V2	B 66V2	926	B 11V4	B 19V2
798	B 52V4	B 58V4	841	B 25V2	B 30V2	884	B 34V2	B 66V2	927	B 20V2	B 77V4
799	B 51V2	B 58V4	842	B 25V2	B 30V2	885	B 31V2	B 50V2	928	B 20V2	B 77V4
800	B 51V2	B 58V4	843	B 2V2	B 76V2	886	B 31V2	B 50V2	929	B 20V2	B 76V2
801	B 18V2	B 39V2	844	B 2V2	B 76V2	887	B 31V2	B 44V2	930	B 20V2	B 76V2
802	B 18V2	B 39V2	845	B 2V2	B 77V4	888	B 31V2	B 44V2	931	B 36V2	B 73V4

(continued)

Table J.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
932	B 36V2	B 73V4	976	B 25V2	B 58V4	1020	B 34V2	B 36V2	1064	B 17V2	B 19V2
933	B 19V2	B 29V4	977	B 59V4	B 66V2	1021	B 22V2	B 61V2	1065	B 45V4	B 73V4
934	B 19V2	B 29V4	978	B 59V4	B 66V2	1022	B 22V2	B 61V2	1066	B 45V4	B 73V4
935	B 18V2	B 34V2	979	B 45V4	B 66V2	1023	B 28V2	B 45V4	1067	B 59V4	B 73V4
936	B 18V2	B 34V2	980	B 45V4	B 66V2	1024	B 28V2	B 45V4	1068	B 59V4	B 73V4
937	B 10V2	B 55V4	981	B 30V2	B 73V4	1025	B 28V2	B 59V4	1069	B 23V2	B 61V2
938	B 10V2	B 55V4	982	B 30V2	B 73V4	1026	B 28V2	B 59V4	1070	B 23V2	B 61V2
939	B 10V2	B 54V4	983	B 23V2	B 48V2	1027	B 40V2	B 77V4	1071	B 2V2	B 28V2
940	B 10V2	B 54V4	984	B 23V2	B 48V2	1028	B 40V2	B 77V4	1072	B 2V2	B 28V2
941	B 34V2	B 59V4	985	B 6V2	B 28V2	1029	B 40V2	B 76V2	1073	B 30V2	B 40V2
942	B 34V2	B 59V4	986	B 6V2	B 28V2	1030	B 40V2	B 76V2	1074	B 30V2	B 40V2
943	B 34V2	B 45V4	987	B 36V2	B 38V2	1031	B 2V2	B 66V2	1075	B 20V2	B 66V2
944	B 34V2	B 45V4	988	B 36V2	B 38V2	1032	B 2V2	B 66V2	1076	B 20V2	B 66V2
945	B 18V2	B 73V4	989	B 34V2	B 77V4	1033	B 19V2	B 73V4	1077	B 34V2	B 55V4
946	B 18V2	B 73V4	990	B 34V2	B 77V4	1034	B 19V2	B 73V4	1078	B 34V2	B 55V4
947	B 17V2	B 60V2	991	B 34V2	B 76V2	1035	B 6V2	B 31V2	1079	B 34V2	B 54V4
948	B 17V2	B 60V2	992	B 34V2	B 76V2	1036	B 6V2	B 31V2	1080	B 34V2	B 54V4
949	B 51V2	B 73V4	993	B 21V2	B 48V2	1037	B 38V2	B 76V2	1081	B 19V2	B 51V2
950	B 51V2	B 73V4	994	B 21V2	B 48V2	1038	B 38V2	B 76V2	1082	B 19V2	B 51V2
951	B 52V4	B 73V4	995	B 50V2	B 66V2	1039	B 10V2	B 28V2	1083	B 19V2	B 52V4
952	B 52V4	B 73V4	996	B 50V2	B 66V2	1040	B 10V2	B 28V2	1084	B 19V2	B 52V4
953	B 19V2	B 25V2	997	B 44V2	B 66V2	1041	B 39V2	B 77V4	1085	B 23V2	B 35V2
954	B 19V2	B 25V2	998	B 44V2	B 66V2	1042	B 39V2	B 77V4	1086	B 23V2	B 35V2
955	B 18V2	B 31V2	999	B 28V2	B 44V2	1043	B 59V4	B 61V2	1087	B 35V2	B 48V2
956	B 18V2	B 31V2	1000	B 28V2	B 44V2	1044	B 59V4	B 61V2	1088	B 35V2	B 48V2
957	B 44V2	B 57V2	1001	B 28V2	B 50V2	1045	B 45V4	B 61V2	1089	B 5V2	B 48V2
958	B 44V2	B 57V2	1002	B 28V2	B 50V2	1046	B 45V4	B 61V2	1090	B 5V2	B 48V2
959	B 44V2	B 58V4	1003	B 36V2	B 39V2	1047	B 18V2	B 28V2	1091	B 4V4	B 48V2
960	B 44V2	B 58V4	1004	B 36V2	B 39V2	1048	B 18V2	B 28V2	1092	B 4V4	B 48V2
961	B 50V2	B 57V2	1005	B 31V2	B 55V4	1049	B 1V4	B 17V2	1093	B 40V2	B 51V2
962	B 50V2	B 57V2	1006	B 31V2	B 55V4	1050	B 1V4	B 17V2	1094	B 40V2	B 51V2
963	B 50V2	B 58V4	1007	B 31V2	B 54V4	1051	B 18V2	B 29V4	1095	B 40V2	B 52V4
964	B 50V2	B 58V4	1008	B 31V2	B 54V4	1052	B 18V2	B 29V4	1096	B 40V2	B 52V4
965	B 10V2	B 77V4	1009	B 25V2	B 73V4	1053	B 10V2	B 31V2	1097	B 40V2	B 73V4
966	B 10V2	B 77V4	1010	B 25V2	B 73V4	1054	B 10V2	B 31V2	1098	B 40V2	B 73V4
967	B 10V2	B 76V2	1011	B 50V2	B 73V4	1055	B 8V2	B 26V4	1099	B 29V4	B 50V2
968	B 10V2	B 76V2	1012	B 50V2	B 73V4	1056	B 8V2	B 26V4	1100	B 29V4	B 50V2
969	B 8V2	B 35V2	1013	B 44V2	B 73V4	1057	B 18V2	B 66V2	1101	B 29V4	B 44V2
970	B 8V2	B 35V2	1014	B 44V2	B 73V4	1058	B 18V2	B 66V2	1102	B 29V4	B 44V2
971	B 2V2	B 10V2	1015	B 21V2	B 61V2	1059	B 8V2	B 48V2	1103	B 1V4	B 60V2
972	B 2V2	B 10V2	1016	B 21V2	B 61V2	1060	B 8V2	B 48V2	1104	B 1V4	B 60V2
973	B 17V2	B 25V2	1017	B 10V2	B 20V2	1061	B 30V2	B 34V2	1105	B 47V4	B 60V2
974	B 17V2	B 25V2	1018	B 10V2	B 20V2	1062	B 30V2	B 34V2	1106	B 47V4	B 60V2
975	B 25V2	B 58V4	1019	B 34V2	B 36V2	1063	B 17V2	B 19V2	1107	B 27V4	B 77V4

(continued)

Table J.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
1108	B 27V4	B 77V4	1152	B 11V4	B 45V4	1196	B 8V2	B 21V2	1240	B 5V2	B 23V2
1109	B 27V4	B 76V2	1153	B 11V4	B 59V4	1197	B 5V2	B 8V2	1241	B 4V4	B 23V2
1110	B 27V4	B 76V2	1154	B 11V4	B 59V4	1198	B 5V2	B 8V2	1242	B 4V4	B 23V2
1111	B 22V2	B 35V2	1155	B 27V4	B 55V4	1199	B 4V4	B 8V2	1243	B 17V2	B 47V4
1112	B 22V2	B 35V2	1156	B 27V4	B 55V4	1200	B 4V4	B 8V2	1244	B 17V2	B 47V4
1113	B 4V4	B 26V4	1157	B 27V4	B 54V4	1201	B 17V2	B 73V4	1245	B 25V2	B 27V4
1114	B 4V4	B 26V4	1158	B 27V4	B 54V4	1202	B 17V2	B 73V4	1246	B 25V2	B 27V4
1115	B 5V2	B 26V4	1159	B 17V2	B 29V4	1203	B 20V2	B 27V4	1247	B 55V4	B 61V2
1116	B 5V2	B 26V4	1160	B 17V2	B 29V4	1204	B 20V2	B 27V4	1248	B 55V4	B 61V2
1117	B 10V2	B 45V4	1161	B 48V2	B 61V2	1205	B 6V2	B 27V4	1249	B 54V4	B 61V2
1118	B 10V2	B 45V4	1162	B 48V2	B 61V2	1206	B 6V2	B 27V4	1250	B 54V4	B 61V2
1119	B 10V2	B 59V4	1163	B 61V2	B 66V2	1207	B 51V2	B 60V2	1251	B 51V2	B 76V2
1120	B 10V2	B 59V4	1164	B 61V2	B 66V2	1208	B 51V2	B 60V2	1252	B 51V2	B 76V2
1121	B 19V2	B 30V2	1165	B 27V4	B 30V2	1209	B 52V4	B 60V2	1253	B 52V4	B 76V2
1122	B 19V2	B 30V2	1166	B 27V4	B 30V2	1210	B 52V4	B 60V2	1254	B 52V4	B 76V2
1123	B 17V2	B 51V2	1167	B 25V2	B 60V2	1211	B 2V2	B 43V4	1255	B 52V4	B 77V4
1124	B 17V2	B 51V2	1168	B 25V2	B 60V2	1212	B 2V2	B 43V4	1256	B 52V4	B 77V4
1125	B 17V2	B 52V4	1169	B 26V4	B 48V2	1213	B 30V2	B 60V2	1257	B 51V2	B 77V4
1126	B 17V2	B 52V4	1170	B 26V4	B 48V2	1214	B 30V2	B 60V2	1258	B 51V2	B 77V4
1127	B 21V2	B 35V2	1171	B 8V2	B 22V2	1215	B 1V4	B 25V2	1259	B 11V4	B 60V2
1128	B 21V2	B 35V2	1172	B 8V2	B 22V2	1216	B 1V4	B 25V2	1260	B 11V4	B 60V2
1129	B 21V2	B 59V4	1173	B 27V4	B 52V4	1217	B 26V4	B 35V2	1261	B 10V2	B 27V4
1130	B 21V2	B 59V4	1174	B 27V4	B 52V4	1218	B 26V4	B 35V2	1262	B 10V2	B 27V4
1131	B 21V2	B 45V4	1175	B 27V4	B 51V2	1219	B 19V2	B 60V2	1263	B 43V4	B 55V4
1132	B 21V2	B 45V4	1176	B 27V4	B 51V2	1220	B 19V2	B 60V2	1264	B 43V4	B 55V4
1133	B 10V2	B 61V2	1177	B 8V2	B 23V2	1221	B 61V2	B 77V4	1265	B 43V4	B 54V4
1134	B 10V2	B 61V2	1178	B 8V2	B 23V2	1222	B 61V2	B 77V4	1266	B 43V4	B 54V4
1135	B 59V4	B 77V4	1179	B 21V2	B 73V4	1223	B 61V2	B 76V2	1267	B 11V4	B 27V4
1136	B 59V4	B 77V4	1180	B 21V2	B 73V4	1224	B 61V2	B 76V2	1268	B 11V4	B 27V4
1137	B 45V4	B 76V2	1181	B 23V2	B 73V4	1225	B 23V2	B 29V4	1269	B 22V2	B 26V4
1138	B 45V4	B 76V2	1182	B 23V2	B 73V4	1226	B 23V2	B 29V4	1270	B 22V2	B 26V4
1139	B 45V4	B 77V4	1183	B 6V2	B 61V2	1227	B 5V2	B 21V2	1271	B 2V2	B 61V2
1140	B 45V4	B 77V4	1184	B 6V2	B 61V2	1228	B 5V2	B 21V2	1272	B 2V2	B 61V2
1141	B 59V4	B 76V2	1185	B 20V2	B 43V4	1229	B 4V4	B 21V2	1273	B 1V4	B 47V4
1142	B 59V4	B 76V2	1186	B 20V2	B 43V4	1230	B 4V4	B 21V2	1274	B 1V4	B 47V4
1143	B 29V4	B 59V4	1187	B 2V2	B 27V4	1231	B 11V4	B 23V2	1275	B 10V2	B 21V2
1144	B 29V4	B 59V4	1188	B 2V2	B 27V4	1232	B 11V4	B 23V2	1276	B 10V2	B 21V2
1145	B 29V4	B 45V4	1189	B 5V2	B 22V2	1233	B 11V4	B 21V2	1277	B 5V2	B 35V2
1146	B 29V4	B 45V4	1190	B 5V2	B 22V2	1234	B 11V4	B 21V2	1278	B 5V2	B 35V2
1147	B 11V4	B 17V2	1191	B 4V4	B 22V2	1235	B 4V4	B 61V2	1279	B 4V4	B 35V2
1148	B 11V4	B 17V2	1192	B 4V4	B 22V2	1236	B 4V4	B 61V2	1280	B 4V4	B 35V2
1149	B 17V2	B 30V2	1193	B 1V4	B 19V2	1237	B 5V2	B 61V2	1281	B 11V4	B 22V2
1150	B 17V2	B 30V2	1194	B 1V4	B 19V2	1238	B 5V2	B 61V2	1282	B 11V4	B 22V2
1151	B 11V4	B 45V4	1195	B 8V2	B 21V2	1239	B 5V2	B 23V2	1283	B 35V2	B 61V2

(continued)

Table J.3 (continued)

No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus	No.	From bus	To bus
1284	B 35V2	B 61V2	1307	B 6V2	B 21V2	1330	B 6V2	B 23V2	1353	B 23V2	B 54V4
1285	B 43V4	B 77V4	1308	B 6V2	B 21V2	1331	B 21V2	B 54V4	1354	B 23V2	B 54V4
1286	B 43V4	B 77V4	1309	B 27V4	B 43V4	1332	B 21V2	B 54V4	1355	B 2V2	B 21V2
1287	B 43V4	B 76V2	1310	B 27V4	B 43V4	1333	B 21V2	B 55V4	1356	B 2V2	B 21V2
1288	B 43V4	B 76V2	1311	B 17V2	B 27V4	1334	B 21V2	B 55V4	1357	B 27V4	B 47V4
1289	B 10V2	B 22V2	1312	B 17V2	B 27V4	1335	B 19V2	B 27V4	1358	B 27V4	B 47V4
1290	B 10V2	B 22V2	1313	B 23V2	B 25V2	1336	B 19V2	B 27V4	1359	B 2V2	B 22V2
1291	B 19V2	B 23V2	1314	B 23V2	B 25V2	1337	B 25V2	B 54V4	1360	B 2V2	B 22V2
1292	B 19V2	B 23V2	1315	B 21V2	B 25V2	1338	B 25V2	B 54V4	1361	B 19V2	B 47V4
1293	B 20V2	B 61V2	1316	B 21V2	B 25V2	1339	B 25V2	B 55V4	1362	B 19V2	B 47V4
1294	B 20V2	B 61V2	1317	B 25V2	B 47V4	1340	B 25V2	B 55V4	1363	B 2V2	B 23V2
1295	B 6V2	B 43V4	1318	B 25V2	B 47V4	1341	B 10V2	B 48V2	1364	B 2V2	B 23V2
1296	B 6V2	B 43V4	1319	B 19V2	B 35V2	1342	B 10V2	B 48V2	1365	B 6V2	B 48V2
1297	B 23V2	B 26V4	1320	B 19V2	B 35V2	1343	B 22V2	B 25V2	1366	B 6V2	B 48V2
1298	B 23V2	B 26V4	1321	B 6V2	B 22V2	1344	B 22V2	B 25V2	1367	B 4V4	B 10V2
1299	B 19V2	B 21V2	1322	B 6V2	B 22V2	1345	B 25V2	B 61V2	1368	B 4V4	B 10V2
1300	B 19V2	B 21V2	1323	B 19V2	B 22V2	1346	B 25V2	B 61V2	1369	B 19V2	B 48V2
1301	B 10V2	B 23V2	1324	B 19V2	B 22V2	1347	B 22V2	B 55V4	1370	B 19V2	B 48V2
1302	B 10V2	B 23V2	1325	B 27V4	B 60V2	1348	B 22V2	B 55V4	1371	B 67V4	B 71V4
1303	B 21V2	B 26V4	1326	B 27V4	B 60V2	1349	B 22V2	B 54V4	1372	B 67V4	B 71V4
1304	B 21V2	B 26V4	1327	B 10V2	B 43V4	1350	B 22V2	B 54V4			
1305	B 8V2	B 61V2	1328	B 10V2	B 43V4	1351	B 23V2	B 55V4			
1306	B 8V2	B 61V2	1329	B 6V2	B 23V2	1352	B 23V2	B 55V4			

[a] The length of any candidate line may be readily calculated from geographical characteristics of the sending and receiving buses. For details, see problem 6 of Chap. 7

Table J.4 Generation data

No.	Bus name	P_G (p.u.)	V_{set} (p.u.)	$\bar{P}_G$ (p.u.)	$\underline{Q}$ (p.u.)	$\overline{Q}$ (p.u.)
1[a]	B 1V4	1.66	1.0103	3.75	-1.15	2.81
2	B 4V4	3.37	1.0292	4.00	-1.22	3.00
3	B 5V2	0.43	1.0134	0.80	-0.24	0.60
4	B 8V2	0.01	0.9826	0.80	-0.25	0.60
5	B 9V2	2.00	0.9653	3.03	-0.88	0.50
6[a]	B 11V4	13.05	1.0092	24.00	-5.70	9.00
7	B 19V2	0.40	0.9615	0.75	-0.23	0.56
8	B 20V2	0.09	0.9558	0.75	-0.23	0.56
9	B 22V2	5.60	1.0097	7.09	-1.86	3.50
10	B 26V4	2.06	1.0405	2.50	-0.75	1.88
11	B 30V2	0.90	0.9925	1.36	-0.72	0.47
12	B 34V2	0.70	0.9477	0.84	-0.40	0.20
13	B 36V2	0.41	0.9918	0.56	-0.39	0.39
14	B 39V2	7.34	0.9684	12.63	-3.25	5.40
15	B 40V2	4.77	0.9642	7.08	-1.10	1.31
16	B 43V4	0.14	1.0041	2.50	-0.75	1.88
17	B 44V2	6.88	1.0057	12.00	-2.55	4.50
18	B 45V4	6.08	1.0107	9.60	-1.53	3.06
19	B 47V4	2.36	1.0165	3.00	-0.95	2.25
20	B 54V4	9.25	1.0146	12.52	-3.69	6.80
21	B 55V4	7.46	1.0150	12.99	-2.79	3.90
22	B 56V2	5.50	0.9758	12.85	-3.40	3.35
23	B 60V2	0.42	0.9035	0.75	-0.23	0.56
24	B 67V4	1.01	1.0192	5.00	-1.50	3.75
25	B 68V2	0.38	0.9455	0.63	-0.18	0.12
26	B 72V2	0.63	0.9533	0.90	-0.20	0.42

[a] Slack bus

Appendix K
Numerical Details of the Hybrid Approach

The details of the hybrid approach, as discussed and tested on the 77-bus dual voltage level test system (see Chap. 9, Sect 9.6, Table 9.9) are given here (as Tables K.1, K.2, K.3, K.4, K.5 and K.6).

Table K.1 The detailed results of the backward stage

No.[a]	From bus	To bus	Length[b] (km)	Voltage level (kV)	No. of lines[c]	Capacity limit (p.u.)	Line flow (p.u.)	Maximum line flow in contingency conditions		
								Flow on line (p.u.)	Relevant contingency	
1	B 44V2	B 50V2	1	230	2	2.2	1.908	2.628	B 44V2	B 50V2
35	B 24V2	B 65V2	1.11	230	2	2.2	-1.482	2.027	B 7V2	B 24V2
38	B 11V4	B 29V4	2.13	400	2	6.6	0.93	1.454	B 11V4	B 29V4
39	B 39V2	B 40V2	2.86	230	2	2.2	1.124	1.477	B 28V2	B 76V2
41	B 46V2	B 64V4	3.33	230	2	2.2	-0.639	0.985	B 46V2	B 64V4
42	B 46V2	B 64V4	3.33	400	2	6.6	-2.555	2.971	B 11V4	B 70V4
43	B 46V2	B 63V2	3.33	230	2	2.2	-0.213	0.87	B 33V4	B 64V4
47	B 63V2	B 68V2	3.5	230	2	2.2	-0.677	1.044	B 63V2	B 68V2
48	B 63V2	B 68V2	3.5	400	2	6.6	-2.709	3.62	B 33V4	B 64V4
49	B 41V2	B 64V4	4.24	230	2	2.2	-1.579	2.004	B 41V2	B 63V2
55	B 9V2	B 65V2	4.44	230	2	2.2	1.428	2.335	B 65V2	B 69V2
58	B 3V2	B 71V4	4.51	400	2	6.6	-2.749	3.919	B 11V4	B 70V4
62	B 3V2	B 70V4	4.51	400	2	6.6	-2.749	3.919	B 11V4	B 71V4
63	B 42V2	B 69V2	4.65	230	2	2.2	1.451	1.713	B 11V4	B 70V4
65	B 9V2	B 42V2	4.65	230	2	2.2	-1.188	2.044	B 65V2	B 69V2
67	B 46V2	B 53V2	5.03	230	2	2.2	0.839	1.374	B 11V4	B 70V4
69	B 38V2	B 40V2	5.03	230	2	2.2	-0.754	0.974	B 38V2	B 39V2
71	B 38V2	B 39V2	5.2	230	2	2.2	-1.347	1.621	B 38V2	B 39V2
73	B 3V2	B 49V2	5.55	230	2	2.2	1.599	1.986	B 3V2	B 49V2
75	B 9V2	B 24V2	5.55	230	2	2.2	1.439	1.93	B 65V2	B 69V2
84	B 7V2	B 16V4	5.84	400	2	6.6	-4.314	5.71	B 14V4	B 45V4

(continued)

Table K.1 (continued)

No.[a]	From bus	To bus	Length[b] (km)	Voltage level (kV)	No. of lines[c]	Capacity limit (p.u.)	Line flow (p.u.)	Maximum line flow in contingency conditions		
								Flow on line (p.u.)	Relevant contingency	
86	B 7V2	B 14V4	5.84	400	2	6.6	-4.439	5.842	B 45V4	B 59V4
101	B 12V2	B 24V2	6.7	230	2	2.2	-1.415	2.641	B 12V2	B 72V2
119	B 39V2	B 62V2	7.16	230	2	2.2	0.696	0.958	B 44V2	B 62V2
124	B 7V2	B 68V2	7.2	400	2	6.6	4.662	5.823	B 7V2	B 24V2
125	B 38V2	B 74V2	7.55	230	2	2.2	1.98	2.456	B 39V2	B 74V2
131	B 33V4	B 68V2	7.72	230	2	2.2	0.925	2.209	B 33V4	B 64V4
143	B 32V2	B 37V2	8.23	230	2	2.2	-1.331	2.164	B 64V4	B 75V4
155	B 36V2	B 72V2	9.02	230	2	2.2	0.917	1.189	B 11V4	B 70V4
181	B 34V2	B 38V2	9.29	230	2	2.2	-1.553	1.787	B 38V2	B 74V2
288	B 37V2	B 75V4	12.17	400	2	6.6	-4.618	6.108	B 64V4	B 75V4
296	B 44V2	B 59V4	12.2	400	2	6.6	-3.289	4.904	B 14V4	B 45V4
398	B 42V2	B 58V4	14.32	400	2	6.6	-5.901	6.619	B 11V4	B 70V4
417	B 54V4	B 76V2	14.56	230	2	2.2	1.775	2.17	B 55V4	B 76V2
421	B 55V4	B 76V2	14.56	230	2	2.2	1.817	2.477	B 54V4	B 55V4
519	B 31V2	B 40V2	17.18	230	2	2.2	-1.453	2.053	B 28V2	B 76V2
657	B 44V2	B 62V2	23.77	230	2	2.2	1.904	2.297	B 14V4	B 45V4
659	B 50V2	B 62V2	23.77	230	2	2.2	1.824	2.21	B 14V4	B 45V4
701	B 6V2	B 55V4	26.96	230	2	2.2	-2.016	2.408	B 6V2	B 76V2
731	B 39V2	B 44V2	28.89	230	2	2.2	-1.394	1.748	B 14V4	B 45V4

[a] The number shown is taken from the candidate line number given in Table J.3
[b] As X and Y are known for each bus, the line length can be readily calculated. For details, see problem 6 of Chap. 7
[c] Two lines are considered in each corridor

Table K.2 The detailed results of the backward stage (transformers)[a]

No.	Bus name	Voltage Level	Transformer capacity (p.u.)
1	B 44V2	400 kV:230 kV	5.50
2	B 37V2	400 kV:230 kV	8.25
3	B 46V2	400 kV:230 kV	5.50
4	B 33V4	400 kV:230 kV	2.75
5	B 3V2	400 kV:230 kV	8.25
6	B 63V2	400 kV:230 kV	5.50
7	B 42V2	400 kV:230 kV	8.25
8	B 54V4	400 kV:230 kV	2.75
9	B 55V4	400 kV:230 kV	5.50
10	B 64V4	400 kV:230 kV	2.75
11	B 68V2	400 kV:230 kV	2.75
12	B 7V2	400 kV:230 kV	5.50

[a] It should be mentioned that the transformers as detailed in Tables K.2, K.4 and K.6 are justified based on the following steps (The steps are described for a typical row 1, Table K.2)

- From Table K.1, a 400 kV line no. 296 (from B 44V2 to B 59V4) is justified. As the former bus is a 230 kV bus, while the latter is a 400 kV one, a 400 kV:230 kV substation is required
- In terms of the transformer (substation) capacity, it is determined based on the maximum flow (for both normal and contingency conditions) through the above mentioned line (line no. 296). This flow is 5.50 p.u

Table K.3 The detailed results of the forward stage

No.	From bus	To bus	Length (km)	Voltage level (kV)	No. of lines	Capacity limit (p.u.)	Line flow (p.u.)	Maximum line flow in contingency conditions		
								Flow on line (p.u.)	Relevant contingency	
1	B 44V2	B 50V2	1	230	2	2.2	1.521	2.094	B 44V2	B 50V2
35	B 24V2	B 65V2	1.11	230	2	2.2	-1.282	1.758	B 12V2	B 72V2
38	B 11V4	B 29V4	2.13	400	2	6.6	0.99	1.541	B 11V4	B 29V4
39	B 39V2	B 40V2	2.86	230	2	2.2	0.911	1.17	B 28V2	B 76V2
41	B 46V2	B 64V4	3.33	230	2	2.2	-0.45	0.67	B 46V2	B 64V4
42	B 46V2	B 64V4	3.33	400	2	6.6	-1.798	2.268	B 32V2	B 46V2
43	B 46V2	B 63V2	3.33	230	2	2.2	0.162	0.341	B 63V2	B 68V2
46	B 64V4	B 68V2	3.5	400	2	6.6	0.071	1.031	B 64V4	B 75V4
47	B 63V2	B 68V2	3.5	230	2	2.2	-0.565	0.826	B 63V2	B 68V2
48	B 63V2	B 68V2	3.5	400	2	6.6	-2.258	2.622	B 7V2	B 24V2
49	B 41V2	B 64V4	4.24	230	2	2.2	-1.637	2.086	B 41V2	B 63V2
55	B 9V2	B 65V2	4.44	230	2	2.2	1.28	1.996	B 65V2	B 69V2
58	B 3V2	B 71V4	4.51	400	2	6.6	-2.567	3.079	B 11V4	B 70V4
62	B 3V2	B 70V4	4.51	400	2	6.6	-1.588	2.064	B 11V4	B 71V4
63	B 42V2	B 69V2	4.65	230	2	2.2	1.103	1.315	B 12V2	B 72V2
65	B 9V2	B 42V2	4.65	230	2	2.2	-0.757	1.43	B 65V2	B 69V2
67	B 46V2	B 53V2	5.03	230	2	2.2	-0.06	0.46	B 12V2	B 72V2
69	B 38V2	B 40V2	5.03	230	2	2.2	-0.407	0.547	B 38V2	B 39V2
71	B 38V2	B 39V2	5.2	230	2	2.2	-0.894	1.068	B 38V2	B 39V2
73	B 3V2	B 49V2	5.55	230	2	2.2	1.536	1.904	B 3V2	B 49V2
75	B 9V2	B 24V2	5.55	230	2	2.2	1.281	1.666	B 65V2	B 69V2
84	B 7V2	B 16V4	5.84	400	2	6.6	-2.933	3.631	B 14V4	B 45V4
86	B 7V2	B 14V4	5.84	400	2	6.6	-3.002	3.687	B 16V4	B 59V4
101	B 12V2	B 24V2	6.7	230	2	2.2	-0.288	2.066	B 12V2	B 72V2
119	B 39V2	B 62V2	7.16	230	2	2.2	1.599	1.782	B 40V2	B 62V2
124	B 7V2	B 68V2	7.2	400	2	6.6	2.818	3.688	B 7V2	B 24V2
125	B 38V2	B 74V2	7.55	230	2	2.2	1.502	1.738	B 39V2	B 74V2
131	B 33V4	B 68V2	7.72	230	2	2.2	0.912	2.047	B 33V4	B 64V4
143	B 32V2	B 37V2	8.23	230	2	2.2	-0.573	1.224	B 64V4	B 75V4
155	B 36V2	B 72V2	9.02	230	2	2.2	0.355	0.996	B 30V2	B 51V2
169	B 2V2	B 20V2	9.15	230	2	2.2	-0.324	0.627	B 2V2	B 76V2
172	B 67V4	B 71V4	90.58	230	2	2.2	1.052	1.129	B 54V4	B 75V4
181	B 34V2	B 38V2	9.29	230	2	2.2	-0.721	0.951	B 34V2	B 45V4
288	B 37V2	B 75V4	12.17	400	2	6.6	-3.413	4.594	B 64V4	B 75V4
296	B 44V2	B 59V4	12.2	400	2	6.6	-3.122	4.185	B 16V4	B 59V4
364	B 32V2	B 74V2	13.71	400	2	6.6	-3.476	4.021	B 14V4	B 45V4
398	B 42V2	B 58V4	14.32	400	2	6.6	-3.864	4.269	B 12V2	B 72V2
417	B 54V4	B 76V2	14.56	230	2	2.2	1.495	1.823	B 55V4	B 76V2
421	B 55V4	B 76V2	14.56	230	2	2.2	1.526	1.99	B 54V4	B 55V4
438	B 34V2	B 74V2	14.89	400	2	6.6	1.247	1.574	B 34V2	B 74V2
519	B 31V2	B 40V2	17.18	230	2	2.2	-1.238	1.714	B 28V2	B 76V2

(continued)

Table K.3 (continued)

No.	From bus	To bus	Length (km)	Voltage level (kV)	No. of lines	Capacity limit (p.u.)	Line flow (p.u.)	Maximum line flow in contingency conditions		
								Flow on line (p.u.)	Relevant contingency	
657	B 44V2	B 62V2	23.77	230	2	2.2	1.017	1.16	B 44V2	B 74V2
659	B 50V2	B 62V2	23.77	230	2	2.2	0.953	1.093	B 44V2	B 74V2
671	B 13V2	B 44V2	25.19	230	2	2.2	-1.347	1.569	B 13V2	B 44V2
701	B 6V2	B 55V4	26.96	230	2	2.2	-1.173	1.426	B 6V2	B 10V2
705	B 53V2	B 74V2	27.61	230	2	2.2	-0.787	0.892	B 64V4	B 75V4
710	B 11V4	B 57V2	28.08	400	2	6.6	4.432	5.193	B 29V4	B 58V4
731	B 39V2	B 44V2	28.89	230	2	2.2	-0.441	0.605	B 44V2	B 74V2
772	B 44V2	B 74V2	32.66	400	2	6.6	3.518	3.914	B 16V4	B 59V4
900	B 29V4	B 72V2	44.55	400	2	6.6	3.348	3.761	B 11V4	B 70V4
938	B 10V2	B 55V4	48.65	400	2	6.6	-1.484	1.688	B 6V2	B 55V4
944	B 34V2	B 45V4	48.99	400	2	6.6	-2.754	3.263	B 14V4	B 45V4
1219	B 19V2	B 60V2	117.4	230	2	2.2	0.24	0.42	B 17V2	B 60V2
1230	B 4V4	B 21V2	121.71	400	2	6.6	1.044	1.507	B 4V4	B 59V4
1237	B 5V2	B 61V2	123.65	230	2	2.2	0.518	0.595	B 4V4	B 59V4
1310	B 27V4	B 43V4	158.74	400	2	6.6	-0.274	0.341	B 54V4	B 77V4

Table K.4 The detailed results of the forward stage (transformers)

No.	Bus name	Voltage level	Transformer capacity (p.u.)
1	B 10V2	400 kV:230 kV	2.75
2	B 57V2	400 kV:230 kV	5.5
3	B 72V2	400 kV:230 kV	5.5
4	B 44V2	400 kV:230 kV	2.75
5	B 32V2	400 kV:230 kV	5.5
6	B 21V2	400 kV:230 kV	2.75
7	B 74V2	400 kV:230 kV	2.75
8	B 37V2	400 kV:230 kV	5.5
9	B 46V2	400 kV:230 kV	2.75
10	B 33V4	400 kV:230 kV	2.75
11	B 34V2	400 kV:230 kV	2.75
12	B 3V2	400 kV:230 kV	5.5
13	B 63V2	400 kV:230 kV	2.75
14	B 42V2	400 kV:230 kV	5.5
15	B 54V4	400 kV:230 kV	2.75
16	B 55V4	400 kV:230 kV	5.5
17	B 64V4	400 kV:230 kV	2.75
18	B 67V4	400 kV:230 kV	2.75
19	B 68V2	400 kV:230 kV	2.75
20	B 71V4	400 kV:230 kV	2.75
21	B 7V2	400 kV:230 kV	5.5

Table K.5 The detailed results of the decrease stage

No.	From bus	To bus	Length (km)	Voltage level (kV)	No. of lines	Capacity limit (p.u.)	Line flow (p.u.)	Maximum line flow in contingency conditions		
								Flow on line (p.u.)	Relevant contingency	
1	B 44V2	B 50V2	1	230	2	2.2	1.565	2.155	B 44V2	B 50V2
35	B 24V2	B 65V2	1.11	230	2	2.2	-1.321	1.79	B 12V2	B 72V2
39	B 39V2	B 40V2	2.86	230	2	2.2	0.877	1.172	B 28V2	B 76V2
41	B 46V2	B 64V4	3.33	230	2	2.2	-0.54	0.827	B 46V2	B 64V4
42	B 46V2	B 64V4	3.33	400	2	6.6	-2.158	2.713	B 32V2	B 46V2
43	B 46V2	B 63V2	3.33	230	1	1.1	0.031	0.144	B 63V2	B 68V2
46	B 64V4	B 68V2	3.5	400	2	6.6	-0.072	1.35	B 64V4	B 75V4
47	B 63V2	B 68V2	3.5	230	2	2.2	-0.59	0.88	B 63V2	B 68V2
48	B 63V2	B 68V2	3.5	400	2	6.6	-2.36	2.741	B 7V2	B 24V2
49	B 41V2	B 64V4	4.24	230	2	2.2	-1.625	2.067	B 41V2	B 63V2
55	B 9V2	B 65V2	4.44	230	2	2.2	1.211	1.924	B 65V2	B 69V2
58	B 3V2	B 71V4	4.51	400	1	3.3	-2.353	2.862	B 11V4	B 70V4
62	B 3V2	B 70V4	4.51	400	1	3.3	-1.652	2.332	B 3V2	B 71V4
63	B 42V2	B 69V2	4.65	230	2	2.2	1.208	1.444	B 12V2	B 72V2
65	B 9V2	B 42V2	4.65	230	1	1.1	-0.489	0.96	B 65V2	B 69V2
67	B 46V2	B 53V2	5.03	230	1	1.1	0.408	0.773	B 49V2	B 53V2
69	B 38V2	B 40V2	5.03	230	1	1.1	-0.308	0.445	B 38V2	B 39V2
71	B 38V2	B 39V2	5.2	230	1	1.1	-0.539	0.692	B 38V2	B 39V2
73	B 3V2	B 49V2	5.55	230	2	2.2	1.536	1.908	B 3V2	B 49V2
75	B 9V2	B 24V2	5.55	230	2	2.2	1.233	1.601	B 65V2	B 69V2
84	B 7V2	B 16V4	5.84	400	2	6.6	-3.158	3.973	B 14V4	B 45V4
86	B 7V2	B 14V4	5.84	400	2	6.6	-3.249	4.043	B 16V4	B 59V4
101	B 12V2	B 24V2	6.7	230	2	2.2	-0.349	2.053	B 12V2	B 72V2
119	B 39V2	B 62V2	7.16	230	2	2.2	1.534	1.716	B 15V2	B 39V2
124	B 7V2	B 68V2	7.2	400	2	6.6	3.154	4.066	B 7V2	B 24V2
125	B 38V2	B 74V2	7.55	230	2	2.2	1.55	1.85	B 39V2	B 74V2
131	B 33V4	B 68V2	7.72	230	2	2.2	0.929	2.108	B 33V4	B 64V4
155	B 36V2	B 72V2	9.02	230	1	1.1	0.315	0.911	B 30V2	B 51V2
169	B 2V2	B 20V2	9.15	230	1	1.1	-0.31	0.627	B 2V2	B 76V2
172	B 67V4	B 71V4	90.58	230	1	1.1	0.805	1.013	B 54V4	B 67V4
181	B 34V2	B 38V2	9.29	230	1	1.1	-0.427	0.851	B 34V2	B 45V4
288	B 37V2	B 75V4	12.17	400	2	6.6	-3.142	4.126	B 64V4	B 75V4
296	B 44V2	B 59V4	12.2	400	2	6.6	-3.396	4.633	B 14V4	B 45V4
364	B 32V2	B 74V2	13.71	400	2	6.6	-3.621	4.21	B 14V4	B 45V4
398	B 42V2	B 58V4	14.32	400	2	6.6	-3.845	4.247	B 12V2	B 72V2
417	B 54V4	B 76V2	14.56	230	2	2.2	1.519	1.852	B 55V4	B 76V2
421	B 55V4	B 76V2	14.56	230	2	2.2	1.549	1.991	B 54V4	B 55V4
519	B 31V2	B 40V2	17.18	230	2	2.2	-1.26	1.751	B 28V2	B 76V2
657	B 44V2	B 62V2	23.77	230	2	2.2	1.117	1.304	B 44V2	B 74V2
659	B 50V2	B 62V2	23.77	230	2	2.2	1.052	1.234	B 44V2	B 74V2
671	B 13V2	B 44V2	25.19	230	1	1.1	-0.82	0.985	B 13V2	B 44V2

(continued)

Table K.5 (continued)

No.	From bus	To bus	Length (km)	Voltage level (kV)	No. of lines	Capacity limit (p.u.)	Line flow (p.u.)	Maximum line flow in contingency conditions		
								Flow on line (p.u.)	Relevant contingency	
701	B 6V2	B 55V4	26.96	230	2	2.2	-1.179	1.43	B 6V2	B 10V2
710	B 11V4	B 57V2	28.08	400	2	6.6	4.492	5.258	B 29V4	B 58V4
731	B 39V2	B 44V2	28.89	230	1	1.1	-0.27	0.378	B 44V2	B 74V2
772	B 44V2	B 74V2	32.66	400	2	6.6	4.027	4.553	B 16V4	B 59V4
900	B 29V4	B 72V2	44.55	400	2	6.6	3.514	3.971	B 11V4	B 70V4
938	B 10V2	B 55V4	48.65	400	2	6.6	-1.506	1.711	B 6V2	B 55V4
944	B 34V2	B 45V4	48.99	400	1	3.3	-1.631	2.092	B 45V4	B 59V4
1219	B 19V2	B 60V2	117.4	230	1	1.1	0.202	0.42	B 17V2	B 60V2
1230	B 4V4	B 21V2	121.71	400	2	6.6	1.054	1.512	B 4V4	B 59V4
1237	B 5V2	B 61V2	123.65	230	2	2.2	0.517	0.593	B 4V4	B 59V4
1310	B 27V4	B 43V4	158.74	400	1	3.3	-0.224	0.279	B 54V4	B 77V4

Table K.6 The detailed results of the decrease stage (transformers)

No.	Bus name	Voltage level	Transformer capacity (p.u.)
1	B 10V2	400 kV:230 kV	2.75
2	B 57V2	400 kV:230 kV	5.50
3	B 72V2	400 kV:230 kV	5.50
4	B 44V2	400 kV:230 kV	2.75
5	B 32V2	400 kV:230 kV	5.50
6	B 21V2	400 kV:230 kV	2.75
7	B 74V2	400 kV:230 kV	2.75
8	B 37V2	400 kV:230 kV	5.50
9	B 46V2	400 kV:230 kV	2.75
10	B 33V4	400 kV:230 kV	2.75
11	B 34V2	400 kV:230 kV	2.75
12	B 3V2	400 kV:230 kV	5.50
13	B 63V2	400 kV:230 kV	2.75
14	B 42V2	400 kV:230 kV	5.50
15	B 54V4	400 kV:230 kV	2.75
16	B 55V4	400 kV:230 kV	5.50
17	B 64V4	400 kV:230 kV	2.75
18	B 67V4	400 kV:230 kV	2.75
19	B 68V2	400 kV:230 kV	2.75
20	B 71V4	400 kV:230 kV	2.75
21	B 7V2	400 kV:230 kV	5.50

Appendix L
Generated Matlab M-files Codes

L.1 GEP1.m

a) "GEP1" M-file code

```
clear
clc
%% Required Input data
%% Required load nodes data:
Gen_Data = xlsread('Gep.xls', 'Gen-Data');
%% Required substations data:
Add_Data = xlsread('Gep.xls', 'Add-Data');
%% Data retrieval from input data
No_Gen = Gen_Data(:,1); % Generator type number
Capaci_Gen = Gen_Data(:,2); % Capacity type plants
% Investment cost type plants:
Invest_Gen = Gen_Data(:,3)*1000;
Life_Gen = Gen_Data(:,4); % Life type plants
FuelCost_Gen = Gen_Data(:,5); % Fuel cost type plants
% Operation and maintenance cost type plants:
O_MCost_Gen = Gen_Data(:,6)*1000*12;
Load = Add_Data(1,1); % Maximum network load (MW)
Reserv = Add_Data(1,2)/100; % Reserve ratio
% Coefficient of annual interest:
Interest_rate = Add_Data(1,3)/100;
Exist_Cap = Add_Data(1,4); % Capacity of existing plants
% Existing power plants, fuel costs:
Exist_FuelCost = Add_Data(1,5);
if isempty(Capaci_Gen)
  fprintf('Input argument "Capaci_Gen" determining');
  fprintf(' capacity type plants.\n');
  error('"Capaci_Gen" is undefined and must be determined.');
end
if isempty(Invest_Gen)
  fprintf('Input argument "Invest_Gen" determining');
  fprintf(' investment cost type plants.\n');
  error('"Invest_Gen" is undefined and must be determined.');
end
if isempty(Life_Gen)
  fprintf('Input argument "Life_Gen" determining');
  fprintf(' life type plants.\n');
  error('"Life_Gen" is undefined and must be determined.');
```

```
end
if isempty(FuelCost_Gen)
  fprintf('Input argument "FuelCost_Gen" determining');
  fprintf('fuel cost type plants.\n');
  error('"FuelCost_Gen" is undefined and must be deter-
mined.');
end
if isempty(O_MCost_Gen)
  fprintf('Input argument "O_MCost_Gen" determining');
  fprintf(' operation and maintenance cost type plants.\n');
  error('"O_MCost_Gen" is undefined and must be determined.');
end
if isempty(Load)
  fprintf('Input argument "Load" determining');
  fprintf(' maximum network load(MW).\n');
  error('"Load" is undefined and must be determined.');
end
if isempty(Reserv)
  fprintf('Input argument "Reserv" determining');
  fprintf(' reserve ratio.\n');
  error('"Reserv" is undefined and must be determined.');
end
if isempty(Interest_rate)
  fprintf('Input argument "Interest_rate" determining');
  fprintf(' coefficient of annual interest.\n');
  error('"Interest_rate" is undefined & must be determined.');
end
if isempty(Exist_Cap)
  fprintf('Input argument "Exist_Cap" determining');
  fprintf(' capacity of existing plants.\n');
  error('"Exist_Cap" is undefined and must be determined.');
end
if isempty(Exist_FuelCost)
  fprintf('Input argument "Exist_FuelCost" determining');
  fprintf(' existing power plants, fuel costs.\n');
  error('"Exist_FuelCost" is undefined & must be determined');
end
%
if (Capaci_Gen==0)
  fprintf('Input argument "Capaci_Gen" determining');
  fprintf(' capacity type plants.\n');
  error('"Capaci_Gen" should not be zero.');
end
if (find(Life_Gen==0))
  fprintf('Input argument "Life_Gen" determining');
  fprintf(' life type plants.\n');
  error('"Life_Gen" should not be zero.');
end
if (Load==0)
  fprintf('Input argument "Load" determining');
  fprintf(' maximum network load(MW).\n');
  error('"Load" should not be zero.');
end
if (Reserv<0)
```

```
    fprintf('Input argument "Reserv" determining');
    fprintf(' reserve ratio.\n');
    error('"Reserv" should not be less than zero.');
  end
  if (Interest_rate<=0)
    fprintf('Input argument "Interest_rate" determining');
    fprintf(' coefficient of annual interest.\n');
    error('"Interest_rate" should be greater than zero.');
  end
  %%
  Gepp;
  %% Print obtained results in command window and results.txt
  Print_GEPP;
```

b) "Gepp" M-file code

```
%% Problem outputs
%Best_Gen: The best units selected
%% Problem inputs
%Capaci_Gen; Capacity type plants
%Invest_Gen; Investment cost type Plants
%Life_Gen; Life type plants
%FuelCost_Gen; Fuel cost type plants
%O_MCost_Gen; Operation and maintenance cost type plants
%Load; Maximum network load(MW)
%Energy; Annual energy consumption(MWh)
%Reserv; Reserve ratio
%Interest_rate; Coefficient of annual interest
%Exist_Cap; Capacity of existing plants
%Exist_FuelCost; Existing power plants, fuel costs

%%Choose the cheapest power plants to produce
CheapFuel = FuelCost_Gen;
CheapFuel(4) = Exist_FuelCost;
[CheapFuel,ICheapFuel] = sort(CheapFuel);
A = (1+Interest_rate);
for i = 1:3
   A_P(i) = (A^Life_Gen(i,1))*Interest_rate;
   A_P(i) = A_P(i)/(A^Life_Gen(i,1)-1);
end
B = zeros (3,1331);
m = 0;
%Create all the solution space
for i = 0:10
    for j = 0:10
        for k = 0:10
            m = m+1;
            B(1,m) = i;
            B(2,m) = j;
            B(3,m) = k;
        end
    end
end
%Calculate the cost of each choice
```

```
for i = 1:1331
    Total_Cap = Exist_Cap+B(1,i)*Capaci_Gen(1)+B(2,i)*...
        Capaci_Gen(2)+B(3,i)*Capaci_Gen(3);
    if Total_Cap < Load*(1+Reserv)
        Total_Cost(i) = 1.0e12;
    else
        Total_Cost(i) = 0.0;
%Calculate the energy production plant
        Energy = Load*8760;
        Energy1 = Energy;
        for j = 1:3
           Energy_Gen(j) = B(j,i) * Capaci_Gen(j) * 8760;
        end
        Energy_Gen(4)=Exist_Cap*8760;
        for j = 1:4
           ii = ICheapFuel(j);
           Energy1 = Energy1-Energy_Gen(ii);
           if Energy1<0.0
              Energy_Gen(ii) = Energy1+Energy_Gen(ii);
              if Energy_Gen(ii)<0.0
                 Energy_Gen(ii) = 0.0;
              end
           end
        end
        if Energy1<=0.0
           for j = 1:3
               Total_Cost(i) = Total_Cost(i)+...
                B(j,i)*Capaci_Gen(j)*...
                (Invest_Gen(j)*A_P(j)+O_MCost_Gen(j))...
                +FuelCost_Gen(j)*Energy_Gen(j);
           end
           Total_Cost(i) = Total_Cost(i)+...
               Exist_FuelCost*Energy_Gen(4);
        else
           Total_Cost(i) = 1.0e12;
        end
    end
end
%Choose the best option
[Solution,II] = min(Total_Cost);
Best_Gen(1) = B(1,II);
Best_Gen(2) = B(2,II);
Best_Gen(3) = B(3,II);
Energy1 = Energy;
for j = 1:3
   Energy_Gen(j) = Best_Gen(j)*Capaci_Gen(j)*8760;
end
Energy_Gen(4) = Exist_Cap*8760;
for j = 1:4
  ii = ICheapFuel(j);
  Energy1 = Energy1-Energy_Gen(ii);
  if Energy1<0.0
     Energy_Gen(ii) = Energy1+Energy_Gen(ii);
     if Energy_Gen(ii)<0.0
```

```
          Energy_Gen(ii) = 0.0;
        end
     end
  end
```

c) "Print_GEPP" M-file code

```
%% Print different costs and optimal capacity of each plant
clc
fprintf('\n Optimal Capacity_Plant1 = %4i',Best_Gen(1));
fprintf('\n Optimal Capacity_Plant1 = %4i',Best_Gen(2));
fprintf('\n Optimal Capacity_Plant1 = %4i',Best_Gen(3));
InvestCost = 0.0;
FuelCost = 0.0;
O_MCost = 0.0;
for i = 1:3
   InvestCost = InvestCost+A_P(i)*Best_Gen(i)*...
       Capaci_Gen(i)*Invest_Gen(i);
   FuelCost = FuelCost+FuelCost_Gen(i)*Energy_Gen(i);
   O_MCost = O_MCost+Best_Gen(i)*Capaci_Gen(i)*...
       O_MCost_Gen(i);
end
FuelCost = FuelCost+Exist_FuelCost*Energy_Gen(4);
Total_Cost1 = InvestCost+FuelCost+O_MCost;
fprintf('\n\n*******************************************');
fprintf('**Result***************************************');
fprintf('************\n');
fprintf('| Capacity_Plant1 | Capacity_Plant2 | Capacity');
fprintf('_Plant3 | Investment cost | Fuel cost | Fixed O');
fprintf('&M cost |\n');
fprintf('|      (Mw)         |      (Mw)         |      (Mw) ');
fprintf('      |      (R/yr)      |  (R/yr)    |    (R/yr) ');
fprintf('      |');
fprintf('\n|     %6.2f       |   %6.2f          |     %6.2f',...
    Best_Gen(1)*Capaci_Gen(1),Best_Gen(2)*Capaci_Gen(2),...
    Best_Gen(3)*Capaci_Gen(3));
fprintf('      |   %10.2E    |%10.2E |  %10.2E    |\n',...
    InvestCost,FuelCost,O_MCost);
fprintf('***********************************************');
fprintf('***********************************************');
fprintf('*********\n');
fprintf('\n Total Cost(R) = %10.2E \n',Total_Cost1);

fid = fopen('result.txt', 'wt');
fprintf(fid,'\n Optimal Capacity_Plant1 = %4i',Best_Gen(1));
fprintf(fid,'\n Optimal Capacity_Plant2 = %4i',Best_Gen(2));
fprintf(fid,'\n Optimal Capacity_Plant3 = %4i',Best_Gen(3));
fprintf(fid,'\n\n**************************************');
fprintf(fid,'***Result*********************************');
fprintf(fid,'***********************\n');
fprintf(fid,'\n| Capacity_Plant1 | Capacity_Plant2 | ');
fprintf(fid,'Capacity_Plant3 | Investment cost | Fuel');
fprintf(fid,' cost| Fixed O&M cost |\n');
fprintf(fid,'|      (Mw)         |      (Mw)         |      ');
```

```
fprintf(fid,'(Mw)        |      (R/yr)       |  (R/yr)    | ');
fprintf(fid,'   (R/yr)       |');
fprintf(fid,'\n|     %6.2f       |     %6.2f        |      '...
    ,Best_Gen(1)*Capaci_Gen(1),Best_Gen(2)*Capaci_Gen(2));
fprintf(fid,'%6.2f       |   %10.2E    |%10.2E |  %10.2E'...
    ,Best_Gen(3)*Capaci_Gen(3),InvestCost,FuelCost,O_MCost);
fprintf(fid,''    |\n');
fprintf(fid,'\n*******************************************');
fprintf(fid,'*******************************************');
fprintf(fid,'*******************\n');
fprintf(fid,'\n Total cost(R) = %10.2E \n',Total_Cost1);
fclose(fid);
```

L.2 GEP2.m

a) "GEP2" M-file code

```
clear
clc
%%Required input data
Busdata = xlsread('Gepdata.xls', 'Busdata');
Linedata = xlsread('Gepdata.xls', 'Linedata');
Candidatesdata = xlsread('Gepdata.xls', 'Candidatesdata');
%Maximum capacity that line i can be enhanced:
Biu = xlsread('Gepdata.xls', 'Biu');
%Investment cost for transmission lines enhancement(R/p.u.km)
Ga = xlsread('Gepdata.xls', 'Gama');
%% Data retrieval from input data
%Candidate buses for generation expansion:
Candidates = Candidatesdata(:,1);
%Beta(i):Investment factor cost of generation expansion
    %in bus i
Beta = Candidatesdata(:,2);
%PGmax(i):Maximum generation expansion limit of bus i
PGmax = Candidatesdata(:,3);
%PGmin(i):Minimum generation expansion limit of bus i
PGmin = Busdata(:,3);
Nlin = Linedata(:,1); %Line number
Nl = Linedata(:,2); %Nl:From bus
Nr = Linedata(:,3); %Nr:To bus
R = Linedata(:,4); %R(i):Line resistance
X = Linedata(:,5); %X(i):Line reactance
%Smax(i):Maximum thermal rating of line i
Smax = Linedata(:,6);
%Length(i):Path length of line i
Length = Linedata(:,7);
Busn = Busdata(:,1); %Bus number
Btype = Busdata(:,2); %Type of bus 1-Slack, 2-PV, 3-PQ
Pg = Busdata(:,3); %Pg(i):Generation of bus i
Pl = Busdata(:,4); %Pl(i):Load of bus i
Nc = setxor(Busn,Candidates); %Nc:Non-candidate buses
```

```
[Ybus] = ybus(Busdata, Linedata); %Computing Ybus
%%
[Gi,Ol,To,Ef] = GEPP(Candidates,Nc,Beta,PGmax,...
PGmin,X,Btype,Nl,Nr,Smax,Length,Biu,Ga,Pg,Pl,Ybus);
%Gi:Generation of candidate buses after expansion
%Ol:Overloaded lines after expansion
%To:Total overload after expansion
%%
if Ef==1
    Print_Gep
else
    fprintf('There is no feasible solution.\n');
end
```

b) "ybus" M-file code

```
function [Ybus] = ybus (Busdata, Linedata)

nbus = size(Busdata,1);
nl = Linedata(:,2);
nr = Linedata(:,3);
Ld = Linedata;
%%
j = sqrt(-1);
X = Ld(:,5);
nbr = length(Ld(:,1));
%Z = R + j*X;
Z = (j*X);
y = ones(nbr,1)./Z;     %Branch admittance
%for n = 1:nbr
Ybus = zeros(nbus,nbus);%Initialize Ybus to zero
%%
%Formation of the off diagonal elements
for k = 1:nbr;
       Ybus(nl(k),nr(k))  = Ybus(nl(k),nr(k))-y(k);
       Ybus(nr(k),nl(k))  = Ybus(nl(k),nr(k));
end
%%
%Formation of the diagonal elements
for n = 1:nbus
    for m = (n+1):nbus
        Ybus(n,n)  = Ybus(n,n)-Ybus(n,m);
    end
    for m = 1:n-1
        Ybus(n,n)  = Ybus(n,n)-Ybus(n,m);
    end
end
```

c) "GEPP" M-file code

```
function [Gi, Ol, To, Ef]= GEPP (Candidates, Nc, Beta,...
    PGmax,PGmin, X, Btype, Nl, Nr, Smax, Length, Biu,...
    Ga, Pg, Pl, Ybus)
if isempty(Ybus)
  error('Input argument "Ybus" is undefined.');
```

```
end
if isempty(Pg)
  fprintf('Input argument "Pl" determining');
  fprintf(' load demand of buses.\n');
  error('"Pl" is undefined and must be determined.');
end
if isempty(Pg)
  fprintf('Input argument "Pg" determining');
  fprintf(' generation of buses.\n');
  error('"Pg" is undefined and must be determined.');
end
if isempty(Ga)
  fprintf('Input argument "Ga" determining Investment ');
  fprintf('cost of transmission lines enhancement.\n');
  warning('"Ga" is undefined and is set to a default value.');
  Ga = 20;
end
if isempty(Biu)
  fprintf('Input argument "Biu" determining ');
  fprintf('maximum capacity of lines enhancement.\n');
  warning('"Biu" is undefined & is set to a default value.');
  Biu = 1.1;
end
if isempty(Length)
  fprintf('Input argument "Length" determining');
  fprintf(' path length of lines.\n');
  error('"Length" is undefined and must be determined.');
end
if isempty(Smax)
  fprintf('Input argument "Smax" defining');
  fprintf(' lines thermal loading before expansion.\n');
  error('"Smax" is undefined and must be determined.');
end
if isempty(Nr) || isempty(Nl)
  fprintf('Input argument "NL" & "Nr" defining');
  fprintf(' lines sending and ending buses.\n');
  error('"NL" & "Nr" are undefined and must be determined.');
end
if isempty(Btype)
  fprintf('Input argument "Btype" defining');
  fprintf(' information of bus types.\n');
  error('"Btype" is undefined and must be determined.');
end
if isempty(X)
  fprintf('Input argument "X" containing');
  fprintf(' data of lines reactance.\n');
  error('"X" is undefined and must be determined.');
end
if isempty(PGmin)
  fprintf('Input argument "PGmin" defining minimum ');
  fprintf('generation expansion limit of candidate buses.\n');
  error('"PGmin" is undefined and must be determined.');
end
if isempty(PGmax)
```

```
    fprintf('Input argument "PGmax" defining maximum ');
    fprintf('generation expansion limit of candidate buses.\n');
    error('"PGmax" is undefined and must be determined.');
  end
  if isempty(Beta)
    fprintf('Input argument "Beta" defining investment cost');
    fprintf('of generation expansion in candidate buses.\n');
    error('"Beta" is undefined and must be determined.');
  end
  if isempty(Candidates)
    fprintf('Input argument "Candidates" defining');
    fprintf('candidate buses.\n');
    error('"Candidates" is undefined and must be determined.');
  end
  %% Problem outputs
  %Gi:Generation of candidate buses after expansion
  %Ol:Overloaded lines after expansion
  %To:Total overload after expansion
  %Ef:Exit flag, integer identifying the reason the algorithm
      %is terminated. Ef is 1, if there is a feasible solution
  %% Problem Inputs
  %Candidates:Candidate buses for generation expansion
  %Beta(i):investment cost of generation expansion in bus i
  %PGmax(i):Maximum generation expansion limit of bus i
  %PGmin(i):Minimum generation expansion limit of bus i
  %Nlin:Line number
  %Nl:Line from bus
  %Nr:Line to bus
  %R(i):Line resistance
  %X(i):Line reactance
  %Smax(i):Maximum thermal rating of line i
  %Length(i):Path Length of Line i
  %Busn:Bus number
  %Btype:Type of bus 1-Slack, 2-PV, 3-PQ
  %Pg(i):Generation of bus i
  %Pl(i):load of bus i
  %Nc:Non-candidate buses
  %%Obtaining Ybus matrix
  %%
  Ps = (Pg-Pl);
  Na = size (Pg, 1);
  M = size (X, 1);
  %%%%
  [Nons] = find(Btype~=1);
  Nx = length(Nons);
  B = zeros (Nx,Nx);
  for k = 1:Nx
      for j = 1:Nx
          Ymn = Ybus(Nons(k),Nons(j));
          B(k,j) = -imag(Ymn);
      end
  end
  E = inv (B);
  Binv = zeros (Na,Na);
```

```
for k = 1:Nx
    an = Nons(k);
    for j = 1:Nx
        am = Nons(j);
        Binv(an,am) = E(k,j);
    end
end
%% Computing branch admittance calculation (b)
%The admittance matrix in which bii is the admittance
    % of line i and non-diagonal elements are zero
jay = sqrt(-1);
Z = (jay*X);
Y = ones(M,1)./Z;
b = zeros (M,M);
for i = 1:M
    b(i,i) = -imag(Y(i));
end
%% Computing connection matrix (A)
% The connection matrix (M*N) in which aij is 1, if a
    % line exists from bus i to bus j; otherwise zero.
A = zeros (M, Na);
for i = 1:M
    nl = Nl(i);
    nr = Nr(i);
    A(i, nl) = 1;
    A(i, nr) = -1;
end
%% Computing sensitivity matrix (a)
theta = Binv*Ps;
a = b*A*Binv;
%% The line flows are calculated as follows:
Pli = zeros (M,1);
for i = 1:M
    for k = 1:Na
        Pli(i,1) = Pli(i,1)+(a(i,k)*(Pg(k,1)-Pl(k,1)));
    end
end
%% Generation expansion cost of each bus
Pmax = zeros (Na,1);
beta = zeros (Na,1);
for j = 1:length (Nc)
    Inc = Nc(j);
    beta(Inc) = 10^10;
    Pmax(Inc) = 0.000001;
end
for j = 1:length (Candidates)
    Ica = Candidates(j);
    beta(Ica) = Beta(j);
    Pmax(Ica) = PGmax(j);
end
Beta = beta;
PGmax = Pmax;
%% Investment cost for transmission lines enhancement (R/MW)
Gama = Ga*Length;
```

```
%% Maximum possible capacity expansion of each line
Biu = Biu.*ones(M,1);
%% Thermal rating of each line
Pcu = Smax; %Upper bound of thermal rating of each line
Pcl = -Pcu; %Lower bound of thermal rating of each line
%% Defining objective function
for k = 1:Na
    OF(k) = Beta(k);
end
for i = 1:M
    I = i+Na;
    OF(I) = Gama (i);
end
%% First set of inequality constraints: determining
    %% minimum permissible thermal rating of each line
for i = 1:M
    C(i) = (-a(i,:)*Pg)+Pli(i);
end
GH1 = zeros (M, M+Na);
bGH1 = zeros (M,1);
for i = 1:M
    for k = 1:Na
        GH1(i,k) = -a(i,k);
    end
    I = i+Na;
    GH1(i,I) = Pcl(i);
    bGH1(i,1) = C(i);
end
%% Second set of inequality constraints: determining
    %% maximum permissible thermal rating of each line
GH2 = zeros (M, M+Na);
bGH2 = zeros (M,1);
for i = 1:M
    for k = 1:Na
        GH2(i,k) = a(i,k);
    end
    I = i+Na;
    GH2(i,I) = -Pcu(i);
    bGH2(i,1) = -C(i);
end
%% Integrating all inequality constraints
    %% to one matrix, called An & bn here
for i = 1:M
    An(i,:) = GH1(i,:);
    bn(i) = bGH1(i);
    I = i+M;
    An(I,:) = GH2(i,:);
    bn(I) = bGH2(i);
end
%% Determining upper and lower bounds of
    %% decision variables, called lb & ub here
lb = zeros (M+Na,1);
ub = zeros (M+Na,1);
for k = 1:Na
```

```
        lb(k,1) = PGmin(k);
        ub(k,1) = PGmax(k);
    end
    for i = 1:M
        I = i+Na;
        lb(I,1) = 1;
        ub(I,1) = Biu(i);
    end
    %% Defining equality constraint
        %% (Total generation = Total demand)
    Aeq = zeros (1, Na+M);
    for k = 1:Na
        Aeq(1,k) = 1;
    end
    beq = sum (Pl);
    %% Solving the problem and finding the optimal point
    [Dv, Fval, Ef] = linprog(OF,An,bn,Aeq,beq,lb,ub);
    To = 0;
    if Ef~=1
        fprintf('\nWARNING: No feasible solution was found.')
        Gi = zeros(size(Candidates,1),1);
        Ol = zeros(M,1);
    else
        for k = 1:size(Candidates,1)
            Gi (k,1) = Candidates(k,1);
            Gi (k,2) = Dv(k);
        end
        for i = 1:M
            I = i+Na;
            Ol (i,1) = Nl(i);
            Ol (i,2) = Nr(i);
            Ol (i,3) = Dv(I,1)-1;
            To = To+(Dv(I)-1);
        end
    end
```

d) "Print_Gep" M-file code

```
clc
fprintf('**************************************************');
fprintf('***************\n');
fprintf('Generation of each candidate bus after expansion');
fprintf(' is as follows: \n');
fprintf('**************************************************');
fprintf('***************\n');
fprintf('             |Bus number|          |Gi (p.u.)|');
for i = 1:size(Gi,1)
    fprintf('\n %18.0f % 22.2f', Gi(i,1), Gi(i,2));
end
fprintf('\n\n*********************************************');
fprintf('********************\n');
fprintf('      Total overload value and enhanced lines ');
fprintf('are as follows\n');
fprintf('*************************************************');
```

```
fprintf('****************\n');
fprintf('                      |Total overload| \n');
fprintf('%31.2f \n', To);
if To>=0.0001
    El = find (Ol(:,3)>=0.0001);
    Sel = length(El);
    fprintf('********************************************');
    fprintf('********************\n');
    fprintf('                      |Enhanced lines|      ');
    fprintf('                \n');

    fprintf('     |From bus|           |To bus|         ');
    fprintf('|Enhancement(%%)|');
    for i = 1:Sel
        fprintf('\n %10i  %18i  % 19.2f \n',...
            Ol(El(i),1), Ol(El(i),2), Ol(El(i),3)*100);
    end
    fprintf('\n********************************************');
    fprintf('********************\n');
else
    fprintf('\n                      No enhanced line    ');
    fprintf('                   \n');
end

%% Printing the results in results.txt
fid = fopen('results.txt', 'wt');
fprintf(fid,'*********************************************');
fprintf(fid,'*******************\n');
fprintf(fid,...
    'Generation of each candidate bus after expansion');
fprintf(fid,'is as follows: \n');
fprintf(fid,'**********************************************');
fprintf(fid,'******************\n');
fprintf(fid,'             |Bus number|           |Gi (p.u.)|');
for i = 1:size(Gi,1)
    fprintf(fid,'\n %18.0f % 22.2f', Gi(i,1), Gi(i,2));
end
fprintf(fid,'\n\n*****************************************');
fprintf(fid,'***********************\n');
fprintf(fid,'      Total overload value and enhanced lines');
fprintf(fid,' are as follows\n');
fprintf(fid,'*********************************************');
fprintf(fid,'*******************\n');
fprintf(fid,'                      |Total overload| \n');
fprintf(fid,'%31.2f \n', To);
if To>=0.0001
    El = find (Ol(:,3)>=0.0001);
    Sel = length(El);
    fprintf(fid,'******************************************');
    fprintf(fid,'**********************\n');
    fprintf(fid,'                      |Enhanced lines|    ');
    fprintf(fid,'                   \n');

    fprintf(fid,'     |From bus|           |To bus|        ');
```

```
    fprintf(fid,'   |Enhancement(%%)|');
    for i = 1:Sel
        fprintf(fid,'\n %10i  %18i  % 19.2f \n',...
            Ol(El(i),1), Ol(El(i),2), Ol(El(i),3)*100);
    end
    fprintf(fid,'\n************************************');
    fprintf(fid,'*************************\n');
else
    fprintf(fid,'\n                       No enhanced line ');
    fprintf(fid,'                     \n');
end
```

L.3 SEP.m

a) "SEP.m" M-file code

```
clear
clc
%% Required Input data
%Required load nodes data:
Ln = xlsread('Sepdata.xls', 'Load nodes');
%Required substations data:
Sub = xlsread('Sepdata.xls', 'Substations');
%Maximum possible distance between load nodes and substations:
Dmax = xlsread('Sepdata.xls', 'Dmax');
%% Data retrieval from input data
Iln = Ln (:,1);%Load node number
%Geographical position of load nodes
    %in terms of X (Lx) and Y (Ly):
Lx = Ln(:,2); Ly = Ln(:,3);
Sl = Ln(:,4); %Sl(i):The load i magnitude in MVA
%The cost of downward feeder unit length (e.g. 1 km)
    %per one unit power transfer capability (e.g. 1 MVA):
Gl = Ln(:,5);
Isub = Sub (:,1); %Substation number
%Geographical position of substations
    %in terms of X (Sx) and Y (Sy):
Sx = Sub(:,2); Sy = Sub(:,3);
Cexis = Sub(:,4); %Existing capacity of substations
%Smax(j):Maximum capacity of the jth substation
Smax = Sub(:,5);
%Gsf(j):The fixed cost of a substation (land cost)
    %for the jth candidate location
Gsf = Sub(:,6);
%Gsv(j):The variable cost of jth substation per MVA
Gsv = Sub(:,7);
%%
[Soc, Cstat_var, Cdown_line, Cstat_fix, Ctotal, XX, Ef]=...
    SEPP (Iln, Lx, Ly, Sl, Gl, Isub, Sx, Sy,...
    Cexis, Smax, Gsf, Gsv, Dmax);
%SOC:Geographical position and optimal capacity of
```

```
    %HV substations after expansion
%Cstat_var:Variable cost of HV substations
%Cstat_fix:Fixed cost of HV substations
%Cdown_line:Downward grid cost
%XX(i,j):1 means the jth load center is
    %connected to the ith substaion
if Ef == 1
    %% Printing the obtained results in the
        %% command window and results.txt
    Print_SEPP
    %% Plotting the expansion results
    Plot_SEPP
else
    fprintf...
      ('\nThere is no feasible solution for this case.\n');
End
```

b) "SEPP" M-file code

```
function [Soc, Cstat_var, Cdown_line, Cstat_fix, Ctotal,...
    XX, Ef]=SEPP (Iln, Lx, Ly, Sl, Gl, Isub, Sx, Sy, ...
    Cexis, Smax, Gsf, Gsv, Dmax);
if isempty(Dmax), Dmax=50; end
if isempty(Gsv)
  fprintf('Input argument "Gsv" containing');
  fprintf(' the variable cost of substations');
  warning('"Gsv" is undefined and is set to a default value');
  Gsv = 2500*ones(size(Sub,1),1);
end
if isempty(Gsf)
  fprintf('Input argument "Gsf" containing');
  fprintf(' the fixed cost of substations');
  warning('"Gsf" is undefined and is set to a default value');
  Gsf = 1700000*ones(size(Sub,1),1);
end
if isempty(Smax)
  fprintf('Input argument "Smax" containing');
  fprintf(' the maximum capacity of substations');
  warning('"Smax" is undefined & is set to a default value');
  Smax = 100*ones(size(Sub,1),1);
end
if isempty(Cexis)
  fprintf('Input argument "Cexis" containing');
  fprintf(' the existing capacity of substations');
  error('"Cexis" is undefined and must be determined');
end
if isempty(Sx) || isempty(Sy)
  fprintf('Input arguments "Sx" & "Sy" containing');
  fprintf(' the geographical position of substations');
  error('"Sx" & "Sy" are undefined and must be determined');
end
if isempty(Isub), Isub=1:size(Sub,1); end
if isempty(Gl)
  fprintf('Input argument "Gl" containing');
```

```
    fprintf(' the cost of downward feeder');
    warning('"Gl" is undefined and is set to a default value');
    Gl = 80*ones(size(Ln,1),1);
  end
  if isempty(Sl)
    fprintf('Input argument "Sl" containing');
    fprintf(' the load magnitude of each load node');
    error('"Sl" is undefined and must be determined');
  end
  if isempty(Lx) || isempty(Ly)
    fprintf('Input arguments "Lx" & "Ly" containing');
    fprintf(' the geographical position of load nodes');
    error('"Lx" & "Ly" are undefined and must be determined');
  end
  if isempty(Iln), Iln=1:size(Ln,1); end
  %% Problem outputs
  %SOC:Geographical position and optimal capacity
      % of HV substations after expansion
  %Cstat_var:Variable cost of HV substations
  %Cstat_fix:Fixed cost of HV substations
  %Cdown_line:Downward grid cost
  %% Problem Inputs
  %Iln:Load node number
  %Lx & Ly:geographical position of load nodes
      % in terms of X  and Y
  %Sl(i)=The load i magnitude in MVA
  %Gl:The cost of downward feeder unit length (e.g. 1 km)
      %per one unit power transfer capability (e.g. 1 MVA)
  %Isub:Substation number
  %Sx & Sy:Geographical position of substations
      % in terms of X (Sx) and Y (Sy)
  %Cexis:Current capacity of substations
  %Smax(j):Maximum capacity of the jth substation
  %Gsf(j):The fixed cost of a substation (land cost)
      % for the jth candidate location
  %Gsv(j):The variable cost of jth substation per MVA
  %Dmax:Maximum permissible distance between
      % load nodes and substations
  %%
  Nl = size(Iln,1); % Number of load nodes
  Ns = size(Isub,1); % Number of substations
  Nls = Nl*Ns;
  %% Distance matrix (computing distances between
      %%  the load nodes and the substations)
  for i = 1:Nl
      for j = 1:Ns
          D(i,j) = sqrt(((Sx(j)-Lx(i))^2)+((Sy(j)-Ly(i))^2));
          if D(i,j)>Dmax
              D(i,j) = 100000000000000;
          end
      end
  end
  %% Objective function (forming the objective
      %% function of sep problem)
```

```
for i = 1:Nl
    for j = 1:Ns
        b = ((i-1)*Ns)+j;
        bb = (j)+(Nls);
        fc_total(b) = (Gsv(j)*Sl(i))+(Gl(i)*D(i,j));
        %       fc_total(bb) = (Gsf(j))-(Cexis(j)*Gsv(j));
        fc_total(bb) = (Gsf(j));
        fcstat_var(b) = (Gsv(j)*Sl(i));
        fcdown_line(b) = (Gl(i)*D(i,j));
    end
end
%% Forming constraints
%% Forming equality constraints
Aeq = zeros(Nl,((Nls)+Ns));
for i = 1:Nl
    for j = 1:Ns
        p = ((i-1)*Ns)+(j);
        Aeq(i,p) = 1;
    end
end
Beq=ones(Nl,1);
%% Defining different components of inequality constraints
%% Defining constraints corresponding
    %% with maximum capacity of each substaion
A2 = zeros(Ns,((Nls)+Ns));
for j = 1:Ns
    for i = 1:Nl
        bb = ((i-1)*Ns)+j;
        A2(j,bb) = ((Sl(i)));
    end
    b2(j,1) = Smax(j,1);
end
%% Defining constraints corresponding
    %% with presence of candidate substation
A3 = zeros(Ns,(Nls+Ns));
for j = 1:Ns
    for i = 1:Nl
        bb = ((i-1)*Ns)+j;
        A3(j,bb) = 1;
    end
    A3(j,(Nls+j)) = -Nl;
    b3(j,1) = 0;
end
%% Integrating all inequality constraints
    %% to one matrix, called A & B here
A = zeros((2*Ns), (Nls+Ns));
B = zeros((2*Ns), 1);
for M = 1:Ns
    A(M,:) = A2(M,:);
    B(M,1) = b2(M,1);
end
for m = 1:Ns
    MM = m+Ns;
    A(MM,:) = A3(m,:);
```

```
        B(MM,1) = b3(m,1);
    end
    %% Solving the problem by branch and bound solver
    [x, Fval, Ef] = bintprog(fc_total, A, B, Aeq, Beq);
    if Ef~=1
        fprintf('\nWARNING: No feasible solution was found ')
        Soc(:,1) = Isub(:,1); Soc(:,2) = Sx;
        Soc(:,3) = Sy; Soc(:,4) = zeros(Ns,1);
        Cstat_var = 0; Cstat_fix = 0; Cdown_line = 0;
        Ctotal = 0; XX = zeros(Nl,Ns);
    else
        %% Calculating the optimal capacity of substations
            %% based on the obtained decision variables in 'x'
        for i = 1:Nl
            for j = 1:Ns
                xx(i,j) = x((((i-1)*Ns)+j),1);
            end
        end
        xx = xx'; XX = xx; %Decision variables
        clear m n
        for m = 1:Nls
            xls(m,1) = x(m,1);
        end
        for n = 1:Ns
            xs(n,1) = x(n+Nls,1);
        end
       %Computing optimal capacity of substations after expansion:
        oc = xx*Sl;
        Soc(:,1) = Isub(:,1); Soc(:,2) = Sx;
        Soc(:,3) = Sy; Soc(:,4) = oc(:,1);
        %% Calculating different components of total cost
        [iaab] = find(Cexis|0);
        iq = 0;
        for q = 1:length(iaab)
            if oc(iaab(q))<Cexis(iaab(q))
                iq = iq+1;
                ip(iq) = iaab(q);
            end
        end
        for jj = 1:iq
            for ii = 1:Nl
                bq = (((ip(jj)-1)*Nl)+ii);
                Cstat_var(bq) = 0;
            end
        end

        for jjj = 1:length(iaab)
            Cstat_fix(iaab(jjj)) = (0);
        End
        %Variable cost of installed substations:
        Cstat_var = (fcstat_var*xls)-((Cexis')*Gsv);
        %Fixed cost of installed substations:
        Cstat_fix = ((Gsf')*xs);
        %Variable cost of lines:
```

```
    Cdown_line = (fcdown_line*xls);
    Ctotal = Cstat_var+Cstat_fix+Cdown_line;
end
%%
```

c) "Print_SEPP" M-file code

```
%% Printing different costs
%% Printing optimal capacity of each substation
clc
Nl = size(Iln,1); Ns = size(Isub,1);
fprintf('******************************Costs************');
fprintf('********************************\n');
fprintf('||Cstat_var      ||  Cstat_fix       ||    ');
fprintf('Cdown_line    ||    Ctotal    ||\n');
fprintf('||    (R)         ||       (R)          ||         ');
fprintf('(R)        ||        (R)        ||');
fprintf('\n %10.1f  %18.1f %19.1f % 19.1f \n',...
    Cstat_var, Cstat_fix, Cdown_line, Ctotal);

%% Printing the optimal capacity of substations
%% Printing the locations of substations
fprintf('\n');
fprintf('************************************************');
fprintf('********************************\n');
fprintf('***The position and optimal capacity of installed');
fprintf(' substations after expansion***\n');
fprintf('***********************************************');
fprintf('*********************************\n');
fprintf(' |Sub_number|    |X|          |Y|    |Optimal ');
fprintf('capacity|\n');
for i = 1:Ns
    if Soc(i,4)~=0
        fprintf('%8.f', Soc(i,1));
        fprintf('    %8.f', Soc(i,2));
        fprintf('    %8.f', Soc(i,3));
        fprintf('     %8.1f',Soc(i,4));
        fprintf('\n');
    end
end
fprintf('***********************************************');
fprintf('********\n');

%% Connected loads nodes
    %% to the selected substation after expansion
for i = 1:Ns
    if Soc(i,4)~=0
        Cln = find (XX(i,:)~=0);
        fprintf('****************************************');
        fprintf('***************\n');
        fprintf(' Connected load nodes to the substation');
        fprintf('%3.0f  are:  \n', Soc(i,1));
        fprintf('|Load_node|    |X|       |Y|    |');
        fprintf('Magnitude(MVA)|\n');
```

```
        for i = 1:length(Cln)
            fprintf('\n %6.0f % 10.0f % 8.0f %10.1f',...
                Cln(i), Lx(Cln(i)), Ly(Cln(i)), Sl(Cln(i)));
        end
        fprintf('\n************************************');
        fprintf('*****************\n');
    end
end
%% Printing the results in results.txt
fid = fopen('results.txt', 'wt');
Nl = size(Iln,1); Ns = size(Isub,1); %
fprintf(fid,'******************************Costs********');
fprintf(fid,'*************************************\n');
fprintf(fid,'||Cstat_var       ||  Cstat_fix       ||    ');
fprintf(fid,'Cdown_line     ||    Ctotal     ||\n');
fprintf(fid,'||     (R)         ||       (R)         ||    ');
fprintf(fid,'    (R)        ||        (R)        ||');
fprintf(fid,'\n %10.1f  %18.1f %19.1f % 19.1f \n',...
    Cstat_var, Cstat_fix, Cdown_line, Ctotal);

fprintf(fid,'\n');
fprintf(fid,'*****************************************');
fprintf(fid,'*************************************\n');
fprintf(fid,'***The position and optimal capacity of ');
fprintf(fid,'installed substations after expansion***\n');
fprintf(fid,'*****************************************');
fprintf(fid,'*************************************\n');
fprintf(fid,' |Sub_number|    |X|         |Y|    |Optimal');
fprintf(fid,' capacity| \n');
for i = 1:Ns
    if Soc(i,4)~=0
        fprintf(fid,'%8.f', Soc(i,1));
        fprintf(fid,'    %8.f', Soc(i,2));
        fprintf(fid,'    %8.f', Soc(i,3));
        fprintf(fid,'     %8.1f',Soc(i,4));
        fprintf(fid,'\n');
    end
end
fprintf(fid,'*****************************************');
fprintf(fid,'*************\n');
for i = 1:Ns
    if Soc(i,4)~=0
        Cln = find (XX(i,:)~=0);
        fprintf(fid,'*********************************');
        fprintf(fid,'*********************\n');
        fprintf(fid,' Connected load nodes to the ');
        fprintf(fid,'substation %3.0f  are:  \n', Soc(i,1));
        fprintf(fid,'|Load_node|     |X|       |Y|    ');
        fprintf(fid,'|Magnitude(MVA)|\n');
        for i = 1:length(Cln)
            fprintf(fid,'\n %6.0f % 10.0f % 8.0f %10.1f',...
                Cln(i), Lx(Cln(i)), Ly(Cln(i)), Sl(Cln(i)));
        end
        fprintf(fid,'\n*******************************');
```

```
        fprintf(fid,'**********************\n');
    end
end
%%
```

d) "Plot_SEPP" M-file code

```
hold off
format short
xx = XX;
[jjj,iii] = find(xx==1);
for I = 1:Ns
    [II] = find(jjj==I);
    for J = 1:length(II)
        S_LC(J,I) = iii(II(J));
    end
    clear II
end
z = sum(xx,2);
iz = find(z|0);
izn = find(z==0);
niz = length(iz);
nizn = length(izn);
for bb = 1:niz
subposx(1,bb) = Sx((iz(bb)),1);
subposy(1,bb) = Sy((iz(bb)),1);
end
for bb1 = 1:nizn
  nsubposx(1,bb1) = Sx((izn(bb1)),1);
  nsubposy(1,bb1) = Sy((izn(bb1)),1);
end
for ba = 1:Nl
    loadposx(1,ba) = Lx(ba,1);
    loadposy(1,ba) = Ly(ba,1);
end
Aa = cell(niz,2);
for ia = 1:niz
    Aax = xx(iz(ia),:);
    [iax] = find(Aax==1);
    niax = length(iax);
    for ja = 1:niax
        jab = (2*ja)-1;
        Aaa(1,jab) = Sx(iz(ia));
        Bbb(1,jab) = Sy(iz(ia));
        jaa = (2*ja);
        Aaa(1,jaa) = Lx(iax(ja));
        Bbb(1,jaa) = Ly(iax(ja));
        Aa{ia,1} = Aaa;
        Aa{ia,2} = Bbb;
    end
     clear Aaa Bbb
end
%% Plotting the location of installed & current substations
figure(1)
```

```
subplot(2,2,1)
plot(subposx,subposy,'sb')
xlabel('X Axis')
ylabel('Y Axis')
axis([0 100 0 100])
title('Location of selected candidate substations')
grid on
%% Plotting the location of uninstalled candidate substations
subplot(2,2,2)
plot(nsubposx,nsubposy,'sr')
xlabel('X Axis')
ylabel('Y Axis')
axis([0 100 0 100])
title('Location of unselected candidate substations')
grid on
%% Plotting the location of load nodes
subplot(2,2,3)
plot (loadposx,loadposy,'ok')
xlabel('X Axis')
ylabel('Y Axis')
axis([0 100 0 100])
title('Location of load nodes')
grid on
%% Plotting the location of selected substations, load nodes
    %% & downward lines
subplot(2,2,4)
plot(nsubposx,nsubposy,'sr')
xlabel('X Axis')
ylabel('Y Axis')
axis([0 100 0 100])
title('Selected substations, load nodes & downward lines')
hold on
plot(subposx,subposy,'sb')
hold on
plot(nsubposx,nsubposy,'sr')
hold on
plot (loadposx,loadposy,'ok')
grid on
%% Plotting the Position of selected substations, load nodes
    %% & downward lines
hold on
for iia = 1:niz
    Aab = Aa{iia,1};
    Bba = Aa{iia,2};
    plot(Aab,Bba,'m')
    hold on
end
figure (2)
plot(nsubposx,nsubposy,'sr')
xlabel('X Axis')
ylabel('Y Axis')
axis([0 100 0 100])
title('Selected substations, load nodes & downward lines')
hold on
```

```
plot(subposx,subposy,'sb')
hold on
plot(nsubposx,nsubposy,'sr')
hold on
plot (loadposx,loadposy,'ok')
hold on
for iia = 1:niz
    Aab = Aa{iia,1};
    Bba = Aa{iia,2};
    plot(Aab,Bba,'m')
    hold on
end
grid on
```

L.4 NEP.m

a) "Hybridsearch" M-file code

```
clear
clc
%% Reading the input data %%
%% Reading data of the network buses:
Busdata = xlsread('Nepdata.xls', 'Busdata');
%% Reading data of the network lines:
Linedata = xlsread('Nepdata.xls', 'Linedata');
%% Reading data of the candidate lines:
Candid = xlsread('Nepdata.xls', 'CandidateLinedata');
%% Reading the information of defined line types:
Linetype = xlsread('Nepdata.xls', 'LineType');
inputs = xlsread('Nepdata.xls', 'Otherinputs');
%% Lg: load growth rate (Lg=1 means 100% load growth):
Lg = inputs(1,1);
%% Mof: minimum fitness, which is kept at high value for
    % the first iteration of the forward search algorithm
Mof = inputs(1,2);
Solution = ones (size(Candid,1),1);
%% Calling the hybrid search algorithm to solve the NEP prob-
lem
[Os, Adline, Noll, Coll, Angle, Mof]=...
    HS(Busdata, Linedata, Candid, Linetype, Solution, Lg,
Mof);
%% Os: optimal solution of the NEP problem
%% Adline: final set of selected candidate lines
    % among all candidates
%% Noll: overload of the existing and selected candidate
    % lines in normal condition after adding optimal candidate
    % line in each iteration (or in order of priority)
%% Coll: overload of the existing and selected candidate lines
    % in N-1 condition after adding optimal candidate line
    % in each iteration
%% Angle: voltage phase of all buses for adding the best
```

```
    % candidate line to the network
%%  Printing and saving the obtained results in result.txt
    % in the corresponding directory
Print_Nep
```

b) "Forwardsearch" M-file code

```
clear
clc
%% Reading the Input Data %%
%% Reading data of the network buses
Busdata = xlsread('Nepdata.xls', 'Busdata');
%% Reading data of the network lines
Linedata = xlsread('Nepdata.xls', 'Linedata');
%% Reading data of the candidate lines
Candid = xlsread('Nepdata.xls', 'CandidateLinedata');
%% Reading the information of defined line types
Linetype = xlsread('NEPdata.xls', 'LineType');
inputs = xlsread('Nepdata.xls', 'Otherinputs');
%% Lg: load growth rate Lg=1 means 100% load growth
Lg = inputs(1,1);
%% Mof: minimum fitness, which is kept at high value for
    % the first iteration of the forward search algorithm
Mof = inputs(1,2);
%% Contingency=1 means the problem is solved,
    % considering N-1 condition
Contingency = inputs(1,3);
%% Forward search starts with base network (no candidate line
    % is added to the base network at the beginning)
Solution=zeros (size(Candid,1),1);
%% Calling the forward search algorithm
    % to solve the NEP problem
[Os, Adline, Noll, Coll, Angle] = FS(Busdata, Linedata, ...
    Candid, Linetype, Solution, Contingency, Lg, Mof);
%% Os: optimal solution of the NEP problem
%% Adline: final set of selected candidate lines
    % among all candidates
%% Noll: overload of the existing and selected candidate lines
    % in normal condition after adding optimal candidate line
    % in each iteration (or in order of priority)
%% Coll: overload of the existing and selected candidate lines
    % in N-1 condition after adding optimal candidate line
    % in each iteration
%% Angle: voltage phase of all buses for adding the best
    % candidate line to the network

%%  Printing and saving the obtained results in result.txt in
    % the corresponding directory and in the command window
Print_NEP
```

c) "Backwardsearch" M-file code

```
clear
clc
%% Reading the input data %%
```

```
%% Reading data of the network buses
Busdata = xlsread('Nepdata.xls', 'Busdata');
%% Reading data of the network lines
Linedata = xlsread('Nepdata.xls', 'Linedata');
%% Reading data of the candidate lines
Candid = xlsread('Nepdata.xls', 'CandidateLinedata');
%% Reading the information of defined line types
Linetype = xlsread('NEPdata.xls', 'LineType');
inputs=xlsread('Nepdata.xls', 'Otherinputs');
%% Lg: load growth rate, Lg=1 means 100% load growth
Lg = inputs(1,1);
%% Mof: minimum fitness, which is kept at high value for
    % the first iteration of the forward search algorithm
Mof = inputs(1,2);
%% Contingency=1 means the problem is solved, considering
    % N-1 condition.
Contingency = inputs(1,3);
%% Backward search starts with considering all candidate
    % lines added to the base network at the beginning)
Solution = ones (size(Candid,1),1);
%% Calling the backward search algorithm to
    % solve the NEP problem
[Os, Adline, Noll, Coll, Angle, Mof] = BS(Busdata,...
    Linedata, Candid, Linetype, Solution, ...
    Contingency, Lg, Mof);
%% Os: optimal solution of the NEP problem
%% Adline: final set of selected candidate lines
    % among all candidates
%% Noll: overload of the existing and selected candidate lines
    % in normal condition after adding optimal candidate line
    % in each iteration (or in order of priority)
%% Coll: overload of the existing and selected candidate lines
    % in N-1 condition after adding optimal candidate line
    % in each iteration
%% Angle: voltage phase of all buses for adding the best
    % candidate line to the network

%% Printing and saving the obtained results in result.txt
    % in the corresponding directory
Print_NEP
```

d) "HS" M-file code

```
function [Os, Adline, Noll, Coll, Angle,Mof] = ...
    HS(Busdata, Linedata, Candid, Linetype, Solution, Lg, Mof)
if nargin<7 | isempty(Mof), Mof = 10^20;          end
if nargin<6 | isempty(Lg),  Lg = 0; Mof = 10^20; end
if nargin<5 | isempty(Solution)
    Solution = ones(size(Candid,1),1); Lg = 0; Mof = 10^20;
end
if nargin<4 | isempty(Linetype)
    fprintf('Input argument "Linetype" containing the');
    fprintf(' information of different types of lines.');
    error('"Linetype" is undefined.');
```

```
end
if nargin<3 | isempty(Candid)
    fprintf('Input argument "Candid" containing');
    fprintf(' the information of candidate lines.');
    error('"Candid" is undefined.');
end
if nargin<2 | isempty(Linedata)
    fprintf('Input argument "Linedata" containing');
    fprintf(' the information of existing lines.');
    error('"Linedata" is undefined.');
end
%% Problem outputs:

%% Os: Optimal solution of the NEP problem
%% Adline: final set of selected candidate lines among all
    % candidates.
%% Noll: overload of the existing and selected candidate lines
    % in normal condition after adding optimal candidate line
    % in each iteration (or in order of priority)
%% Coll: overload of the existing and selected candidate lines
    % in N-1 condition after adding optimal candidate line
    % in each iteration
%% Angle: voltage phase of all buses for adding the best
    % candidate line to the network

%% Problem inputs:

%% Busdata: data of the network buses
%% Linedata: data of the network lines
%% Candid: data of candidate lines
%% Linetype: data of different line types
%% Solution: the initial solution, which is a zero vector
    % for hybrid search algorithm
%% Contingency: if contingency=1, the problem is solved by
    % considering N-1 condition.
%% Lg: load growth rate
%% Mof: minimum fitness, which is kept at high value for
    % the first iteration of the forward search algorithm

contingency = 0;
[OSB, added_lineB, NOLLB, COLLB, AngleB, MOFB] = ...
    BS(Busdata, Linedata, Candid, Linetype, Solution,...
        contingency, Lg, Mof);
contingency = 1;
[Os, Adline, Noll, Coll, Angle, Mof] = FS(Busdata, ...
    Linedata, Candid, Linetype, OSB, contingency, Lg, Mof);
if sum(Os-OSB) == 0
    Angle = AngleB;
    Noll = NOLLB;
    Coll = COLLB;
    Mof = MOFB;
    Adline = added_lineB
    Os = OSB;
End
```

e) "BS" M-file code

```
function[Os, Adline, Noll, Coll, Angle, Mof] = BS ...
    (Busdata, Linedata, Candid, Linetype, Solution, ...
    Contingency, Lg, Mof);
if nargin<8 | isempty(Mof), Mof = 10^20;         end
if nargin<7 | isempty(Lg),  Lg = 0; Mof = 10^20; end
if nargin<6 | isempty(Contingency)
    Contingency = 0; Lg = 0; Mof = 10^20;
end
if nargin<5 | isempty(Solution)
    Solution = ones (size(Candid,1),1);
    Contingency = 0; Lg = 0; Mof = 10^20;
end
if nargin<4 | isempty(Linetype)
    fprintf('Input argument "Linetype" containing the');
    fprintf(' information of different types of lines.');
    error('"Linetype" is undefined.');
end
if nargin<3 | isempty(Candid)
    fprintf('Input argument "Candid" containing the');
    fprintf(' information of candidate lines.');
    error('"Candid" is undefined.');
end
if nargin<2 | isempty(Linedata)
    fprintf('Input argument "Linedata" containing');
    fprintf(' the information of existing lines.');
    error('"Linedata" is undefined.');
end

%% Problem outputs:

%% Os: optimal solution of the NEP problem
%% Adline: final set of selected candidate lines among
    % all candidates.
%% Noll: overload of the existing and selected candidate
    % lines in normal condition after adding optimal candidate
    % line in each iteration (or in order of priority)
%% Coll: overload of the existing and selected candidate lines
    % in N-1 condition after adding optimal candidate line
    % in each iteration
%% Angle: voltage phase of all buses for adding the best
    % candidate line to the network

%% Problem inputs:

%% Busdata: data of the network buses
%% Linedata: data of the network lines
%% Candid: data of candidate lines
%% Linetype: data of different line types
%% Solution: the initial solution, which is a zero vector
    % for hybrid search algorithm
%% Contingency: if contingency = 1, the problem is solved
    % by considering N-1 condition
```

```
%% Lg: load growth rate
%% Mof: minimum fitness, which is kept at high value for
    % the first iteration of the forward search algorithm

nc = size (find(Solution ~= 0),1);

%% Backward search algorithm %%
%% Initialization
diff = 1; SID = 0; j = 1;
ii = 0; jj = 0; kk = 0;
Noll = null(1); Coll = null(1);

while diff>0 | j<=2^nc
    Solution1 = Solution;
    [isol] = find(Solution1 ~= 0);
    best_sol = null(1);

%% Adding all candidate lines to the present set of lines and
    % finding the best possible candidate to be eliminated
    % from the set of present and added candidate lines.
    % This step is iterated untill the the obtained fitness
    % function doesn't decreas.
    for i = 1:length (isol)
        Isol = isol(i);
        Solution1 (Isol) = 0;
%% Updating corresponding line data and bus data according
    % to the eliminated candidate line; constructing ybus;
    % computing number of islands
    % after each candidate is eliminated from the network.
        [Ybus, linedata, busdata, nIs, nbus, bus_number] ...
            = ybus_calculation(Busdata, Linedata, ...
            Solution1, Candid, Linetype, Lg);
%% busdata:Updated bus data after considering new candidates
%% linedata:Updated line data after considering new andidates
%% Running DC Power flow for updated line data and bus data
    % to obtain total overload in the normal condition
       [angle_r, angle_d, PF, OL, SOL] = ...
           dcpf(busdata, linedata, Ybus);
%% NOL{i,1}: total overload in case of eliminating the i-th
    % candidate line among the added candidates
        NOL{i,1} = OL;
        angle{i,1} = angle_r;
%% Computing the total cost (TC) after eliminating
    % each candidate line
        Isoln = find(Solution1~=0);
%% TC: Total Cost
        [TC] = Total_Cost(Isoln, Solution1, Candid, Linetype);
%% Computing total overload in N-1 condition after eliminating
    % each candidate line

%% If N-1 condition is considered in the algorithm and there
    % is no island in normal condition
        if Contingency == 1 & nIs == 0
            [COL, CnIs, OLF] = contingency(linedata, busdata);
```

```
%% OOLF{i,1}: total overload in N-1 condition, in case of
    % eliminating the i-th candidate line among not
    % selected candidates
            OOLF{i,1} = OLF;
        else
            COL = 0; CnIs = 0;
        end
        nline = size (linedata,1);
%% Formation of fitness function (OF: NEP Objective Function)
        OF = TC+(10^9*((SOL)+COL))+(10^12*((nIs)+(CnIs)));

        if OF < Mof
            diff = (Mof-OF);
            Mof = OF;
            best_sol = Isol;
            j = j+1;
        else
            j = j+1;
        end

%%  Eliminating the worst candidate line from the set of
    % candidate lines; retrieval the power flow and
    % overloaddata corresponding with the selected candidate
    % of each iteration
        Solution1(Isol) = Candid(Isol,6);
    end
    best_sol_index = isempty(best_sol);
    if best_sol_index == 1;
        break
    else
        Solution(best_sol) = 0;
        ii = ii+1;
        best(ii,1) = best_sol;
        best(ii,2) = Mof;
        if Contingency == 1
            jj = jj+1;
            bsol = find (isol == best_sol);
            Coll{jj,1} = OOLF{bsol,1};
            kk = kk+1;
            Noll{kk,1} = NOL{bsol,1};
            Angle{kk,1} = angle{bsol,1};
            clear angle NOL
        else
            kk = kk+1;
            bsol = find(isol == best_sol);
            Noll{kk,1} = NOL{bsol,1};
            Angle{kk,1} = angle{bsol,1};
            clear angle NOL
        end
    end
end
%% Adline: final set of selected candidate lines
    % among all candidates
Os = Solution; % Optimal solution
```

```
al = find(Os~=0);
if length(al)~=0;
    lb = length(best);
    for i = 1:length(al)
        Adline(i,1) = Candid(al(i),2);
        Adline(i,2) = Candid(al(i),3);
    end
    for i = 1:lb
        removed_line(i,1) = Candid(best(i),2);
        removed_line(i,2) = Candid(best(i),3);
        removed_line(i,3) = (best(i,2)/10^7);
    end
else
    Adline = null(1);
end
```

f) "FS" M-file code

```
function[Os, Adline, Noll, Coll, Angle, Mof] = FS...
    (Busdata, Linedata, Candid, Linetype, Solution, ...
    Contingency, Lg, Mof)
if nargin<8 | isempty(Mof), Mof = 10^9;          end
if nargin<7 | isempty(Lg),  Lg = 0; Mof = 10^9; end
if nargin<6 | isempty(Contingency)
    Contingency = 0; Lg = 0; Mof = 10^9;
end
if nargin<5 | isempty(Solution)
    Solution = zeros(size(Candid,1),1);
    Contingency = 0; Lg = 0; Mof = 10^9;
end
if nargin<4 | isempty(Linetype)
    fprintf('Input argument "Linetype" containing the');
    fprintf(' information of different types of lines.');
    error('"Linetype" is undefined.');
end
if nargin<3 | isempty(Candid)
    fprintf('Input argument "Candid" containing the');
    fprintf(' information of candidate lines.');
    error('"Candid" is undefined.');
end
if nargin<2 | isempty(Linedata)
    fprintf('Input argument "Linedata" containing the');
    fprintf(' information of existing lines.');
    error('"Linedata" is undefined.');
end

%% Problem outputs:

%% Os: optimal solution of the NEP problem
%% Adline: final set of selected candidate lines
    % among all candidates
%% Noll:overload of the existing and selected candidate lines
    % in normal condition after adding optimal candidate line
    % in each iteration (or in order of priority)
```

```
%% Coll:overload of the existing and selected candidate lines
    % in N-1 condition after adding optimal candidate line
    % in each iteration
%% Angle: voltage phase of all buses for adding the best
    % candidate line to the network

%% Problem inputs:

%% Busdata: data of the network buses
%% Linedata: data of the network lines
%% Candid: data of candidate lines
%% Linetype: data of different line types
%% Solution: the initial solution, which is a zero vector for
    % hybrid search algorithm
%% Contingency: if contingency=1, the problem is solved by
    % considering N-1 condition
%% Lg: load growth rate
%% Mof: minimum fitness, which is kept at high value for
    % the first iteration of the forward search algorithm

ncr = length (find(Solution == 0));
%% Forward search algorithm %%
%% Initialization
diff = 1; j = 1; ii = 0; jj = 0;
kk = 0; Noll = null(1); Coll = null(1);
best = null(1); Angle = null(1);
%%
while diff>0 |j<=2^ncr
    Solution1 = Solution;
    %% Finding not selected candidate lines
    [isol] = find(Solution1 == 0);
    best_sol = null(1);

%% Adding each candidate among non-selected candidates to
    % the previously selected lines and finding the best
    % possible candidate for joining to the set of current
    % and previously selected lines.
    % This step is iterated untill the the obtained fitness
    % function doesn't decreas.
    for i = 1:length (isol)
        Isol = isol(i); % Selecting a candidate
        Solution1(Isol) = Candid(Isol,6);
%% Updating corresponding line data and bus data according
    % to the added candidate line; constructing ybus;
    % computing number of islands after each candidate is
    % added to the network

        [Ybus, linedata, busdata, nIs, nbus, bus_number]...
            = ybus_calculation (Busdata, Linedata, ...
            Solution1, Candid, Linetype, Lg);
%% busdata:updated bus data after considering new candidates
%% linedata:updated line data after considering new andidates
%% Running DC Power flow for updated line data and bus data
    % to obtain total overload in the normal condition
```

```
        [angle_r,angle_d, PF, OL, SOL] = ...
            dcpf(busdata, linedata, Ybus);
        NOL{i,1} = OL;
        angle{i,1} = angle_r;
%% Computing Total Cost (TC) for adding each candidate line
        Isoln = find(Solution1~=0);
%% TC: Total Cost
        [TC] = Total_Cost(Isoln,Solution1,Candid,Linetype);
%% Computing total overload in N-1 condition for
    % adding each candidate line
%% If N-1 condition is considered in the algorithm and there
    % is no island in normal condition
if Contingency == 1 & nIs == 0
            [COL,CnIs,OLF] = contingency(linedata,busdata);
%% OOLF{i,1}: total overload in N-1 condition, in case of
    % adding the i-th candidate line among not
    % selected candidates
            OOLF{i,1} = OLF;
        else
            COL = 0; CnIs = 0;
        end
%% Formation of fitness function
%% OF: NEP Objective Function
        OF = TC+(10^9*((SOL)+COL))+(10^12*((nIs)+(CnIs)));
%% Finding the best candidate (which has the least fitness)
    % by comparing OF (fitness of the i-th candidate) with
    % Mof (minimum fitness)
        if OF < Mof
            diff = (Mof-OF);
            Mof = OF; %%
            best_sol = Isol;
            j = j+1;
        else
            j = j+1;
        end
        Solution1(Isol) = 0;
    end
%% Adding the best candidate line to the set of present and
    % previously selected lines by eliminating the selected
    % candidate from the set of candidate lines retrieval
    % the power flow and overload data corresponding with
    % the selected candidate of each iteration
    best_sol_index = isempty(best_sol);
    if best_sol_index == 1;
        break
    else
        Solution(best_sol) = Candid(best_sol,6);
        ii = ii+1;
        best(ii,1) = best_sol;
        best(ii,2) = Mof;
        if Contingency == 1
           jj = jj+1;
           bsol = find (isol == best_sol);
           Coll{jj,1} = OOLF{bsol,1};
```

```
            kk = kk+1;
            Noll{kk,1} = NOL{bsol,1};
            Angle{kk,1} = angle{bsol,1};
            clear angle NOL
         else
            kk = kk+1;
            bsol = find (isol == best_sol);
            Noll{kk,1} = NOL{bsol,1};
            Angle{kk,1} = angle{bsol,1};
            clear angle NOL
         end
     end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% added_line: final set of selected candidate lines
    % among all candidates.
Os = Solution;  % Optimal solution
al = find(Os~=0);
if length(al)~=0;
    lb = length(best);
    for i = 1:length(al)
        Adline(i,1) = Candid(al(i),2);
        Adline(i,2) = Candid(al(i),3);
    end
else
    Adline = null(1);
end
```

g) "print_NEP" M-file code

```
fid = fopen('results.txt', 'wt');
fprintf(fid, '---------------------------------------');
fprintf(fid,'\n Added candidate lines are as follows:\n');
fprintf(fid, '--------------------------------------\n');
fprintf(fid, '    From bus     To bus\n');
fprintf(fid, '    --------        ------');
fprintf(fid, '\n %10.0f %15.0f', Adline');
fprintf(fid, '\n\n****************************************');
fprintf(fid, '******************************');
fprintf(fid,'\n                              Normal');
fprintf(fid,' condition\n');
fprintf(fid, '*********************************************');
fprintf(fid, '************************\n');
if (isempty(Noll) == 1)
    fprintf(fid, 'No overloaded in normal condition\n');
else
    NNOLL = Noll{size (Noll,1),1};
    NL = size (Linedata,1);
    NS = length (find (Os ~= 0));
    fprintf(fid, '\Overload at normal');
    fprintf(fid,'Total overload of normal condition is');
    fprintf(fid,'3.5f pu\n',sum(NNOLL(:,4)));
    fprintf(fid, '****************************************');
    fprintf(fid, '******************************');
```

```
    fprintf(fid, '\n*********************** Candidate');
    fprintf(fid, 'branches *************************\n');
    fprintf(fid, '****************************************');
    fprintf(fid, '******************************\n');
    fprintf(fid, '\n  From bus      To bus       Line flow');
    fprintf(fid, ' (pu)      Overload (pu)\n');
    for i = 1:NS
        fprintf(fid, '\n %10.0f %15.0f %20.5f %20.5f\n',...
            NNOLL(i+NL,:));
    end
    fprintf(fid, '***************************************');
    fprintf(fid, '********************************');
    fprintf(fid, '\n************************ Existing');
    fprintf(fid, ' branches **************************\n');
    fprintf(fid, '***************************************');
    fprintf(fid, '********************************\n');
    fprintf(fid, '\n  From bus      To bus       Line flow');
    fprintf(fid, ' (pu)      Overload (pu)\n');
    for i=1:NL
        fprintf(fid, '\n %10.0f %15.0f %20.5f %20.5f\n',...
            NNOLL(i,:));
    end
            fprintf(fid, '\n******************************');
            fprintf(fid, '********************************');
            fprintf(fid, '*********\n');
end
nco = size (Coll,1);
oc = Coll{nco,1};
nc = size (oc,1);
ocl = 0;
for i = 1:nc
    occ = oc{i,1};
    ocl = ocl+occ(1,3);
end
if (ocl == 0)
    fprintf(fid, '\n No overload in N-1 condition\n');
    fprintf(fid, '\n************************************\n');
else
            fprintf(fid, '\n******************************');
            fprintf(fid, '********************************');
            fprintf(fid, '*********\n');
    LCOLL = Coll{size (Coll,1),1};
    fprintf(fid, '               Overloaded lines in');
    fprintf(fid, ' N-1 condition');
    for i = 1: size (LCOLL,1)
        iLCOLL = LCOLL{i,1};
        iL = iLCOLL(1,:);
        if iL(1,3) ~= 0
            fprintf(fid, '\n******************************');
            fprintf(fid, '********************************');
            fprintf(fid, '*********\n');
            fprintf(fid,'Total overload for outage of line');
            fprintf(fid, ': from bus');
            fprintf(fid, '%3.0f to bus %3.0f is %6.5f\n',iL);
```

```
                fprintf(fid, '**********************************');
                fprintf(fid, '**********************************');
                fprintf(fid, '*******\n');
                fprintf(fid,' Following lines are overloaded');
                fprintf(fid,' in this outage\n');
                fprintf(fid, '*******************************');
                fprintf(fid, '*******************************');
                fprintf(fid, '***********\n');
                fprintf(fid, '  From bus  To bus');
                fprintf(fid, '        Overload (pu)\n');
                fprintf(fid, '  *******  ******        *****');
                fprintf(fid, '******');
                for j = 2:size (iLCOLL,1);
                    fprintf(fid, '\n %6.0f %7.0f %18.5f\n',...
                        iLCOLL(j,:));
                end
            end
        end
    end
    LAngle = Angle{size (Angle,1),1};
    fprintf(fid,'\n****************************************\n');
    fprintf(fid,'\n***************Bus data*****************\n');
    fprintf(fid, '     No. bus          Voltage angle (Rad)\n');
    fprintf(fid, '***************       *******************\n');
    for i = 1:size (Busdata,1);
        fprintf(fid, '\n %10.0f %27.5f \n', i, LAngle(i,:));
    end
    fclose(fid);
    fid = fopen('results1.txt', 'wt');
    fprintf(fid, '---------------------------------------');
    fprintf(fid, '\n Added candidate lines are as follows:\n');
    fprintf(fid, '---------------------------------------\n');
    fprintf(fid, '    From bus     To bus\n');
    fprintf(fid, '    --------     ------');
    fprintf(fid, '\n %8.0f %11.0f', Adline');
    fprintf(fid, '\n\n*****************************************');
    fprintf(fid, '******************************');
    fprintf(fid, '\n                              Normal');
    fprintf(fid, ' condition\n');
    fprintf(fid, '*********************************************');
    fprintf(fid, '*************************\n');
    if (isempty(Noll) == 1)
        fprintf(fid, 'No overloaded in normal condition\n');
    else
        NNOLL = Noll{size (Noll,1),1};
        NL = size (Linedata,1);
        NS = length (find (Os~=0));
        fprintf(fid, '\Overload at normal');
        fprintf(fid, 'Total overload of normal condition is');
        fprintf(fid, '%3.5f pu\n',sum(NNOLL(:,4)));
        fprintf(fid, '*****************************************');
        fprintf(fid, '*******************************');
        fprintf(fid, '\n************************ Candidate');
        fprintf(fid, ' branches *************************\n');
```

```
    fprintf(fid, '*************************************');
    fprintf(fid, '*********************************\n');
    fprintf(fid, '\n  From bus      To bus       Line flow');
    fprintf(fid, ' (pu)      Overload (pu)\n');
    for i=1:NS
        fprintf(fid, '\n %6.0f %10.0f %20.5f %20.5f\n', ...
            NNOLL(i+NL,:));
    end
    fprintf(fid, '****************************************');
    fprintf(fid, '*******************************');
    fprintf(fid, '\n************************ Existing');
    fprintf(fid, ' branches **************************\n');
    fprintf(fid, '****************************************');
    fprintf(fid, '*******************************\n');
    fprintf(fid, '\n  From bus      To bus       Line flow');
    fprintf(fid, ' (pu)      Overload (pu)\n');
    for i=1:NL
        fprintf(fid, '\n %6.0f %10.0f %20.5f %20.5f\n',...
            NNOLL(i,:));
    end
        fprintf(fid, '\n********************************');
        fprintf(fid, '**********************************');
        fprintf(fid, '*****\n');
end
nco=size (Coll,1);
oc=Coll{nco,1};
nc=size (oc,1);
ocl=0;
for i=1:nc
    occ=oc{i,1};
    ocl=ocl+occ(1,3);
end
if (ocl==0)
    fprintf(fid, '\n No overload in N-1 condition\n');
    fprintf(fid, '\n************************************\n');
else
    fprintf(fid, '\n**************************************');
    fprintf(fid, '*********************************\n');
    LCOLL=Coll{size (Coll,1),1};
    fprintf(fid, '               Overloaded lines in N-1');
    fprintf(fid, ' condition');
    for i=1: size (LCOLL,1)
        iLCOLL=LCOLL{i,1};
        iL=iLCOLL(1,:);
        if iL(1,3)~=0
            fprintf(fid, '\n*******************************');
            fprintf(fid, '*********************************');
            fprintf(fid, '*******\n');
            fprintf(fid,' Total overload for outage of line');
            fprintf(fid, ': from bus');
            fprintf(fid, ' %3.0f to bus %3.0f is%6.5f\n',iL);
            fprintf(fid, '*********************************');
            fprintf(fid, '*********************************');
            fprintf(fid, '*****\n');
```

```
                fprintf(fid,' Following lines are overloaded in');
                fprintf(fid, ' this outage\n');
                fprintf(fid, '*********************************');
                fprintf(fid, '*********************************');
                fprintf(fid, '*****\n');
                fprintf(fid, '  From bus  To bus');
                fprintf(fid, '        Overload (pu)\n');
                fprintf(fid, '  *******  ******        *******');
                fprintf(fid, '****');
                for j=2:size (iLCOLL,1);
                    fprintf(fid, '\n %6.0f %7.0f %18.5f\n',...
                        iLCOLL(j,:));
                end
            end
        end
    end
    LAngle=Angle{size (Angle,1),1};
    fprintf(fid,'\n****************************************\n');
    fprintf(fid,'\n**************Bus Data*****************\n');
    fprintf(fid, '     No. Bus          Voltage Angle (Rad)\n');
    fprintf(fid, '***************       ******************\n');
    for i=1:size (Busdata,1);
        fprintf(fid, '\n %10.0f %27.5f \n', i, LAngle(i,:));
    end
    fclose(fid);
    clc
    type results1.txt
    delete results1.txt
```

h) "ybus_calculation" M-file code

```
function [Ybus, linedata, busdata, nIs, nbus, bus_number]...
     = ybus_calculation(Busdata, Linedata, Solution, ...
     CandidateLinedata, LineType, Lg);
if isempty(Lg), Lg = 0; end
if isempty(Linedata)
    fprintf('Input argument "Linedata" containing the');
    fprintf(' information of network lines.');
    error('"Linedata" is undefined.');
end
if isempty(Busdata)
    fprintf('Input argument "Busdata" containing the');
    fprintf(' information of network buses.');
    error('"Busdata" is undefined.');
end
%if nargin<3 | isempty(Solution), linedata = Linedata; end ??

%% Problem outputs:

%% Ybus: admittance matrix
%% Bdata: data of network buses after considering load growth
%% Ldata: data of network lines after adding candidate lines
%% Nis: number of islands in the base network
%% Nbus: number of buses
```

```
%% Problem inputs:

%% Busdata: data of the network buses
%% Linedata: data of the network lines
%% Candid: candidate lines
%% Linetype: data of different line types
%% Lg: load growth rate

Bd = Busdata;
Ld = Linedata;
Sol = Solution;
Cl = CandidateLinedata;
Lt = LineType;
%% Finding suggested solutions %%
Iz = find (Solution~=0);
nIz = length(Iz);
nline = size (Linedata,1);

for i = 1:nIz
    can(1,1) = size (Linedata,1)+i; can(1,2) = Cl(Iz(i),2);
    can(1,3) = Cl(Iz(i),3);
    %candid(1,4)=(Lt((Cl(Iz(i),4)),2)*Cl(Iz(i),5))/...
    %    (Cl(Iz(i),6));
    can(1,4) = 0;
    can(1,5) = (Lt((Cl(Iz(i),4)),3)*Cl(Iz(i),5))/...
        (Cl(Iz(i),6));
    can(1,6) = (Lt((Cl(Iz(i),4)),4))*(Cl(Iz(i),6));
    can(1,7) = Cl(Iz(i),5);
    Ld(nline+i,:) = can(1,:);
end
linedata = Ld;
exl = size (Linedata,1);
%% Islanding detedtion and uodating busdata
busnumber = Bd(:,1);
nl = Ld(:,2);
nr = Ld(:,3);
nlr = union(nl,nr);
%Is = setdiff(nlr,busnumber);
Is = setxor(nlr,busnumber);
bus_number = setxor(busnumber,Is);
nbus = length(bus_number);
nIs = length (Is);
for i = 1:nbus
    busdata (i,:) = Bd(bus_number(i),:);
end
busdata(:,4) = busdata(:,4).*(1+Lg); busdata(:,5) = ...
    busdata(:,5).*(1+Lg);

j = sqrt(-1);
i = sqrt(-1);
X = Ld(:,5);
nbr = length(Ld(:,1));
Z = (j*X);
```

```
y = ones(nbr,1)./Z;          % Branch admittance
Ybus = zeros(nbus,nbus);    % Initialize Ybus to zero
%% Formation of the off diagonal elements
for k = 1:nbr;
       Ybus(nl(k),nr(k)) = Ybus(nl(k),nr(k))-y(k);
       Ybus(nr(k),nl(k)) = Ybus(nl(k),nr(k));
end
%% Formation of the diagonal elements
for n = 1:nbus
    for m = (n+1):nbus
        Ybus(n,n) = Ybus(n,n)-Ybus(n,m);
    end
    for m = 1:n-1
        Ybus(n,n) = Ybus(n,n)-Ybus(n,m);
    end
end
```

i) "dcpf" M-file code

```
function [angle_r,angle_d, PF, OL, SOL] = ...
    dcpf(busdata, linedata, Ybus)
if nargin<3 | isempty(Ybus)
    error('Input argument "Ybus" is undefined');
end
if nargin<2 | isempty(linedata)
    fprintf('Input argument "Linedata" containing the');
    fprintf(' information of lines.');
    error('"Linedata" is undefined.');
end
if isempty(busdata)
    fprintf('Input argument "busdata" containing the');
    fprintf(' information of buses.');
    error('"busdata" is undefined.');
end

%% Problem outputs:

%% angle_r: voltage angle based on radian
%% angle_d: voltage angle based on degree
%% PF: power flow data of lines
%% OL: overload information of lines
%% SOL: total overload of the network

%% Problem inputs:

%% busdata: required data of network buses
%% busdata: required data of network lines
%% Ybus: computed ybus of the netowrk

nbus = size (busdata,1);
nl = linedata(:,2);
nr = linedata(:,3);
Smax = linedata(:,6);
nbr = length(nl);
```

```
%% Computing net power of buses
Ps1 = (busdata(:,3)-busdata(:,4));
%% Finding non-slack buses in the busdata matrix
code = busdata(:,2);
[aa] = find(code~=3);
%% Forming Network suceptance matrix (B)
for n = 1:length(aa)
    for m = 1:length(aa)
            Ymn = Ybus(aa(n),aa(m));
            B(n,m) = -imag(Ymn);
    end
    Ps(n,1) = Ps1((aa(n)),1);
end
%% Computing voltage angle values of all buses
Binv = inv(B);
ang1 = Binv*Ps;
%% angle_r: volatge angle based on radian
angle_r = zeros(nbus,1);
for i=1: length(aa)
    aaa = aa(i);
    angle_r(aaa) = ang1(i);
end
%% angle_d: voltage angle based on degree
angle_d = angle_r*(180/pi);
%% Computing Power flow and overload of all lines
jay = sqrt(-1);
for i = 1:nbr
     PF(i,1) = nl(i); OL(i,1) = nl(i);
     PF(i,2) = nr(i); OL(i,2) = nr(i);
     PF(i,3) = (angle_r(nl(i))-angle_r(nr(i)))/...
         (linedata(i,5));
    if abs(PF(i,3))>Smax(i)
        OL(i,3) = abs(PF(i,3));
        OL(i,4) = abs(PF(i,3))-Smax(i);
    else
        OL(i,3) = PF(i,3);
        OL(i,4) = 0;
    end
end
%% Computing total overload of the network
SOL = sum(OL(:,4));
```

j) "Total_Cost" M-file code

```
function [TC]=Total_Cost(Isolnew, Solution, candid, LineType)
In=Isolnew;
TC=0;
for i=1:length (In)
    TC=TC+(LineType(candid(In(i),4),6))*candid(In(i),5)*...
        Solution(In(i));
end
```

k) "contingency" M-file code

```
function [COL, Cnis, OLD] = contingency(linedata, busdata)
```

```
if isempty(busdata)
    fprintf('Input argument "busdata" containing the');
    fprintf(' information of buses.');
    error('"busdata" is undefined.');
end
if isempty(linedata)
    fprintf('Input argument "linedata" containing the');
    fprintf(' information of lines.');
    error('"linedata" is undefined.');
end

%% Problem outputs

%% COL: total overload of each contingency
%% Cnis: total number of islands in each contingency
%% OLD: over load and power flow data of all lines
    % in each contingency

%% Problem inputs:

%% busdata: required data of network buses
%% linedata: required data of network lines

%% Computing overload and power flow data in each contingency
    % (each iteration) and summing all overloads (COL);
Cnis = 0;
COL = 0;
for i = 1:size (linedata, 1)
%% Updating linedata after outage of each line
    esl = setxor (linedata (:,1), i); % Exsiting lines
    ulinedata = linedata; ulinedata(i,4) = 10^10;
    ulinedata(i,5) = 10^10; ULD = ulinedata;
    ulinedata1 = linedata (esl,:); ULD1 = ulinedata1;
%% Computing number of islands in each contingency
    nl = ULD1(:,2); nr = ULD1(:,3);
%% Exsiting buses:
    nbs = intersect (busdata (:,1), union(nl,nr));
    Is = setxor(nbs,busdata (:,1));    % Islanded buses
    UBD = busdata;
    nbus = size(busdata,1);
    Cnis = Cnis+length(Is);    % Number of islands
%% Computing Ybus for updated bus data (UBD) and updated
    % line data (ULD) for each contingency
    [Ybus]= ybus_calculation(UBD, ULD, [], [],[], []);
%% Running dc power flow for UBD and ULD
    [angle_r,angle_d, PF, OL, SOL]= dcpf(UBD, ULD, Ybus);
%% Computing overload and power flow data of all lines
    % in each contingency (each iteration)
    COL=COL+SOL;
    OL(:,3)=[];
    idOL=find(OL(:,3)~=0);
    OLF=OL(idOL,:); IOL(1,1)=linedata(i,2);
    IOL(1,2)=linedata(i,3); IOL(1,3)=SOL;
    for j=2:size(OLF,1)+1
```

```
        IOL(j,:)=OLF(j-1,:);
    end
    OLD{i,1}=IOL;
    clear IOL
end
```

L.5 DCLF.m

a) "DCLF" M-file code

```
clear
clc
%% Problem inputs:
Busdata = xlsread('DCLFDATA.xls', 'Busdata');
%% Busdata: Required bus data:
%% Busdata(:,1): bus number
%% Busdata(:,2): bus type 3=slack bus, 2=PV buses 1=PQ buses
%% Busdata(:,3): bus generation
%% Busdata(:,4): bus load

Linedata = xlsread('DCLFDATA.xls', 'Linedata');
%% Linedata: required branch data:
%% Linedata(:,1): branch ID
%% Linedata(:,2): branch source bus
%% Linedata(:,3): branch destination bus
%% Linedata(:,4): branch resistance
%% Linedata(:,5): branch reactance
%% Linedata(:,6): branch thermal loading
%% Linedata(:,7): branch circuit ID
%% Lg: load growth
Lg = xlsread('DCLFDATA.xls', 'Load growth');

%% Problem outputs:
% Normal condition
[Angle_r,Angle_d, Pf, Ol, Sol] = Dcpf(Busdata, Linedata, Lg);
%% Angle_r: voltage phase (radian)
%% Angle_d: voltage phase (degree)
%% Pf: flow of branches
%% Ol: over load amount of each branches
%% Sol: sum of all overloads

% N-1 condition
[Col, Old] = Contingency(Busdata, Linedata, Lg);
%% Col: total overload of each contingency
%% Old: over load and power flow data of all branchs
    % in each contingency

%% Printing the obtained results in both command window and
    % in result1.txt in the ANEP directory
print_DCLF
```

b) " Dcpf " M-file code

```
function [Angle_r,Angle_d, Pf, Ol, Sol] = ...
    Dcpf(Busdata, Linedata, Lg)
if nargin<3 | isempty(Lg), Lg = 0; end

%% Problem outputs:

%% Angle_r: voltage phase (radian)
%% Angle_d: voltage phase (degree)
%% Pf: flow of branches
%% Ol: over load amount of each branches
%% Sol: sum of all overloads

%% Problem inputs:

%% Busdata: required bus data:
%% Busdata(:,1): bus number
%% Busdata(:,2): bus type 3=slack bus, 2=PV buses 1=PQ buses
%% Busdata(:,3): bus generation
%% Busdata(:,4): bus load

%% Linedata: required branch data:
%% Linedata(:,1): branch ID
%% Linedata(:,2): branch source bus
%% Linedata(:,3): branch destination bus
%% Linedata(:,4): branch resistance
%% Linedata(:,5): branch reactance
%% Linedata(:,6): branch thermal loading
%% Linedata(:,7): branch circuit ID

%% Lg: load growth

%% Conversion block; to convert buses names
    % to consecutive numbers
Busname=Busdata(:,1);
nbus = length(Busname);
Busnumber = 1:nbus;
NL = Linedata(:,2);
NR = Linedata(:,3);
save namedata Busname Busnumber NL NR
for i = 1:length(NL)
    for j = 1:length(Busnumber);
        if NL(i) == Busname(j)
            nnl(i) = Busnumber(j);
        end
        if NR(i) == Busname(j)
            nnr(i) = Busnumber(j);
        end
    end
end
LD = Linedata; LD(:,2) = nnl; LD(:,3) = nnr';
BD = Busdata; BD(:,1) = Busnumber;
```

```
%% Ybus calculation
[Ybus, linedata, busdata] = Ybuscal(BD, LD, Lg);
%% Load flow calculation
nbus = size (busdata,1);
nl = linedata(:,2);
nr = linedata(:,3);
Smax = linedata(:,6);
nbr = length(nl);
Ps1 = (busdata(:,3)-busdata(:,4));
%% Finding non-slack buses in the busdata matrix
code = busdata(:,2);
[aa]=find(code~=3);
for n = 1:length(aa)
    for m = 1:length(aa)
            Ymn = Ybus(aa(n),aa(m));
            B(n,m) = -imag(Ymn);
    end
    Ps(n,1) = Ps1((aa(n)),1);
end
Binv = inv(B);
ang1 = Binv*Ps;
Angle_r = zeros(nbus,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%end
%% Calculation Power flow and over load values
for i = 1 : length(aa)
    aaa = aa(i);
    Angle_r(aaa) = ang1(i);
end
Angle_d = Angle_r*(180/pi);
jay = sqrt(-1);
for i = 1:nbr
     Pf(i,1) = NL(i); Ol(i,1)=NL(i);
     Pf(i,2) = NR(i); Ol(i,2)=NR(i);
     Pf(i,3) = Linedata(i,7); Ol(i,3)=Linedata(i,7);
     Pf(i,4) = (Angle_r(nl(i))-Angle_r(nr(i)))/...
         (linedata(i,5));
    if abs(Pf(i,4))>Smax(i)
        Ol(i,4) = Pf(i,4);
        Ol(i,5) = abs(Pf(i,4))-Smax(i);
    else
        Ol(i,4) = Pf(i,4);
        Ol(i,5) = 0;
    end
end
Sol = sum(Ol(:,5));
```

c) "Contingency" M-file code

```
function [Col, Old] = Contingency(Busdata, Linedata, Lg)
if nargin<3 | isempty(Lg), Lg = 0; end

%% Problem inputs:

%% Busdata: required data of network buses:
```

```
%% Busdata(:,1): bus number
%% Busdata(:,2): bus type 3=slack bus, 2=PV buses 1=PQ buses
%% Busdata(:,3): bus generation
%% Busdata(:,4): bus load
%% Linedata: required data of network branches:
%% Linedata(:,1): branch ID
%% Linedata(:,2): branch source bus
%% Linedata(:,3): branch destination bus
%% Linedata(:,4): branch resistance
%% Linedata(:,5): branch reactance
%% Linedata(:,6): branch thermal loading
%% Linedata(:,7): branch circuit ID
%% Lg: load growth

%% Outputs

%% Col: total overload of each contingency
%% Cnis: total number of islands in each contingency
%% Old: over load and power flow data of all branchs
    % in each contingency

%% Computing overload and power flow data in each contingency
    % (each iteration) and summing all overloads (Col);
Col = 0;
for i = 1:size (Linedata, 1)
%%  Updating Linedata after outage of each branch
    esl = setxor (Linedata (:,1), i);    %Exsiting branchs
    ulinedata = Linedata; ulinedata(i,4) = 10^10;
    ulinedata(i,5) = 10^10; ULD = ulinedata;
    ulinedata1 = Linedata(esl,:); ULD1 = ulinedata1;
    UBD = Busdata; nbus = size(Busdata,1);
%%  Running dc power flow for updated bus data (UBD) and
    % updated line data (ULD)
    [angle_r,angle_d, PF, OL, SOL] = Dcpf(UBD, ULD, Lg);
%% Computing overload and power flow data of all branchs
    % in each contingency (each iteration)
    Col = Col+SOL;
    OL(:,4) = [];
    idOL = find(OL(:,4) ~= 0);
    OLF = OL(idOL,:);
    IOL(1,1) = Linedata(i,2); IOL(1,2) = Linedata(i,3);
    IOL(1,3) = Linedata(i,7); IOL(1,4) = SOL;
    for j = 2:size(OLF,1)+1
      IOL(j,:) = OLF(j-1,:);
    end
    Old{i,1} = IOL;
    clear IOL
end
```

d) "print_DCLF" M-file code

```
fid = fopen('results.txt', 'wt');
fprintf(fid, '**********************************************');
fprintf(fid, '*************************');
```

```
fprintf(fid,'\n                             Normal');
fprintf(fid,' condition\n');
fprintf(fid, '*********************************************');
fprintf(fid,'*************************\n');
fprintf(fid,'\n***************Bus data******************\n');
fprintf(fid, '     No. bus          Voltage angle (Rad)\n');
fprintf(fid, '***************       ********************\n');
for i = 1:size (Busdata,1);
    fprintf(fid, ' %10.0f %27.5f \n', ...
        Busdata(i,1), Angle_r(i,1));
end
if Sol == 0
    fprintf(fid, '\n No overload in normal condition\n');
else
    NL = size (Linedata,1);
    fprintf(fid, '\n Overload at normal:');
    fprintf(fid,'\n Total overload of normal condition is');
    fprintf(fid,' %3.5f pu\n',Sol);
    fprintf(fid, '***************************************');
    fprintf(fid, '*************************************');
    fprintf(fid,'\n*************** Power flow and overload');
    fprintf(fid,' values of branches *****************\n');
    fprintf(fid, '****************************************');
    fprintf(fid, '************************************\n');
    fprintf(fid, '\n From bus      To bus      Circuit ID ');
    fprintf(fid, '     Line flow (pu)      Overload (pu)\n');
    for i = 1 : NL
      fprintf(fid,'\n %1.0f %14.0f %15.0f %20.5f %20.5f\n'...
           , Ol(i,:));
    end
end
fprintf(fid, '\n\n****************************************');
fprintf(fid, '********************************************');
fprintf(fid, '*******');
fprintf(fid,'\n                             N-1 condition\n');
fprintf(fid, '**********************************************');
fprintf(fid, '*********************************************');

if (Col == 0)
    fprintf(fid, '\n No overload in N-1 condition\n');
    fprintf(fid, '\n*************************************\n');
else
    fprintf(fid, '\n*************************************');
    fprintf(fid, '***************************************');
    fprintf(fid, '*************\n');
    LCOL=Old{size (Col,1),1};
    fprintf(fid, '               Overload values of ');
    fprintf(fid, 'branches in N-1 condition');
    for i = 1 : size (Old,1)
       iOLD = Old{i,1};
       iL = iOLD(1,:);
       fprintf(fid, '\n********************************');
       fprintf(fid, '**********************************');
       fprintf(fid, '***********************\n');
```

```
            fprintf(fid,' Total overload for outage of line: ');
            fprintf(fid, '');
            fprintf(fid,'From bus %3.0f to bus %3.0f and',iL(1:2));
            fprintf(fid,' circuit ID %3.0f is %6.5f pu\n',iL(3:4));
            fprintf(fid, '*************************************');
            fprintf(fid, '*************************************');
            fprintf(fid, '*****************\n');
            fprintf(fid,' following lines are overloaded in');
            fprintf(fid,' this outage\n');
            fprintf(fid, '**************************************');
            fprintf(fid,'***************************************');
            fprintf(fid,'**************\n');
            fprintf(fid, '  From Bus To Bus  Circuit ID         ');
            fprintf(fid,'Overload (pu)\n');
            fprintf(fid, '  *******  ******  **********         ');
            fprintf(fid,'***********');
            for j = 2 : size (iOLD,1);
                fprintf(fid, '\n %5.0f %10.0f %8.0f %18.5f\n',...
                    iOLD(j,:));
            end
        end
    end
    fclose(fid);
    %% Print in the command window
    fprintf('**************************************************');
    fprintf('********************');
    fprintf('\n                            Normal condition\n');
    fprintf('**************************************************');
    fprintf('********************\n');
    fprintf('\n***************Bus data******************\n');
    fprintf('     No. bus          Voltage angle (Rad)\n');
    fprintf('***************       *******************\n');
    for i = 1 : size (Busdata,1);
        fprintf(' %10.0f %27.5f \n', Busdata(i,1), Angle_r(i,1));
    end
    if Sol == 0
        fprintf('\n No overload in normal Condition\n');
    else
        NL = size (Linedata,1);
        fprintf('\n overload at Normal:');
        fprintf('\n Total overload of normal condition is ');
        fprintf('%3.5f pu\n',Sol);
        fprintf('*********************************************');
        fprintf('*********************************');
        fprintf('\n**************** Power flow and overload ');
        fprintf('values of branches ****************\n');
        fprintf('********************************************');
        fprintf('*********************************\n');
        fprintf('\n From bus      To bus      Circuit ID    ');
        fprintf('   Line flow (pu)      Overload (pu)\n');
        for i = 1 : NL
            fprintf('\n %6.0f %12.0f %14.0f %16.5f %19.5f\n',...
                Ol(i,:));
        end
```

```
end
fprintf('\n\n**********************************************');
fprintf('*******************************************');
fprintf('\n                             N-1 condition\n');
fprintf('**********************************************');
fprintf('***************************************\n');
if (Col == 0)
    fprintf('\n No overload in N-1 condition\n');
    fprintf('\n*************************************\n');
else
    fprintf('\n******************************************');

fprintf('*********************************************\n');
    LCOL=Old{size (Col,1),1};
    fprintf('               Overload values of branches in');
    fprintf(' N-1 condition');
    for i = 1 : size (Old,1)
        iOLD = Old{i,1};
        iL = iOLD(1,:);
        fprintf('\n*************************************');
        fprintf('****************************************');
        fprintf('*********\n');
        fprintf(' Total overload for outage of line: from ');
        fprintf('bus %3.0f to bus %3.0f and ',iL(1:2));
        fprintf('circuit ID %3.0f is %6.5f pu\n',iL(3:4));
        fprintf('****************************************');
        fprintf('****************************************');
        fprintf('*******\n');
        fprintf(' Following lines are overloaded ');
        fprintf('in this outage\n');
        fprintf('****************************************');
        fprintf('****************************************');
        fprintf('*******\n');
        fprintf('  From Bus To Bus  Circuit ID        ');
        fprintf('Overload (pu)\n');
        fprintf('  *******  ******  **********        ');
        fprintf('***********');
        for j = 2 : size (iOLD,1);
            fprintf('\n %5.0f %10.0f %8.0f %18.5f\n',...
                iOLD(j,:));
        end
    end
end
```

e) "Ybuscal" M-file code

```
function [Ybus, linedata, busdata] = ...
    Ybuscal(busdata, linedata, Lg);
if nargin<3 | isempty(Lg), Lg = 0; end
%%
busdata(:,4) = busdata(:,4).*(1+Lg);
busdata(:,5) = busdata(:,5).*(1+Lg);
%% Computation of admittance of all branches
j = sqrt(-1);
```

```
i = sqrt(-1);
X = linedata(:,5);
nbr = length(linedata(:,1));
nbus = size (busdata,1);
nl = linedata(:,2); nr = linedata(:,3);
Z = (j*X);
y = ones(nbr,1)./Z;          % Branch admittance
Ybus = zeros(nbus,nbus);     % Initialize Ybus to zero
%% Formation of the off diagonal elements
for k = 1 : nbr;
       Ybus(nl(k),nr(k)) = Ybus(nl(k),nr(k))-y(k);
       Ybus(nr(k),nl(k)) = Ybus(nl(k),nr(k));
end
%% Formation of the diagonal elements
for n = 1 : nbus
    for m = (n+1) : nbus
        Ybus(n,n) = Ybus(n,n)-Ybus(n,m);
    end
    for m = 1 : n-1
        Ybus(n,n) = Ybus(n,n)-Ybus(n,m);
    end
end
```

L.6 ACLF.m

a) "ACLF" M-file code

```
clear
clc
%% Load Data
Linedata = xlsread('ACLFDATA.xls', 'Linedata');
Busdata = xlsread('ACLFDATA.xls', 'Busdata');
Setdata = xlsread('ACLFDATA.xls', 'Loadflowsetting');
Basemva = Setdata (1,1);            % Base MVA
MIter = Setdata (1,2);              % Maximum iteration
Acc = Setdata (1,3);                % Accuracy
%% Voltage acceptable deadband
Vmin = Setdata (1,4); Vmax = Setdata (1,5);

%% busdata(:,1): bus number
%% busdata(:,2): type of bus 1-Slack, 2-PV, 3-PQ
%% busdata(:,3): voltage of PV buses
%% busdata(i,5): active power Load in bus i
%% busdata(i,6): reactive power Load in bus i
%% busdata(i,7): active power generation in bus i
%% busdata(i,8): reactive power generation in bus i
%% busdata(i,9): Qmin; minimum reactive power limit of bus i
%% busdata(i,10): Qmax; maximum reactive power limit of bus i
%% busdata(i,11): injected reactive power to bus i

for i = 1 : size (Busdata,1)
```

```
        if Busdata(i,2) == 3
            Vini(i) = 1.0;
        else
            Vini(i) = Busdata(i,3);
        end
    end

    [Vb0, Fij0, nfij0,Vprof0, SID0] = Acpf(Busdata, ...
        Linedata, Basemva, MIter, Acc, 0, Vmin, Vmax, Vini);
    %% Calculating voltage stability index (Pstab)
        % in normal condition
    slstep = 0.005;                 % Small step length
    llstep = 0.05;                  % Large step length
    mstep = 1000;                   % Mmaximum step
    if  SID0 == 0
        fprintf('\n ******************************************');
        fprintf('*********************************************');
        fprintf('******************')
        fprintf('\nWARNING: The load flow solution did not ');
        fprintf('converged At Base Case ')
        fprintf('\n ******************************************');
        fprintf('*********************************************');
        fprintf('****************** \n')
    else
        i = 1;
        SID = 1;
        while i <= mstep & SID == 1
            LR = i*llstep;
            [Vb, Fij, nfij, Vprofc, SID] = Acpf(Busdata,...
                Linedata, Basemva, MIter, Acc, LR, Vmin,...
                Vmax, Vini);
            DelV{i,1} = Vb(:,2); DelV{i,2} = Vprofc;
            if SID ~= 1
                SID = 1;
                LR = (LR-llstep);
                j = 0;
                j = i+j;
                while SID == 1
                    LR = LR+slstep;
                    [Vb, Fij, nfij, Vprofc, SID] = Acpf(...
                        Busdata, Linedata, Basemva, MIter, ...
                        Acc, LR, Vmin, Vmax, Vini);
                    DelV{j,1} = Vb(:,2); DelV{j,2} = Vprofc;
                    j = j+1;
                end
                save DelV DelV
                Pstab = LR+1;
                break
            end
            i = i+1;
        end
        for j=1:size (DelV,1)
            A=DelV{j,1};
            for i=1:size (Busdata,1)
```

```
            C(i,1)=Busdata (i,1);
            k=j+1;
            C(i,k)=A (i,1);
         end
      end

  %% Calculating voltage profile index (Vprof) voltage
      % stability index (Pstab) in N-1 condition
      linenumber = size(Linedata, 1);
      lineno = 1 : linenumber;
      for i = 1 : linenumber
          esl = setxor (lineno, i);    % Exsiting lines
          ulinedata = Linedata; ulinedata(i,3) = 10^10;
          ulinedata(i,4) = 10^10; ulinedata(i,5) = 0;
          ULD = ulinedata; ulinedata1 = Linedata (esl,:);
          ULD1 = ulinedata1; UBD = Busdata;
          nbus = size(Busdata,1);
          [Vbc0, Fijc0, nfijc0,Vprofc0, SID0] = Acpf(UBD,...
              ULD1, Basemva, MIter, Acc, 0, Vmin, Vmax,Vini);

          if  SID0 == 0
             TC{i,1} = Linedata(i,1);
             TC{i,2} = Linedata(i,2);
             TC{i,3} = -1;
             TC{i,4} = 0;
             TC{i,5} = C;
             continue
          else
             ii = 1;
             SID = 1;
             while ii <= mstep & SID == 1
                   LRc = ii*llstep;
                   [Vbc, Fijc, nfijc,Vprofc, SID] = Acpf(...
                       UBD, ULD1, Basemva, MIter, Acc, LRc,...
                       Vmin, Vmax,Vini);
                   DelVc{ii,1} = Vbc(:,2);
                   DelVc{ii,2} = Vbc(:,3);
                   if SID ~= 1
                      SID = 1;
                      LRc = (LRc-llstep);
                      jj = 0;
                      jj = ii+jj;
                      while SID == 1
                          LRc = LRc+slstep;
                          [Vbc, Fijc, nfijc,Vprofc, SID] =...
                              Acpf(UBD, ULD1, Basemva, ...
                              MIter, Acc, LRc, Vmin,...
                              Vmax,Vini);
                          DelVc{jj,1} = Vbc(:,2);
                          DelVc{jj,2} = Vbc(:,2);
                          jj = jj+1;
                      end
                      save DelVc DelVc
                      break
```

```
                    end
                    ii = ii+1;
                end
            end
            clc
            Pstabc = LRc+1;
            for j = 1 : size (DelVc,1)
                A = DelVc{j,1};
                %% Voltage profile in different iteration
                B = DelVc{j,2};
                for iii = 1:size(Busdata,1)
                    C(iii,1) = Busdata (iii,1);
                    k = j+1;
                    C(iii,k) = A(iii,1);
                end
            end

            TC{i,1} = Linedata(i,1);
            TC{i,2} = Linedata(i,2);
            TC{i,3} = Vprofc0;
            TC{i,4} = Pstabc;
            TC{i,5} = C;

        end
        print_rpp
    end
```

b) "Acpf" M-file code

```
function[Vb, Fij, nfij,Vprof, convergence] = Acpf(Busdata...
    , Linedata, baseMVA, MIter, Acc, LR, Vmin, Vmax,Vini)
%% Program for Newton-Raphson load flow analysis
%% Assumption, bus 1 is considered as slack bus

%% Calling ybusppg.m to get bus admittance matrix
%% Y = ybusppg();

%% Calling busdata30.m to get bus datas
%% busdata = busdata30();

%% Base MVA
%% baseMVA = 100;

%% Outputs
%% Vb: voltage of buses
%% Fij: line flow data
%% vprof: voltage profile
%% convergence: load flow convergence indication

%% Inputs
%% Basemva: Base MVA
%% MIter: maximum iteration of solving load flow
%% Acc: load flow solving telorance
%% Vmin<V< Vmax; voltage acceptable dead band for calculating
```

```
    % voltage profile index
%% LR: load growth
%% Linedata: network line data
%% Linedata: network bus data
%% Conversion block for converting  buses names to numbers
Busname = Busdata(:,1);
nbus = length(Busname);
Busnumber = 1 : nbus;
nl = Linedata(:,1);
nr = Linedata(:,2);
save namedata Busname Busnumber nl nr
for i = 1 : length(nl)
    for j = 1 : length(Busnumber);
        if nl(i) == Busname(j)
            nnl(i) = Busnumber(j);
        end
        if nr(i) == Busname(j)
            nnr(i) = Busnumber(j);
        end
    end
end
linedata = Linedata; linedata(:,1) = nnl;
linedata(:,2) = nnr'; busdata = Busdata;
busdata(:,1) = Busnumber;
%% Ybus calculation
[Y] = LFYBUS(linedata,busdata, baseMVA);
%% Data retrivial from busdata
bus = busdata(:,1);        % Bus number
type = busdata(:,2);       % Type of bus 1-Slack, 2-PV, 3-PQ
%% Type of bus 1-Slack, 0-PV, 2-PQ
%% type = busdata(:,2);
V = busdata(:,3);          % Specified voltage
del = busdata(:,4);        % Voltage angle
Pli = busdata(:,5);        % PLi
Qli = busdata(:,6);        % QLi
Pg = busdata(:,7);         % PGi
Qg = busdata(:,8);         % QGi
pv = find(type == 2);      % Index of PV buses
pq = find(type == 3);      % Index of PQ buses
Pl = Pli*(1+LR);           % Load growth consideration
Ql = Qli*(1+LR);
npv = length(pv);          % Number of PV buses
npq = length(pq);          % Number of PQ buses
Qmin = busdata(:,9);       % Minimum reactive power limit
Qmax = busdata(:,10);      % Maximum reactive power limit
nbus = max(bus);           % To get no. of buses
%% Computing net power of each bus
P = Pg - Pl;               % Pi = PGi - PLi
Q = Qg - Ql;               % Qi = QGi - QLi
%% P = Pl - Pg;            % Pi = PGi - PLi
%% Q = Ql - Qg;            % Qi = QGi - QLi
P = P/baseMVA;             % Converting to p.u.
Q = Q/baseMVA;
Qmin = Qmin/baseMVA;
```

```
Qmax = Qmax/baseMVA;
%% Tol = 10;                % Tolerence kept at high value
%% Iter = 1;                % Iteration starting
%% Pre-specified value of active and reactive power
Psp = P;
Qsp = Q;
G = real(Y);                % Conductance
B = imag(Y);                % Susceptance

%% Beginning of the load flow calculation
convergence = 1;
Tol = 10;                   % Tolerence kept at high value
Iter = 1;                   % Iteration starting
IIII = 1;
%% Iteration starting
while (Tol > Acc | IIII == 1) & Iter <= MIter
    P = zeros(nbus,1);
    Q = zeros(nbus,1);
%% Calculate P and Q
    for i = 1:nbus
        for k = 1:nbus
            P(i) = P(i)+V(i)*V(k)*(G(i,k)*cos(del(i)...
                -del(k)) + B(i,k)*sin(del(i)-del(k)));
            Q(i) = Q(i)+V(i)*V(k)*(G(i,k)*sin(del(i)...
                -del(k)) - B(i,k)*cos(del(i)-del(k)));
        end
    end

%% Checking Q-limit violations
%% Only checked up to 7th iterations
%% if Iter <= 7 && Iter > 4
    if Iter >= 5
        IIII = 0;
        for n = 2 : nbus
            if type(n) == 2
                QG = Q(n)+Ql(n)/baseMVA;
                % QG = Q(n);
                if QG < Qmin(n)
                    V(n) = V(n) + 0.001;
                    IIII = 1;
                elseif QG > Qmax(n)
                    V(n) = V(n) - 0.001;
                    IIII = 1;
                end
            end
        end
    end

%% Calculate change from specified value
    dPa = Psp-P;
    dQa = Qsp-Q;
    k = 1;
    dQ = zeros(npq,1);
    for i = 1:nbus
```

```
            if type(i) == 3
                dQ(k,1) = dQa(i);
                k = k+1;
            end
        end
        dP = dPa(2:nbus);
        M = [dP; dQ];          % Mismatch vector

    %% Jacobian
    %% J1: derivative of real power injections with angles
        J1 = zeros(nbus-1,nbus-1);
        for i = 1:(nbus-1)
            m = i+1;
            for k = 1:(nbus-1)
                n = k+1;
                if n == m
                    for n = 1:nbus
                        J1(i,k) = J1(i,k) + V(m)* V(n)*...
                        (-G(m,n)*sin(del(m)-del(n)) + B(m,n)...
                        *cos(del(m)-del(n)));
                    end
                    J1(i,k) = J1(i,k) - V(m)^2*B(m,m);
                else
                    J1(i,k) = V(m)* V(n)*(G(m,n)*sin(del(m)...
                        -del(n)) - B(m,n)*cos(del(m)-del(n)));
                end
            end
        end

    %% J2: derivative of real power injections with V
        J2 = zeros(nbus-1,npq);
        for i = 1:(nbus-1)
            m = i+1;
            for k = 1:npq
                n = pq(k);
                if n == m
                    for n = 1:nbus
                        J2(i,k) = J2(i,k) + V(n)*(G(m,n)*...
                            cos(del(m)-del(n)) + B(m,n)*...
                            sin(del(m)-del(n)));
                    end
                    J2(i,k) = J2(i,k) + V(m)*G(m,m);
                else
                    J2(i,k) = V(m)*(G(m,n)*cos(del(m)-del(n))...
                        + B(m,n)*sin(del(m)-del(n)));
                end
            end
        end

    %% J3: derivative of reactive power injections with angles
        J3 = zeros(npq,nbus-1);
        for i = 1:npq
            m = pq(i);
            for k = 1:(nbus-1)
```

```
                n = k+1;
                if n == m
                    for n = 1:nbus
                        J3(i,k) = J3(i,k) + V(m)* V(n)*(G(m,n)...
                            *cos(del(m)-del(n)) + B(m,n)*...
                            sin(del(m)-del(n)));
                    end
                    J3(i,k) = J3(i,k) - V(m)^2*G(m,m);
                else
                    J3(i,k) = V(m)* V(n)*(-G(m,n)*cos(del(m)...
                        -del(n)) - B(m,n)*sin(del(m)-del(n)));
                end
            end
        end

    %% J4: derivative of reactive power injections with V
        J4 = zeros(npq,npq);
        for i = 1:npq
            m = pq(i);
            for k = 1:npq
                n = pq(k);
                if n == m
                    for n = 1:nbus
                        J4(i,k) = J4(i,k) + V(n)*(G(m,n)*sin...
                            (del(m)-del(n)) - B(m,n)*cos...
                            (del(m)-del(n)));
                    end
                    J4(i,k) = J4(i,k) - V(m)*B(m,m);
                else
                    J4(i,k) = V(m)*(G(m,n)*sin(del(m)-del(n))...
                        - B(m,n)*cos(del(m)-del(n)));
                end
            end
        end
        J = [J1 J2; J3 J4];    % Jacobian
        X = inv(J)*M;          % Correction vector
        dTh = X(1:nbus-1);     % Change in voltage angle
        dV = X(nbus:end);      % Change in voltage magnitude

    %% Updating state vectors
        del(2:nbus) = dTh + del(2:nbus);     % Voltage angle
        k = 1;
        for i = 2:nbus
            if type(i) == 3
                V(i) = dV(k) + V(i);         % Voltage magnitude
                k = k+1;
            end
        end
        Tol = max(abs(M));                   % Tolerance.
        if Iter==MIter & Tol > Acc
            convergence=0;
            break
        else
        Iter = Iter + 1;
```

```
        end
    end
    Iter;                       % Number of iterations took
    Vs = V;                     % Bus voltage magnitudes in p.u.
    Del = 180/pi*del;           % Bus voltage angles in degree

    %% Outputs

    %% Line power flow data
    jay = sqrt (-1);
    Vmr = V.*cos(del); Vmi = V.*sin(del);
    Vm = Vmr + jay*(Vmi);
    Iij = zeros(nbus,nbus);     % Line current
    Sij = zeros(nbus,nbus);     % Line flow
    Si = zeros(nbus,1);         % Bus power injections
    busdata(:,3) = Vs;
    [Y] = LFYBUS(linedata,busdata, baseMVA);
    %% Line power flows
    ii = 0;
    Fij = zeros (nbus,4);
    for m = 1:nbus
        for n = m+1:nbus
            Iij(m,n) = -(Vm(m) - Vm(n))*Y(m,n);
            Iij(n,m) = -Iij(m,n);
            Sij(m,n) = Vm(m)*conj(Iij(m,n));
            Sij(n,m) = -Sij(m,n);
            if Sij(m,n) ~= 0
                ii = ii+1;
                %% Fij (ii,1) = m; Fij (ii,2) = n;
                %% Fij (ii,3) = real (Sij(m,n));
                %% Fij (ii,4) = imag (Sij(m,n));
                Fij (ii,1) = Busname(m);
                Fij (ii,2) = Busname(n);
                Fij (ii,3) = real (Sij(m,n));
                Fij (ii,4) = imag (Sij(m,n));
            end
        end
    end
    nfij=ii;
    %% Bus power injections..
    for i = 1 : nbus
        for k = 1 : nbus
            Si(i) = Si(i) + conj(Vm(i))* Vm(k)*Y(i,k);
        end
    end
    Pi = real(Si); Qi = -imag(Si);
    %% Bus data information
    for i = 1 : nbus
        Vb(i,1) = Busname(i); Vb(i,2) = V(i); Vb(i,3) = del(i);
        Vb(i,4) = Pi(i); Vb(i,5) = Qi(i);
    end
    %% Computing voltage profile
    Vprof = 0;
    for i = 1 : length(busdata(:,1))
```

```
        if Vs(i) >= Vmax
           Vprof = Vprof+((Vs(i)-Vini(i))^2);
        else
          if Vs(i) <= Vmin
              Vprof = Vprof+((Vs(i)-Vini(i))^2);
          end
        end
    end
```

c) "LFYBUS" M-file code

```
function[Ybus,nbr,nl,nr,nbus] = ...
    LFYBUS(linedata,busdata, baseMVA);
j = sqrt(-1);
i = sqrt(-1);
ai = sqrt(-1);
nl = linedata(:,1);
nr = linedata(:,2);
R = linedata(:,3);
X = linedata(:,4);
Bc = j*linedata(:,5);
a = linedata(:,6);
nbr = length(linedata(:,1));
nbus = max(max(nl), max(nr));
Z = R + j*X;
y = ones(nbr,1)./Z;        % Branch admittance
v = busdata(:,3);
Qinj = busdata(:,11)./baseMVA;
rrb = ai.*(Qinj./(v.^2));
for n = 1 : nbr
if a(n) <= 0  a(n) = 1; else end
Ybus = zeros(nbus,nbus);
%% Obtaining nondiagonal elements
for k = 1 : nbr;
       Ybus(nl(k),nr(k)) = Ybus(nl(k),nr(k))-y(k)/a(k);
       Ybus(nr(k),nl(k)) = Ybus(nl(k),nr(k));
    end
end
%% Formation of the diagonal elements
for  n = 1 : nbus
     for k = 1 : nbr
         if nl(k) == n
         Ybus(n,n) = Ybus(n,n) + y(k)/(a(k)^2) + Bc(k);
         elseif nr(k) == n
         Ybus(n,n) = Ybus(n,n) + y(k) + Bc(k);
         else, end
     end
     Ybus(n,n) = Ybus(n,n) + rrb(n);
end
clear Pgg
```

d) "print_rpp" M-file code

```
%clc
fid = fopen('results.txt', 'wt');
```

```
NL = size (Fij,1);
fprintf(fid, '\n\n\n************************* Normal ');
fprintf(fid, 'condition ************************');
if Vprof0<0
    fprintf(fid,' The load flow does not converge ');
    fprintf(fid, 'in normal condition\n');
    fprintf(fid, '-------------------------------------');
    fprintf(fid, '--------------\n\n');
else
    fprintf(fid,' \n Voltage profile index(Vprof) in normal');
    fprintf(fid, 'condition is %6.5f\n',Vprof0);
    fprintf(fid,' \n Voltage stability index (Pstab) in ');
    fprintf(fid, 'normal condition is %6.5f\n',Pstab);
    fprintf(fid, '-----------------------------------------');
    fprintf(fid, '---------------------------\n\n');
    fprintf(fid, '************************** Load flow ');
    fprintf(fid, 'results **********************\n');
    fprintf(fid, '*****************************************');
    fprintf(fid, '***************************\n');
    fprintf(fid, '               Bus data                \n');
    fprintf(fid, '---------------------------------------\n');
    fprintf(fid, '    Bus number      Voltage      Phase\n');
    fprintf(fid, '    ----------      -------      -----');
    for i=1:size (Busdata,1);
        fprintf(fid, '\n %10.0f %13.3f %13.3f', Vb0(i,1),...
            Vb0(i,2), Vb0(i,3));
    end
    fprintf(fid, '\n---------------------------------------');
    fprintf(fid, '\n\n************************ Power flow ');
    fprintf(fid, 'of branches**********************\n');
    fprintf(fid, '-----------------------------------------');
    fprintf(fid, '---------------------------');
    fprintf(fid, '\n  From bus      To bus       Active ');
    fprintf(fid, 'power (Pu)      Reactive power (Pu)');
    fprintf(fid, '\n  --------      ------       ---------');
    fprintf(fid, '-------       ---------------- ');
    for i=1:NL
        fprintf(fid, '\n %5.0f %14.0f %15.3f %22.3f',...
            Fij0(i,:));
    end
    fprintf(fid, '\n  ------------------------------------');
    fprintf(fid, '------------------------------');
    fprintf(fid, '\n\n\n**********************************');
    fprintf(fid, '*************');
end
fprintf(fid, '\n               N-1 condition              ');
fprintf(fid, '                      \n');
fprintf(fid, '********************************************');
fprintf(fid, '***\n\n');
for i = 1 : NL
    if TC{i,3} < 0
        fprintf(fid,' !!!!\n');
        fprintf(fid,' For outage of the line from bus ');
        fprintf(fid,'%3.0f to bus %3.0f\n',TC{i,1},TC{i,2});
```

```
            fprintf(fid,' The load flow does not converge\n');
            fprintf(fid, '------------------------------------');
            fprintf(fid, '-----------\n\n');
        else
            fprintf(fid,' For outage of the line from bus ');
            fprintf(fid,'%3.0f to bus %3.0f \n',TC{i,1},TC{i,2});
            fprintf(fid,' Voltage profile index (Vprof) is ');
            fprintf(fid,'%6.5f and\n',TC{i,3});
            fprintf(fid,' Voltage stability index (Pstab) is');
            fprintf(fid,' %6.5f\n',TC{i,4});
            fprintf(fid, '------------------------------------');
            fprintf(fid, '-----------\n\n');
        end
    end
    %% Printing in the command window
    fprintf('\n************************** Normal condition ***');
    fprintf('*********************');
    if Vprof0<0
        fprintf(' The load flow does not converge in normal ');
        fprintf('condition\n');
        fprintf('---------------------------------------------');
        fprintf('-------\n\n');
    else
        fprintf(' \n Voltage profile index(Vprof) in normal ');
        fprintf('condition is %6.5f\n', Vprof0);
        fprintf(' \n Voltage stability index (Pstab) in normal');
        fprintf('condition is %6.5f\n',Pstab);
        fprintf('----------------------------------------------');
        fprintf('----------------------\n\n');
        fprintf('************************** Load flow ');
        fprintf('results ***********************\n');
        fprintf('*********************************************');
        fprintf('**********************\n');
        fprintf('               Bus data                \n');
        fprintf('---------------------------------------\n');
        fprintf('    Bus number      Voltage      Phase\n');
        fprintf('    ----------      -------      -----');
        for i = 1 : size (Busdata,1);
            fprintf('\n %10.0f %13.3f %13.3f',...
                Vb0(i,1), Vb0(i,2), Vb0(i,3));
        end
        fprintf('\n---------------------------------------');
        fprintf('\n\n*********************** Power flow of ');
        fprintf('branches**********************\n');
        fprintf('--------------------------------------------');
        fprintf('------------------------');
        fprintf('\n  From bus      To bus       Active power ');
        fprintf('(Pu)      Reactive power (Pu)');
        fprintf('\n  --------      ------       --------------');
        fprintf('--       ----------------');
        for i = 1 : NL
            fprintf('\n %5.0f %14.0f %15.3f %22.3f', Fij0(i,:));
        end
        fprintf('\n  -----------------------------------------');
```

```
        fprintf('-------------------------');
        fprintf('\n\n\n***************************************');
        fprintf('********');
    end
    fprintf('\n               N-1 condition                   ');
    fprintf('                 \n');
    fprintf('*********************************************');
    fprintf('**\n\n');
    for i = 1 : size(TC,1)
        if TC{i,3} < 0
            fprintf(' !!!!\n');
            fprintf(' For outage of the line from bus ');
            fprintf('%3.0f to bus %3.0f\n',TC{i,1},TC{i,2});
            fprintf(' The load flow does not converge\n');
            fprintf('---------------------------------------');
            fprintf('--------\n\n');
        else
            fprintf(' For outage of the line from bus ');
            fprintf('%3.0f to bus %3.0f \n',TC{i,1},TC{i,2});
            fprintf(' Voltage profile index (Vprof) is ');
            fprintf('%6.5f and\n',TC{i,3});
            fprintf(' Voltage stability index (Pstab) is ');
            fprintf('%6.5f\n',TC{i,4});
            fprintf('---------------------------------------');
            fprintf('--------\n\n');
        end
    end
```

Index

S

T