Economic dispatch - shift factor evaluation and load profile power flow simulations in Excel VBA macro

Abstract. Calculating shift factor (SF) with power injection is essential for dispatch demand network analysis. Similarly, load profile simulation in an electrical power flow network is essential to determine powers and voltages transmission in the network. Many available specialized commercial software packages for this purpose are expensive and licensed. The purpose of this paper is to present the solution of the above two mentioned problems simply and in readily available Microsoft Excel Visual Basic Applications (VBA) Macros. Efforts are made here not to describe the theory behind electrical dispatch demand and power flow networks, but to describe VBA Macro programming for this purpose. Further, the website link for the application program is provided for teaching, learning, and expansion of the program.

Introduction

Excel is widely used as a spreadsheet tool for data storage, accounting, finance in the business organization. And is less used for engineering application which requires repetitive and iterative programming approaches for algebra and matrix equations, complex numbers, optimization specially in Electric power related field. The Excel programming capabilities are available with embedded VBA Macro tool to tackle engineering related problems. The work presented in [1] shows network flow model to determine how much power should be allowed to flow across each transmission lines with minimal losses. In [2] “Microsoft Excel based Power System Load Flow Analysis” (MSEBPSLF) is presented, which is used by teachers, undergraduate, graduate students and practicing engineers, who are interested in using spreadsheet tool for carrying out load flow analysis. In [3] the objective of the power-flow problem is described to calculate the voltage magnitudes and phase angles at each bus or node in a given power system using Gauss-Seidel or the Newton-Raphson methods to determine the power flows in a small-scale power system. In [4] an Example of Using Microsoft Excel Solver for Power Network Calculations is presented. In [5], the authors present radial distribution method with HV forward and backward convergence. Excel spreadsheet related electric market problems and examples are excellently presented in [6].

However, in all above references most of the works are accomplished in Excel spread sheet and VBA macro programming was not applied. In this paper basic work for economic dispatch (ED) describe in [6], [7] is implemented in Excel Macro and shows a new method to find inverse power injection for binding transmission constraints. Further the one-point load power network calculations in [4] is extended to one day load profile data network calculation with Excel Macro programming techniques.

Economic dispatch

The In era of open electric sale market, the suppliers and bidders are bound to meet the customer demand in least expensive way. The term ED is used to define the interest of all parties involved in electric market sale. ED calculation method is based on sensitivity factors to approximate the change of branch power flow depending on the change of nodal generation or outage of another branch. Values of sensitivity factors can be obtained by calculation based on the solving DC circuit. Factors are divided into two basic types depending on the type of modeled outage – GSF (Generation Shift Factor) and LODF (Line Outage Distribution Factor) [8]. All SFs are computed relative to the reference or swing bus. GSF shows how the flow in the branch will change if the injection at the bus changes by one MW. SF gives the fraction of a change in injection at bus i that appears on branch k, Figure 1.

Fig.1. Four bus (i) and five Branch (k) network

A. Economic dispatch formulation

ED problem formulation is defined by objective function shown in equation (1) - Minimize total Pgi generation cost for i=1,2,...,N injection bus, C_i is incremental energy price of P_gi at Generation injection bus i. P_ij in equation (2) system balance is defined \(\sum P_{g_i} = \sum P_{l_j}\) for demand load j=1,2,...,L. In equation (3) S_ki is a SF from branch k to the bus i is less than transmission limit, T_k of kth branch.
that defined by equation (4) for lower and upper generators limits.

\[ \text{Min} \sum_{i=1}^{N} C_i \cdot P_{gi} \]

\[ \sum_{i=1}^{N} P_{gi} - \sum_{j=1}^{L} P_{ij} = 0, \text{Losses} = 0 \]

\[ \sum_{i=1}^{N} S_{ki} \cdot P_{gi} \leq T_{k}^{max}, k = 1,2, ..., K \]

\[ \sum_{i=1}^{N} S_{ki} \cdot P_{gi} \leq T_{k}^{max}, k = 1,2, ..., K \]

### Shift factor (Excel sheet 1)

Simple four (N) bus and five (M) Branch network [7] shown in Figure 1 is used in this paper to demonstrate VBA macro programming techniques. Once the macro programming is understood it is possible to extend the network and improve the programming techniques described here.

Table 1 and 2 are filled in Excel spreadsheet by observing the impedance and power in network branches. For admittance matrix in Table 3, Excel VBA ‘shift factor’ macro program [9] is used.

Note the reduction of DC power flow Ybus (4x4) matrix dimension to B (3x3) is necessary for power flow equations. The reasons being that the Ybus (4x4) matrix is not a full rank matrix and its inverse is not possible. By removing first row and column of Ybus and considering it as ‘0’ reference, full rank condition it fulfilled for inversion. SF [7] for all five branches with increase in injection at bus 2 and decrease injection at bus 3 is gives as:

\[ \begin{bmatrix} s_{12} & s_{13} & s_{14} & s_{15} \\ s_{22} & s_{23} & s_{24} & s_{25} \\ s_{32} & s_{33} & s_{34} & s_{35} \end{bmatrix} = (D \times A)[B]^{-1} \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} \]

\(D_{5x5}\) is a matrix, which has diagonal elements of susceptance in Table 1, and zeros non-diagonal element. Matrix \(A_{3x3}\) is incidence arc matrix given in Table 2 range (“Q8:S12”). Which shows power entering the bus as negative and leaving bus (node) as positive, excluding the reference bus 1. Matrix \(B_{3x3}\) is admittance matrix given in Table 3 range (“D17:F19”). Column matrix at right is \(P_{gi} = P_{ij}\) input in Figure 3 range (“O33:O36”). Resulting Shift factor output (SFxP_{inj}) shown in Table 4 Range (“L21:L25”) [9]. Appendix I shows the ‘shift factor’ macro flow chart.

### A. Long economic dispatch and transmission limit for case

Once the SF is determined and with given cost \(C_i\), supply \(P_{gi}\), demands \(P_{lj}\) and limits \(P_{gi}^{min}\)limits in Figure 3, the ED problem can be easily solved with Excel solver in Figure 2. Transmission limit is a constraint of transmission branch as shown in Table 4. Since the limits (\(T_{min}, T_{max}\)) exceeds the SF, they are not binding. Binding multiple branches for ED problem reduces the flow and change the economic cost. Button in Figure 2 is used to activate macro ‘shift_factor_solver’ for SF evaluation. Appendix I shows ‘shift_factor_solver’ macro flow chart.

### Table 1. Branch impedance and admittance

<table>
<thead>
<tr>
<th>R</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>No. Bus = 4</td>
<td>No. Branch = 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>From</td>
<td>To</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>0.1j</td>
<td>-10j</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>3</td>
<td>0.1j</td>
<td>-10j</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>4</td>
<td>0.1j</td>
<td>-10j</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>3</td>
<td>0.1j</td>
<td>-10j</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>4</td>
<td>0.1j</td>
<td>-10j</td>
</tr>
</tbody>
</table>

### Table 2. Admittance matrix jB with Macro

<table>
<thead>
<tr>
<th>R</th>
<th>C</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Node Branch</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1 to 4</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1 to 2</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2 to 3</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4 to 3</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1 to 2</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3. Admittance matrix jB with Macro

<table>
<thead>
<tr>
<th>R</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>20i</td>
<td>-10i</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>-10i</td>
<td>30i</td>
<td>-10i</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>-10i</td>
<td>20i</td>
<td></td>
</tr>
</tbody>
</table>

### B. Economic dispatch and transmission limit general case

Solution of general ED problem equation (1) for network in Figure 1 is shown in Figure 4. The \(P_{gi}\) and \(P_{lj}\) inputs are shown in Figure 4. Demand at buses 2 and 3 is 70MW and 100MW respectively. Generation at buses 1, 2 and 4 are evaluated by optimizer to solve on base of lowest cost and generation \(P_{gi}^{min}\) and \(P_{gi}^{max}\) limits. Due to lowest cost \(C2\), cell (“N34”) shows that after 70MW demand at bus 2, excess of 80MW (150MW = \(P_{g2}^{max}\)) is produced for bus 3. The solver minimizes total cost cell(“F36”). In Table 5, the new SF is shown for general ED case with no binding limit constrains.

By binding the \(T_{max}\) for branch (1 to 2) 2 and (1 to 3) 3 from \(T_{max}=100\) to \(T_{max}=50\) and 30 respectively total reduction will be 18.750+37.50 = 56 MW. The ‘shift_factor_solver’ is invoked again to update SF to new limits in Table 6. Button in cell (“A32:B35”) is to activate macro ‘shift_factor_solver’ for ED evaluation.

### Table 5. Binding SF with transmission limits

<table>
<thead>
<tr>
<th>T_{min}</th>
<th>Branch</th>
<th>SFxP_{inj}</th>
<th>T_{max}</th>
</tr>
</thead>
<tbody>
<tr>
<td>-10</td>
<td>&lt;= 1 to 4</td>
<td>33.750</td>
<td>&lt;= 100</td>
</tr>
<tr>
<td>-10</td>
<td>&lt;= 1 to 2</td>
<td>50.000</td>
<td>&lt;= 50</td>
</tr>
<tr>
<td>-10</td>
<td>&lt;= 2 to 3</td>
<td>-1.250</td>
<td>&lt;= 100</td>
</tr>
<tr>
<td>-10</td>
<td>&lt;= 4 to 3</td>
<td>33.750</td>
<td>&lt;= 100</td>
</tr>
<tr>
<td>-10</td>
<td>&lt;= 1 to 3</td>
<td>30.000</td>
<td>&lt;= 30</td>
</tr>
</tbody>
</table>
Least square inverse power injection

Binding SF in Table 6 can be solved to determine the $P_{\text{inj}}$. However, matrix $q(5\times3)$ in equation (7) is non square matrix and cannot be inverted. Using pseudoinverse method, we can compute a "best fit" (least squares) solution for new $P^*_{\text{inj}}$.

Table 6. SF and transmission line flow limits for general case

<table>
<thead>
<tr>
<th>R</th>
<th>C</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>-10</td>
<td>&lt;= 1</td>
<td>to</td>
<td>4</td>
<td>33.75 &lt;= 100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>-10</td>
<td>&lt;= 1</td>
<td>to</td>
<td>2</td>
<td>68.75 &lt;= 100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>-10</td>
<td>&lt;= 2</td>
<td>to</td>
<td>3</td>
<td>-1.25 &lt;= 100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>-10</td>
<td>&lt;= 4</td>
<td>to</td>
<td>3</td>
<td>33.75 &lt;= 100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>-10</td>
<td>&lt;= 1</td>
<td>to</td>
<td>3</td>
<td>67.5 &lt;= 100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(6) $SF^* = (P \times A)[E']^{-1} \cdot P^*_{\text{inj}}$

(7) $P^*_{\text{inj}} = [q'q]^{-1}q'b$

(8) $q = (P \times A)[E']^{-1}$

(9) $b = SF^*$

New $P^*_{\text{inj}}$ is evaluated using equation (7) in Figure 5 Range (“N41:N44”) for transmission binding constraint as shown in Table 6. As expected, total supply is reduced by 170-56=114MW to bus 2 and 3. The solver again assign the lowest $P_{\text{sc}}$ cost of production. Constrained optimization in Figure 5 is invoked by button in Cell("A42:B44") and Macro ‘$P_{\text{inj}}$ Limit_const’. Appendix I shows the flow chart of later macro.

Excel VBA Macro

Binding VBA Macro is very useful tool for Excel spread sheet programming. Like all other programming language, it automates the coding for repetitive programming and keep visual spread sheet clean and tidy. It is enabled in Excel from file-> options-> Add-ins ‘Analysis ToolPak – VBA’ https://www.excel-easy.com/vba/create-a-macro.html. After installing in Excel 2019 a ‘developer’ tab is added in the main ribbon. Macro prompt is visible under this tab. If files from [9] is downloaded and opened, then Macro developed for this paper will be visible and can be opened with ‘Edit’ option.

Excel VBA Macro

Binding VBA Macro is very useful tool for Excel spread sheet programming. Like all other programming language, it automates the coding for repetitive programming and keep visual spread sheet clean and tidy. It is enabled in Excel from file-> options-> Add-ins ‘Analysis ToolPak – VBA’ https://www.excel-easy.com/vba/create-a-macro.html. After installing in Excel 2019 a ‘developer’ tab is added in the main ribbon. Macro prompt is visible under this tab. If files from [9] is downloaded and opened, then Macro developed for this paper will be visible and can be opened with ‘Edit’ option.

Excel VBA Macro

Binding VBA Macro is very useful tool for Excel spread sheet programming. Like all other programming language, it automates the coding for repetitive programming and keep visual spread sheet clean and tidy. It is enabled in Excel from file-> options-> Add-ins ‘Analysis ToolPak – VBA’ https://www.excel-easy.com/vba/create-a-macro.html. After installing in Excel 2019 a ‘developer’ tab is added in the main ribbon. Macro prompt is visible under this tab. If files from [9] is downloaded and opened, then Macro developed for this paper will be visible and can be opened with ‘Edit’ option.

It is not possible to discuss all aspect of VBA programming here; https://www.excel-easy.com/vba.html but some basic interaction between spread sheet and Macro will be discussed for benefit of beginners. Data such as ‘No. Branch’ in Table 1 can be read in macro from spreadsheet using bus = Range("D5") or B = Range ("F8:F12") to read susceptance in Table 1.

To write from Macro to spread sheet, Range("L20").Offset(i,0).Value = Ans(i,1) will write SF to Table 5 with iterative ‘i’ from 1 to 5_branches. The ‘Offset(i,0)’ start writing from i=1, at Cell("L21") and ‘0’ column offset until i=5 or Cell("L25"). To learn further about macros used in this application, use ‘Breakpoint’ command in debugging mode and use ‘F8’ step into command and observe changes in spread sheet at each step. https://www.excel-easy.com/vba/examples/debugging.html.

ED minimization solver

VBA Solver can be invoked from Excel after installation from file-> options-> Add-ins ‘Solver Add-In’ or writing
directly in macro as in this paper. Excel solver is used in two macros ‘shift_factor_solver’ and ‘Pinj_Limit_const’. All the solver parameters are to be described such as SetCell:="$F$36",MaxMinVal:=2, ByChange:= "$D$33: $D$35, $L$21:$L$25,$N$33:$N$36", Engine:=1, EngineDesc:="GRG Nonlinear" followed by the constraints in equation (2) - equation (4) which are given in Table 4 Cell("I21:I25") and Cell("F21:F25"). SetCell:="$F$36"=SUM(F33:F35) is the cost minimization cell eq(1a) in Figure 3-4, MaxMinVal:=2 is to set the solver to minimum value by changing variables $D$33:$D$35, $L$21 :$L$25, $N$33 : $N$36, in Figure 3 - 4 and Table 4. Engine:=1, EngineDesc:= "GRG Nonlinear" is used to select Generalized Reduced Gradient method, in which solver determines the slope of objective function at every step change of ‘ByChange’ variables. Appendix I describe the flow chart of solver used in this paper.

Load profile simulation (Sheet 2)

To demonstrate Excel Macro programming for load profile to determine voltages and steady state power flow in network Figure 6. Here similar point load power network calculations [4] are performed repeatedly over one day load profile data with 15min interval. Load data (p.u) is injected to PQ bus Pp3 in cell (F11:G11) with other data as shown in Figure 6. Power flow in network expressing $P_{ij}$ real and $Q_{ij}$ reactive power is given as:

\[
P_{ij} = \sum_{j=1}^{n} \left| U_i \right| \left| U_j \right| \left( G_{ij} \cos(\theta_i-\theta_j) + B_{ij} \sin(\theta_i-\theta_j) \right)
\]

\[
Q_{ij} = \sum_{j=1}^{n} \left| U_i \right| \left| U_j \right| \left( G_{ij} \cos(\theta_i-\theta_j) + B_{ij} \sin(\theta_i-\theta_j) \right)
\]

Load profile data $P_{p3}$, swing bus $P_{g1}$ real and $Q_{g1}$ reactive power plots are shown in Figure 11. Appendix II shows the flow chart of load profile ‘gridsim’ macro.

![Fig.6. Load profile power flow 1 network](image)

![Fig.7. Power flow data simulation input and output table for four bus and five branch](image)

Indexes $i$ and $j$ are; from and to nodes notations and $t$ is the time stamp of profile load data $P_{t,i}$ at bus 3 with increment of $\Delta t=15$ min for total $T=48$ steps. Other power and load parameters except $P_{t,i}$ are kept constant as shown in Figure 7. Empty fields are the variables of network which are evaluated by the solver. Conductance matrix $B_{ij}$ is given at Cell("G16:J19") and susceptance matrix $G_{ij}$ is given in at Cell("B16:E19") in sheet 2. $P_{t,i}$ injection is shown in Cell ("F11:G11"). Load profile data is stored in Cell("B37:C85") and transferred to cell ("F11:G11") at each time stamp. The Solver is run at each time stamp and table in Figure 6 is populated with buses power flow data which are transferred to Cell("D37:I68") before new data is transferred and whole cycle is repeated until the load data ends at 48th step at Cell("B85:C85").

GRG solver is programmed in macro ‘gridsim’ and iteratively solved at each $T=48$ time stamps of load profile data. Power flow expressions in equation (10) and equation (11) are entered in Figure 8 - 9, Cell(C23:C26) for $P_{t=1...4}$ and Cell(G23:G26) for $Q_{t=1...4}$. In Figure 8 - 9 error difference $\Delta P_{t,i}$ between injected power and computed power in each bus and similarly reactive power differences $\Delta Q_{t,i}$ are given.

\[
\Delta P_{t,i} = P_{t,i} - P_{t,i}^*
\]

\[
\Delta Q_{t,i} = Q_{t,i} - Q_{t,i}^*
\]

Solver GRG will solve the power flow equations (12) and (13) according to load profile input $P_{t,p3}$ at bus 3 by balancing $P_{t,i} = P_{t,i}^*$ and $Q_{t,i} = Q_{t,i}^*$ and making $\Delta P_{t,i} = \Delta Q_{t,i} = 0$. $P_{t,i}^*$ and $Q_{t,i}^*$ are current computed values by the solver. Equations (12) and (13) can be represented with Cell(C29:C32)= Cell(D9:D12)-Cell(F9:F12)-Cell(C23:C26) and Cell(G29:G32)= Cell(E9:E12)-Cell(F9:F12)-Cell(G23:G26) respectively. The objective function to minimize errors in equations (12) and (13) is shown Figure 9 cell("J24") as:

\[
\text{error} = \sqrt{\sum_{i=1}^{4} \Delta P_{t,i}^2 + \sum_{i=1}^{4} \Delta Q_{t,i}^2}
\]
Fig.11. Load profile and voltages at load and generated buses for four bus and 5 branch network shown in Figure 6.

A. Load profile simulation 2

Five bus and seven branch power networks shown in Figure 12 is simulated for the same load as in sec.7.0. All baseMW=100MW, baseMVA =100MVA and baseMVAr =100KVA values are considered at V = 1 p.u. for simulation purpose.

Profile load injection is made at PQ bus P_{P4} cell(F12:G12) in Figure 12. Other generations and loads are kept at steady state as shown in Figure 13. Results of simulation with all traces of generation and load data is shown in Figure 14.

Fig.12. Five bus and seven branch network with load profile input at bus 4

Fig.13. Power flow data simulation input and output table for five bus and seven branch network

Fig.14. Load profile and voltages at generated and loads for five bus and seven branch network in Figure 12
Conclusion

The work presented in this paper demonstrate application of Excel macro to analyze ED problem and simulate load profile data with power flow equations. The work describe here is extension of work available in literature [1]–[4], [7], [9]. However due to conciseness many other important aspects of macros used in this work cannot be described. To encourage and support readers, all excel macro are shared in link [9]. Although the network demonstrated in this work are simple, but after going through the provided macros codes complex network can be analyzed and ED macro can be further expanded with additional code to include locational marginal pricing (LMP) and shadow prices.

Acknowledgement

The authors would like to thank the Ministry of Higher Education Malaysia (MOHE) for funding this research. The MOHE has fully funded this research through the Fundamental Research Grant Scheme (FRGS), with the project code: FRGS/1/2020/TK0/UNITEN/03/4.

REFERENCES