Power Systems

MATLAB Assignment

EEP-225

Line Transmissions Project

Author:
Omar Hany El-sayed
19016102
Omar Wael Mohamed
19016103
Amr Ayman Elmaghraby
19016107
Amr Mohamed Abdelhamed
19016117
Toka Mohamed Bishara
19015541
sherin shabaan Hammad
19015848
Eman Abd-ElFataah Mahmoud
19015497
Maya Emad Abass
190162367
Somaya Ibrahim Ishiba
19015795
Manar Abdelmnem Elmeslmany
19016700

Submitted to:
Dr. Hossam Kotb

0.1 Introduction

The project aims to calculate transmission line parameters Resistance, Inductance and Capacitance per phase.

First, The program asked User to input the conductor resistivity, conductor length, and conductor diameter. After that the User choose whether the transmission system is symmetrical or unsymmetrical then will choose Number of bundles per phase and calculate the parameters.

Second, Depends on the length given by the user in the First the program will choose if the model is short or medium then calculate **ABCD constants** of each model.

Third, The user will choose between 2 Cases. Case I, the program will draw a curve that relates the efficiency with the active power and another curve that relates the voltage regulation with the active power if the receiving end load is 0.8 pf lag with active power varying from 0 to 100 kw. Case II, the program will draw a curve that relates the efficiency with the power factor and another curve that relates the voltage regulation with the power factor if the receiving end power is 100 kw but with power factor varying from 0.3 lag to UPF and from 0.3 lead to UPF. The source code is written in **Matlab** and the program is equipped with a set of user input protection which is discussed in later sections.

0.2 Task 1

The code is taking from User the conductor resistivity, conductor length per Km, and conductor diameter per Cm. Then calculate Area and resistance as shown in fig.1

```
%%%Task 1%%%
clc;
close all;
%calculate Resistance
Re=input('Enter conductor resistivity :\n ');        %calc resistivity
len=input('Enter conductor length (Km) :\n ');       %Entering Length in Km

   % Making sure the length is between 0 and 250 km
    while len==0 || len > 250

        fprintf('\n[Invalid input]: Please renter your choice\n');
        len=input('Enter conductor length (Km) :\n ');

    end

Dcm=input('Enter conductor diameter (cm) :\n ');     %calc diameter in cm
Dm=Dcm/100;
A=pi*((Dm/2).^(2));                             %cross sectionalarea of theconductor in m2
R=(Re*len)/A;          %resistance dc
```

Figure 1: Code of Entered Parameters

Then the User will choose model is symmetrical or unsymmetrical then will choose Number of bundles per phase and calculate Inductance per phase and Capacitance per phase ,So Here we got Resistance, Inductance and Capacitance per phase as shown in fig.2.

```
while 1
    system=input('\n Select the desired system :\n\n -> 1  symmetrical \n -> 2  unsymmetrical\n');

    switch system

      case 1 %symmetrical

        while 1
            bundle=input('\n select number bundle per phase :\n ->1 one \n->2 two \n->3 Three \n->4 Four\n');

            switch bundle

            case 1
              %one bundle
              D1=input('\nEnter D of three phase line :\n ');
              r=Dm/2;
              GMR1=r*exp(-0.25);
              Lphase=(2*10^(-7))*log(D1/GMR1);
              Cphase=(2*pi*8.85*10^(-12))/(log(D1/r));
              break;

            case 2
              %Two bundle
              D2=input('\nEnter D of three phase line :\n');
              d=input('\nEnter d between bundles :\n');
              r=Dm/2;
              re=r*exp(-.25);
              GMR2=sqrt(d*re);
              Lphase=(2*10^(-7))*log(D2/GMR2);
              Cphase=(2*pi*8.85*10^(-12))/(log(D2/sqrt(d*r)));
              break;

            case 3
              %Three bundle

              D3=input('\nEnter D of three phase line :\n');
              d=input('\nEnter d between bundles :\n');
              r=Dm/2;
              re=r*exp(-.25);
              GMR3=nthroot(d*d*re,3);
              Lphase=(2*10^(-7))*log(D3/GMR3);
              Cphase=(2*pi*8.85*10^(-12))/(log(D3/nthroot(d*d*r,3)));
              break;
```

Figure 2: Code of Choosing Model

Here, there is an Example of our task as shown in fig.3 which is Example of symmetrical transmission system. As shown in fig.4 Example of unsymmetrical trasmission system.

```
Enter conductor resistivity :
 1.59e-8
Enter conductor length (m) :
 70000
Enter conductor diameter (cm) :
 2.48

 Select the desired system :

 -> 1  symmetrical
 -> 2  unsymmetrical
1

 select number bundle per phase :
 ->1 one
->2 two
->3 Three
->4 Four
1

 select number bundle per phase :
 ->1 one
->2 two
->3 Three
->4 Four
1

Enter D of three phase line :
 5

Lphase =

   1.2499e-06


Cphase =

   9.2685e-12
```

Figure 3: Sumple Run Of Symmetrical

```
Enter conductor resistivity :
 1.59e-8
Enter conductor length (m) :
 70000
Enter conductor diameter (cm) :
 1.25

 Select the desired system :

 -> 1  symmetrical
 -> 2  unsymmetrical
2

 select number bundle per phase :
 ->1 one
->2 two
->3 Three
->4 Four
1

Enter D1 of three phase line :
 2.5

Enter D2 of three phase line :
 2

Enter D3 of three phase line :
 4.5

Lphase =

   1.2726e-06


Cphase =

   9.0964e-12
```

Figure 4: Sumple Run Of Unsymmetrical

0.3 Task 2

After calculating the resistance, inductance and capacitance, the program will ask the user about the voltage frequency and calculate the total Z and Y values as shown in fig.5.

```
f=input('Enter frequancy (Hz) :\n ');

len = L *0.001; % converting the length into km
L_tot = Lphase * L ; %calculating the total inductance
C_tot = Cphase * L ; %calculating the total capacitance
Z = R + 1i*2*pi*f*L_tot;
Y = 1i*2*pi*f*C_tot;
```

Figure 5: Code of Entered frequency and Y, Z Calculations

Then the program will calculate the ABCD parameters based on the length entered before and if the transmission line is medium the program will ask whether it is T or π Model fig.6.

```
if len < 80
    fprintf('\n The Transmission line is Short Model\n');
    A= 1 ;
    B= Z ;
    C= 0 ;
    D= 1;

elseif 80 <= len && len<= 250
   while 1
     model=input('choose your model:\n (1)Pi model\n (2)T model\n ');
    switch model
        case 1
            fprintf('\n The Transmission line is pi Model\n');
            A= 1+ Z*Y./2;
            B= Z;
            C= Y.*(1+Z*Y./4);
            D= 1+ Z*Y./2;
           break;
        case 2
            fprintf('\n The Transmission line is T Model\n');
            A= 1+ Z*Y./2;
            B = Z.*(1+Z*Y./4);
            C = Y;
            D= 1+ Z*Y./2;
            break;
        otherwise
            fprintf('\n[Invalid input]: Please renter your choice\n');
```

Figure 6: Code of Choosing π or T Model

After that the program will print the type of transmission medium and the ABCD parameters as shown in fig.7.

```
Enter frequancy (Hz)  :
 50

 The Transmission line is Short Model

 Parameter A = 1.0000 + 0.0000i
 Parameter B = 2.3041 + 27.4867i
 Parameter C = 0.0000 + 0.0000i
 Parameter D = 1.0000 + 0.0000i
```

Figure 7: Command line user interface

0.4 Task 3

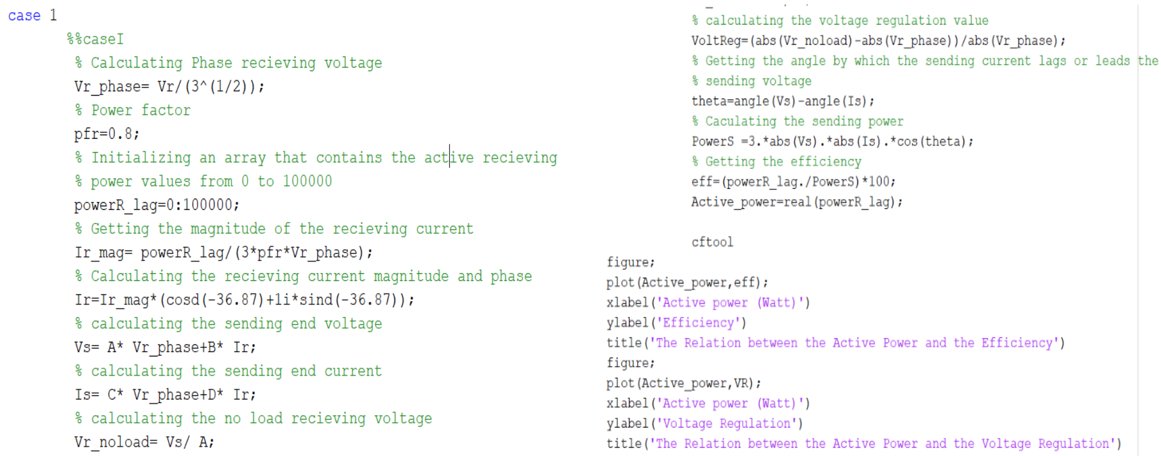
At this point the user will enter the receiving voltage, then choose the desired Case between Case 1 and Case 2 as shwon in fig.8.

```
%Task3
% Taking Recieving line voltage
Vr=input('\nEnter recieving line voltage(KV): \n');Vr=Vr*1000;
% Asking the user to select either Case I or Case II
selectedCase = input('choose your case:\n (1)Case I\n (2)Case II:\n ');
% Making sure the user selected a valid case
while selectedCase ~= 1 && selectedCase ~= 2
    selectedCase = input('[INVALID INPUT],choose your case:\n (1)Case I\n (2)Case II:\n  ');
end
```

Figure 8: Code for the user to enter Vr and the case number

0.4.1 Case 1

This case used to calculate all voltages and currents with power change over range 0 : 100 KW in model then plotting Efficiency with Power and Voltage Regulation with Power as shown in fig.9.

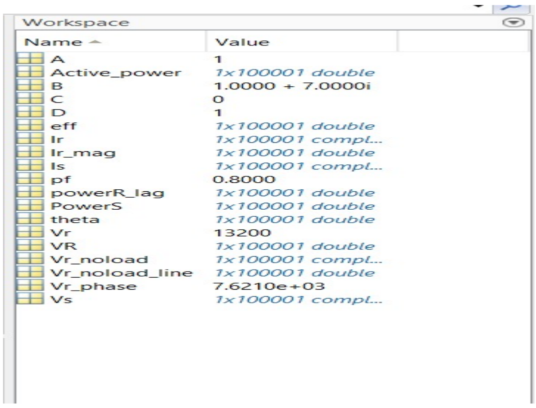


```
case 1
    %%caseI
    % Calculating Phase recieving voltage
    Vr_phase= Vr/(3^(1/2));
    % Power factor
    pfr=0.8;
    % Initializing an array that contains the active recieving
    % power values from 0 to 100000
    powerR_lag=0:100000;
    % Getting the magnitude of the recieving current
    Ir_mag= powerR_lag/(3*pfr*Vr_phase);
    % Calculating the recieving current magnitude and phase
    Ir=Ir_mag*(cosd(-36.87)+1i*sind(-36.87));
    % calculating the sending end voltage
    Vs= A* Vr_phase+B* Ir;
    % calculating the sending end current
    Is= C* Vr_phase+D* Ir;
    % calculating the no load recieving voltage
    Vr_noload= Vs/ A;
    % calculating the voltage regulation value
    VoltReg=(abs(Vr_noload)-abs(Vr_phase))/abs(Vr_phase);
    % Getting the angle by which the sending current lags or leads the
    % sending voltage
    theta=angle(Vs)-angle(Is);
    % Caculating the sending power
    PowerS =3.*abs(Vs).*abs(Is).*cos(theta);
    % Getting the efficiency
    eff=(powerR_lag./PowerS)*100;
    Active_power=real(powerR_lag);

    cftool
figure;
plot(Active_power,eff);
xlabel('Active power (Watt)')
ylabel('Efficiency')
title('The Relation between the Active Power and the Efficiency')
figure;
plot(Active_power,VR);
xlabel('Active power (Watt)')
ylabel('Voltage Regulation')
title('The Relation between the Active Power and the Voltage Regulation')
```

Figure 9: Code of case 1

Here, there is an Example of our task as shown in fig.10 and the output graphs are in fig.11,13,12,14.



Name	Value
A	1
Active_power	1x100001 double
B	1.0000 + 7.0000i
C	0
D	1
eff	1x100001 double
Ir	1x100001 compl...
Ir_mag	1x100001 double
Is	1x100001 compl...
pf	0.8000
powerR_lag	1x100001 double
PowerS	1x100001 double
theta	1x100001 double
Vr	13200
VR	1x100001 double
Vr_noload	1x100001 compl...
Vr_noload_line	1x100001 double
Vr_phase	7.6210e+03
Vs	1x100001 compl...

Figure 10: Working Space Of Case 1

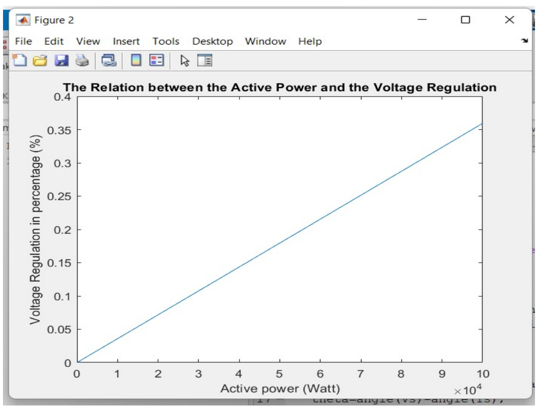


Figure 11: Plotted Graph of Active power and V.R

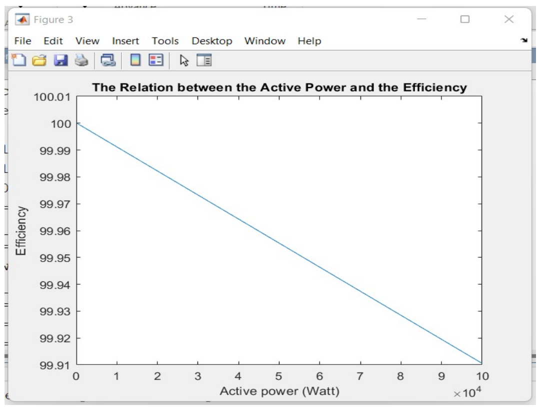


Figure 12: Plotted Graph of Active power and Efficiency

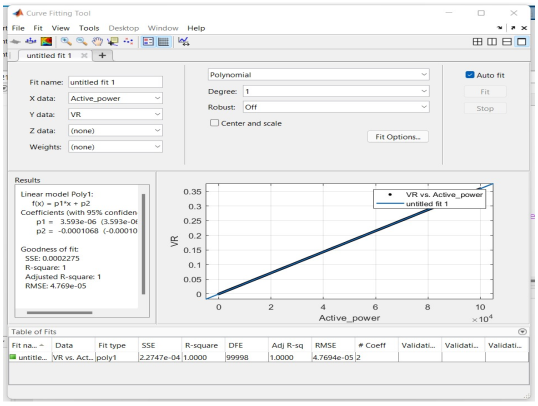


Figure 13: cf-tool Graph of Active power and V.R

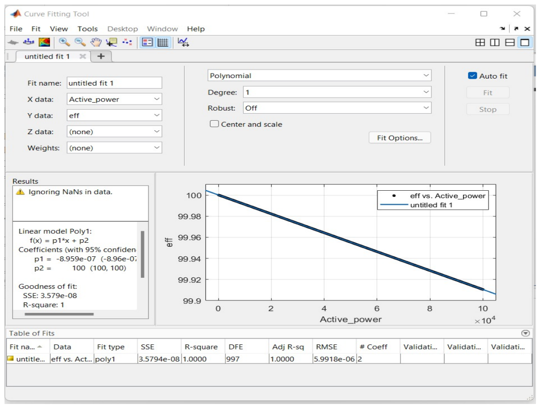


Figure 14: cf-tool Graph of Active power and Efficiency

0.4.2 Case 2

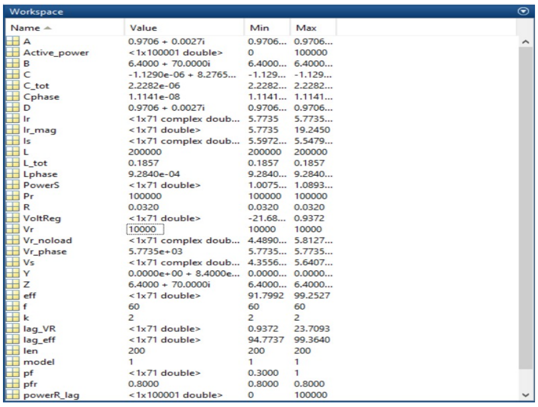
After User Enter V_R the receiving voltage, This case used to calculate all voltages and currents with power factor over range 0.3 : 1 leading and lagging as shown in fig.15.

```
case 2
    %case 2
      % Active power is constant
      Pr=100e3;
      % Power factor values in an array from 0.3 to 1
      pf=0.3:0.01:1;
      % Getting the phase recieving voltage
      Vr_phase= Vr/(3^(1/2));
      % getting the magnitude of the recieving end current
      Ir_mag = Pr./(3*pf*Vr_phase);
      %for lagging pf
      lag_VR = 0;
      lag_eff = 0;
      % solving for both lead and lag
      for k=1:2
          % Getting the recieving end current
          Ir=Ir_mag.*(cos((-1)^(k)*acos(pf))+1i*sin((-1)^(k)*acos(pf)));
          % Getting the sending end voltage
          Vs= A* Vr_phase+B.* Ir;
          % Getting the sending end current
          Is= C* Vr_phase+D.* Ir;
          % Getting the no load recieving end voltage
          Vr_noload= Vs./ A;
```

```
% getting the voltage regulation value
VoltReg=((abs(Vr_noload)-abs(Vr_phase))/abs(Vr_phase))*100;
% Getting the angle by which the sending current lags or leads the
% sending voltage
theta=angle(Vs)-angle(Is);
% Caculating the sending power
PowerS =3.*abs(Vs).*abs(Is).*cos(theta);
% Getting the efficiency
eff=(Pr./PowerS).*100;
% Plotting
cftool
figure;
subplot(2,1,1);
plot(pf,eff);
xlabel('Effeciency versus Power Factor');
subplot(2,1,2);
plot(pf,VoltReg);
xlabel('Voltage Regulation versus Power Factor');
if k==1
    suptitle('Lagging power factor');
    lag_VR = VoltReg;
    lag_eff = eff;
else
    suptitle('Leading power factor');
end
```

Figure 15: Code Of case 2

After After calculating the ABCD constants, if the receiving end power is 100 kW but with power factor varying from 0.3 lag to UPF and from 0.3 lead to UPF. The program will draw a curve that relates the efficiency with the power factor and another curve that relates the voltage regulation with the power factor as shown in fig.18, 19, 20, 21 and fig.17



Workspace

Name	Value	Min	Max
A	0.9706 + 0.0027i	0.9706...	0.9706...
Active_power	<1x100001 double>	0	100000
B	6.4000 + 70.0000i	6.4000...	6.4000...
C	-1.1290e-06 + 8.2765...	-1.129...	-1.129...
C_tot	2.2282e-06	2.2282...	2.2282...
Cphase	1.1141e-08	1.1141...	1.1141...
D	0.9706 + 0.0027i	0.9706...	0.9706...
Ir	<1x71 complex doub...	5.7735	5.7735...
Ir_mag	<1x71 double>	5.7735	19.2450
Is	<1x71 complex doub...	5.5972...	5.5479...
L	200000	200000	200000
L_tot	0.1857	0.1857	0.1857
Lphase	9.2840e-04	9.2840...	9.2840...
PowerS	<1x71 double>	1.0075...	1.0893...
Pr	100000	100000	100000
R	0.0320	0.0320	0.0320
VoltReg	<1x71 double>	-21.68...	0.9372
Vr	10000	10000	10000
Vr_noload	<1x71 complex doub...	4.4890...	5.8127...
Vr_phase	5.7735e+03	5.7735...	5.7735...
Vs	<1x71 complex doub...	4.3556...	5.6407...
Y	0.0000e+00 + 8.4000e...	0.0000...	0.0000...
Z	6.4000 + 70.0000i	6.4000...	6.4000...
eff	<1x71 double>	91.7992	99.2527
f	60	60	60
k	2	2	2
lag_VR	<1x71 double>	0.9372	23.7093
lag_eff	<1x71 double>	94.7737	99.3640
len	200	200	200
model	1	1	1
pf	<1x71 double>	0.3000	1
pfr	0.8000	0.8000	0.8000
powerR_lag	<1x100001 double>	0	100000

Figure 16: Working Space Of Case 2

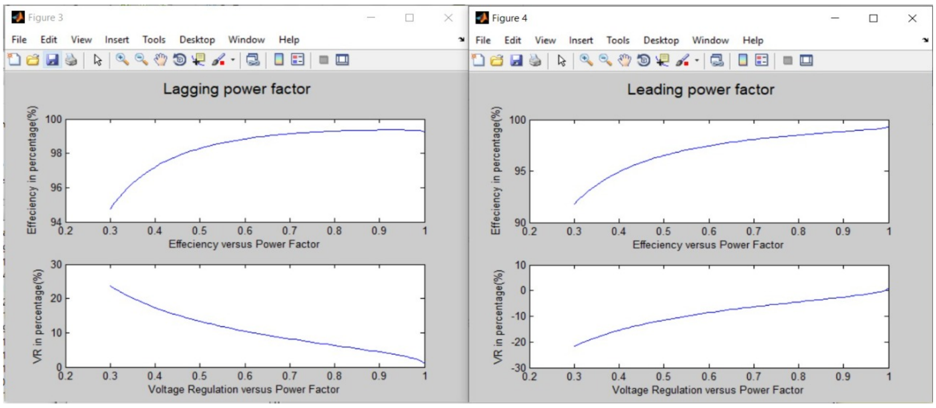


Figure 17: plotting graph for leading and lagging pf with $\mathrm{V_R}$ and $\mathrm{e_{ff}}$

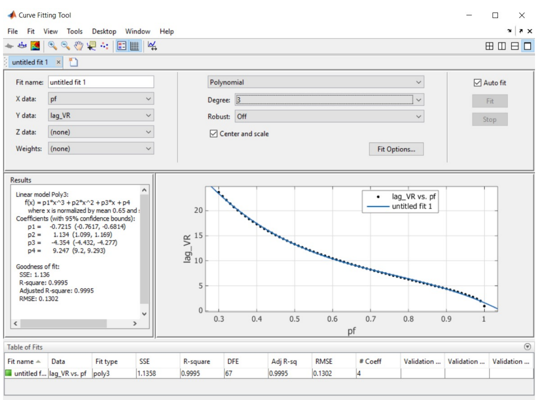


Figure 18: lag pf cf-tool Graph of $\mathrm{V_R}$ and $\mathrm{P_f}$

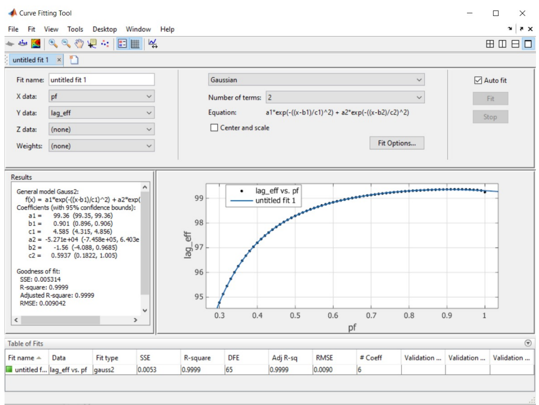


Figure 19: lag pf cf-tool Graph of Efficiency and pf

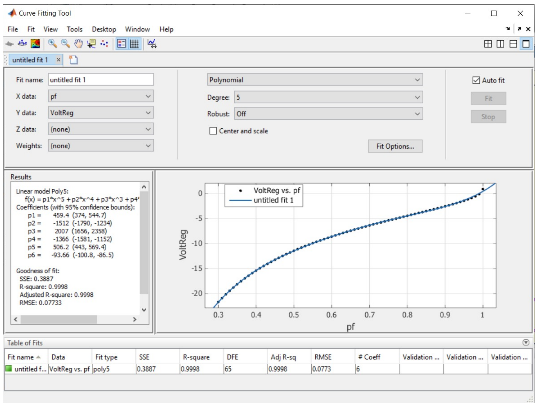


Figure 20: lead cf-tool Graph of V_R and P_f

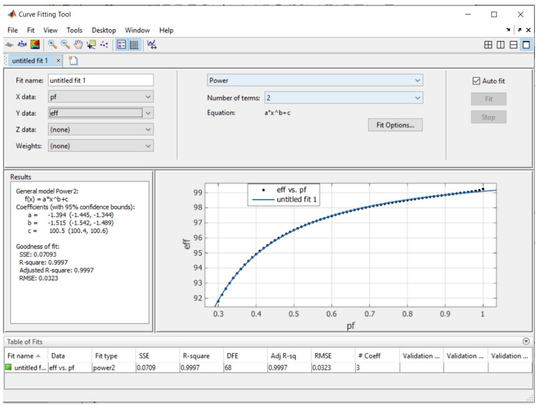


Figure 21: cf-tool Graph of Efficiency and P_f

0.5 User Input Protection

The program is designed to handle miss-entered data by the user such as a blank input line which can cause the program to collapse or generate garbage output.

0.5.1 Invalid Length

The program prevent the user to input Length is greater than 250 Km so, the model of transmission will be short or medium as shown in fig.22

```
% Making sure the length is between 0 and 250 km
while len==0 || len > 250

    fprintf('\n[Invalid input]: Please renter your choice\n');
    len=input('Enter conductor length (Km) :\n ');

end
```

Figure 22: Check of Length

0.5.2 Number Of Bundles Not Available

The User may Enter Number of bundles is grater than 4 or Negative Number whether in symmetrical and Unsymmetrical so, That section prevent that problem as shown in fig.23. Also, That code is putting for not make User choosing Number Non equal to 1 or 2 which assigned to Symmetrical and Unsymmetrical.

```
otherwise
 fprintf('\n[Invalid input]: Please renter your choice\n');
 continue;
```

Figure 23: Check of Number Bundles

0.5.3 Invalid Selected Case

In Task3 The User must chosse between case i or case ii so, That make sure User input 1 fot case i or 2 for case ii and if choose another number , code will print invalid input untill enter valid number of selected case as shown in fig.24

```
% Making sure the user selected a valid case
while selectedCase ~= 1 && selectedCase ~= 2
    selectedCase = input('[INVALID INPUT],choose your case:\n (1)Case I\n (2)Case II:\n  ');
end
```

Figure 24: Check of Selected case