RFVLSI Lab

Tutorial – Performing Model Fitting and Circuit Optimization by CMAES and LSQNONLIN Functions

Author：廖以諾

Outline

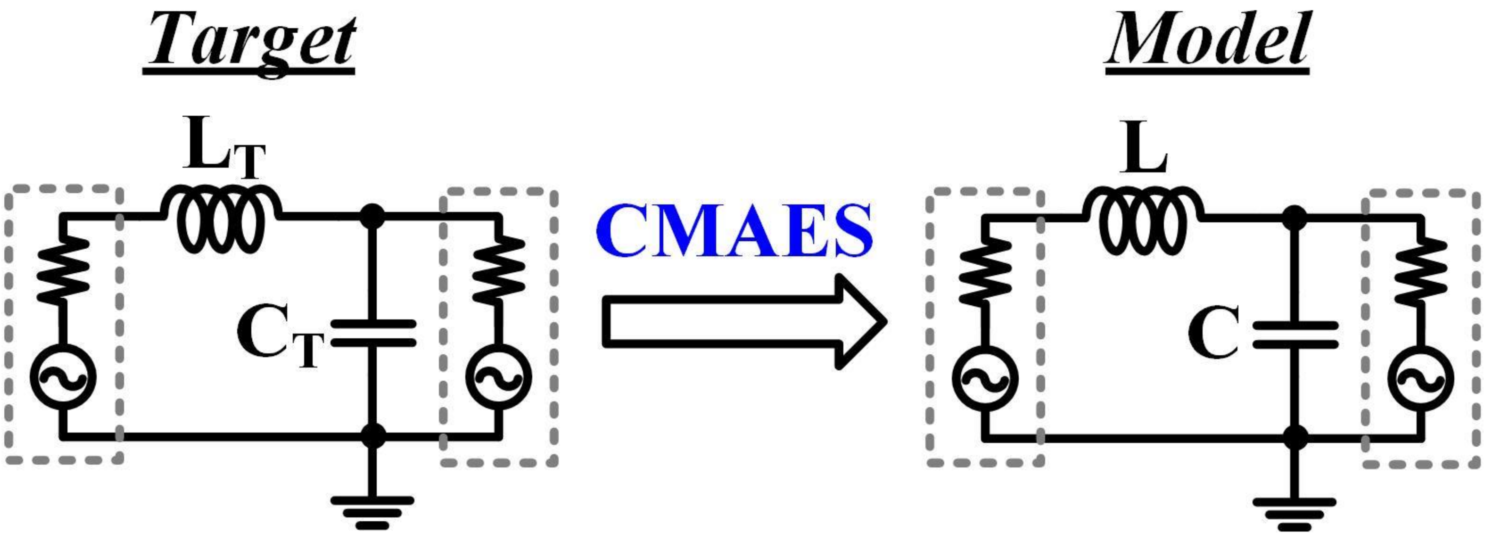
- **Introduction**
  - **What are CMAES and LSQNONLIN?**
  - **How does optimizer help?**
- **Example**
  - **CMAES - LC Network Model Fitting**
  - **CMAES - LCL Network Model Fitting**
  - **CMAES - Supplement**
- **Different Optimizers**
- **Performance Comparison**

Introduction

- **What is CMAES?**
  - **CMAES stands for Covariance Matrix Adaptation Evolution Strategy**
  - **It solves numerical optimization of non-linear or non-convex continuous optimization problems**
- **What is LSQNONLIN?**
  - **LSQNONLIN solves nonlinear least-squares problems**
- **How does these optimizers help?**
  - **In circuit design field, CMAES can be used in**
    - **Model Fitting**
    - **Circuit Optimization**

CMAES – LC Network Model Fitting

- **Target is the real network needed to be fitted**
  - Our *Target* has $L_T = 100\ pH$ and $C_T = 250\ fF$
- **By means of CMAES to fit out target**
  - The main goal is to fit the s parameters (SP Simulation)

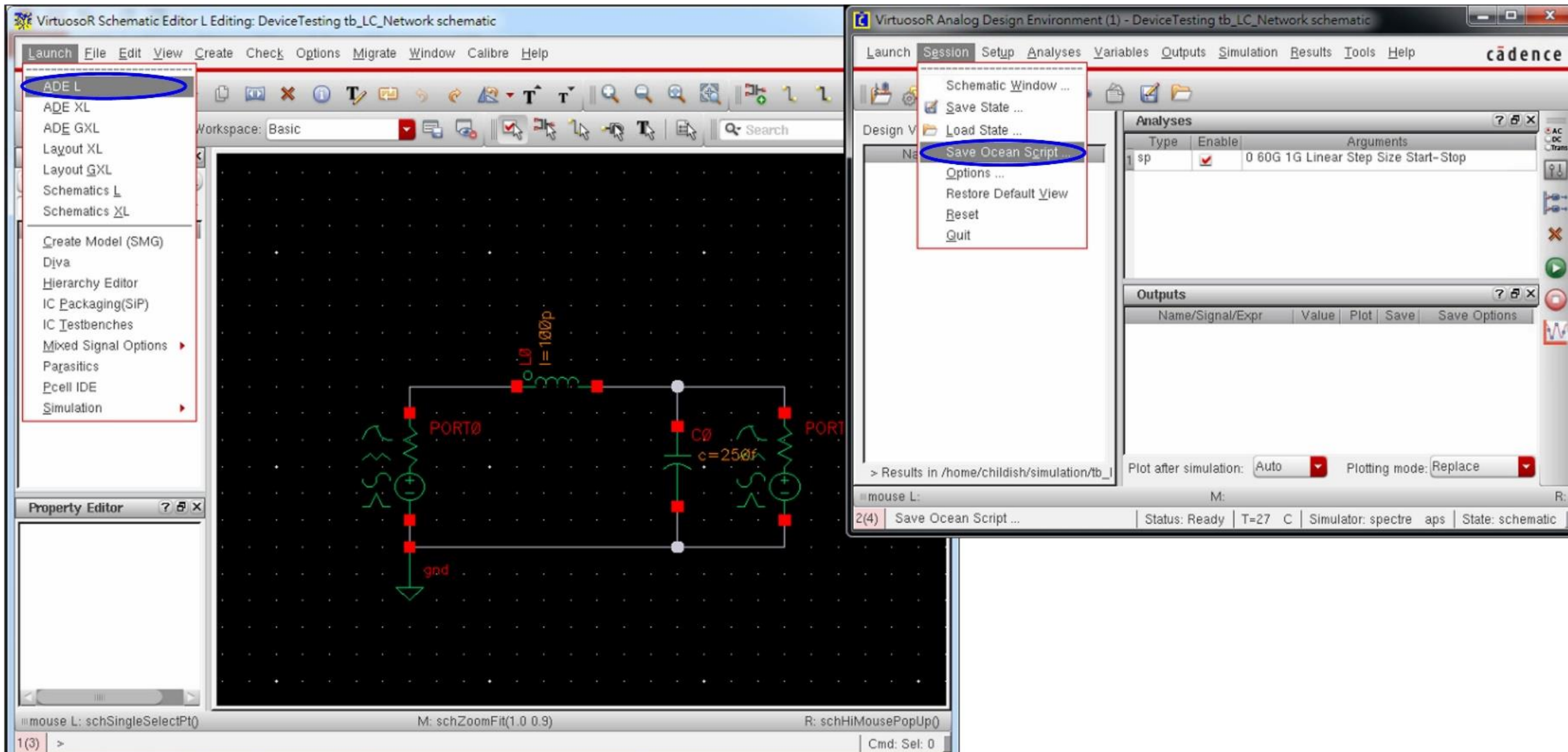


CMAES – LC Network Model Fitting

- **Several programs are required in the model fitting flow**
- **Two Ocean Scripts**
  - **1. Target (Named as *schematic.ocn*)**
    - ***schematic.ocn* outputs the target sp data called *schematic_sp.txt***
  - **2. Model (Names as *schematic_Fit.ocn*)**
    - ***schematic_Fit.ocn* outputs the fitting sp data called *schematic_Fit_sp.txt***
- **Matlab Program**
  - **Start SpectreRF simulation on both *schematic.ocn* and *schematic_Fit.ocn***
  - **Calculate the difference between two sp data**
    - **The difference is called *"Result"***
  - **CMAES minimizes *"Result"***
  - **Once the task is done, the program returns the fitting values**

CMAES – LC Network Model Fitting

- **Step 1: Set *Target* Ocean scripts**
  - **Create schematic ⇒ Click *ADE* ⇒ Setting simulation environment ⇒ Click *Session* ⇒ Click *Save Ocean Script***



CMAES – LC Network Model Fitting

- **Step 1: Set *Target* Ocean scripts *(Cont,)***
  - **Add instructions for printing sp data to *schematic_sp.txt***

Target* Ocean script *schematic.ocn

E:\Graduate_School\RF_VLSI_Lab\RFID_T65\RFID_v2\Designs\Ocean_Scripts\tb_LC_Network\schematic.ocn - Notepad++

```
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_rfind")
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_rfmvar_25")
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_na25")
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_lvt")
)
analysis('sp ?ports list("/PORT0" "/PORT1")  ?start "0"  ?stop "60G"
        ?step "1G"  )
envOption(
    'analysisOrder  list("sp")
)
temp( 27 )
run()

;;;;; Begin ;;;;;
printf("Begin   \n")
results()
selectResult( 'sp )
s11 = sp(1 1)
s12 = sp(1 2)
s21 = sp(2 1)
s22 = sp(2 2)
printf("Writing Files ...    \n")
OutputFile = outfile("/home/childish/OA65/Ocean_Scripts/tb_LC_Network/schematic_sp.txt" "w")
ocnPrint(?output OutputFile ?numberNotation 'none s11 s12 s21 s22)
close(OutputFile)
printf("End")
exit    ; Terminate Spectre Operation
;;;;; End ;;;;;
```

Print sp data to file named *schematic_sp.txt*

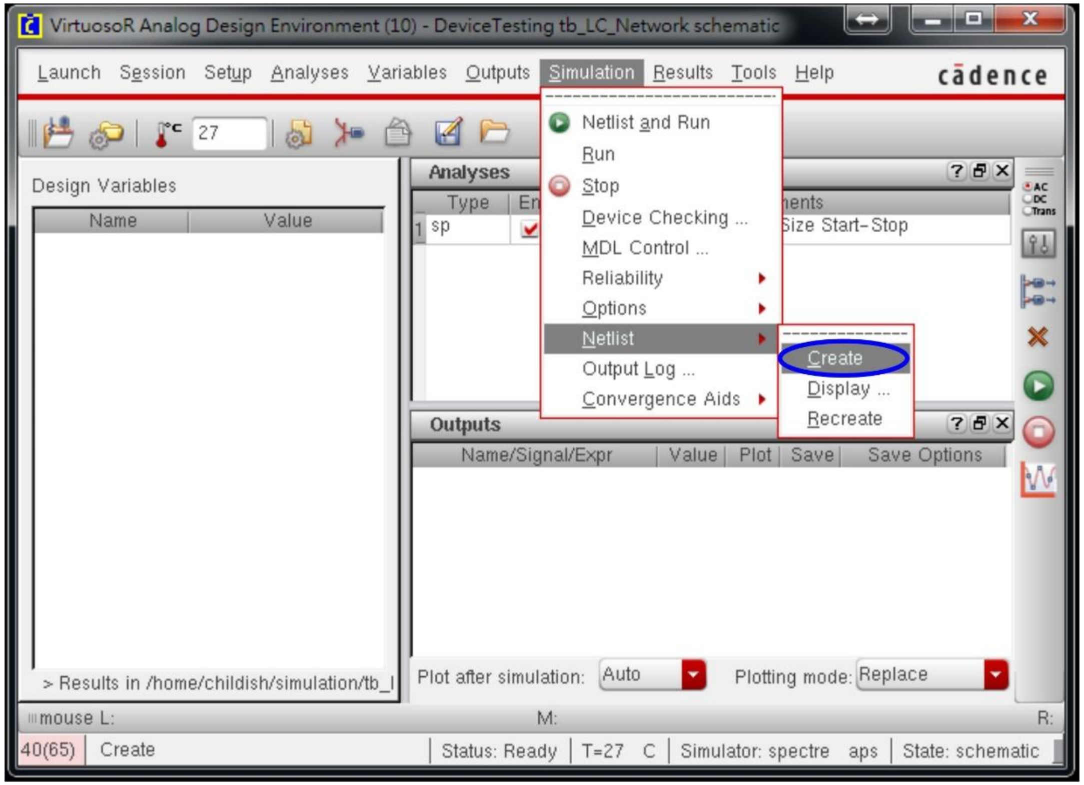
***"ocnPrint"* helps you print sp data to a specified file**

Exit SpectreRF

Normal text file | length : 5071 lines : 80 | Ln : 59 Col : 1 Sel : 0 | 0 | UNIX | ANSI as UTF-8 | INS

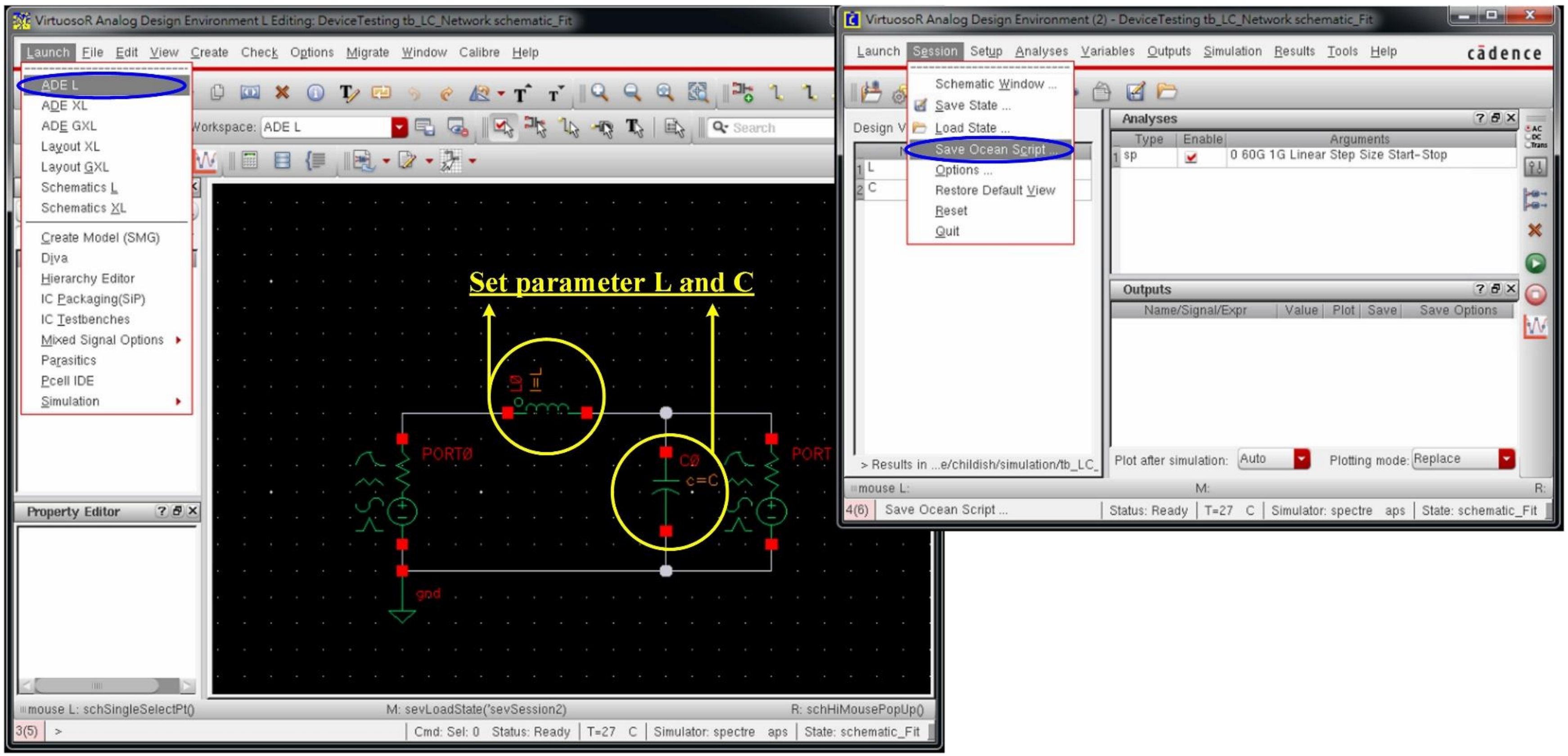
CMAES – LC Network Model Fitting

- **Step 2: Create *Target* Netlist**
  - **Schematic ⇒ Click *ADE* ⇒ Setting simulation environment ⇒ Click *Simulation* ⇒ Click *Netlist* ⇒ Click *Create***



CMAES – LC Network Model Fitting

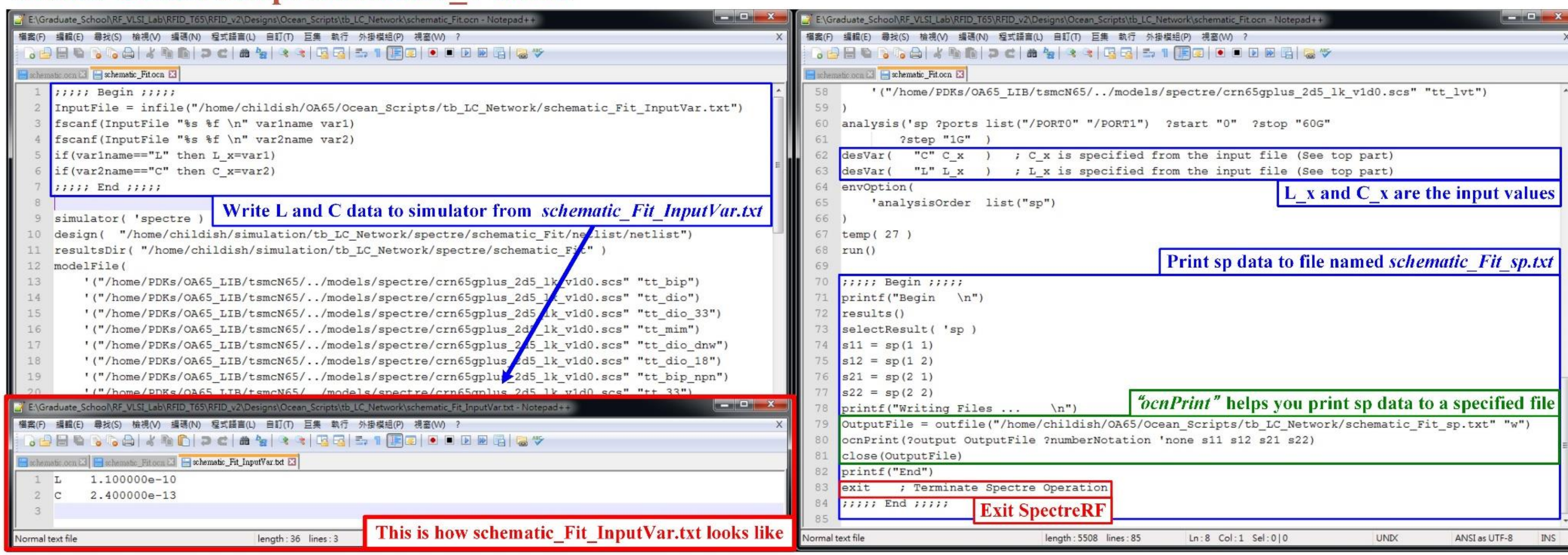
- **Step 3: Set *Model* Ocean scripts**
  - **Set parameters and save Ocean scripts**



CMAES – LC Network Model Fitting

- **Step 3: Set *Model* Ocean scripts *(Cont,)***
  - **Give input data from *schematic_Fit_InputVar.txt***
  - **Top Part ⇒ Access input data**
  - **Tail Part ⇒ Print sp data to *schematic_Fit_sp.txt***

Model* Ocean script *schematic_Fit.ocn




```
;;;;; Begin ;;;;;
InputFile = infile("/home/childish/OA65/Ocean_Scripts/tb_LC_Network/schematic_Fit_InputVar.txt")
fscanf(InputFile "%s %f \n" var1name var1)
fscanf(InputFile "%s %f \n" var2name var2)
if(var1name=="L" then L_x=var1)
if(var2name=="C" then C_x=var2)
;;;;; End ;;;;;

simulator( 'spectre )
design(  "/home/childish/simulation/tb_LC_Network/spectre/schematic_Fit/netlist/netlist")
resultsDir( "/home/childish/simulation/tb_LC_Network/spectre/schematic_Fit" )
modelFile(
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_bip")
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_dio")
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_dio_33")
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_mim")
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_dio_dnw")
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_dio_18")
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_bip_npn")
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_33")
```

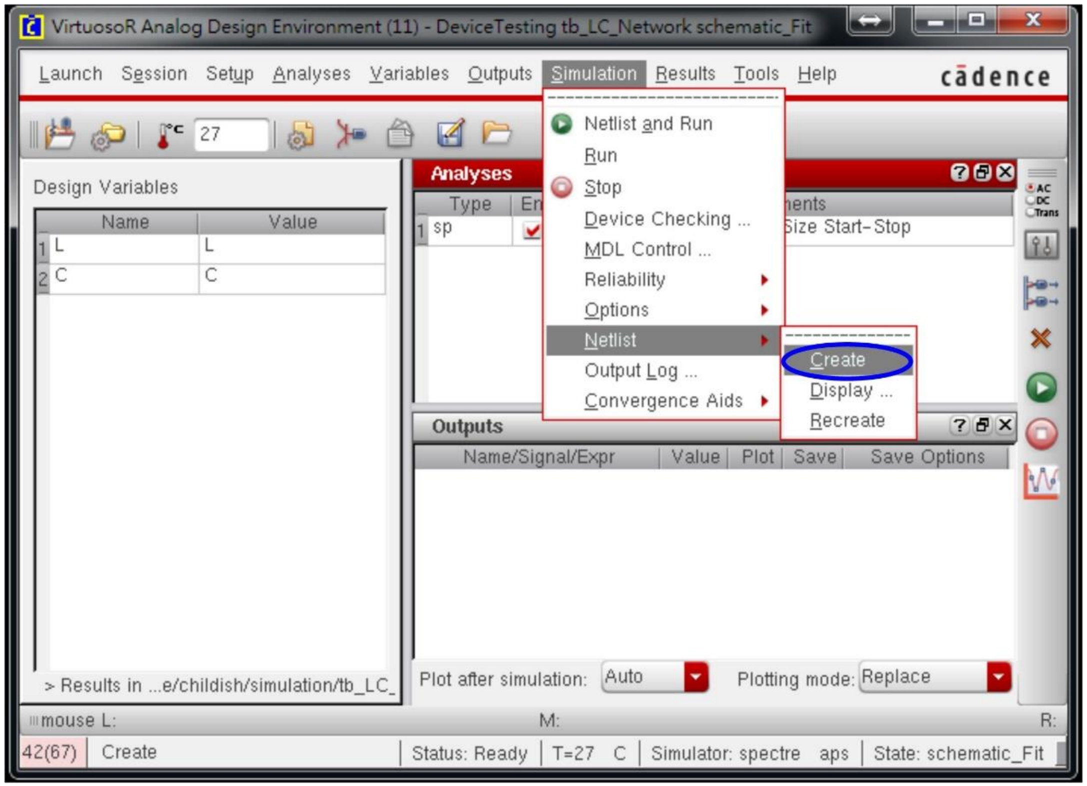
```
L     1.100000e-10
C     2.400000e-13
```

```
    '("/home/PDKs/OA65_LIB/tsmcN65/../models/spectre/crn65gplus_2d5_lk_v1d0.scs" "tt_lvt")
)
analysis('sp ?ports list("/PORT0" "/PORT1")  ?start "0"  ?stop "60G"
        ?step "1G"  )
desVar(   "C" C_x    )   ; C_x is specified from the input file (See top part)
desVar(   "L" L_x    )   ; L_x is specified from the input file (See top part)
envOption(
    'analysisOrder  list("sp")
)
temp( 27 )
run()

;;;;; Begin ;;;;;
printf("Begin    \n")
results()
selectResult( 'sp )
s11 = sp(1 1)
s12 = sp(1 2)
s21 = sp(2 1)
s22 = sp(2 2)
printf("Writing Files ...     \n")
OutputFile = outfile("/home/childish/OA65/Ocean_Scripts/tb_LC_Network/schematic_Fit_sp.txt" "w")
ocnPrint(?output OutputFile ?numberNotation 'none s11 s12 s21 s22)
close(OutputFile)
printf("End")
exit     ; Terminate Spectre Operation
;;;;; End ;;;;;
```

Top Part **Tail Part**

CMAES – LC Network Model Fitting

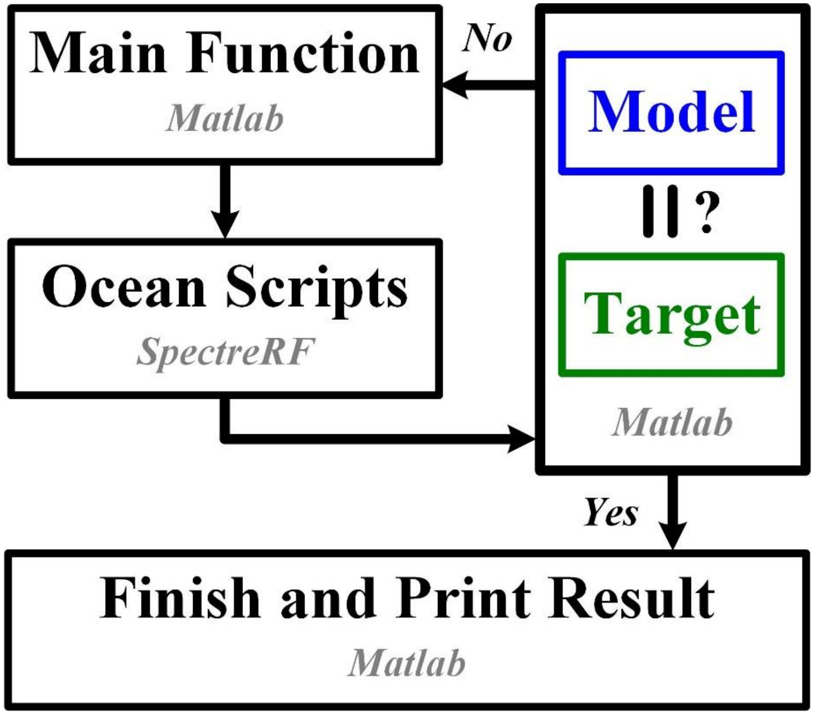
- **Step 4: Create *Model* Netlist**
  - **Schematic ⇒ Click *ADE* ⇒ Setting simulation environment ⇒ Click *Simulation* ⇒ Click *Netlist* ⇒ Click *Create***



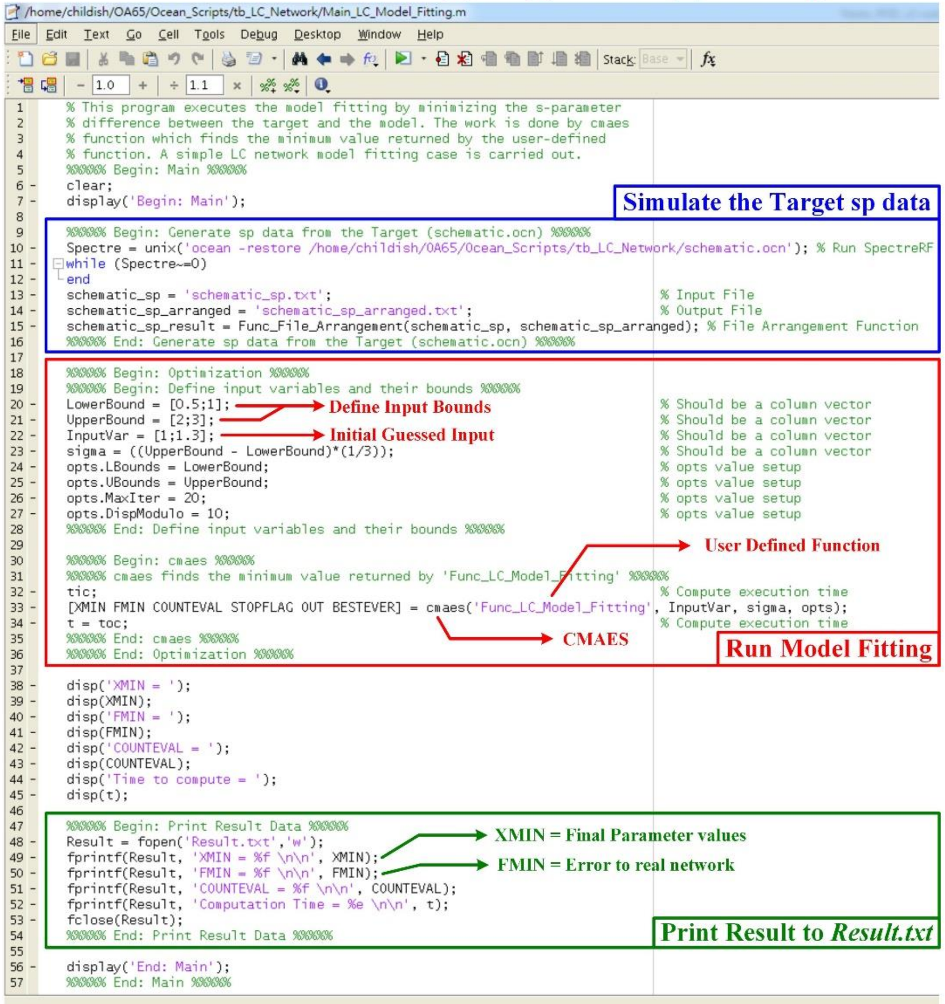
CMAES – LC Network Model Fitting

- **Step 5: Matlab Program**

Model Fitting Flow Chart



Main Function: *Main_LC_Model_Fitting.m*




```matlab
% This program executes the model fitting by minimizing the s-parameter
% difference between the target and the model. The work is done by cmaes
% function which finds the minimum value returned by the user-defined
% function. A simple LC network model fitting case is carried out.
%%%%% Begin: Main %%%%%
clear;
display('Begin: Main');

%%%%% Begin: Generate sp data from the Target (schematic.ocn) %%%%%
Spectre = unix('ocean -restore /home/childish/OA65/Ocean_Scripts/tb_LC_Network/schematic.ocn'); % Run SpectreRF
while (Spectre~=0)
end
schematic_sp = 'schematic_sp.txt';                        % Input File
schematic_sp_arranged = 'schematic_sp_arranged.txt';      % Output File
schematic_sp_result = Func_File_Arrangement(schematic_sp, schematic_sp_arranged); % File Arrangement Function
%%%%% End: Generate sp data from the Target (schematic.ocn) %%%%%

%%%%% Begin: Optimization %%%%%
%%%%% Begin: Define input variables and their bounds %%%%%
LowerBound = [0.5;1];                                     % Should be a column vector
UpperBound = [2;3];                                       % Should be a column vector
InputVar = [1;1.3];                                       % Should be a column vector
sigma = ((UpperBound - LowerBound)*(1/3));                % Should be a column vector
opts.LBounds = LowerBound;                                % opts value setup
opts.UBounds = UpperBound;                                % opts value setup
opts.MaxIter = 20;                                        % opts value setup
opts.DispModulo = 10;                                     % opts value setup
%%%%% End: Define input variables and their bounds %%%%%

%%%%% Begin: cmaes %%%%%
%%%%% cmaes finds the minimum value returned by 'Func_LC_Model_Fitting' %%%%%
tic;                                                      % Compute execution time
[XMIN FMIN COUNTEVAL STOPFLAG OUT BESTEVER] = cmaes('Func_LC_Model_Fitting', InputVar, sigma, opts);
t = toc;                                                  % Compute execution time
%%%%% End: cmaes %%%%%
%%%%% End: Optimization %%%%%

disp('XMIN = ');
disp(XMIN);
disp('FMIN = ');
disp(FMIN);
disp('COUNTEVAL = ');
disp(COUNTEVAL);
disp('Time to compute = ');
disp(t);

%%%%% Begin: Print Result Data %%%%%
Result = fopen('Result.txt','w');
fprintf(Result, 'XMIN = %f \n\n', XMIN);
fprintf(Result, 'FMIN = %f \n\n', FMIN);
fprintf(Result, 'COUNTEVAL = %f \n\n', COUNTEVAL);
fprintf(Result, 'Computation Time = %e \n\n', t);
fclose(Result);
%%%%% End: Print Result Data %%%%%

display('End: Main');
%%%%% End: Main %%%%%
```

CMAES – LC Network Model Fitting

Step 5: Matlab Program *(Cont,)*

Func_LC_Model_Fitting.m

/home/childish/OA65/Ocean_Scripts/tb_LC_Network/Func_LC_Model_Fitting.m

File Edit Text Go Cell Tools Debug Desktop Window Help

Stack: Base

```
% This function computes the difference between model and the tatget
% "Result" (Type: Number) = Difference between model and the target
% "InputVar" (Type: Array) = Input variables for the SpectreRF simulation
% "Target" (Type: Array) = Model S parameters
function Result = Func_LC_Model_Fitting(InputVar)

    %%%%% Begin: Arrange file and generate array (Target) %%%%%
    schematic_sp = 'schematic_sp.txt';                      % Input File
    schematic_sp_arranged = 'schematic_sp_arranged.txt';    % Output File
    schematic_sp_result = Func_File_Arrangement(schematic_sp, schematic_sp_arranged); % File Arrangement Function
    %%%%% End: Arrange file and generate array (Target) %%%%%

    %%%%% Begin: Assign input values to input file for SpectreRF simulation %%%%%
    L = round(InputVar(1)*10)/10*100*1e-12;                    % Unit: H
    C = round(InputVar(2)*10)/10*100*1e-15;                    % Unit: F
    schematic_Fit_InputVar = fopen('schematic_Fit_InputVar.txt', 'w+');
    if schematic_Fit_InputVar ~= -1
        fprintf(schematic_Fit_InputVar, 'L \t %e \n', L);     % Assign valus to input variables
        fprintf(schematic_Fit_InputVar, 'C \t %e \n', C);     % "%e" is the scientific notation
    end
    fclose(schematic_Fit_InputVar);
    %%%%% End: Assign input values to inpu

    %%%%% Begin: Generate sp data from the model fitting (schematic_Fit.ocn) %%%%%
    Spectre = unix('ocean -restore /home/childish/OA65/Ocean_Scripts/tb_LC_Network/schematic_Fit.ocn'); % Run SpectreRF
    while (Spectre~=0)
    end
    %%%%% Begin: Arrange file and generate array (Model Fitting) %%%%%
    schematic_Fit_sp = 'schematic_Fit_sp.txt';                      % Input File
    schematic_Fit_sp_arranged = 'schematic_Fit_sp_arranged.txt';    % Output File
    schematic_Fit_sp_result = Func_File_Arrangement(schematic_Fit_sp, schematic_Fit_sp_arranged); % File Arrangement Function
    %%%%% End: Arrange file and generate array (Model Fitting) %%
    %%%%% End: Generate sp data from the model fitting (schematic

    %%%%% Begin: sp data computation %%%%%
    Indicator = abs(schematic_sp_result - schematic_Fit_sp_result);   % Take positive value (Type: Array)
    Result = sum(Indicator(:));                                         % Sum all elements (Type: Number)
    %%%%% End: sp data computation %%%%%
    display(Result);

end
```

Turn Target sp data into an array

Print L and C input values to *schematic_Fit_InputVar.txt*

Simulate Model Fitting *schematic_Fit.ocn*

Get difference between target and model

This function compares the sp data difference between the real network and the model

The difference result is then returned back to CMAES

Then, CMAES generates another input values to this function

And so on…

CMAES – LC Network Model Fitting

- **Step 5: Matlab Program *(Cont,)***
  - **This function turns sp data into an array so that we can easily manipulate the data**

Func_File_Arrangement.m

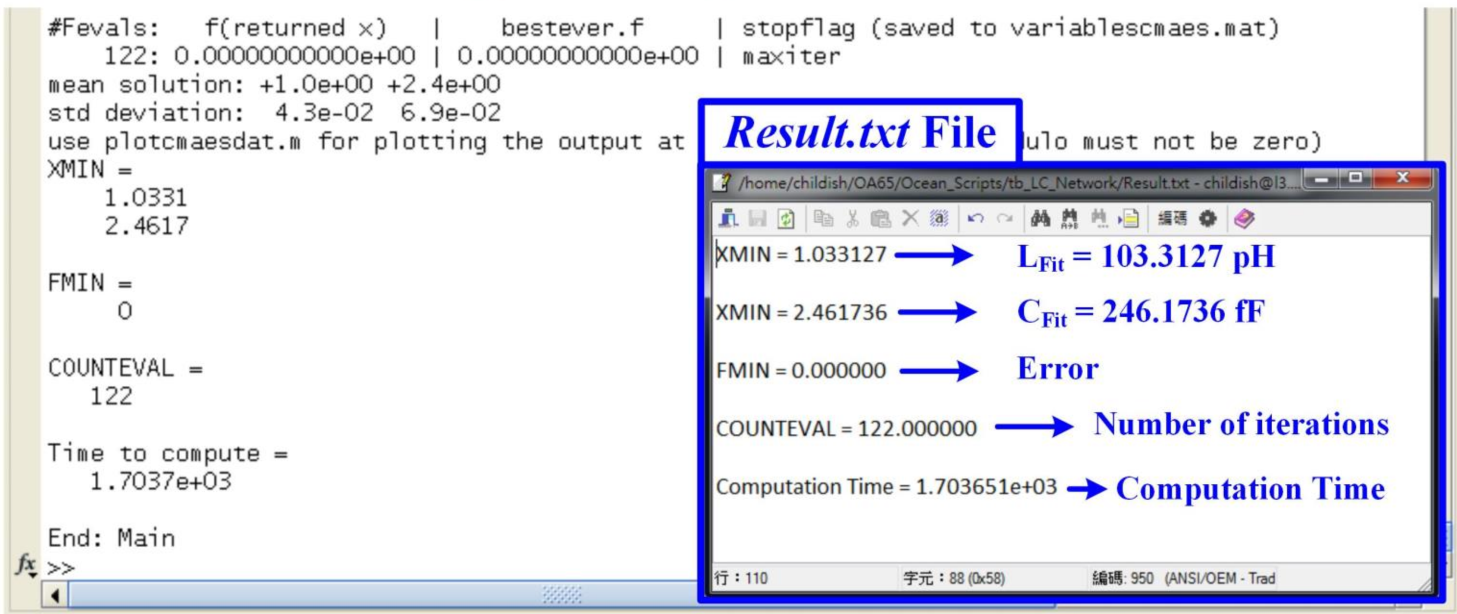
/home/childish/OA65/Ocean_Scripts/tb_LC_Network/Func_File_Arrangement.m

```matlab
% This function rearranges the sp file generated by SpectreRF to an array
% type file and generate the sp data in array "OutputArray"
% "OutputArray" (Type: Array) = Result data array
% "Input_String" (Type: String) = Name of the input file (Before rearranged)
% "Output_String" (Type: String) = Name of the output file
function OutputArray = Func_File_Arrangement(Input_String, Output_String)
    InputFile = fopen(Input_String);                % Open the orginal file from ocnPrint (Call it A)
    OutputFile = fopen(Output_String, 'w');         % Open a new file. Save new data to this file (Call it B)
    Count = 0;                                      % Top 3 lines in File A are not desired
    while ~feof(InputFile)                          % Check End of File
        Line = fgetl(InputFile);                    % Get line information
        Count = Count + 1;
        if Count > 4                                % Top 4 lines in File A are not desired
            fprintf(OutputFile, '%s\n', Line);      % Write data to File B
            continue;                               % Move on to the next line (while)
        end
    end
    fclose(InputFile);
    fclose(OutputFile);

    %%%%% Begin: Turn the sp file into an array type %%%%%
    Content = fileread(Output_String);
    Parsed = textscan(Content,'%f complex( %f , %f ) complex( %f , %f ) complex( %f , %f ) complex( %f , %f )');
    Parsed_1 = Parsed{1,1};                         % Frequency
    Parsed_2 = Parsed{1,2};                         % Real(S11)
    Parsed_3 = Parsed{1,3};                         % Imag(S11)
    Parsed_4 = Parsed{1,4};                         % Real(S12)
    Parsed_5 = Parsed{1,5};                         % Imag(S12)
    Parsed_6 = Parsed{1,6};                         % Real(S21)
    Parsed_7 = Parsed{1,7};                         % Imag(S21)
    Parsed_8 = Parsed{1,8};                         % Real(S22)
    Parsed_9 = Parsed{1,9};                         % Imag(S22)
    OutputArray = [Parsed_2+i*Parsed_3 Parsed_4+i*Parsed_5 Parsed_6+i*Parsed_7 Parsed_8+i*Parsed_9];
    %%%%% End: Turn the sp file into an array type %%%%%
end
```

CMAES – LC Network Model Fitting

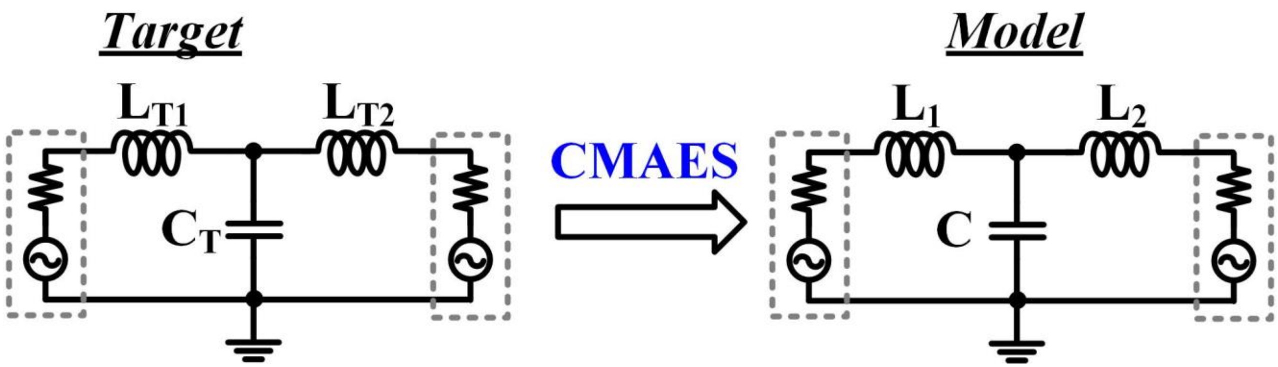
- **Step 6: Check *Result.txt***
  - **After the program is complete, check the *Result.txt* file**
  - **Use the modeled values to run sp simulation and confirm the correctness**

Model Fitting Complete!!



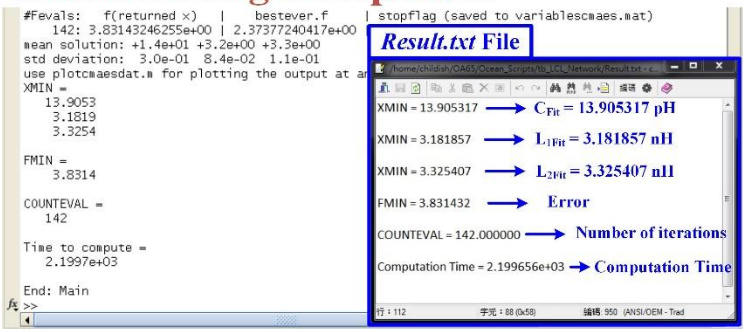
CMAES – LCL Network Model Fitting

- **Flow is the same as the previous example except for adjusting the input variable information (LC⇒LCL)**
  - Out *Target* has $L_{T1} = L_{T2} = 3.19\,nH$ and $C_T = 14\,pF$



- Model Fitting Result:
  - $L_1 = 3.181857\,nH$
  - $L_2 = 3.325407\,nH$
  - $C = 13.905317\,pH$

Model Fitting Complete!!



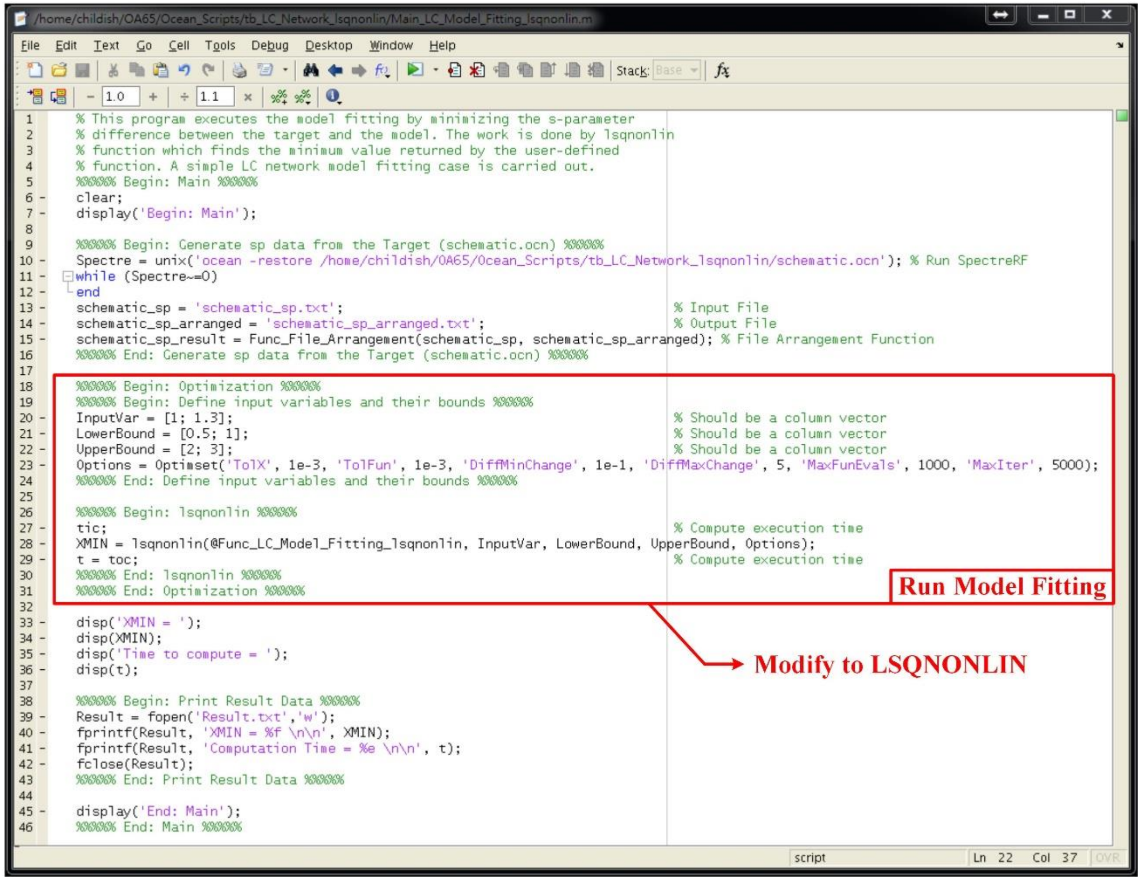
CMAES – Supplement

- **How to apply CMAES to other applications?**
  - **CMAES needs an user-defined function, for instance**
    - ***Result = Func_Example(InputVar)***
    - ***InputVar* is an array which contains your input information**
    - ***Result* is a number which is returned to CMAES**
  - **CMAES targets on minimizing *Result***
- **For example, if one would like to maximize the efficiency of a rectifier**
  - **Set** $\boldsymbol{Result = -Efficiency = -\left(\frac{P_{Out}}{P_{In}}\right)}$
  - **CMAES minimizes this *Result* so that the best efficiency can be found**

Different Optimizers

- **For Example, using LSQNONLIN**
- **TolX**
  - **Termination tolerance on XMIN (Cannot be too small)**
- **TolFun**
  - **Termination tolerance on the function value**
- **DiffMinChange**
  - **Minimum change in variables (Cannot be too small)**
- **MaxFunEvals**
  - **Maximum function evaluations**
- **MaxIter**
  - **Maximum iterations**

Main Function: *Main_LC_Model_Fitting.m*

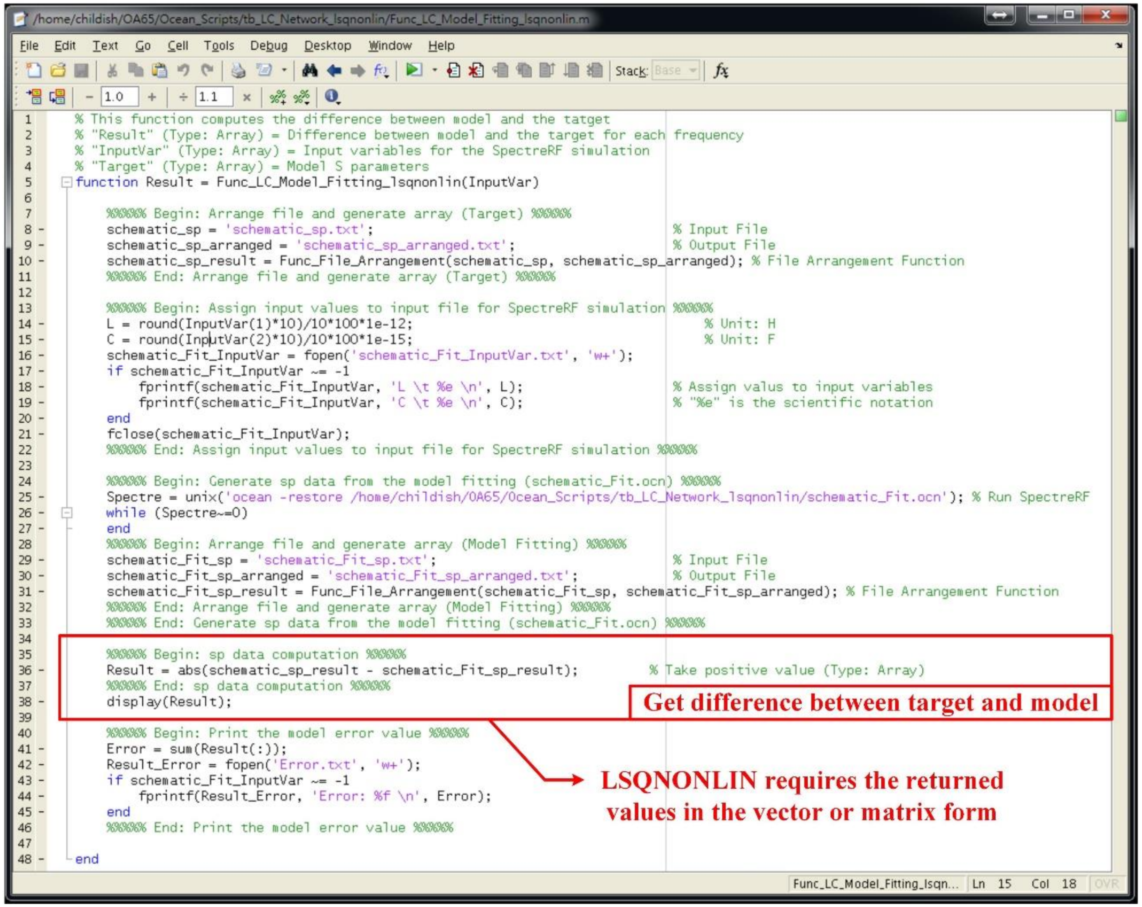



```matlab
% This program executes the model fitting by minimizing the s-parameter
% difference between the target and the model. The work is done by lsqnonlin
% function which finds the minimum value returned by the user-defined
% function. A simple LC network model fitting case is carried out.
%%%%% Begin: Main %%%%%
clear;
display('Begin: Main');

%%%%% Begin: Generate sp data from the Target (schematic.ocn) %%%%%
Spectre = unix('ocean -restore /home/childish/OA65/Ocean_Scripts/tb_LC_Network_lsqnonlin/schematic.ocn'); % Run SpectreRF
while (Spectre~=0)
end
schematic_sp = 'schematic_sp.txt';                                      % Input File
schematic_sp_arranged = 'schematic_sp_arranged.txt';                    % Output File
schematic_sp_result = Func_File_Arrangement(schematic_sp, schematic_sp_arranged); % File Arrangement Function
%%%%% End: Generate sp data from the Target (schematic.ocn) %%%%%

%%%%% Begin: Optimization %%%%%
%%%%% Begin: Define input variables and their bounds %%%%%
InputVar = [1; 1.3];                                                    % Should be a column vector
LowerBound = [0.5; 1];                                                  % Should be a column vector
UpperBound = [2; 3];                                                    % Should be a column vector
Options = Optimset('TolX', 1e-3, 'TolFun', 1e-3, 'DiffMinChange', 1e-1, 'DiffMaxChange', 5, 'MaxFunEvals', 1000, 'MaxIter', 5000);
%%%%% End: Define input variables and their bounds %%%%%

%%%%% Begin: lsqnonlin %%%%%
tic;                                                                    % Compute execution time
XMIN = lsqnonlin(@Func_LC_Model_Fitting_lsqnonlin, InputVar, LowerBound, UpperBound, Options);
t = toc;                                                                % Compute execution time
%%%%% End: lsqnonlin %%%%%
%%%%% End: Optimization %%%%%

disp('XMIN = ');
disp(XMIN);
disp('Time to compute = ');
disp(t);

%%%%% Begin: Print Result Data %%%%%
Result = fopen('Result.txt','w');
fprintf(Result, 'XMIN = %f \n\n', XMIN);
fprintf(Result, 'Computation Time = %e \n\n', t);
fclose(Result);
%%%%% End: Print Result Data %%%%%

display('End: Main');
%%%%% End: Main %%%%%
```

Different Optimizers

- **LSQNONLIN asks for results in vector or matrix form**
- **LSQNONLIN also minimize the *Result***
- **From CMAES to LSQNONLIN, one only has to modify the code in the red part**

Func_LC_Model_Fitting.m




```
% This function computes the difference between model and the tatget
% "Result" (Type: Array) = Difference between model and the target for each frequency
% "InputVar" (Type: Array) = Input variables for the SpectreRF simulation
% "Target" (Type: Array) = Model S parameters
function Result = Func_LC_Model_Fitting_lsqnonlin(InputVar)

    %%%%% Begin: Arrange file and generate array (Target) %%%%%
    schematic_sp = 'schematic_sp.txt';                                  % Input File
    schematic_sp_arranged = 'schematic_sp_arranged.txt';                % Output File
    schematic_sp_result = Func_File_Arrangement(schematic_sp, schematic_sp_arranged); % File Arrangement Function
    %%%%% End: Arrange file and generate array (Target) %%%%%

    %%%%% Begin: Assign input values to input file for SpectreRF simulation %%%%%
    L = round(InputVar(1)*10)/10*100*1e-12;                                 % Unit: H
    C = round(InputVar(2)*10)/10*100*1e-15;                                 % Unit: F
    schematic_Fit_InputVar = fopen('schematic_Fit_InputVar.txt', 'w+');
    if schematic_Fit_InputVar ~= -1
        fprintf(schematic_Fit_InputVar, 'L \t %e \n', L);               % Assign valus to input variables
        fprintf(schematic_Fit_InputVar, 'C \t %e \n', C);               % "%e" is the scientific notation
    end
    fclose(schematic_Fit_InputVar);
    %%%%% End: Assign input values to input file for SpectreRF simulation %%%%%

    %%%%% Begin: Generate sp data from the model fitting (schematic_Fit.ocn) %%%%%
    Spectre = unix('ocean -restore /home/childish/OA65/Ocean_Scripts/tb_LC_Network_lsqnonlin/schematic_Fit.ocn'); % Run SpectreRF
    while (Spectre~=0)
    end
    %%%%% Begin: Arrange file and generate array (Model Fitting) %%%%%
    schematic_Fit_sp = 'schematic_Fit_sp.txt';                          % Input File
    schematic_Fit_sp_arranged = 'schematic_Fit_sp_arranged.txt';        % Output File
    schematic_Fit_sp_result = Func_File_Arrangement(schematic_Fit_sp, schematic_Fit_sp_arranged); % File Arrangement Function
    %%%%% End: Arrange file and generate array (Model Fitting) %%%%%
    %%%%% End: Generate sp data from the model fitting (schematic_Fit.ocn) %%%%%

    %%%%% Begin: sp data computation %%%%%
    Result = abs(schematic_sp_result - schematic_Fit_sp_result);      % Take positive value (Type: Array)
    %%%%% End: sp data computation %%%%%
    display(Result);

    %%%%% Begin: Print the model error value %%%%%
    Error = sum(Result(:));
    Result_Error = fopen('Error.txt', 'w+');
    if schematic_Fit_InputVar ~= -1
        fprintf(Result_Error, 'Error: %f \n', Error);
    end
    %%%%% End: Print the model error value %%%%%

end
```

Different Optimizers

- **Notes for using LSQNONLIN**
  - **Be careful in choosing the parameters in *Optimset*, they influence a lot in the overall computation time and in whether the program converges or not**
  - **LSQNONLIN is a suitable function in the model fitting case**
  - **Guessed value influences the computation time a lot when using LSQNONLIN optimizer as proven below:**

LSQNONLIN	Case 1	Case 2
Target	LC Network Fitting: $(100, 250)$ in the unit of (pH, fF)	
Guessed Value	$(100, 130)$	$(80, 220)$
Lower Bound	$(50, 100)$	$(50, 100)$
Upper Bound	$(200, 300)$	$(200, 300)$
Result	$(102.61, 249.88)$	$(98.00, 248.03)$
Computation Time	4.28 minutes	2.4 minutes

Performance Comparison

- **LC network fitting using CMAES and LSQNONLIN**

	CMAES	LSQNONLIN
Target	$(100, 250)$ in the unit of (pH, fF)	
Guessed Value	$(100, 130)$	$(100, 130)$
Lower Bound	$(50, 100)$	$(50, 100)$
Upper Bound	$(200, 300)$	$(200, 300)$
Result	$(102.78, 247.89)$	$(102.61, 249.88)$
Computation Time	29.38 minutes	4.28 minutes

- **For the model fitting task, using LSQNONLIN is much more efficient as proven here**

Performance Comparison

- **LCL network fitting using CMAES and LSQNONLIN**

	CMAES	LSQNONLIN
Target	$(14, 3.19, 3.19)$ in the unit of (pF, nH, nH)	
Guessed Value	$(10, 3, 3)$	$(10, 3, 3)$
Lower Bound	$(5, 1, 1)$	$(5, 1, 1)$
Upper Bound	$(20, 5, 5)$	$(20, 5, 5)$
Result	$(13.9053, 3.1819, 3.3254)$	$(14.0211, 3.2262, 3.2263)$
Computation Time	37.06 minutes	8.15 minutes

- **For the model fitting task, using LSQNONLIN is much more efficient as proven here**