

# Improving Analog Circuit Equation and Jacobian Generator (Parser)

Sudhir Kumar

Indian Institute of Technology, Bombay, India

[13D070036@iitb.ac.in](mailto:13D070036@iitb.ac.in)

Guide: Ljiljana Trajkovic

School of Engineering Science

Simon Fraser Institute, Burnaby

British Columbia, Canada

July 14, 2016

## 1 The Parser

Parser is C++ program[1] which generates the set of equations and Jacobians required by the Matlab code to employ the homotopy method. The Matlab script then calculates all operating points using iterative techniques. The advantage of using homotopy methods is that the output contains all possible operating points as compared to the SPICE output that returns only one result depending on the initial point provided.

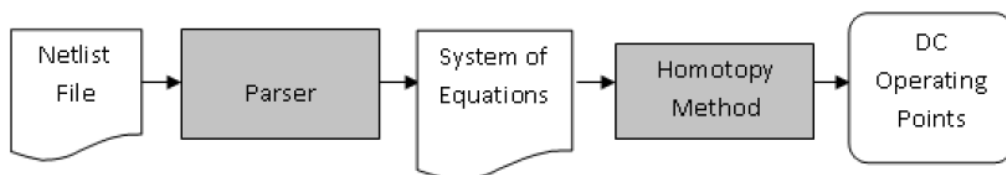
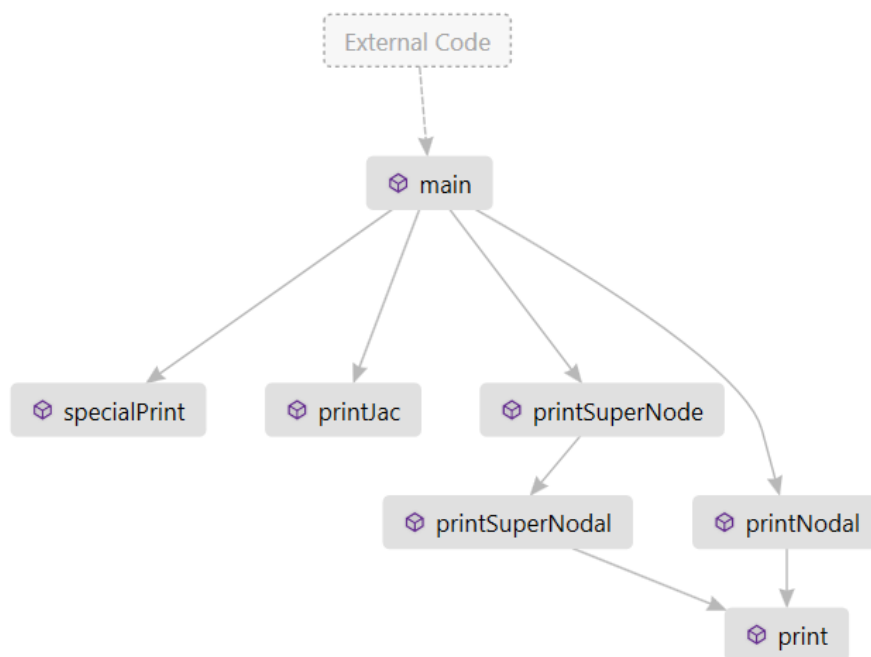


Figure 1: Algorithmic flow for DC operating point calculation.

Implementation of the homotopy methods requires that the set of equations. For some circuits, these equations can be written by hand. However, this is not possible for more complicated circuits. The parser takes the SPICE netlist as the input that describes the elements of the circuit and how they are connected. It then provides the circuit equations for either nodal or modified nodal analysis and their Jacobians.

## 2 Code Flow Map of Parser

The parser originally developed developed by Edward Chan [2] improved by Andrea Dyess, and then recently further improved by Joao Eric Melo [5] and Jatin Vikram Singh basically has following code structure.



## 3 The Extra Equation (Nodal Analysis)

The current parser was able to generate the correct equations for Schmitt trigger circuits and Chua's circuits.

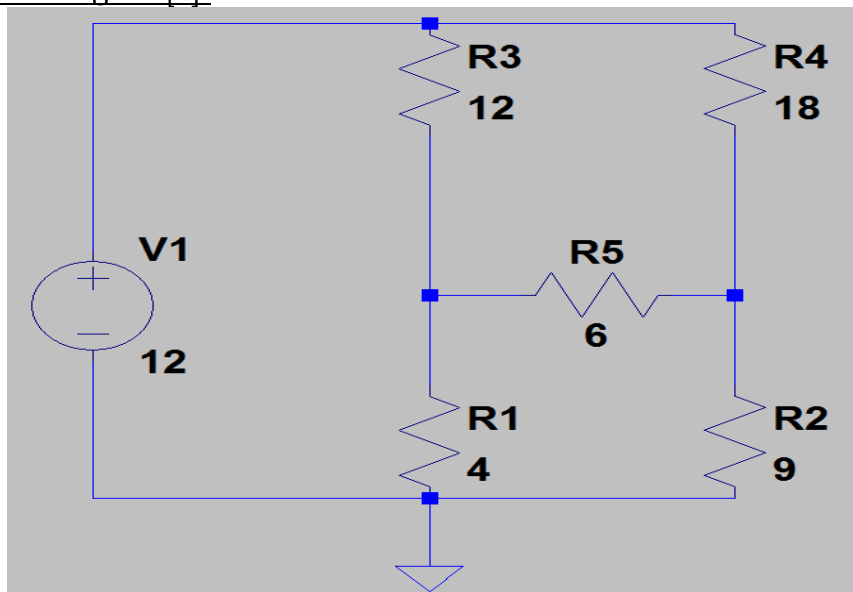
However, it was needed to test the code on a lot more varieties of the circuits. In this attempt we choose some simple dc circuits with and without transistors respectively.

The code[4] generated all the equations and Jacobians for modified nodal analysis correctly, but in nodal analysis we encountered one extra equation, which had no meaning. The extra equations in nodal analysis have been shown in the following pictures.

In both these circuits we have only 3 nodes. So for nodal analysis, we would require only 3 nodal equations but we got fourth equation i.e. evident from the following figures.

### 3.1 Simple DC circuit analysis

Circuit diagram[3]:



Output having one extra equation(F(4)):

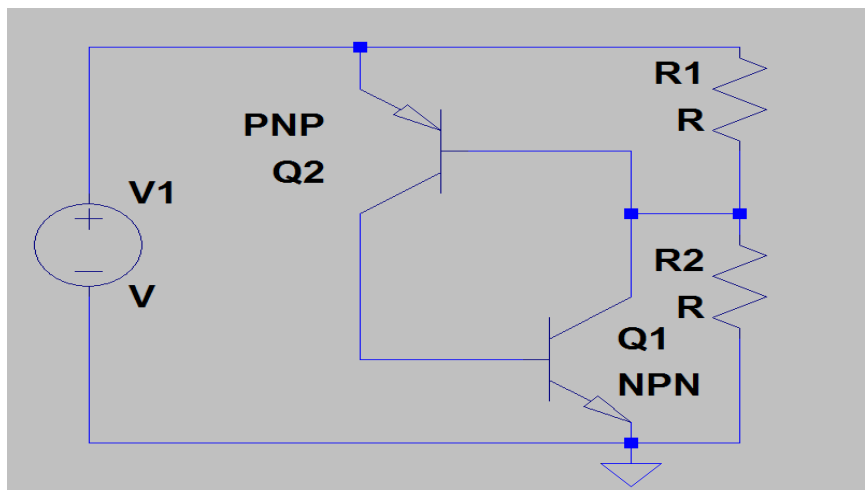
```

out_1nodal.txt
1 %Parser V1.0
2 %Input Spice Deck: C:\Users\sudhirk\Desktop\Netlist_1.txt
3 %Equation Type: NODAL
4 %Datum Node: 2
5
6 %*****
7 % Component Values:
8 r1 = 4;
9 r2 = 9;
10 r3 = 12;
11 r4 = 18;
12 r5 = 6;
13
14 %*****
15 %
16 % Circuit Equations:
17
18 F(1) = (X(2))/r1 + (X(2)-X(1))/r3 + (X(2)-X(3))/r5 ;
19
20 F(2) = (X(3))/r2 + (X(3)-X(1))/r4 + (X(3)-X(2))/r5 ;
21
22 F(3) = (X(1)) -12;
23
24 F(4) = ( (X(1)-X(2))/r3 (X(1)-X(3))/r4 ) + ( (-X(2))/r1 (-X(3))/r2 );
25
26 %*****
27 %
28 % Jacobians:
29
30 JAC(1, 2) = 1/r1 + 1/r3 + 1/r5 ;
31
32 JAC(1, 3) = 0 + 0 + (-1/r5) ;
33
34 JAC(1, 1) = 0 + (-1/r3) + 0 ;
35
36 JAC(2, 2) = 0 + 0 + (-1/r5) ;

```

### 3.2 2-transister circuit analysis

Circuit diagram:



### Output having one extra equation:

```
1 %Parser V1.0
2 %Input Spice Deck: C:\Users\sudhir\Desktop\twoTwoRsameside.txt
3 %Equation Type: NODAL
4 %Datum Node: 2
5
6 %*****
7 % Component Values:
8 q1IS = 1e-016;
9 q1BF = 100;
10 q1BR = 1;
11 q1N = -1.25262e-058;
12 q2IS = 1e-016;
13 q2BF = 100;
14 q2BR = 1;
15 q2N = -1.25262e-058;
16 r1 = 1;
17 r2 = 2;
18
19 %*****
20 % Circuit Equations:
21
22 F(1) = (q1IS)*(exp(-q1N*(-X(3))) -1) + (-q1IS/q1BR)*(1+q1BR)*(exp(-q1N*(X(2)-X(3))) -1) + (q2IS)*(e:
23
24 F(2) = (-q1IS)*(exp(-q1N*(-X(3))) -1) + (q1IS/q1BR)*(1+q1BR)*(exp(-q1N*(X(2)-X(3))) -1) + (q1IS/q1B
25
26 F(3) = (X(1)) -12;
27
28 F(4) = ((q2IS/q2BF)*(1+q2BF)*(exp(q2N*(X(1)-X(2))) -1) - (q2IS)*(exp(q2N*(X(3)-X(2))) -1) + (X(1)-X
29
30 %*****
31 % Jacobians:
32
33
34
35
```

## 4 Output (Nodal Analysis) After Improving the Parser

In the function to print super\_node equations, the code takes compPtr incorrectly, and hence its value is not NULL even after covering all the nodes once. So the code was modified accordingly so that we could get right no of equations in Nodal analysis.

### Output of simple dc circuit Now:

```

1 %Parser V1.0
2 %Input Spice Deck: C:\Users\sudhirk\Dropbox\Intern\Intern_2016\ckt_inout\Netlist_1.txt
3 %Equation Type: NODAL
4 %Datum Node: 2
5
6 %*****
7 %
8 % Component Values:
9 r1 = 4;
10 r2 = 9;
11 r3 = 12;
12 r4 = 18;
13 r5 = 6;
14 %*****
15 %
16 % Circuit Equations:
17 F(1) = (X(2))/r1 + (X(2)-X(1))/r3 + (X(2)-X(3))/r5 ;
18 F(2) = (X(3))/r2 + (X(3)-X(1))/r4 + (X(3)-X(2))/r5 ;
19 F(3) = (X(1)) -12;
20
21 %*****
22 %
23 % Jacobians:
24 JAC(1, 2) = 1/r1 + 1/r3 + 1/r5 ;
25 JAC(1, 3) = 0 + 0 + (-1/r5) ;
26 JAC(1, 1) = 0 + (-1/r3) + 0 ;
27
28
29
30
31
32
33
34

```

Output of 2-transistor circuit Now:

```

1 %Parser V1.0
2 %Input Spice Deck: C:\Users\sudhirk\Desktop\twoTtwoRsameside.txt
3 %Equation Type: NODAL
4 %Datum Node: 2
5
6 %*****
7 %
8 % Component Values:
9 q1IS = 1e-016;
10 q1BF = 100;
11 q1BR = 1;
12 q1N = -1.25262e-058;
13 q2IS = 1e-016;
14 q2BF = 100;
15 q2BR = 1;
16 q2N = -1.25262e-058;
17 r1 = 1;
18 r2 = 2;
19 %*****
20 %
21 % Circuit Equations:
22 F(1) = (q1IS)*(exp(-q1N*(-X(3))) -1) + (-q1IS /q1BR)*(1+q1BR)*(exp(-q1N*(X(2)-X(3))) -1) + (q2IS)*(exp(q
23 F(2) = (-q1IS)*(exp(-q1N*(-X(3))) -1) + (q1IS /q1BR)*(1+q1BR)*(exp(-q1N*(X(2)-X(3))) -1) + (q1IS /q1BF)*
24 F(3) = (X(1)) -12;
25
26 %*****
27 %
28 % Jacobians:
29 JAC(1, 2) = (q1IS * q1N / q1BR)*(1 + q1BR)*(exp(- q1N*(X(2) - X(3)))) + ( - q2IS * q2N)*(exp(q2N*(X(1) - X(
30
31
32
33
34

```

## 5 Netlist Format Requirement

Input to the parser requires the circuit in netlist format. The required netlist can be generated by a lot of softwares present. But the format in which our parser accepts the input has some specifications which are as follows:

- There must be one-line gap after the very first line i.e. \*description
- 3<sup>rd</sup> line should look like “.model nbjt NPN IS=1e-16 BF=100 BR=1”. this format must be followed in every netlist file even if the circuit contains bjt or not.
- If the circuit have npn and pnp both transistors; then the 3<sup>rd</sup> and 4<sup>th</sup> line should look like:  

```
.model nbjt NPN IS=1e-16 BF=100 BR=1
.model pbjt PNP IS=1e-16 BF=100 BR=1
```
- In the circuit description of npn and pnp; nbjt and pbjt must be written after the connections like:  

```
q1 2 3 0 nbjt
q2 3 2 1 pbjt
```
- At the end of the netlist file we must have these two lines:  

```
.nodeset v(0)=0
.op
```
- A specimen netlist file has been attached as follows

```
* twoTwoRsameside.asc

.model nbjt NPN IS=1e-16 BF=100 BR=1
.model pbjt PNP IS=1e-16 BF=100 BR=1

q1 2 3 0 nbjt
q2 3 2 1 pbjt
I1 1 0 12
r1 2 1 1
r2 0 2 2
r3 2 3 3

.nodeset
```

## 6 Conclusion and future work

Till now parser is giving correct equations for the circuits having one voltage or current source.

While working in the last week of my project after rectifying the extra equation problem and implementing new homotopy I found out that parser still is needed to be checked on various circuits containing the floating voltage sources. One insight in this regard is to now check correctly the sequence of “compPtr” in the component list so that we can get the equations for floating sources as well without affecting the nodal analysis of the circuit equations.

And then if time permits extra features of MOSFETs can be added to the code appropriately.

## 7 References

- [1] <https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx>
- [2] E. Chan, Documentation on the parser program, Technical report, UC Berkeley, 1996.
- [3] Ltspice
- [4] [http://www2.ensc.sfu.ca/~ljilja/cnl/projects/Homotopy/parser\\_revised/](http://www2.ensc.sfu.ca/~ljilja/cnl/projects/Homotopy/parser_revised/)
- [5] J. Eric Melo and L. Trajkovic, Improving an electronic circuit simulator based on homotopy methods, Technical report, SFU Burnaby, 2014.