DC analysis

Objective: Describe numerical methods, expose problems and discuss solution techniques

Outline:

1. Problem formulation
2. Solution methods
3. Newton-Raphson scalar case
4. Multivariable Newton-Raphson method
5. Examples - exercises
6. Modifications to Newton-Raphson
7. DC convergence control in SPICE
8. Concluding remarks - suggestions,
9. Handling non-convergence cases.

1. Problem formulation

The mathematical form of circuit equations (MNA, traditional variables, Session. 3, sect. 3.3)

\[ C(x) \frac{dx}{dt} + G(x)x = w \ . \]

**Problem:** for given, constant input, find selected circuit variables at equilibrium, i.e. when

the time derivatives, \( \frac{d}{dt} \), are ignored

\[ G(x)x = w \ . \]

In the circuit terminology this means that for DC analysis the capacitors are open and the

inductors are shorted.

DC analysis is a very fundamental function of all circuit simulators.

It is used in determining: operating point, initial conditions in transients, transfer curves etc.

Presence of nonlinear elements in the network makes the analysis very difficult.

Very often a simulator fails in DC analysis because of numerical complexity.
2. Solution methods

1. Newton - Raphson
   (basic SPICE method)

2. Relaxation techniques
   (Gauss-Seidel, Gauss-Jacobi)

3. Optimization procedures

4. Continuation techniques
   - inaccurate transient analysis (IBM ASTAP)
   - source stepping (implemented as optional in some SPICE’s)
Alternative approaches

 Reformulate problem:

\[ G(x)x - w = f(x) = 0 \]

Newton

\[ \Gamma = \nabla f(x) \]

solve \[ \Gamma \delta = -f(x) \]

substitute \[ x \leftarrow x + \delta \]

and iterate.

The procedure has a local quadratic convergence.

Trust-region Newton’s

same as Newton, with restriction \( \| \delta_k \| \leq \Delta_k \)

this is equivalent to solving \( (\Gamma + \lambda I) \delta = -f(x) \), with some \( \lambda \geq 0 \).

\( \Delta_k \) sufficiently small \( \Rightarrow \delta \) decreases \( \| f(x) \| \),

near solution \( \lambda = 0 \),

retains convergence properties.

Quasi-Newton: estimate \( \Gamma \) from \( f(x_k), f(x_{k+1}), f(x_{k+2}), \ldots \)

can be incorporated in trust-region method.
3. Newton-Raphson scalar case

\[ g(y^*) = 0 \quad \text{---ideal (theoretical) solution.} \]

The iterative process is constructed for the increment, \( \Delta y \), defined as

\[ y^* = y^{(k)} + \Delta y \]

and is based on the linearization

\[ g(y^{(k)} + \Delta y) \approx g(y^{(k)}) + g'(y^{(k)})\Delta y \]

where

\[ g' = \left. \frac{\partial g}{\partial y} \right|_{y=y^{(k)}}. \]

The increment is approximated as follows

\[ \Delta y = y^* - y^{(k)} \Rightarrow \Delta y \approx y^{(k+1)} - y^{(k)} \]

which leads to the iterative equation

\[ g(y^{(k)}) + g'(y^{(k)})(y^{(k+1)} - y^{(k)}) = 0 \]
or else
\[ g'(y^{(k)}) \cdot y^{(k+1)} = g'(y^{(k)}) \cdot y^{(k)} - g(y^{(k)}) \]

\[ k = 0, 1, 2, \ldots \]

The iterative process needs: \( y^{(0)} \) — starting value.

Geometry
Convergence

If $g$ is twice differentiable and $\frac{dg}{dy} \bigg|_{y=y^*} \neq 0$,

Then $y^{(k)} \rightarrow y^*$;

If $y^{(0)}$ is sufficiently close to $y^*$, the convergence is quadratic:

$$e_k = |y^{(k)} - y^*| \quad \text{and} \quad e_{k+1} \leq ce_k^2 \quad \text{where } c \text{— is a constant}$$

In constructing the iterative process we need a derivative. Numerical differentiation is “noisy” and thus we want to use analytical derivatives and avoid numerical differentiation.
4. Multivariable Newton-Raphson algorithm

**Problem:** find the solution to the vector equation: \( f(x) = 0 \)

where \( x \) is a vector and \( f(x) \) is a vector function.

Newton-Raphson iterations require Jacobian

\[
\frac{\partial f}{\partial x} = J(x) .
\]

The iterative process is constructed as follows

\[
J(x^{(k)})x^{(k+1)} = J(x^{(k)})x^{(k)} - f(x^{(k)})
\]

where the Jacobian is computed via evaluation of partial derivatives.
The Jacobian is defined as follows:

\[ J(x^{(k)}) = \left. \frac{\partial f}{\partial x} \right|_{x=x^k} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \]

Multivariable N-R algorithm definitions:

the theoretical solution: \( f(x^*) = 0 \quad x^* \text{--solution} \)

the iterative process:

\[ J(x^{(k)})x^{(k+1)} = J(x^{(k)})x^{(k)} - f(x^{(k)}) \]

\( k = 0, 1, 2, \ldots \)

Convergence conditions:

1. \( J(x) \) is a LIPSCHITZ function: \( \exists L \) such that if for \( \forall x, x' \),

\[ \|J(x) - J(x')\| \leq L\|x - x'\| \]

2. \( J(x^*) \) is non-singular

3. If \( x^{(0)} \) is close enough to \( x^* \), then \( x^{(k)} \rightarrow x^* \) when \( k \rightarrow \infty \).
Observations:

a.) Good estimates of initial (starting) values are important.

In case of difficulties we use: *Source stepping*

The algorithm is based on the assumption that circuit variables are 0 if sources are 0.

b. Device models must be continuous with continuous partial derivatives.
5. Examples_ Exercises

CE: \[ i = I_s (e^{\frac{v}{R}} - 1) = g(v) \]

KCL: \[ I_o = \frac{1}{R} V + i \]
Equation to be solved (D-C) is non-linear:

\[ I_o - \frac{V}{R} = I_s \left( e^\frac{V}{V_r} - 1 \right) \]

We construct an iterative procedure with \( V^{(k)} \) – as a starting value.

Exercise 1:

a.) Derive iterative equation

\[ I_o - \frac{1}{R} V^{(k+1)} = I_s \left( e^\frac{V^{(k)}}{V_r} - 1 \right) + \frac{I_s}{V_t} \left( e^\frac{V^{(k)}}{V_r} \right) \left( V^{(k+1)} - V^{(k)} \right) \]

\[ k = 0, 1, 2, \ldots \]

b.) Assume: \( I_o = 1 \text{[mA]}, \quad R = 40 \text{[k\Omega]}, \quad I_s = 2.5 \cdot 10^{-12} \text{[A]}; \)

Calculate the first three steps \((k=1,2,3)\) of the iteration process starting at two different starting points:

1) \( V^{[0]} = 0 \)
2) \( V^{[0]} = 0.5 \) \text{[V]}
Example of non-convergence in analysis of an operational amplifier

\[ k = 1 \]

\[ \| V^{(1)} - V^{(0)} \| = 5 \]

after \( k=100 \) (default in SPICE), no convergence message was issued.

Try the following brute force approach (increase the iteration limit – ITL1):

```
.OPTION ....... , ITL1=1000, ....
```

Results

```

\[ k = 666 \]

\[ \| V^{(666)} - V^{(665)} \| = 43.02 \cdot 10^3 [V] \]

\[ k = 675 \]

\[ \| V^{(675)} - V^{(674)} \| = 0.25 \cdot 10^{-2} [V] \]

Convergence!
```
6. Modifications to Newton-Raphson algorithm

Limiting equations

Example of diode:

\[ i = I_s (e^{\frac{V}{V_t}} - 1) \]

\[ V = V_{\text{Lim}} \quad \Rightarrow \quad i = i_{mx} \]

\[ i_{mx} = I_s e^{\frac{V_{lim}}{V_t}} \quad V_{lim} \gg V_{tx} \]

\[ V_{lim} = V_t \ln\left(\frac{i_{mx}}{I_s}\right) \]

In the iterations:

If \( V^{(k+1)} \geq V_{lim} \)
then \( \overline{V}^{(k+1)} = V_{lim} \).
Modification to an iterative process will be discussed.

Notation:

\( V_o \) --starting point

\( \hat{V}_1 \) --computed new point

\( V_1 \) --accepted new point.

Three examples of specific procedures:

A.) Fixed departures in voltage

Procedure:

1. \( V_o < 10V \) and \( \hat{V}_1 < 10V \) then \( V_1 = \hat{V}_1 \)

2. \( |V_o - \hat{V}_1| \leq 2V \) then \( V_1 = \hat{V}_1 \)

3. \( V_o > 10V \) and \( \hat{V}_1 < V_o \) then \( V_1 = V_o - 2V \)

4. \( \hat{V}_1 > V_o \) and \( \hat{V}_1 > 10V \) then \( V_1 = \max\{10V, V_o + 2V \} \)
B) Non-linearly limited departures

\[ V_i = V_o + 10V_t \cdot \tanh \left( \frac{\hat{V}_i - V_o}{10V_t} \right) \]

the \( \tanh(\cdot) \) is approximated as follows:

\[
\tanh(x) = \begin{cases} 
\pm 1 & |x| > \frac{3}{2} \\
 x - \frac{x^3}{3} & \frac{1}{2} < |x| \leq \frac{3}{2} \\
x & |x| \leq \frac{1}{2}
\end{cases}
\]

Note that when \( |x| \leq \frac{1}{2} \) than \( \hat{V}_i = V_i \).
Graphical illustration

\[ \tanh(x) \]
C) Iterating in voltage or current
Iteration in current

Second iter. approx. to the diode at $V^{[1]}$

First iteration approx.

Accepted voltage corresp. to $I^{(1)}$

Volt. defined by solu. of first approx. to diode and load line. This volt. is not accepted
Linearized relations

\[
\hat{I} = I_s \left( e^{\frac{v^{(o)}}{V_t}} - 1 \right) + \frac{I_s}{V_t} e^{\frac{v^{(o)}}{V_t}} \left( \hat{V}^{(1)} - V^{(o)} \right)
\]  

(L)
\[
\hat{I} = I^{(1)} \quad \text{in (L)} \\
(\text{Diode}) \quad I^{(1)} = I_s \left(e^{v^{(1)}/V_t} - 1\right) \quad \Rightarrow \quad v^{(I)}
\]

Note: \( I^{(1)} \) is defined by the intersection of (L) and load lines

After algebra (do as an exercise)

\[
V^{(1)} = V^{(0)} + V_t \ln(1 + \frac{\hat{V}^{(1)} - V^{(0)}}{V_t})
\]

When to iterate in current or when to iterate in voltage?

Find \( V_{crit} \)

\[
\hat{V}^{(k+1)} < V_{crit} \quad \text{iterate in current}
\]

\[
\hat{V}^{(k+1)} > V_{crit} \quad \text{iterate in voltage}
\]
**Critical current** (arbitrary choice of value is given below).

Critical current is determined as a current caused by the thermal voltage, $V_t$, applied to the resistor of $\sqrt{2}[\Omega]$.

$$I_{\text{crit}} = \frac{V_t}{\sqrt{2}}$$

this is an arbitrary definition

$$I = I_s (e^{V/V_t} - 1) \quad \frac{V_{\text{crit}}}{V_t} \gg 1$$

$$I_{\text{crit}} = I_s e^{V_{\text{crit}}/V_t}$$

$$V_{\text{crit}} = V_t \ln\left(\frac{V_t}{\sqrt{2}I_s}\right)$$

*Exercise:*

Calculate the first 3 steps in iteration for D-C analysis (use circuit specified in the exercise 1) using the following algorithms:

1. fixed departures in voltage.
2. nonlinearly limited departures
3. iteration in current or voltage as necessary (as indicated by $V_{\text{crit}}$).
Assume $V^{(0)}=0$ as a starting point.

7. DC convergence control in SPICE

Control parameters:

- RELTOL default value is $10^{-3}$
- ABSTOL default value is $1\,[\text{pA}]$
- VNTOL default value is $1\,[\text{\mu V}]$

Convergence criterion

$$\left\| x^{(k+1)} - x^{(k)} \right\| \leq \epsilon_a + \epsilon_r \max\{\left\| x^{(k)} \right\|, \left\| x^{(k+1)} \right\|\}$$

where $\epsilon_a$ represents ABSTOL or VNTOL
$\epsilon_r$ represents RELTOL.

Other parameters:

- ITL1-sets the limit for the number of iterations in DC analysis,
  the default number is 100,
- ITL2-sets the limit for the number of iterations in computing the transfer curve
  the default number is 50.
Details of convergence control in SPICE

A. circuit variables – convergence criteria

--nodal voltages

\[ |v_n^{(k+1)} - v_n^{(k)}| \leq VNTOL + RELTOL \cdot \max\{v_n^{(k+1)},v_n^{(k)}\}\]

--branch currents (computation of \(I_2\) vector)

\[ |i_{\ell}^{(k+1)} - i_{\ell}^{(k)}| \leq ABSTOL + RELTOL \cdot \max\{|i_{\ell}^{(k+1)}|,|i_{\ell}^{(k)}|\}\]

B. I-V characteristics (functions like: \(I_D\) (diode); \(I_B\) (BJT); \(I_{DS}\) (FET))
Example of criterion for diode:

Notation: $V_J$ – junction voltage

Current due to the voltage value determined in the previous $(k^{th})$–iteration

\[ I_D = I_s \left( e^{\frac{V_J^{(k)}}{V_t}} - 1 \right) \]

Current from the linearized equation $(k+1)$--iter.

\[ \tilde{I}_D = I_s \frac{V_J^{(k)}}{V_t} V_J^{(k+1)} + I_D - I_s \frac{V_J^{(k)}}{V_t} V_J^{(k)} \]

convergence criterion for I-V relations

\[ |\tilde{I}_D - I_D| \leq ABSTOL + RELTOL \cdot \max\{I_D, \tilde{I}_D\} \]

Convergence criteria for circuit variables and I-V relations must be satisfied in:

ITL1 –iterations for D-C analysis
ITL2—iterations for D-C transfer curve.
8. Concluding remarks

D-C analysis is performed before the following major analyses:

AC, TRAN, SENS, TRANSF. CURVES

unless it is cancelled by a user.

Sensitivity analysis

.SENS outvar computes sensitivity of “outvar” with respect to every element in the circuit and all D-C parameters of diodes and BJT’s (FET’s are excluded)

Generation of D-C transfer curves (example: I-V)

DC Vname start stop step <vname1 str1 stp1 setp1….>
Help/suggestions for dealing with convergence problems.

1. Nothing helps better than knowledge of circuit and setting good starting values using

   NODESET   V(node)=value < V(node1)=value1......>

2. Check tolerances (defaults) and possibly relax them.

3. Increase ITL1—limit for the number of iterations in DC analysis.

4. Use source ramping (this is automatically initiated in SPICE3 & PSPICE), in SPICE2 one must set ITL6=…..(50).

5. Use “off” flag at the end of transistor lines to indicate the transistors that may be turned off.

To learn more about the computing process, use ACCT option, which gives some additional numerical details.

When DC analysis converges, one can use the flag

   .OP   - which is a request for operating point information such as:
        1. the terminal currents & voltages;
        2. the small signal equivalent conductances for all Non-Linear devices.

The information is listed in the “OPERATING POINT INFO” section of the output.

Note: .OP is done by default when .DC is not followed by any analysis specification.