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.**

Supplemental reading: **Vladimirescu, The SPICE Book,** **Chapters: 4, 9.1 - 9.3, 10.3**

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 $\|\delta_k\| \le \Delta_k$

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

Δ_k **sufficiently small** $\Rightarrow$ δ **decreases** $\|f(x)\|$,

near solution $\lambda = 0$,

retains convergence properties.

Quasi-Newton: **estimate** Γ **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

$$g(y^*) = 0 \qquad y^* \text{---ideal (theoretical) solution.}$$

The iterative process is constructed for the increment, Δy, defined as

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

and is based on the linearization

$$g(y^{(k)} + \Delta y) \cong g(y^{(k)}) + g'(y^{(k)})\Delta y$$

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

The increment is approximated as follows

$$\Delta y = y^* - y^{(k)} \Rightarrow \Delta y \cong y^{(k+1)} - y^{(k)}$$

which leads to the iterative equation

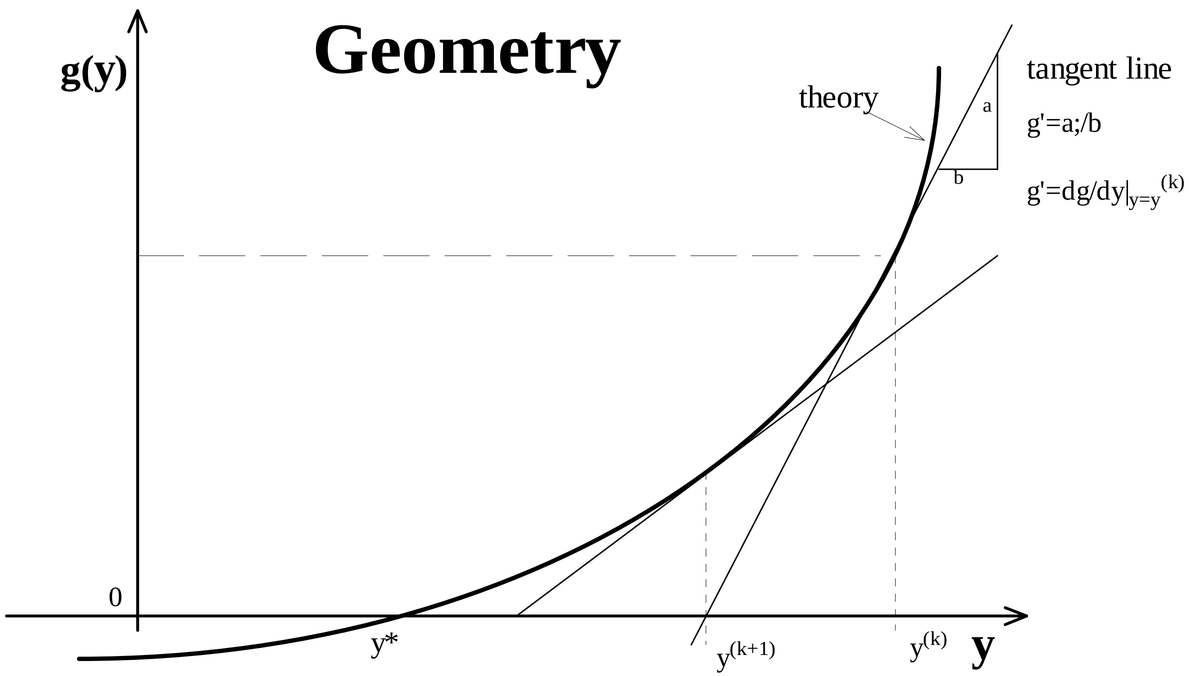
$$g(y^{(k)}) + g'\left(y^{(k)}\right)(y^{(k+1)} - y^{(k)}) = 0$$

or else

$$g'\left(y^{(k)}\right)\cdot y^{(k+1)} = g'\left(y^{(k)}\right)\cdot y^{(k)} - g\left(y^{(k)}\right)$$

$$k = 0, 1, 2, \ldots\ldots$$

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



Convergence

If g is **<u>twice differentiable</u>** and $\left.\frac{dg}{dy}\right|_{y=y^*} \neq 0$,

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

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

$e_k = \left| y^{(k)} - y^* \right|$ and $e_{k+1} \leq c e_k^2$ where c—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***

$$\frac{\partial f}{\partial x} = J(x) \quad .$$

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:

$$J(x^{(k)}) = \frac{\partial f}{\partial x}\bigg|_{x=x^k} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}\Bigg|_{x=x^{(k)}} .$$

Multivariable N-R algorithm definitions:

the theoretical solution: $f(x^*) = 0$ x^*--solution

the iterative process:

$$J(x^{(k)})x^{(k+1)} = J(x^{(k)})x^{(k)} - f(x^{(k)})$$

$$k = 0,1,2,\ldots\ldots$$

Convergence conditions:

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

   $$\|J(x) - J(x')\| \le L\|x - x'\|$$

2. ***J(x*)*** **is non-singular**
3. **If** $x^{[0]}$ **is close enough to** x^***, then** $x^{(k)} \to x^*$ **when** $k \to \infty$ **.**

Observations:

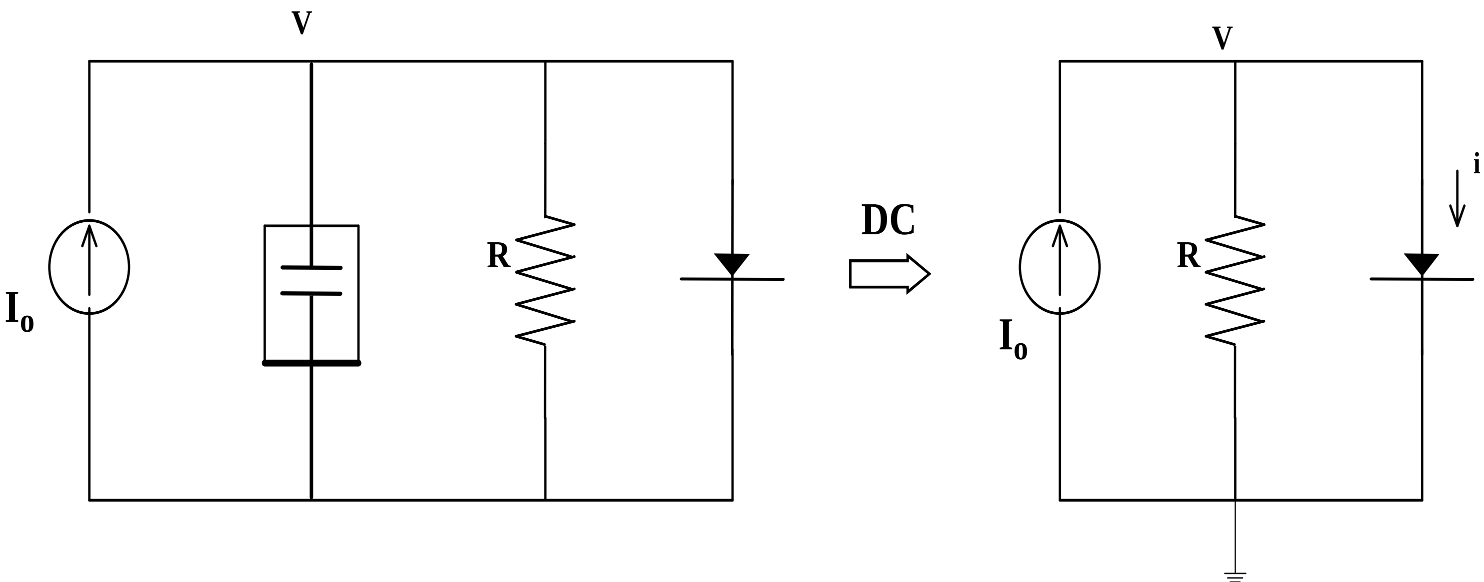
a.) Good estimates of initial (starting) values are important.

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: $$i = I_s\left(e^{\frac{V}{V_t}} - 1\right) = \bar{g}(v)$$

KCL: $$I_o = \frac{1}{R}V + i$$

Equation to be solved (D-C) is non-linear:

$$I_o - \frac{V}{R} = I_s\left(e^{\frac{V}{V_t}} - 1\right)$$

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

Exercise 1:

a.) Derive iterative equation

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

$$k = 0,1,2,.....$$

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

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

1) $V^{[0]}=0$

2) $V^{[0]}=0.5$ **[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: $i = I_s(e^{\frac{V}{V_t}} - 1)$

$$V = V_{Lim} \quad \Rightarrow \quad i = i_{mx}$$

$$i_{mx} = I_s e^{\frac{V_{Lim}}{V_t}} \qquad V_{Lim} >> V_{tx}$$

$$V_{Lim} = V_t \ln(\frac{i_{mx}}{I_s})$$

In the iterations:

If $V^{(k+1)} > V_{Lim}$

then

$$\overline{V}^{(k+1)} = V_{Lim} \quad .$$

Modification to an iterative process will be discussed.

Notation:

V_o **--starting point**

$\hat{V}_1$ **--computed new point**

V_1 **--accepted new point.**

<u>Three examples of specific procedures:</u>

A.)Fixed departures in voltage

Procedure:

1. $V_0 < 10V_t$ and $\hat{V}_1 < 10V_t$ then $V_1 = \hat{V}_1$
2. $\left|V_0 - \hat{V}_1\right| \le 2V_t$ then $V_1 = \hat{V}_1$
3. $V_0 > 10V_t$ and $\hat{V}_1 < V_0$ then $V_1 = V_0 - 2V_t$
4. $\hat{V}_1 > V_0$ and $\hat{V}_1 > 10V_t$ then $V_1 = \max\{10V_t, V_0 + 2V_t\}$

B) Non-linearly limited departures

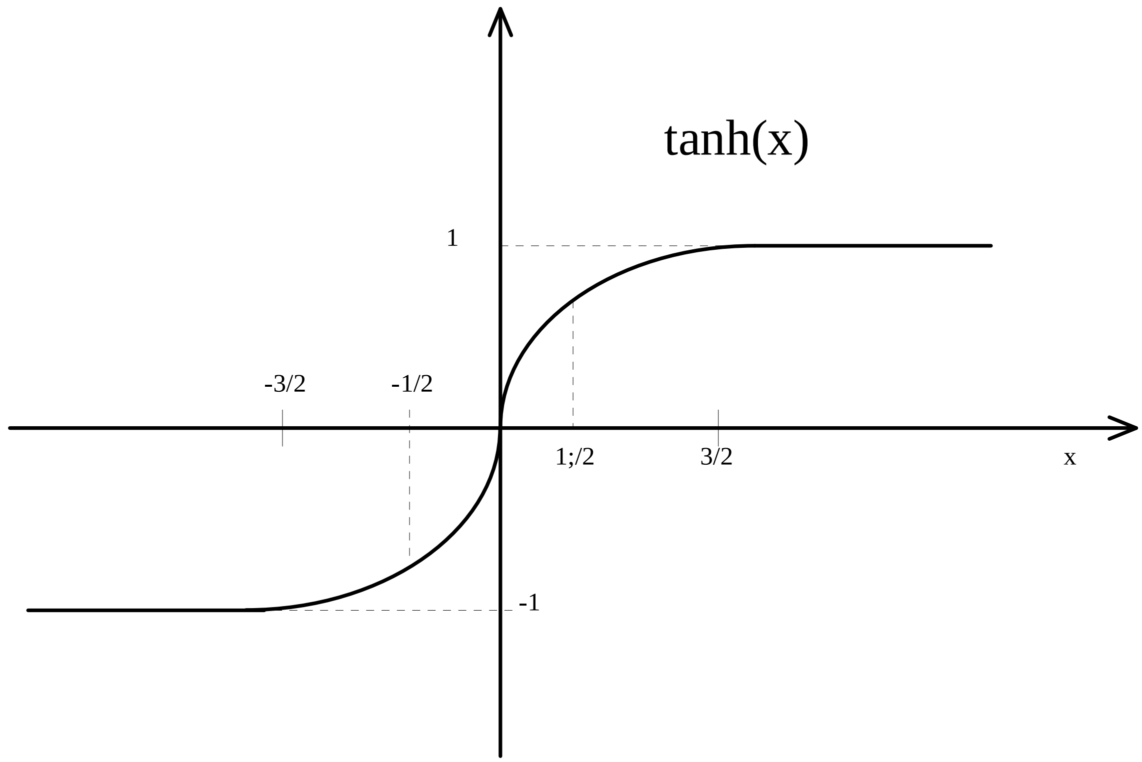
$$V_1 = V_0 + 10V_t \cdot \widetilde{\tanh}\left(\frac{\hat{V}_1 - V_0}{10V_t}\right)$$

the tanh( ) is approximated as follows:

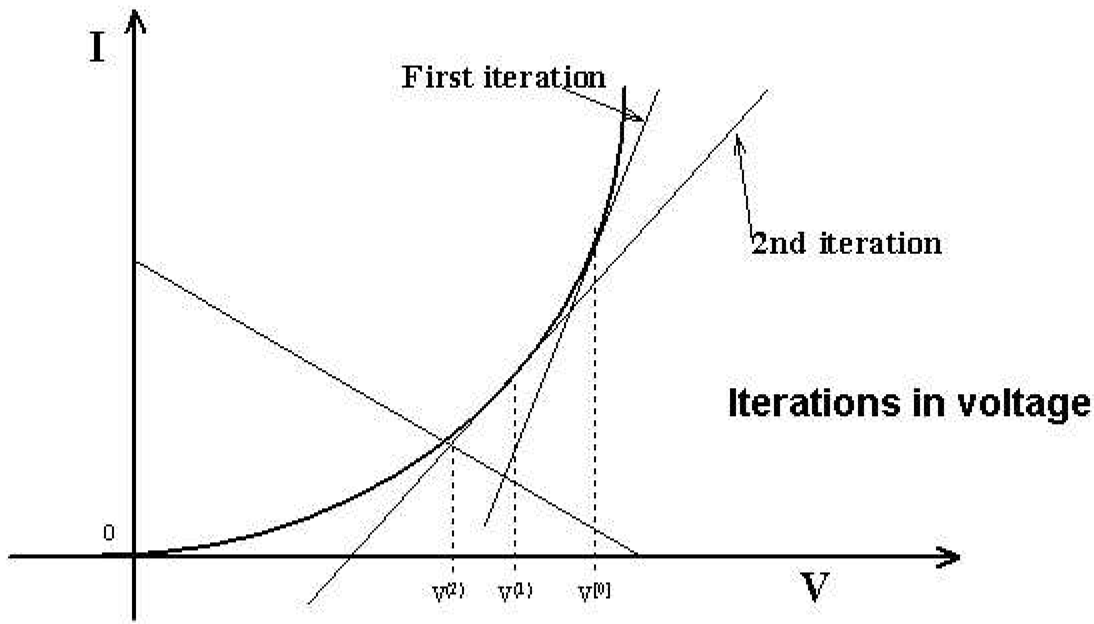
$$\widetilde{\tanh}(x) = \begin{cases} \pm 1 & |x| > \frac{3}{2} \\ x - \frac{x^3}{3} & \frac{1}{2} < |x| \le \frac{3}{2} \\ x & |x| \le \frac{1}{2} \end{cases}$$

Note that when $|x| \le \frac{1}{2}$ **than** $V_1 = \hat{V}_1$ **.**

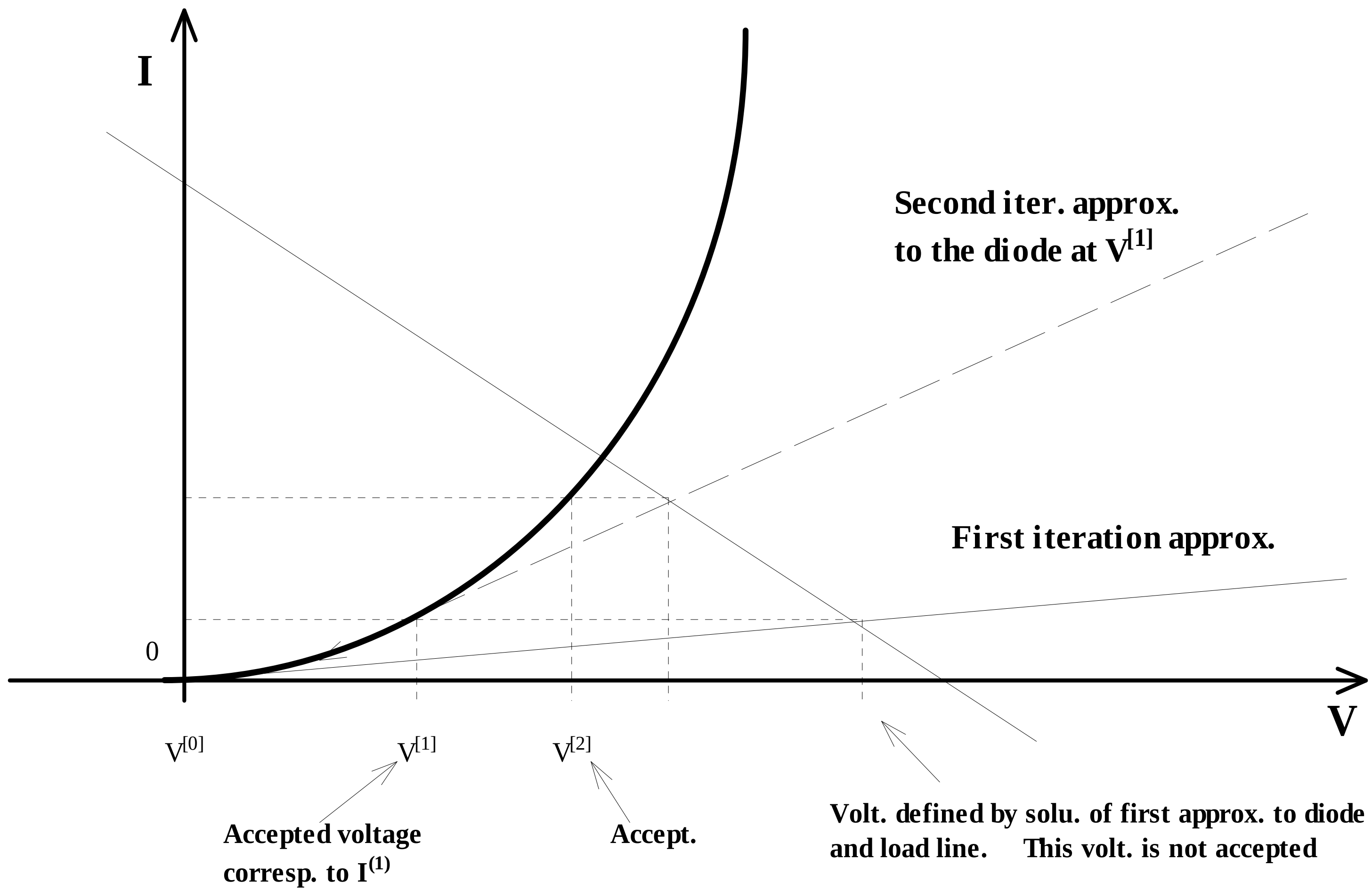
Graphical illustration



C) Iterating in voltage or current

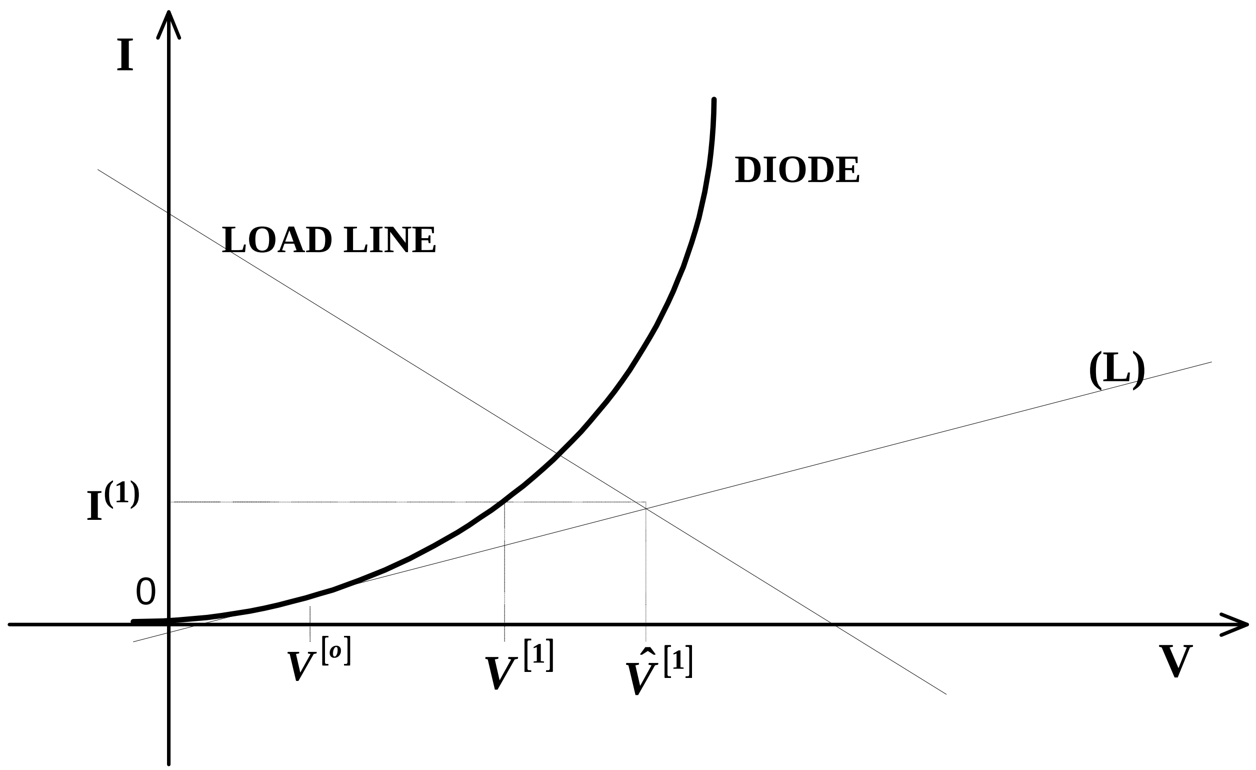


Iteration in current



Linearized relations

$$\hat{I} = I_s\left(e^{\frac{V^{(o)}}{V_t}} - 1\right) + \frac{I_s}{V_t} e^{\frac{V^{(o)}}{V_t}} \left(\hat{V}^{(1)} - V^{(o)}\right) \qquad \text{(L)}$$



$$\hat{I} = I^{(1)} \quad \text{in (L)}$$

$$\text{(Diode)} \quad I^{(1)} = I_s(e^{V^{(1)}/V_t} - 1) \quad \Rightarrow \quad v^{(1)}$$

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

After algebra (do as an exercise)

$$V^{(1)} = V^{(0)} + V_t \ln\left(1 + \frac{\hat{V}^{(1)} - V^{(0)}}{V_t}\right)$$

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 $\sqrt{2}[\Omega]$.

$$I_{crit} = \frac{V_t}{\sqrt{2}} \qquad \text{this is an arbitrary definition}$$

$$I = I_s(e^{V/V_t} - 1) \qquad \frac{V_{crit}}{V_t} >> 1$$

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

$$\Downarrow$$

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

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_{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 [*pA*]
VNTOL	default value is $1[\mu V]$

Convergence criterion

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

where ε_a represents ABSTOL or VNTOL

ε_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

$$\left|v_n^{(k+1)} - v_n^{(k)}\right| \le VNTOL + RELTOL \cdot \max\left\{\left|v_n^{(k+1)}\right|, \left|v_n^{(k)}\right|\right\}$$

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

$$\left|\dot{i}_\ell^{(k+1)} - \dot{i}_\ell^{(k)}\right| \le ABSTOL + RELTOL \cdot \max\left\{\left|\dot{i}_\ell^{(k+1)}\right|, \left|\dot{i}_\ell^{(k)}\right|\right\}$$

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.)

$$\tilde{I}_D = \frac{I_S}{V_t} e^{\frac{V_J^{(k)}}{V_t}} V_J^{(k+1)} + I_D - \frac{I_S}{V_t} e^{\frac{V_J^{(k)}}{V_t}} V_J^{(k)}$$

convergence criterion for I-V relations

$$\left|\tilde{I}_D - I_D\right| \leq ABSTOL + RELTOL \cdot \max\{I_D, \tilde{I}_D\}$$

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:

A**C, 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 strt1 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.