In this HW you will **extend** *myspice* to **solve** a system of **nonlinear equations**, i.e., perform df analysis of nonlinear circuits. You have already implemented the Newton iterative loop in HW# 2.

a) Add the code to **readin**, **setup**, and **load** diodes. The syntax is

\[ D_{\text{name}} \text{ node}(p) \text{ node}(n) \text{ model value} \quad \text{<the area field>} \]

The diode is described by

\[ I = Area \times I_S \left( \frac{V}{V_T} - 1 \right) \quad \text{where} \quad I_S = 1 \times 10^{-16} \text{ A}, \quad V_T = 0.0258 \text{ V}. \]

You do not have to implement a model “readin” and processing and can hardcode the parameter values \((I_S \text{ and } V_T)\).

b) Solve the circuits test[9-10].ckt using *myspice* and note the number of iterations to convergence.

c) Modify the diode stamping such that \( \text{pnjlim} \) is used for limiting the diode voltage before calculations and stamping. The \( \text{pnjlim} \) function from Spice3 is provided. Rerun test[9-10].ckt. How is the convergence of these examples with limiting?

d) Add the code to **readin**, **setup**, and **load** MOSFETs. The MOSFET syntax is

\[ M_{\text{name}} \text{ node}(d) \text{ node}(g) \text{ node}(s) \text{ node}(b) \text{ model value value} \quad \text{<W and L, respectively>} \]

The equations describing a NMOSFET are

\[
I_d = \begin{cases} 
  k \frac{W}{L} (V_{gs} - V_T) V_{ds} - \frac{V_{ds}^2}{2} (1 + \lambda V_{ds}) & \text{linear} \\
  \frac{k}{2} \frac{W}{L} (V_{gs} - V_T) V_{ds} & \text{saturation} \\
  0 & \text{cutoff}
\end{cases}
\]

\[ V_T = V_{TO} + \gamma \left( \sqrt{V_{SB}} + \phi - \sqrt{\phi} \right) \]

The following parameters should be used

- n-ch: \( V_{TO} = 0.7 \text{ V}, \quad k = 100 \mu A/\text{V}^2, \quad \lambda = 0.05 \text{ V}^{-1}, \quad \gamma = 0.5 \text{ V}^{1/2}, \quad \phi = 0.6 \text{ V} \)
- p-ch: \( V_{TO} = -0.9 \text{ V}, \quad k' = 50 \mu A/\text{V}^2, \quad \lambda = 0.05 \text{ V}^{-1}, \quad \gamma = 0.5 \text{ V}^{1/2}, \quad \phi = 0.6 \text{ V} \)

You can hardcode these values (and not implement a model) but must distinguish between a NMOSFET and PMOSFET using the model field that is NMOS or PMOS, respectively. A template is included (mosfet.c, mosfet.h) for the MOSFET load. This is not functional code for *myspice* but is a template for writing the MOSload function with appropriate modifications. The derivatives, current calculations, and stamps in this template are correct!

e) Solve the circuits test[11-16].ckt using *myspice* and note the number of iterations to convergence. List any convergence improvement techniques you had to use to get these circuits to converge. The code for the Spice3 limiting functions \( \text{limvds} \) and \( \text{fetlim} \) is provided.
f) Add the code to **readin**, **setup**, and **load** NPN BJTs. The syntax is

\[
\text{Qname node(c) node(b) node(e) model}
\]

The equations describing a NPN transistor are

\[
I_C = I_S \left( e^{\frac{v_{EB}}{V_T}} - 1 \right) - \frac{I_S}{\alpha_F} \left( e^{\frac{v_{CE}}{V_T}} - 1 \right)
\]

\[
I_E = -\frac{I_S}{\alpha_F} \left( e^{\frac{v_{EB}}{V_T}} - 1 \right) + I_S \left( e^{\frac{v_{CE}}{V_T}} - 1 \right)
\]

\[
I_B = -(I_C + I_E)
\]

The following parameters should be used:

\[V_T = 0.0258 \text{ V}, \quad I_S = 1 \times 10^{-16} \text{ A}, \quad \alpha_F = 0.99, \quad \alpha_R = 0.01\]

You can hardcode these values (and not implement a model). Use **pnjlim** for limiting the forward biased junction voltages before calculations and stamping.

g) Solve the circuits test[17-18].ckt using **myspice** and note the number of iterations to convergence.