模拟集成电路晶体管级仿真器设计项目报告

一、项目内容

基于C++语言,实现SPICE仿真器中的非线性电路求解器以及瞬态分析相关代码。

二、SPICE 语法规范

1. 注释：以 * 开头的行不会被解析
2. 器件模型：暂时只支持 MOSFET 模型，格式为：.MODEL <MODEL_ID> VT <VT> MU <> COX <COX> LAMBDA <> CJ0 <CJ0>
3. 电阻：R <tag> <n1> <n2> <value>
4. 直流电压源：V <tag> <n1> <n2> DC <value>
5. 直流电流源：I <tag> <n1> <n2> DC <value>
6. MOSFET：M <tag> <Drain> <Gate> <Source> <type> <Width> <Length> <model_ID>
7. 节点数据输出：.PLOTNV <node>

三、非线性求解器功能实现

实现了原有代码框架下进行非线性求解所必需的函数，并进一步完成了电压初始化函数 initialValue()的实现。输出结果重定向至与测试网表同名的.lis 文件中。

1. MOSFET 模型建立

本项目中采用的 NMOS 及 PMOS 模型如下：

$NMOS\ model$:

$Cut-off$: $V_{gs} \le V_T$

$$I_{ds} = 0$$

$Linear$: $V_{gs} < V_T \ \ and\ V_{ds} \le V_{gs} - V_T$

$$I_{ds} = \mu C_{ox} \frac{W}{L}\left((V_{gs} - V_T)V_{ds} - \frac{1}{2}V_{ds}^2\right)$$

$Saruation$: $V_{gs} > V_T\ and\ V_{ds} > V_{gs} - V_T$

$$I_{ds} = \frac{1}{2}\mu C_{ox}\frac{W}{L}(V_{gs} - V_T)^2(1 + \lambda V_{ds})$$

$PMOS\ model$:

$Cut-off$: $V_{sg} \le V_T$

$$I_{sd} = 0$$

$Linear$: $V_{sg} < V_T \ \ and\ V_{sd} \le V_{sg} - V_T$

$$I_{ds} = \mu C_{ox} \frac{W}{L}\left((V_{sg} - V_T)V_{sd} - \frac{1}{2}V_{sd}^2\right)$$

$Saruation$: $V_{sg} > V_T\ and\ V_{sd} > V_{sg} - V_T$

$$I_{ds} = \frac{1}{2}\mu C_{ox}\frac{W}{L}(V_{sg} - V_T)^2(1 + \lambda V_{sd})$$

2. 非线性求解器的结构

（1）updateMatrix()

该函数将计算好参数的非线性-线性元件对 std::map<const NonlinearComponent *, std::vector<BasicComponent *> > 传递给建立 MNA 方程的模块 equation.cpp，实现 MNA 方程的更新。

（2）getParameter()

getParameter() 函数输入是类 nonlinearMOS 的对象，根据传入对象中储存的 MOSFET 参数，计算出对应工作点下的线性元件参数，于 MOSFET 而言即为一个电阻的阻值、一个电流源的电流值和一个压控电流源的跨导，将其存储在 std::vector<BasicComponent *> 中，返回给 updateMatrix()函数。

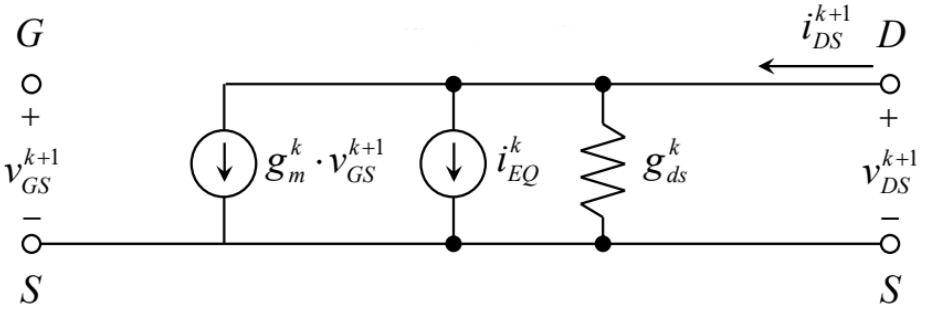


图 1 NMOS 等效线性模型

（3）updateParameter()

根据前一次牛顿迭代的结果更新 mos 器件对应的节点电压值（漏栅源）。

（4）newtonMethod()

牛顿迭代的主体函数，设定最大迭代次数为 21 次，达到最大迭代次数后终止牛顿迭代并输出最终结果，迭代的过程中，每一次迭代均调用线性求解器 linearSolver 的成员函数 solve() 进行求解，综合考虑到线性求解速度和求解精度，在牛顿迭代中默认使用 DirectPrecise 选项进行线性求解。

（5）checkError()

checkError() 函数功能是求解相邻两次牛顿迭代结果解空间向量的第二范数，当该第二范数值小于 $1 \times 10{-}5$ 时，结束牛顿迭代，并输出结果。

（6）outputResults()

用于简单结果输出。

（7）initialValue()

用于设定 mos 管各结点的初始电压值。但仍未实现从标准输入传入形参。在 DC 分析中仍采用预设定的初值，该初始化函数不起作用（从 analyze_dc 传入空的初始电压数组）。而在瞬态分析中将 mos 管电压初始化为上一次迭代的结果。

函数名	功能	依赖关系
updateMatrix()	更新非线性-	getParameter()

	线性元件对	
getParameter()	更新替换用线性元件	/
updateParameter()	存储前一次牛顿迭代节点电压	/
newtonMethod()	牛顿迭代法	checkError()/updateParameter()/updateMatrix()
checkError()	计算前后两次迭代结果误差	/
initialValue()	初值设定	/
outputResults()	结果输出	/

表 1 部分函数功能与依赖关系

3. 工作流程

首先经过一个 for 循环遍历 equation 中存放的 transformedComponents 的元素，在统合了每个 mos 管信息并设定 mos 管节点电压初始值后，将其记录进类私有成员变量 mosComponents，然后利用 updateMatrix 函数更新非线性-线性元件对，在遍历结束后，生成更新后的 MNA 方程，利用设定的初始值和 mos 管参数，调用 newtonMethod() 函数进行牛顿迭代法，在迭代终止之后，利用 outputResults() 输出结果。

三、瞬态分析功能实现

1. 对于电容和电感的处理

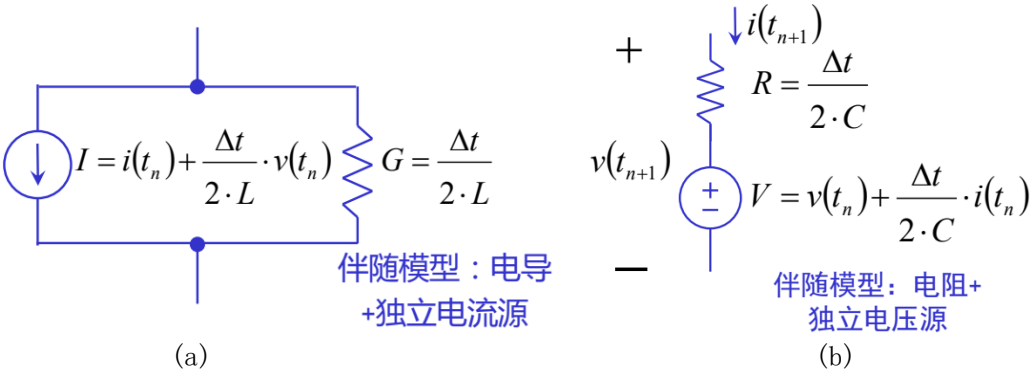


图 2 电感(a)和电容(b)的梯形法伴随模型

在该项目中，统一采用了梯形法的迭代模型来处理电容和电感。对于电感，将其等效为一个电流源和一个电感的并联，这部分的处理比较简单；而对于电容，将其等效为一个电阻和独立电压源的串联，在此时，便会引入一个新节点，需要利用 equation.applyNewNode() 函数来添加新节点。

梯形法的截断误差如下式所示：

电容的截断误差为：

$$\epsilon \approx -\frac{(\Delta t)^2}{12 \cdot C} \cdot [i(t_{n+1}) - i(t_n)]$$

电感的截断误差为：

$$\epsilon \approx -\frac{(\Delta t)^2}{12 \cdot L} \cdot [v(t_{n+1}) - v(t_n)]$$

2. 部分函数实现

（1）newtonMethod()

牛顿迭代的主体函数。用于在时间轴上求解非线性电路的瞬态响应。在该函数当中通过两次循环遍历所有的非线性-线性元件对中的电容电感部分，第一次循环中根据本次迭代的初始值和结果值，求得调整参数 q，若 q 值不满足要求，则放弃本次迭代结果，对时间步长进行**向下调整**并且回溯至上一次迭代的状态。

在时间步长满足要求后，可以进入第二次循环，表明本次迭代结果可以采纳，根据第一次循环中记录的调整参数 q 的最大值对时间步长进行**向上调整**的判断，调用 updateInductor() 和 updateCapacitor() 函数更新电容和电感的非线性-线性元件对，然后调用 equation 类的 equation-AddTransformed() 函数更新 MNA 方程。

（2）timeStepControlC()/timeStepControlL()

timeStepControlC() 和 timeStepControlL() 函数用于控制时间步长，根据电路的特性，时间步长的选取会影响到瞬态分析的精度和效率，此函数的形参包括一个动态步长变换后的时间步长的引用，即 float &tNow，将其和当前的时间步长进行比较，用一个控制信号 flagPaceChanged 传递是否进行步长修改的信息，若 flagPaceChanged 为 true，则说明时间步长需要进行修改，否则不需要进行修改。

（3）getParameterC()/getParameterI()

根据输入的电容和电感器件的节点电压值，对伴随模型中的电压（流）源和电阻值进行相对应的更新，也就是意味着，若该函数被调用则该次的迭代结果被采纳。

（4）updateVC_C()/updateVC_I()

updateVC_C() 和 updateVC_I() 函数从本次迭代非线性求解器输出的解空间向量 Voltages 中提取对应原件的节点电压或电流值，并且更新对应对象的成员变量值，在这里仅提取节点电压值，其节点电流值是通过节点电压对伴随模型部分进行电路分析得到的。

3. 动态步长控制

（1）动态步长控制策略

在该项目中，所用的调整参数 q 的表达式如下所示：

$$q = \left| \frac{\xi^{(n+1)}}{\xi + \xi_r \max\{|x_{n+1}|, |x_n|\}} \right|$$

其中 $\xi(n+1)$ 为截断误差，ξ 为预设的绝对误差限，ξr 为设定的相对误差限，$|xn+1|$，$|xn|$ 为本次迭代和上次迭代的节点电压（流）值。其中，绝对误差限 ξ 和相对误差限 ξr，都应该动态变化而不是定值. 在本项目中将相对误差限设置为该

电（容）感的 100000 分之一，绝对误差限设置为该电（容）感的 1000 分之一。

若 q>1，则说明局部截断误差大于设定的误差限，步长偏大；若 q<1，则说明局部截断误差已经小于设定误差限，若 q 远小于 1 则说明步长过小。 根据 q 对步长 h 进行动态调整。此外，为了避免时间步长变化过于剧烈，同时考虑到输出图形的平滑程度，为每次的调整范围设置了上下限：

$$q > 1 \text{ 时，} h_{n+1} = h_n \cdot \max\left(\left(\frac{1}{q}\right)^{\frac{1}{2}},\ 0.3\right)$$

$$q < 0.2 \text{ 时，} h_{n+1} = \min\left(2h_n,\ h\_oringin\right)$$

$h_oringin$为程序运行时规定的时间步长。

将 q>1 和 q<0.2 分开进行讨论的原因在于：在求解实际电路时，如果电路中含有多个非线性元件，在牛顿迭代的第一个循环中（动态控制步长）若同时对每个元件进行步长过小和步长过大的判断，对时间步长的调整易发生冲突，导致程序陷入死循环。

为避免上述问题的发生，在牛顿迭代的第一个循环中只进行时间步长缩小的判断与调整，并记录本次循环中 q 的最大值 q_max。而当迭代结果可以被采纳时，再基于此次循环的 q_max 进行时间步长增大的判断与调整。

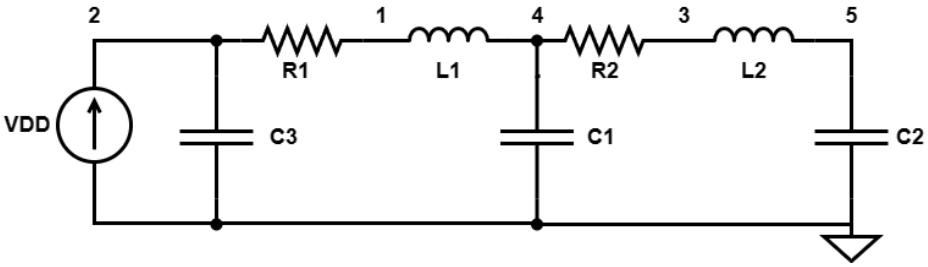
此外，为了程序的稳定性以及避免过于频繁的调整导致运行速度下降，在增大步长时采用了较为保守的判定条件（q<0.2）和步长调整方法(在原步长的基础上乘 2，而不是$(1/q)^{1/2}$)。

（2）关于动态步长控制的效果

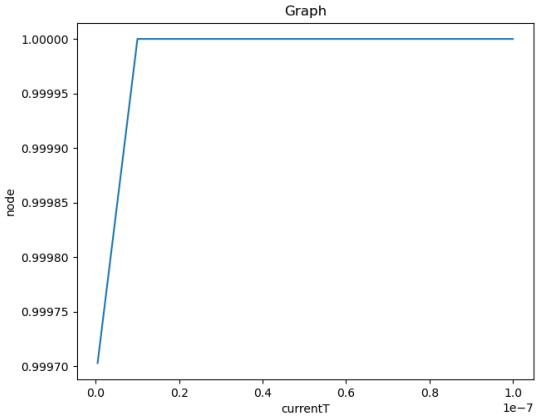
“对于 test8.sp，其电容电感均处于 1e-12 数量级，电阻最小值为 0.05，因此估算该电路的时间常数约在 1e-13 至 1e-14 数量级，在未添加动态步长控制功能的情况下，要使瞬态分析顺利进行，时间步长至少要选取在 1e-15 数量级，而添加了动态步长控制功能后，可以将时间步长甚至选取在 1e-8 数量级，虽然会有一定偏差，但是基本符合理论预期并且可以成功收敛，并且在仿真时间上有极大的提升。”

在原项目文档中的这段描述基础上，动态时间步长调整有缩小和增大两种情况。若只对步长进行缩小的调整，则程序一定能够收敛（不会发生多个器件），且在稳态下输出的稳定性较好，缺点是迭代次数多，运行时间长；而若加入增大步长的调整，则可以显著减少迭代次数和运行时间，缺点是输出数据稳定性相对较差。

另外，本项目对 test8.sp 进行仿真的迭代次数和花费时间比原项目少了约一半，且误差也相对较小。猜测原因在于动态增大时间步长的策略的具体实现不同。

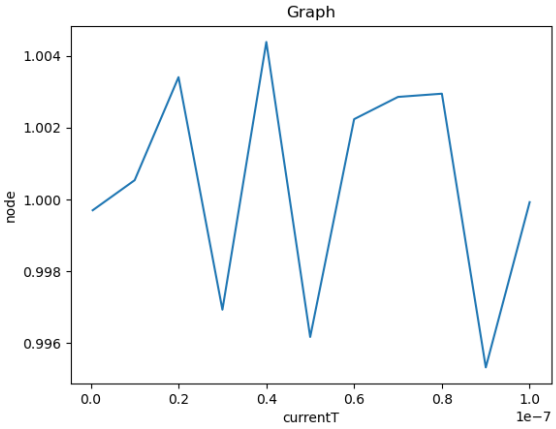


（a）test8.sp 对应的电路图




```
Transient analysis setting:
deltaT: 1e-08 s
tFinal: 1e-07 s

After 666918 iterations, Newton Method was ended.
The voltages of the nodes:
node.1  1
node.2  1
node.3  1
node.4  1
node.5  1
Running time = 165.353s
```

（b）只对步长进行缩小的调整




```
Transient analysis setting:
deltaT: 1e-08 s
tFinal: 1e-07 s

After 113226 iterations, Newton Method was ended.
The voltages of the nodes:
node.1  0.997563
node.2  1
node.3  0.997077
node.4  0.997297
node.5  0.999929
Running time = 22.0008s
```

（c）加入步长增大的调整

```
E:\AnalogSPICE\AnalogSPICE\cmake-build-debug\AnalogSPICE.exe ./tests/test8.sp
test started
Please choose the mode, 1 for DC Analyze and 2 for Transient Analyze
2
Please input the deltaT:
1e-8
Please input the tFinal:
1e-7
After 224361 iterations, Newton Method was ended.
The voltages of the nodes:
node.1  1.0846
node.2  1
node.3  1.02624
node.4  1.02606
node.5  0.943662
Running time = 52.246s
```

（d）原项目运行结果

图 3 关于 test8.sp 中的不同动态步长控制策略仿真结果

（3）关于仿真结果的观测

可通过.PLOTNV <NODE>命令输出多个指定结点的电压图像数据。求解完成后联合 python 脚本，根据提示输入需处理的.lis 文件名和需要观察的节点即可完成作图。但目前只实现了电压数据的输出，电流数据的输出仍待后续改进。

四、基于状态预测对于瞬态分析的改进

为提高瞬态分析的速度，基于原项目的框架，在非线性求解器中增加了电压初始化 initialValue()部分。其作用在于：每一次瞬态分析迭代中对非线性电路进行分析时，基于上一次迭代的结果对非线性电路进行 MNA 方程的求解。

对于仿真时间较长的 buffer 电路（测试文件中的 buffer.sp）而言，若每次迭代都从人工设定的初值开始求解，非线性求解器需要 11 次迭代才能得到当前时刻电路的 MNA 方程解；而若基于上一次迭代的求解结果进行分析，每次求解 MNA 方程实际只需要 2 次迭代，迭代次数大大减少，而这对瞬态分析的速度提升是显著的。

在步长为 10^{-4}s,仿真时长为 10^{-2}s 时，由图 11 可知，对该电路进行瞬态分析所花费的时间缩短了 4 倍！

```
currentT: 0.00990018
node: 0.328866
iteration reached: 30000
currentT: 0.0100002
node: 0.331394
After 30022 iterations, Newton Method was ended.
The voltages of the nodes:
node.101    3
node.102    3
node.103    3
node.104    0.343018
node.107    -6.54754e-05
node.115    1.21865
node.116    0.345541
node.117    0.560037
node.118    0.331394
Running time = 66.9426s
```

```
After 2 iterations, Newton Method was ended.
After 2 iterations, Newton Method was ended.
After 2 iterations, Newton Method was ended.
After 2 iterations, Newton Method was ended.
currentT: 0.0100001
node: 0.331372
After 29988 iterations, Newton Method was ended.
The voltages of the nodes:
node.101    3
node.102    3
node.103    3
node.104    0.34302
node.107    -7.05693e-05
node.115    1.21193
node.116    0.345539
node.117    0.565622
node.118    0.331372
Running time = 16.5836s
```

图 4 优化前（左）与优化后（右）瞬态分析结果比较

此外，当仿真时长设置为 0.5s 时，若每次迭代都从初始情况重新求解，在虚拟机中仿真时长达到约 0.2s 就需要花费大于 1 个小时的时间，且程序还会在 0.2s 附近崩溃，原因不明。而优化后的程序则不会发生这种情况，可顺利完成仿真时长为 0.5s 的要求，花费的时间约为 13 分钟（774.393 秒）（见图 12）。

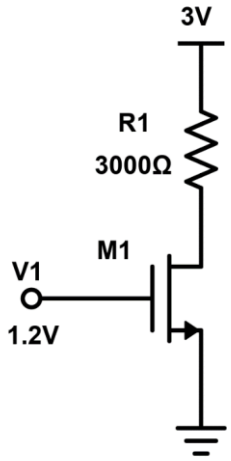
经验证，该优化方法同样对对其他电路的仿真速度有较大提升。

五、测试样例

1. 非线性电路 DC 分析

（1）带上拉电阻的 NMOS 电路

带上拉电阻的 NMOS 电路，漏节点为 2，栅节点为 1，高电平节点为 3，对应 tests 文件夹中的 test3.sp。

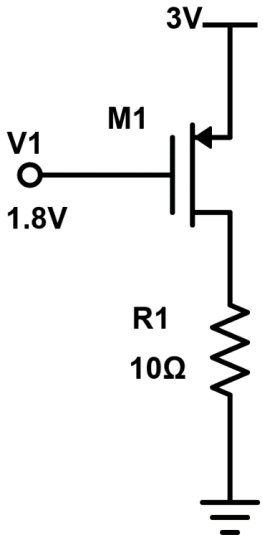



```
DC analysis started...
After 3 iterations, Newton Method was ended.
The voltages of the nodes:
node.1  1.2
node.2  2.96968
node.3  3
Running time = 0.000236s
```

图 5 电路图及 DC 仿真结果

（2）带下拉电阻的 PMOS 电路

带下拉电阻的 PMOS 电路，漏节点为 2，栅节点为 1，高电平节点为 3，对应 tests 文件夹中的 test4.sp。

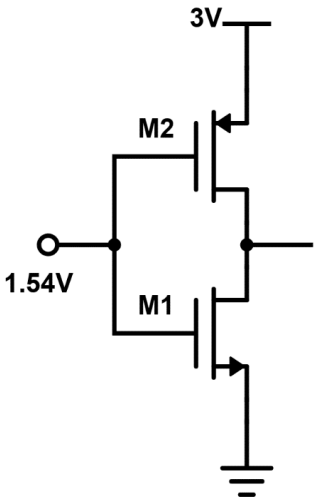



```
DC analysis started...
After 3 iterations, Newton Method was ended.
The voltages of the nodes:
node.1  1.8
node.2  4.99017e-05
node.3  3
Running time = 0.000193s
```

图 6 电路图及 DC 仿真结果

（3）CMOS 反相器电路

简单的反相器，输入节点为 1，输出节点为 2，高电平节点为 3，对应 tests 文件夹中的 inverter.sp

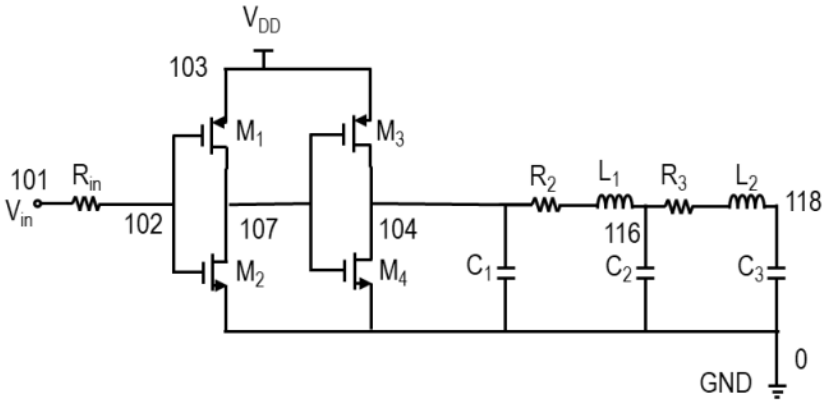



```
DC analysis started...
After 6 iterations, Newton Method was ended.
The voltages of the nodes:
node.1   1.54
node.2   1.5
node.3   3
Running time = 0.000379s
```

图 7 电路图及 DC 仿真结果

（4）缓冲器 Buffer(两级反相器级联)

缓冲器 Buffer，对应 tests 文件夹中的 buffer.sp。




```
DC analysis started...
After 21 iterations, Newton Method was ended.
The voltages of the nodes:
node.101    3
node.102    3
node.103    3
node.104    2.99999
node.107    0
node.115    2.99999
node.116    2.99999
node.117    2.99999
node.118    2.99999
Running time = 0.002786s
```

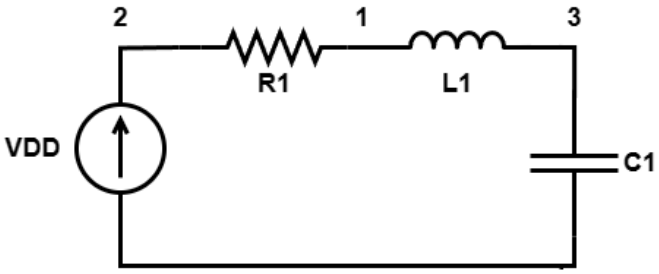
图 8 电路图及 DC 仿真结果

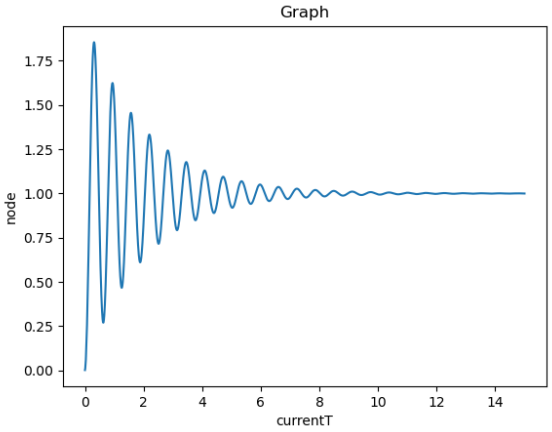
2. 瞬态分析

（1）简单的 RLC 串联电路(test7.sp)

仿真步长：10^{-3}s

仿真时间：15s





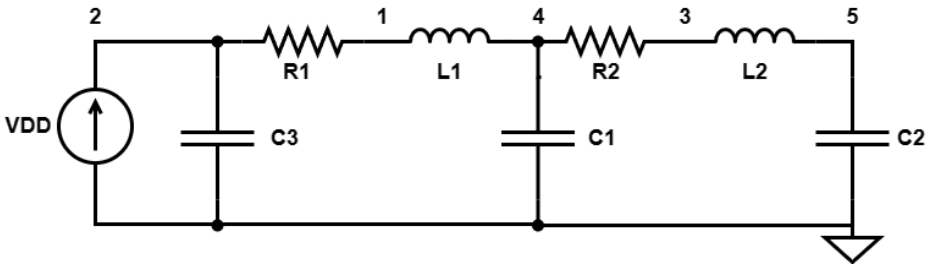

```
Transient analysis setting:
deltaT: 0.001 s
tFinal: 15 s

After 14998 iterations, Newton Method was ended.
The voltages of the nodes:
node.1  1.00004
node.2  1
node.3  0.999747
Running time = 1.24789s
```

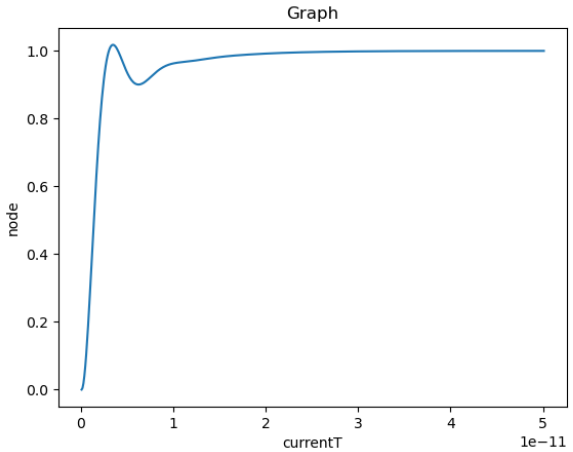
图 9 电路图及瞬态仿真结果，输出结点 3 的电压

（2）较复杂的 RLC 串联电路（test8.sp）

仿真步长：10^{-13}s

仿真时间：5×10^{-11}s






```
Transient analysis setting:
deltaT: 1e-13 s
tFinal: 5e-11 s

After 501 iterations, Newton Method was ended.
The voltages of the nodes:
node.1  0.999942
node.2  1
node.3  0.999699
node.4  0.9997
node.5  0.999944
Running time = 0.137129s
```

图 10 电路图及瞬态仿真结果 1，输出结点 5 的电压

（3）CMOS 反相器（inverter.sp）

输入节点为 1，输出节点为 2，高电平节点为 3，输出节点接 1fF 的电容负载。

仿真步长：10^{-10}s,

仿真步长：5×10^{-7}s

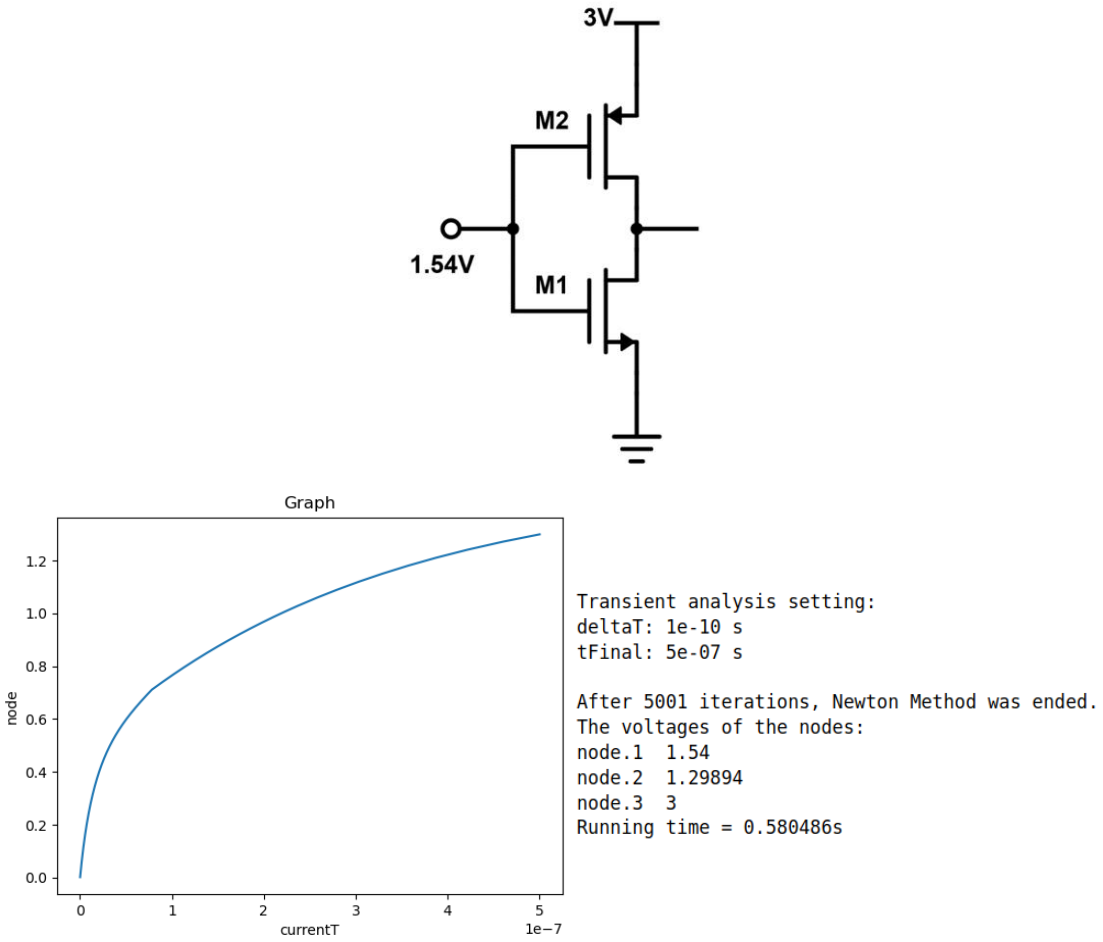
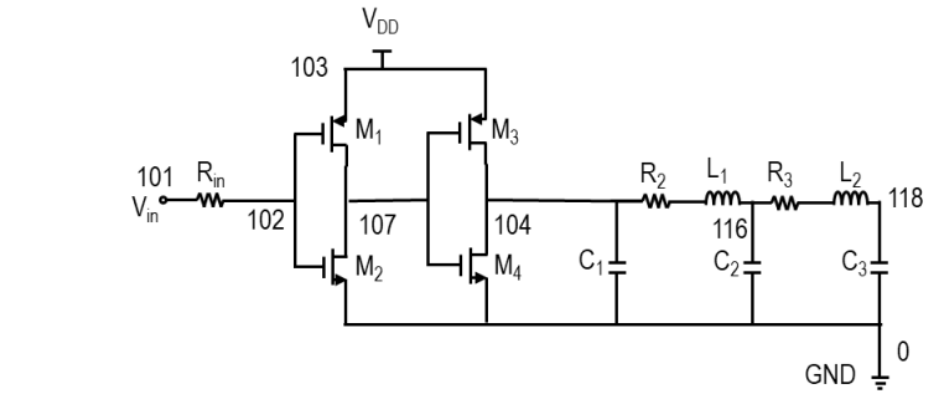


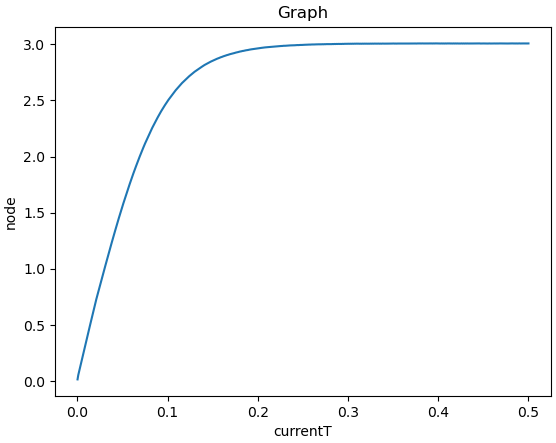
图 11 电路图及瞬态仿真结果，输出结点 2 的电压

（4）缓冲器（buffer.sp）

仿真步长：10^{-3}s

仿真时间：0.5s






```
currentT: 0.5
node: 3.00565
After 1545642 iterations, Newton Method was ended.
The voltages of the nodes:
node.101    3
node.102    3
node.103    3
node.104    2.99944
node.107    0
node.115    3.45004
node.116    3.0051
node.117    2.86476
node.118    3.00565
Running time = 774.393s
```

图 12 电路图及瞬态仿真结果，输出结点 118 的电压