

电磁学小论文
演示文稿

电路的计算机仿真

邱哲儒
PB15000034

目录

一.理论讨论

- 1.从最简单的情况开始
- 2.对恒压源的处理
- 3.受控电源的处理
- 4.非线性电路的迭代求解
- 5.电路的时域动态分析

二.计算机实现

- 1.编写的程序XLWSpice
- 2.应用举例
 - i)包含二极管的算例
 - ii)RC,RLC电路暂态过程的算例
 - iii)大型电阻网络的计算
- 3.前景展望

1.最简单的情况：只有电流源与线性电阻

用节点法处理

回路法需要处理图的问题，确定独立回路的过程较为复杂，节点法建立方程的过程更直观。

据基尔霍夫电流定律与欧姆定律，对节点 k 有： $-u_k \sum_{j,l} G_{jk}^l + \sum_j \sum_l G_{jk}^l u_j = I_{src,k}$

把所有的方程写成矩阵的形式

$$Au = i, \quad i = \begin{bmatrix} I_{src_1} \\ I_{src_2} \\ \vdots \\ I_{src_n} \end{bmatrix}, \quad A_{kp} = \begin{cases} -\sum_{j,l} G_{jk}^l, & k = p \\ \sum_l G_{pk}^l, & k \neq p \end{cases}$$

2.电压源的处理

实际建模中常需要电压源，处理电压源的问题常用的方法包括“列表法”与改进节点电压法。本质都是增加一个节点电压方程，增加一个未知电流。

以下采用改进节点电压法(Modified Nodal Analysis)

- 对每个节点有 $-u_k \sum_{j,l} G_{jk}^l + \sum_j \sum_l G_{jk}^l u_j + \sum_l I_{src_l} (\delta_{node1,k} - \delta_{node0,k}) = I_{src,k}$式(*)
- 对矩阵方程 $Au = i$ 进行扩充,以包含新的未知量与方程。

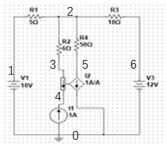
$$M \begin{bmatrix} U_k \\ I_{U_{src}} \end{bmatrix} = \begin{bmatrix} I_{src} \\ I_{U_{src}} \end{bmatrix}$$

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad B_{kl} = \sum_l \delta_{node1,k} - \delta_{node0,k} \quad C = B^T \quad D = I$$

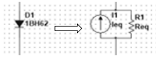
- 对方程组再进行扩充，加入受控电源的方程。
- 此时 $C \neq B^T, D = I$ 也不再成立。

• 举例

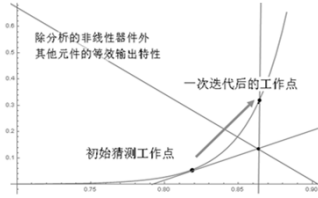
对于如图所示的电路
包含多种线性器件
按前述方法处理



在每次迭代中非线性器件可以等效为一个电流源并联一个电阻



如果取 $G_{eq}(u) = \frac{di}{du} I_{eq}(u) = i(u) - uG_{eq}(u)$ 对电路方程进行求解就与求解(*)式完全相同, 此时问题完全化为线性情况。



5. 电路的时域动态分析

对于伏安特性包含电流或电压导数的动态元件, 比如电容, 电感, 可以简单地通过引入复数参数通过相量法处理问题。但是这样的方式很有局限性

对电路微分代数方程组进行数值积分是更为实用的方法。使用简单的梯形法进行数值积分。

$$x_{n+1} = x_n + timestep \times \left(\frac{\dot{x}_n + \dot{x}_{n+1}}{2} \right) \dots (*)$$

以电容器为例: 其伏安特性为 $I = C \frac{du}{dt}$, 代入(*)式中整理得

$$U_{n+1} = U_n + \frac{timestep}{2C} (I_n + I_{n+1})$$

在每一个时间步长中, 其前一项可以看作一个阻值为 $\frac{timestep}{2C}$ 的电阻, 后一项可以看作一个电流为 $-\frac{2C}{timestep} U_n - I_n$ 的电流源, 两个元件并联。这样在每一次迭代中问题都可以化归为静态问题。

1. 编写程序XLWSpice

• XiaoLunWenSpice

特点:

- 使用C++语言
- 代码精简, 不含外部库只有一千两百余行
- 使用高性能线性代数库Eigen进行稀疏矩阵的运算
 - 通过PardisoSupport模块调用Intel MKL Pardiso LU进行方程求解 (兼顾Eigen的友好接口, MKL的优秀性能)

• 输入为网表数据, 详情见源码注释

• 输出示例:

```

.DC 1e-4 1e-3
VSR1 1 0 10
R 1 2 5
R 2 3 6
I 4 0 1
R 5 2 50
R 2 6 10
VSR2 6 0 12
F 0 5 3 4 1

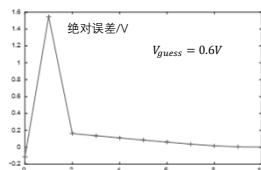
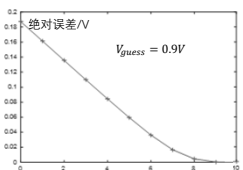
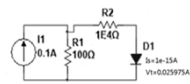
*****NON LINEAR DC ANALYSIS*****
ABSOLUTE CONVERGENCE LIMIT:0.0001
RELATIVE CONVERGENCE LIMIT:0.001
%HAVING NODES:14
%NON LINEAR DC ANALYSIS ITERATION 0
%NODE VOLTAGE
1,1
2,0.122817
3,0.581027
4,0.296156
5,0.245634
6,0.446926
7,0.357962
8,0.317929
9,0.401789
10,0.370829
11,0.35019
12,0.387601
13,0.373614
14,0.361892
%NODE VOLTAGE
1,10
2,17.3333
3,23.3333
4,23.3333
5,67.3333
6,12
%VOLTAGE SOURCES CURRENT
0,1.46667,VSR1
-1,4.1897,VSR2
1,0.533333,VSR2
%STOP ITERATION CRITERIA MET
    
```

2. 算例1: 包含晶体二极管的简单电路

由固体物理知识:

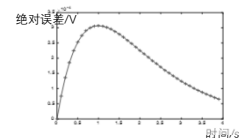
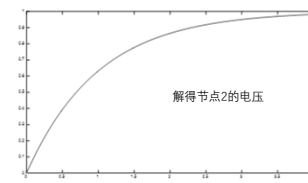
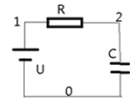
$$i(u) = I_s \left(e^{\frac{u}{V_T}} - 1 \right), I_s, V_T \text{ 为与器件特性有关的常数}$$

$$G_{eq}(u) = \frac{di}{du} = \frac{I_s}{V_T} e^{\frac{u}{V_T}} \quad I_{eq}(u) = I_s \left[\left(1 - \frac{u}{V_T} \right) e^{\frac{u}{V_T}} - 1 \right]$$



算例2: 求解RC电路的暂态过程

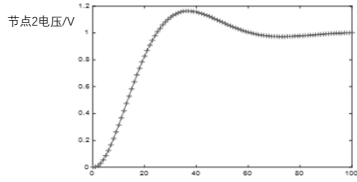
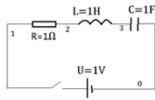
timestep = 0.1s, R = 1Ω, C = 1F, U = 1V
共求解40步
与理论解: $1 - e^{-t}$ 符合很好



算例3:RLC电路的暂态过程

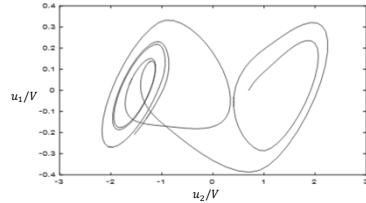
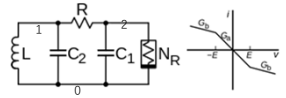
0.1s步长, 求解100步

电路处在欠阻尼状态,结果与理论值符合良好



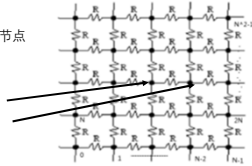
算例4: 蔡氏电路

求解10s步长0.01s
可以明显看到蔡氏电路的混沌式的振荡。
有“双涡旋”结构



算例5: 运用于大型电阻网络问题

- 对于无限大的网格, 两节点间电阻理论值应为 $\frac{R}{2}$ (不妨设 $R=1\Omega$)
- 在这里用有限个电阻去逼近, 设正方形网格一条边上有 N 个节点
- 电压源放在 $(N+1)$ 点与 $(N+1)$ 点之间



对于大型规律性的电阻网络, LU算法直接求解效果最佳。

Intel MKL支持多核并行计算, 大大提高求解性能, PardisoLU较Eigen 内置SparseLU至少快一倍。

BICGSTAB等迭代求解器, 因Eigen库的限制, 矢量矩阵乘法无法并行化, 实验证明对于大型规律性的电阻网络效果不好, $N=750$ 时需要超过50s

用大型电阻网络问题的求解作为性能评价

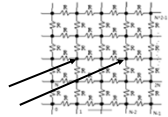
测试环境:
Intel(R) 64 Compiler 16.0.1.146
关键选项/O3 /Qparallel
Eigen 3.2.8
Intel MKL 11.3
Windows Server 2012 R2
Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz
8G RAM

点数N	6	8	10	12	20	40	60	
电阻/ Ω	0.51940	0.50975	0.50592	0.50328	0.50139	0.50035	0.50015	
求解时间 (PardisoLU/s)	0.11	0.12	0.12	0.12	0.12	0.13	0.16	
点数N	80	120	240	320	500	750	1000	1200
求解时间 (PardisoLU/s)	0.18	0.26	0.72	1.23	3.12	7.10	13.0	19.0
求解时间 (SparseLU/s)	0.08	0.17	0.93	1.69	5.31	14.9	33.7	54.1

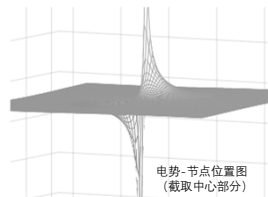
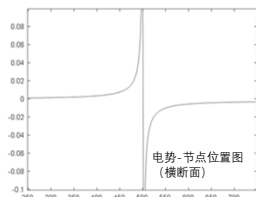
总体性能令人满意

计算相距两个电阻处的等效电阻

在 $N=1000$ 时加电压源: $V = 500499.5005011$
共一百万节点
电流为 $1.37597A$, 相当于 0.72676Ω
与理论值 $(2 - \frac{4}{\pi})\Omega \approx 0.72676\Omega$ (见文献[5]) 一致

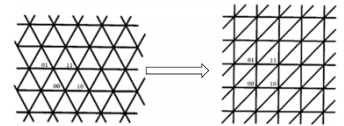


直观分析计算结果

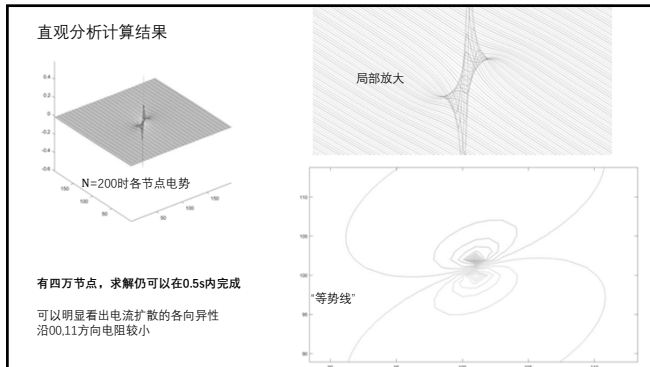


三角形无限网格

在输入中加入对角线项



计算得 $n=20$ 时 $R=0.465807\Omega$, $n=120$ 时 $R=0.461472\Omega$, $n=200$ 时 $R=0.461395\Omega$
与理论值为 $(\frac{8}{3} - \frac{4\sqrt{3}}{\pi})\Omega$ (见文献[5]), 约 0.461351Ω 相吻合
求解耗时不长, 性能满足要求



3.前景展望

- 牛顿法求解非线性电路方程时时有不收敛的情况
 - 尤其是电路中包含二极管等指数伏安特性的器件时，若初值选取不当，迭代过程中会产生非常大的电流和 I_{eq} ，导致求解失败。
 - 目前处理方法：限制迭代时尝试工作点的变化
- 目前时域动态分析使用的梯形法相对粗糙，可以改用高阶的，更稳定的其他算法
 - 三阶、四阶隐式Adams公式
 - $y_{n+1} = y_n + \frac{1}{6}(5f(x_{n+1}, y_{n+1}) + 8f(x_n, y_n) - f(x_{n-1}, y_{n-1}))(三阶)$
- 实现时域动态分析的时间步长(timestep)的自动选取与自适应调整
 - 根据各元件参数量级预估时间步长
 - 用较大的步长尝试求解，若失败或与初值相差过大，再采用更小步长
 - 根据本次迭代中计算结果与上次迭代结果的差决定
- 增加对三端非线性器件的支持，BJT,MOSFET...
- 其他细节上的性能优化
- 改进用户界面

获取源码和完整版演示文稿请访问

<http://home.ustc.edu.cn/~qzx/XLWSpice>

关于Eigen请访问

<http://eigen.tuxfamily.org/>

关于Intel® MKL请访问

<https://software.intel.com/en-us/intel-mkl>



参考文献

- [1] Kincaid D R, Cheney E W. Numerical analysis: mathematics of scientific computing[M]. American Mathematical Soc., 2002.
- [2] 邱关源. 电路(第五版) [M]. 高等教育出版社
- [3] Vladimirescu A. The SPICE Book[M]. 1994. John Wiley&Sons, New York, 1994.
- [4] 张运华等, 数值计算方法与算法(第二版)[M]. 科学出版社
- [5] Atkinson D, Van Steenwijk F J. Infinite resistive lattices[J]. American Journal of Physics, 1999, 67(6): 486-492.
- [6] Denardo B, Earwood J, Sazonova V. Experiments with electrical resistive networks[J]. American Journal of Physics, 1999, 67(11): 981-986.
- [7] Cserti J. Application of the lattice Green's function for calculating the resistance of an infinite network of resistors[J]. American Journal of Physics, 2000, 68(10): 896-906.