DOI: 10.22184/1992-4178.2023.222.1.110.120

Анализ электрических схем по постоянному току методом Ньютона – Рафсона

А. Строгонов, д. т. н.[1]

УДК 621.3.011.7 | ВАК 2.2.2

Для анализа аналоговых схем разработчики широко применяют симулятор SPICE, который входит в состав большинства САПР от ведущих мировых вендоров. Сегодня доступ к бесплатным версиям и исходным кодам этой программы ограничен, в связи с чем актуальной задачей становится поиск альтернативных методов и алгоритмов расчета электрических схем, которые можно было бы реализовать в отечественных системах схемотехнического проектирования ИС. В статье на примере расчета электрических схем по постоянному току методом Ньютона – Рафсона показана возможность применения известных алгоритмов для анализа электрических схем, выполнена оценка их точности в сравнении с современными средствами проектирования.

Основные расчетные методы анализа электрических схем изложены в широко известных работах [1, 2], посвященных вопросам автоматизации схемотехнического проектирования ИС. Эти методы не потеряли актуальность и в наши дни, поскольку заложены в основу практически всех современных SPICE-подобных симуляторов аналоговых схем.

Симулятор SPICE был разработан в Калифорнийском университете в Беркли в 1972 году, в 1975 году вышел следующий релиз – SPICE2, а в 1985 году появился SPICE3, исходный код которого был преобразован с языка программирования FORTRAN на C. В конце 1970-х годов появились индустриальные версии SPICE.

В конце 1990-х годов исходные коды программ SPICE3 и XSPICE (Georgia Tech Research Institute) на языке C для UNIX были размещены в Интернете. Сегодня скачать ранее открытые исходные коды программы невозможно по причине закрытия доступа к соответствующим ресурсам (таким как www.aboutspice.com). Закрыт также доступ к бесплатному симулятору LTspice, разработанному компанией Linear Technologies (принадлежит Analog Devices), а также к учебной версии Capture CIS Lite, которая входит в состав пакета OrCad от Cadence.

В вузах широко используют отечественную программу uSpice (http://www.uspice.ru/) для аналогового и цифрового схемотехнического моделирования, для задач цифровой обработки сигналов и для анализа радиочастотных цепей [4]. Учебная программа uSpice была создана с использованием переписанного для Windows открытого кода пакетов SPICE3 и XSPICE. Имеется также руководство пользователя uSpice на русском языке. Не закрытым остается доступ к учебному симулятору AIM-Spice без схемотехнического редактора (http://www.aimspice.com), описание которого приведено в [5].

Оценим возможность применения расчетных методов анализа схем, как альтернативы SPICE-моделированию, на примере реализации метода Ньютона – Рафсона для анализа схем по постоянному току и сравним полученные результаты с результатами SPICE-моделирования. Для разработки собственного алгоритма («счетного ядра») по методу Ньютона – Рафсона воспользуемся учебным пособием [3].

МЕТОД НЬЮТОНА – РАФСОНА ДЛЯ АНАЛИЗА ЭЛЕКТРИЧЕСКИХ СХЕМ

Метод Ньютона – Рафсона – это итерационный численный метод нахождения корня заданной функции, то есть решения уравнений вида $f(x)=0$.

Рассмотрим систему n нелинейных уравнений для f_i с n переменными x_i:

$$
\begin{aligned}
&f_1(x_1, x_2, \ldots, x_n) = 0,\\
&f_2(x_1, x_2, \ldots, x_n) = 0,\\
&\ldots\\
&f_n(x_1, x_2, \ldots, x_n) = 0,
\end{aligned}
$$

[1] Воронежский государственный технический университет, профессор кафедры полупроводниковой электроники и наноэлектроники, тел. +79102471470, andreistrogonov@mail.ru.

или в матричной форме: $f(x)=0$, где f – вектор функций, а x – вектор переменных.

После разложения в ряд Тейлора система уравнений в линеаризованной форме принимает вид:

$$f(x^*)\approx f(x)+M(x^*-x),$$

где x^* – истинное решение; M – матрица Якоби функции f:

$$M=\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ . & . & \cdots & . \end{bmatrix}.$$

Если удается подобрать истинное решение, то $f(x^*)\to 0$, тогда $f(x)+M(x^*-x)=0$ или $f(x^k)+M(x^{k+1}-x^k)=0$, отсюда $x^{k+1}=x^k-M^{-1}f(x^k)$. На практике M^{-1} никогда не обращают. Делают следующее:

$$x^{k+1}-x^k=\Delta x^k=-M^{-1}f(x^k),$$

тогда систему линейных уравнений $M\Delta x^k=-f(x^k)$, где Δx^k – шаг метода, можно решить методом Гаусса, LU- или QR-разложением. Новые значения x^{k+1} можно найти по формуле: $x^{k+1}=x^k+\Delta x^k$.

j-й столбец матрицы Якоби $M(x_k)$ аппроксимируется центральными разностями [3]:

$$M(x_k)_j=\frac{f(x_k+h_j)-f(x_k-h_j)}{2h_j}.$$

Условиями останова, например, могут быть [3]:

1. в простейшем случае справедливо, чтобы $f(x^{k+1})$ стремилось к нулю, то есть при подстановке найденных x^{k+1} в правую часть норма должна уменьшаться: $||f(x^{k+1})||\le||f(x^k)||$;
2. второй критерий останова:

$$\max\left\{\frac{|x_+[i]-x_k|}{\max\{|x_+[i]|\}}\right\}\le steptol,\text{ где } steptol=macheps^{2/3}.$$

ДЕМОНСТРАЦИОННАЯ ПРОГРАММА

Для оценки возможности применения расчетных методов анализа схем была разработана демонстрационная программа на языке PASCAL, которая решает системы нелинейных уравнений методом Ньютона – Рафсона с использованием алгоритмов, приведенных в работах [1, 3]. Программу можно применять для анализа по постоянному току для двух и более контуров в цепи, содержащей диоды, без емкостей и индуктивностей. Перед запуском программы необходимо сформировать файл с именем `soft.dat` со следующим содержанием: `0.1  0.1` (начальные приближения). Если анализируемая схема состоит из двух контуров, как в рассматриваемых примерах, то $n=m=2$ (n и m – размерности матрицы Якоби $M(m\times n)$ и размерности вспомогательных векторов, например, вектор правых частей $f(n)$, вектор переменных $x(n)$, Ньютоновский шаг $\Delta x(n)$ и др.).

В программе приняты следующие обозначения: `f[i]` – вектор правых частей; `b0[i]` – начальные приближения; `x0[i]` – начальный пробный шаг; `x[i]` – последующие пробные шаги к точному решению `xc[i]`, где `i` изменяется от 1 до n.

Процедура `kon` обеспечивает решение уравнения Ньютона в итерационной части. Первой работает процедура `kon(b0,x0)`, которая вычисляет `x0` при начальных условиях `b0`. Далее опять вызывается процедура `kon(b,x)` для решения с минимальным шагом (для нахождения последующих пробных решений). Процедура `kon` внутри своего тела использует три другие процедуры: `fvec(b,f)`; `FDJAC(b,f,JAC)`; `QRdecomp(a,f,m1,m2,x)`. QR-разложение, QR- и R-решения разработаны по алгоритмам A3.2.1, A3.2.2 и A3.2.2a, а конечно-разностная аппроксимация якобиана – по алгоритму A5.4.1, приведенным в работе [3].

Критерии останова реализованы в процедуре Nestop `(itncount:ter;b,b0,f:tf; Var termocode:TER)` (рис. 1).

```
procedure Nestop (itncount:ter;b,b0,f:tf; Var termocode:TER);
  var i:integer;
  begin
   termocode:=0;
   for i:=1 to n do f[i]:=abs(f[i]); max1(f,fmax);
   if fmax<epsdop then termocode:=1;
   for i:=1 to n do eps[i]:=abs((b[i]-b0[i])/b[i]); max1(eps,epsmax);
   if epsmax<=epsdop then termocode:=2;
    if itncount>=itnlimit then termocode:=4;
  end;
```

Рис. 1. Процедура Nestop

В программе используются критерии останова `termocode` `0, 1, 2` и `4` (критерий `termocode 3` оставлен про запас):

- `termocode = 0`, если не выполняется ни одно из ниже приведенных условий (`termcode = 1, 2, 4`);
- `termocode = 1`, когда максимальное значение одного из элементов правой части $f(x^{k+1})$ меньше допустимой ошибки `(epsdop=1e–8;): fmax<epsdop`;
- `termocode = 2`, если максимальная вычисленная ошибка по одному из элементов правой части меньше или равна наперед заданной ошибке `epsmax <= epsdop`;
- `termocode = 4`, если число итерационных циклов решения уравнения Ньютона – Рафсона больше или равно наперед заданному числу итераций `(itnlimit = 10): itncount>=itnlimit`. В разработанной программе общее число итерационных циклов определяется как `intcout + 1`.

ПРИМЕРЫ АНАЛИЗА СХЕМ ПО ПОСТОЯННОМУ ТОКУ

Разработанную программу будем использовать для анализа схемы по постоянному току (DC Sweep), который предшествует анализу по переменному току (AC Sweep), определению частотной характеристики и анализу переходных процессов (Time Domain/transient). Полученные в результате анализа данные по постоянному току берутся в качестве входных условий для анализа переходных процессов.

Пример 1

Цепь, состоящая из источника тока и резисторов. Источники напряжений и токов могут быть взаимно преобразованы с использованием теорем Тевенина и Нортона. Будем считать, что такое преобразование произведено, и цепь содержит только источник тока и резисторы. Используя метод узловых потенциалов, определим напряжения в узлах. Метод основан на составлении уравнений по первому закону Кирхгофа. При этом, потенциал одного из узлов цепи принимается равным нулю. Примем узел 0 за базовый и будем считать его потенциал равным нулю.

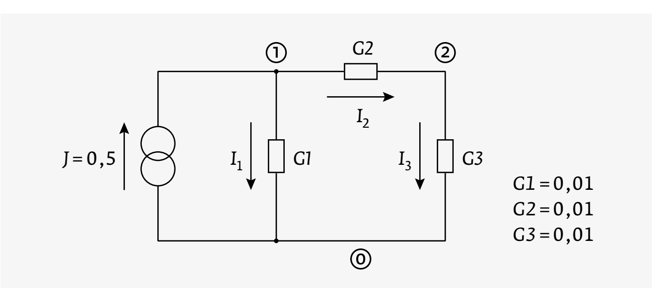


Рис. 2. Линейная цепь, состоящая из источника тока и трех резисторов

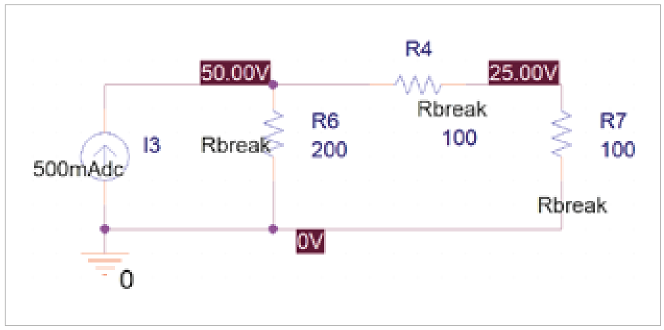


Рис. 3. Анализ линейной цепи, состоящей из источника тока и трех резисторов, в OrCad

Используя закон Ома, составим уравнения для нахождения каждого из токов. За φ_a возьмем потенциал узла, из которого ток выходит, а за φ_b потенциал узла, в который ток входит: $I = \frac{\varphi_a - \varphi_b}{R} = (\varphi_a - \varphi_b)G$. Ток, который входит в узел, берем со знаком минус, ток, который выходит из узла, берем со знаком плюс. За направления токов примем те, которые указаны на схемах. По закону Кирхгофа сумма токов, втекающих в узел равна сумме вытекающих.

На рис. 2 показана линейная цепь, состоящая из источника тока и трех резисторов. Параметры схемы: источник тока $J = 0,5$ А, проводимости $G_1 = 0,01$, $G_2 = 0,01$ и $G_3 = 0,01$ в См (сименс). На рис. 3 показаны результаты анализа схемы по постоянному току (DC Sweep) в САПР OrCad.

Метод узловых потенциалов позволяет составить два уравнения:

для узла 1: $-J + I_1 + I_2 = 0$,

для узла 2: $-I_2 + I_3 = 0$

или

для узла 1: $-0,5 + V_1G_1 + (V_1 - V_2)G_2 = 0$,

для узла 2: $-(V_1 - V_2)G_2 + V_2G_2 = 0$.

Решая эти уравнения $V_1 = (0,5 + 0,01V_2)/0,015$ и $V_2 = 0,5 \cdot V_1$, найдем V_1 и V_2: $V_1 = 50$ В и $V_2 = 25$ В.

Результаты расчетов представлены в табл. 1. Критерий останова программы `termocode = 1`. Разработанная

Таблица 1. Анализ линейной цепи по постоянному току

Расчет по формулам	Разработанная программа	OrCad
$V_1 = 50$ В	$V_1 = 49,999$ В	$V_1 = 50$ В
$V_2 = 25$ В	$V_2 = 24,999$ В	$V_2 = 25$ В

```
procedure FVEC (b:tf;var f:tf);
   var i:integer;
            Begin
           f[1]:=-0.5+0.015*b[1]-0.01*b[2];
           f[2]:=-0.01*b[1]+0.02*b[2];
            End;
```

Рис. 4. Процедура FVEC для линейной цепи, состоящей из источника тока и трех резисторов

программа вычисляет напряжения в узлах за одну итерацию. Для расчета с помощью программы необходимо сформировать вектор правых частей системы линейных уравнений (рис. 4).

Пример 2

Этот пример заимствован из [1]: нелинейная цепь, состоящая из источника тока, двух резисторов и диода (рис. 5). Требуется определить напряжения в узлах схемы. Для практических расчетов удобно ток насыщения диода I_s принять равным единицы. Ток диода определяется формулой: $I_D = e^{40V_D} - 1$. Покажем, что один диод в рассматриваемой цепи приводит к одному нелинейному уравнению.

По методу узловых потенциалов составляем систему из двух уравнений: для узла 1: $-J + I_1 + I_2 = 0$, для узла 2: $-I_2 + I_D = 0$.

Подставив значения токов получим:

1: $-1 + V_1G_1 + (V_1 - V_2)\,G_2 = 0$.
1: $-1 + 3V_1 - 2V_2 = 0$.
1: $V_1 = 1/3 + 2/3V_2$.
2: $-2V_1 + 2V_2 + e^{40V_2} - 1 = 0$.

Подставив V_1 во второе уравнение, получим нелинейное уравнение с одним неизвестным:

$$f(V_2) = 2/3V_2 + 40e^{40V_2} = 0.$$

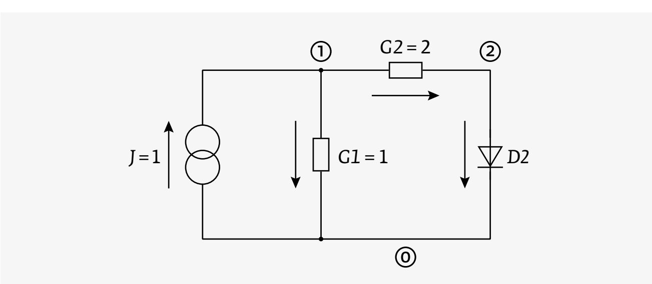


Рис. 5. Нелинейная цепь, состоящая из источника тока, резисторов и диода

Таблица 2. Анализ нелинейной цепи по постоянному току (один диод в контуре)

Расчет из работы [1]	Разработанная программа	OrCad
$V_1 = 0{,}3418$ В	$V_1 = 0{,}3418$ В	$V_1 = 0{,}3421$ В
$V_2 = 0{,}01264$ В	$V_2 = 0{,}0126$ В	$V_2 = 0{,}01308$ В

Для решения такого уравнения воспользуемся методом Ньютона – Рафсона. Начальное значение напряжения положим равным $V_2^0 = 0{,}1$ В. В работе [1] приводится численный расчет (см. табл. 2).

Найдем напряжения в узлах с помощью нашей программы. Аналогично, для расчета с помощью программы необходимо сформировать вектор правых частей системы линейных уравнений (рис. 6).

Проведем анализ схемы по постоянному току в САПР OrCad (рис. 7). Отредактируем Spice-модель диода интегральной схемы Dbreak: `.model Dbreak D Is=1e-14 Cjo=.1pF Rs=.1`.

```
procedure FVEC (b:tf;var f:tf);
   var i:integer;
          Begin
        f[1]:=-1+3*b[1]-2*b[2];
        f[2]:=-2*b[1]+2*b[2]+exp(40*b[2])-1;
          End;
```

Рис. 6. Процедура FVEC для нелинейной цепи, состоящей из источника тока, резисторов и диода

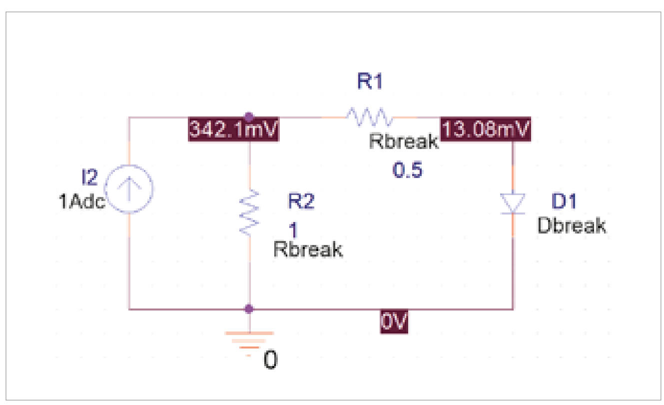


Рис. 7. Анализ нелинейной цепи, состоящей из источника тока, двух резисторов и одного диода, в OrCad

Примем в Spice-модели диода Dbreak параметр $I_s = 1$. Другие параметры модели диода: емкость перехода при нулевом смещении `Cjo = 0`, паразитное сопротивление (последовательное сопротивление) `Rs = 0`. Остальные параметры диода примем по умолчанию. Полный перечень Spice-параметров представлен в [6].

Таким образом, Spice-модель диода для нашего случая: `.model Dbreak D Is=1`.

Результаты расчетов представлены в табл. 2. Число итераций `itncount = 7` (общее число итерационных циклов 8), `termocode = 1`.

Пример 3

Этот пример также заимствован из [1]: нелинейная цепь, состоящая из источника тока, резистора и двух диодов (рис. 8). Два диода в схеме приводят к системе из двух нелинейных уравнений, решение которой потребует применения метода Ньютона – Рафсона. Рассмотрим этот пример более подробно.

Заданы следующие условия: $J = 1$; $G = 1$; $I_{D1} = e^{40V_1} - 1$; $I_{D2} = e^{40V_2} - 1$; $I_{D1} = I_{D2}$.

По методу узловых потенциалов составляем систему из двух уравнений:

1: $-J + I_{D1} + I_l = 0$

2: $-I_1 + I_{D2} = 0$

Подставим значения токов:

1: $-2 + V_1 - V_2 + e^{40V_1} = 0$

2: $-V_1 + V_2 + e^{40V_1} - 1 = 0$

Запишем систему уравнений для определения напряжений:

$$f_1(V_1, V_2) = -2 + V_1 - V_2 + e^{40V_1} = 0;$$

$$f_2(V_1, V_2) = -V_1 + V_2 + e^{40V_1} - 1 = 0.$$

Для составления первой строки матрицы Якоби необходимо функцию $f_1(V_1, V_2)$ дифференцировать по V_1, затем по V_2. Для составления второй строки необходимо $f_2(V_1, V_2)$ дифференцировать по V_1, затем по V_2. Векторная запись примера будет выглядеть следующим образом:

$$\begin{bmatrix} 40e^{40V_1} + 1 & -1 \\ -1 & 40e^{40V_2} + 1 \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} = -\begin{bmatrix} e^{40V_1} + V_1 - V_2 - 2 \\ -V_1 + V_2 + e^{40V_2} - 1 \end{bmatrix},$$

где ΔV_1, ΔV_2 – неизвестные, которые надо найти.

Начальные значения напряжения возьмем равными $V_1^0 = 0{,}1$ В и $V_2^0 = 0{,}1$ В. Для расчета с помощью программы необходимо сформировать вектор правых частей системы линейных уравнений (рис. 9).

```
procedure FVEC (b:tf; var f:tf);
   var i:integer;
            Begin
           f[1]:=exp(40*b[1])+b[1]-b[2]-2;
           f[2]:=-b[1]+b[2]+exp(40*b[2])-1;
            End;
```

Рис. 9. Процедура FVEC для нелинейной цепи, состоящей из источника тока, резистора и двух диодов

Результаты расчетов представлены в табл. 3. Итоговое решение находится за 8 итераций (`itncount = 7`). Расчеты были также проверены в САПР OrCad и с использованием Spice-симулятора LTspice. На рис. 10 показана нелинейная цепь, состоящая из источника тока, резистора и двух диодов, анализируемая в САПР OrCad, а на рис. 11 – в симуляторе LTspice XVII(x64). На рис. 12 показан анализ цепи в системе Matlab/Simulink с применением пакета расширения Simscape, который поддерживает использование Spice-моделей. Как видим, результаты расчетов напряжений в узлах соответствуют расчетам с применением САПР OrCAd и симулятора LTspice.

На рис. 13 (с. 118) приведен код разработанной программы для анализа схемы по постоянному току для примера 3.

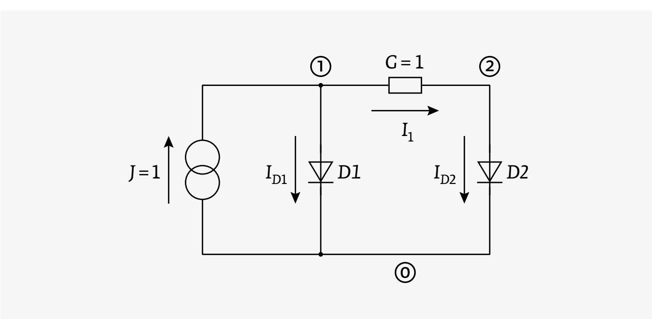


Рис. 8. Нелинейная цепь, состоящая из источника тока, резистора и двух диодов

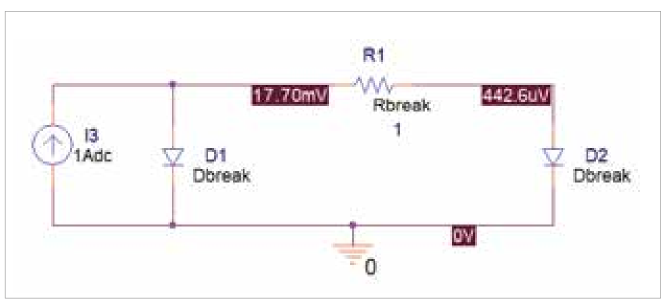


Рис. 10. Анализ нелинейной цепи, состоящей из источника тока, резистора и двух диодов, в OrCad

КРУПНЕЙШИЙ В РОССИИ
КУЛОН
общество с ограниченной ответственностью
завод по производству пассивных электронных компонентов
www.kulon.spb.ru
192019, г. Санкт-Петербург, ул. Профессора Качалова, д. 3, литер К
Тел.: +7 (812) 317-33-04 | E-mail: office@kulon.spb.ru, sale@kulon.spb.ru

Таблица 3. Анализ нелинейной цепи по постоянному току (два диода в контуре)

Расчет из работы [1]	Разработанная программа	OrCad	LTspice XVII(x64)
V_1 = 0,01712 В	V_1 = 0,0171 В	V_1 = 0,0177 В	V_1 = 0,017704 В
V_2 = 0,00041 В	V_2 = 0,000414 В	V_2 = 0,0004426 В	V_2 = 0,000442655 В

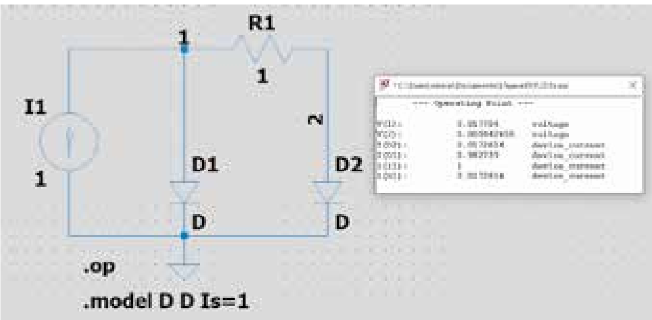


Рис. 11. Анализ нелинейной цепи, состоящей из источника тока, резистора и двух диодов, в симуляторе LTspice XVII(x64)

Таблица 4. Результаты итераций для примера 3 из работы [1]

k	Δv_1^k	Δv_2^k	v_1^{k+1}	v_2^{k+1}
0	−0,02408	−0,02454	0,07592	0,07546
1	−0,02260	−0,02378	0,05331	0,05168
2	−0,01909	−0,02182	0,03423	0,02986
3	−0,01234	−0,01736	0,02188	0,01250
4	−0,00432	−0,00962	0,01757	0,00289
5	−0,00044	−0,00236	0,01712	0,00053
6	−0,00001	−0,00012	0,01712	0,00041
7	−0,00000	−0,00000	0,01712	0,00041

В файл result.out выводятся результаты работы программы. Поясним его более подробно. На рис. 14 показан фрагмент файла отчета для примера 3 (начальные итерации).

Итоговое решение уравнения Ньютона – Рафсона:

```
xc[1]= 1.7118992323625445E-002,
xc[2]= 4.1417076753269045E-004.
```

Число итераций:

```
Itncount = 7.0000000000000000E+000.
```

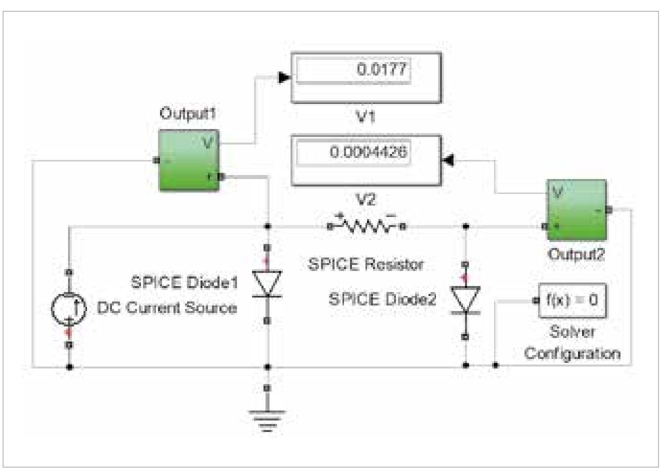
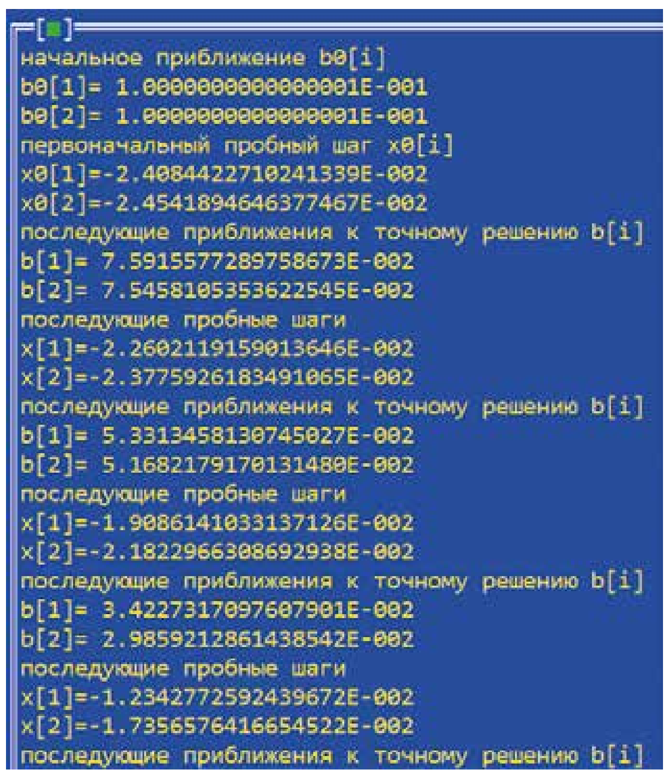


Рис. 12. Анализ цепи в системе Matlab/Simulink (пакет расширения Simscape)



```
начальное приближение b0[i]
b0[1]= 1.0000000000000001E-001
b0[2]= 1.0000000000000001E-001
первоначальный пробный шаг x0[i]
x0[1]=-2.4084422710241339E-002
x0[2]=-2.4541894646377467E-002
последующие приближения к точному решению b[i]
b[1]= 7.5915577289758673E-002
b[2]= 7.5458105353622545E-002
последующие пробные шаги
x[1]=-2.2602119159013646E-002
x[2]=-2.3775926183491065E-002
последующие приближения к точному решению b[i]
b[1]= 5.3313458130745027E-002
b[2]= 5.1682179170131480E-002
последующие пробные шаги
x[1]=-1.9086141033137126E-002
x[2]=-2.1822966308692938E-002
последующие приближения к точному решению b[i]
b[1]= 3.4227317097607901E-002
b[2]= 2.9859212861438542E-002
последующие пробные шаги
x[1]=-1.2342772592439672E-002
x[2]=-1.7356576416654522E-002
последующие приближения к точному решению b[i]
```

Рис. 14. Фрагмент файла отчета работы программы для примера 3 (первые четыре итерации) www.monolit.by

МОНОЛИТ

ВИТЕБСКИЙ ЗАВОД РАДИОДЕТАЛЕЙ

МНОГОСЛОЙНЫЕ КЕРАМИЧЕСКИЕ КОНДЕНСАТОРЫ

ИМПОРТОЗАМЕЩАЮЩАЯ ПРОДУКЦИЯ
для высоконадёжной аппаратуры

ТЕРМОРЕЗИСТОРЫ
с положительным температурным коэффициентом сопротивления

РЕГИСТРЫ НАГРЕВАТЕЛЬНЫЕ

РЕСПУБЛИКА БЕЛАРУСЬ, 210101
г. Витебск, ул. М. Горького, 145

Отдел маркетинга и сбыта
Маркетинг:
Телефон: +375 (212) 36-44-52
E-mail: monolmarket@mail.ru
Сбыт:
Телефон: +375 (212) 36-45-34;
+375 (212) 36-45-42
E-mail: monosbet@mail.ru

www.monolit.by

ЭКСКЛЮЗИВНЫЙ ДИЛЕР НА ТЕРРИТОРИИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Акционерное общество «СПЕЦЭЛЕКТРОНКОМПЛЕКТ»

Почтовый адрес: 125319, г. Москва, а/я 92.
Офис: г. Москва, ул. Краснопролетарская, д. 16, стр. 2
тел.: +7 (495) 234-01-10, факс: +7 (495) 956-33-46
sales@zolshar.ru

СПЕЦЭЛЕКТРОНКОМПЛЕКТ
КОМПЛЕКТНЫЕ ПОСТАВКИ ЭЛЕКТРОННЫХ КОМПОНЕНТОВ

```
{РЕШЕНИЕ СИСТЕМЫ НЕЛИНЕЙНЫХ УРАВНЕНИЙ МЕТОДОМ НЬЮТОНА – РАФСОНА}
{Для решения системы линейных уравнений используется метод QR-разложения}

program LUr (input,output);
uses crt;
const n=2; m=2; itnlimit=10; epsdop=1e-8;
type tx=array [1..m,1..n] of real; tf=array [1..n] of real;
Xm=real; st=real; ter=real;

var i,j:integer;
a,a0,JAC,JAc0:tx;
f,f0,Snak,Snak1,m1,m2,b,b0,b1,x,x0,xc,eps:tf;
Xmaks:Xm; fr,FX: text; Termocode,itncount:Ter;
t:real; fmax, epsmax:St;

      procedure max(a:tx;var Xmaks:Xm);{--отыскание макс.элемента в матрице}
      var i,j:integer;
      begin
      for i:=1 to n do for j:=1 to m do if abs(a[i,j])>=abs(a[1,1]) then
      Xmaks:=abs(a[i,j]);
      end;

       procedure sign(a:tx;var Snak:tf);
       var k:integer;
       begin
            for k:=1 to n do begin
                             if a[k,k]>=0 then Snak[k]:=1
            else Snak[k]:=-1;end;
            end;

       procedure sign1(b:tf;var Snak1:tf);
       var i:integer;
       begin
            for i:=1 to n do begin
                             if b[i]>=0 then Snak1[i]:=1
            else Snak1[i]:=-1;end;
            end;

     procedure QRdecomp (a:tx;f:tf; var m1,m2,x:tf);
         Var i,j,k:integer;sum:real;
         begin  max(a,Xmaks);
         for k:=1 to n-1 do
     begin
                Sum:=0;
                for i:=k to n do
                       begin
                           a[i,k]:=a[i,k]/Xmaks;
                           Sum:=Sum+(sqr(a[i,k]));end;
                           Sum:=sqrt(Sum);
                           sign(a,snak); Sum:=snak[k]*Sum;
                           a[k,k]:=Sum+a[k,k];
                           M1[k]:=Sum*a[k,k];
                           M2[k]:=-Xmaks*Sum;
               for j:=k+1 to n do begin
               Sum:=0;
               for i:=k to n do Sum:=Sum+a[i,k]*a[i,j]/m1[k]; end;
               for i:=k to n do a[i,j]:=a[i,j]-Sum*a[i,k];end;
               m2[n]:=a[n,n];
               {for i:=1 to n do writeln(fx,'M1[',i,']=',M1[i]);
               for i:=1 to n do writeln(fx,'M2[',i,']=',M2[i]);
               for i:=1 to n do for j:=1 to n do writeln (fx,'a[',i,',',j,']=',a[i,J]);}
                  for j:=1 to n-1 do
          begin
                  sum:=0;
```

Рис. 13. Код программы для анализа схемы по постоянному току для примера 3

```
                for i:=j to n do sum:=sum+a[i,j]*f[i];
                sum:=sum/M1[j];
                for i:=j to n do f[i]:=f[i]-sum*a[i,j];
        end;
                f[n]:=f[n]/M2[n];
                for i:=n-1 downto 1 do begin t:=f[i];
                for j:=i+1 to n do T:=(T-a[i,j]*f[j])/m2[i]; f[i]:=t;
                end;
                for i:=1 to n do x[i]:=f[i];
               { for i:=1 to n do writeln(fx,'x[',i,']=',x[i]);}
               END;

procedure FVEC (b:tf;var f:tf);
 var i:integer;
          Begin
         f[1]:=exp(40*b[1])+b[1]-b[2]-2;
         f[2]:=-b[1]+b[2]+exp(40*b[2])-1;
        {writeln(fx,'Значения правой части на каждой итерации');
         for i:=1 to n do writeln(fx,'f[',i,']=',f[i]);}
          End;

procedure FDJAC(b,f:tf; var JAC:tx);
 var i:integer; DIGITS,nn,stepsizej,tempj,sqrteta:real;
 fs,fn:array [1..n] of real;
         Begin
         digits:=2;
         nn:=1e-14;
         sqrteta:=sqrt(nn);
         for j:=1 to n do begin for i:=1 to n do fs[i]:=f[i];
         sign1(b,Snak1);
         stepsizej:=sqrteta*abs(b[i])*Snak1[j];
         tempj:=b[j];
         b[j]:=b[j]+stepsizej;
         stepsizej:=b[j]-tempj;
         Fvec(b,f);
         for i:=1 to n do fn[i]:=f[i];
         for i:=1 to n do jac[i,j]:=(fn[i]-fs[i])/stepsizej;
         for i:=1 to n do f[i]:=fs[i];
         b[j]:=tempj;end;
         {writeln (fx,'Значение матрицы Якоби на каждой итерации');
         for j:=1 to n do for i:=1 to n
         do writeln (fx,'JAC[',i,',',j,']=',JAC[i,J]);}
         End;

procedure max1 (f:tf; var fmax:St);
begin  fmax:=f[1]; for i:=1 to n do if f[i]>fmax then fmax:=f[i];end;

procedure Nestop (itncount:ter;b,b0,f:tf; Var termocode:TER);
var i:integer;
begin
 termocode:=0;
 for i:=1 to n do f[i]:=abs(f[i]); max1(f,fmax);
 if fmax<epsdop then termocode:=1;
 for i:=1 to n do eps[i]:=abs((b[i]-b0[i])/b[i]); max1(eps,epsmax);
 if epsmax<=epsdop then termocode:=2;
  if itncount>=itnlimit then termocode:=4;
end;

procedure Kon (b:tf;var x:tf);{решение ур.Ньютона в итер.части}
 begin
     fvec(b,f);
     FDJAC(b,f,JAC);
     for j:=1 to n do for i:=1 to m do a[i,j]:=JAC[i,j];
     for i:=1 to n do f[i]:=-f[i];
```

Рис. 13. Код программы для анализа схемы по постоянному току для примера 3 (продолжение)

```
      QRdecomp (a,f,m1,m2,x); end;
begin
    ClrScr; Assign(fx,'Resultat.out'); Rewrite(fx);
    Assign(fr, 'Soft.dat'); Reset (fr);
    for i:=1 to n do read (fr, b0[i]);close(fr);
    for i:=1 to n do b[i]:=b0[i];
    writeln (fx,'начальное приближение b0[i]');
    for i:=1 to n do writeln(fx,'b0[',i,']=',b0[i]);
   {--Вычисление первоначальных значений---}
     kon (b0,x0);
     writeln (fx,'первоначальный пробный шаг x0[i]');
     for i:=1 to n do writeln(fx,'x0[',i,']=',x0[i]);
  {--итерационная часть--}
     itncount:=0;
     while termocode=0 do begin
     writeln (fx,'последующие приближения к точному решению b[i]');
     for i:=1 to n do b[i]:=b0[i]+x0[i];
     for i:=1 to n do writeln(fx,'b[',i,']=',b[i]);
     kon(b,x);{решение с шагом hmin}
     writeln (fx,'последующие пробные шаги');
     for i:=1 to n do writeln(fx,'x[',i,']=',x[i]);
     Nestop (itncount,b,b0,f,termocode);
     if termocode>0 then for i:=1 to n do xc[i]:=b[i]
     else  itncount:=itncount+1;
     for i:=1 to n do begin x0[i]:=x[i]; b0[i]:=b[i]; end;
     end;
     for i:=1 to n do writeln(fx,'xc[',i,']=',xc[i]);
     writeln (fx,'Число итераций');
     writeln (fx,'itncount=',itncount);
     writeln (fx,'Критерий останова');
     writeln (fx,'termocode=',termocode);
     close (fx);
    end.
```

Рис. 13. Код программы для анализа схемы по постоянному току для примера 3 (продолжение)

Критерий останова:

`Termocode = 1.0000000000000000E+000.`

В табл. 4 приведены результаты итераций из работы [1]. Используются следующие обозначения: Δv_1^k и Δv_2^k – Ньютоновский шаг (пробные шаги); v_1^{k+1} и v_2^{k+1} – приближения к точному решению. Решение уравнения Ньютона – Рафсона определяется значениями $v_1 = 0{,}01712$ В и $v_2 = 0{,}00041$ В. Сравнивая первые четыре итерации, представленные на рис. 14 и в табл. 4, можно убедиться, что программа работает корректно.

ЗАКЛЮЧЕНИЕ

Реализация в разработанной программе решения системы нелинейных уравнений по методу Ньютона – Рафсона показала возможность применения известных алгоритмов для анализа схем по постоянному току. С учетом закрытия доступа к бесплатным версиям симулятора SPICE алгоритмы и расчетные методы, изложенные в работах [1, 3], не потеряли сегодня свою актуальность.

Рассмотренные примеры показывают, что в симуляторе SPICE используется анализ цепи методом узловых потенциалов для построения системы уравнений, описывающих схему. Метод узловых потенциалов имеет ограничения в работе с индуктивностями, источниками переменного напряжения и различными вариантами управляемых генераторов тока и напряжения. Для устранения недостатков в современных версиях SPICE применяется модифицированный метод узловых потенциалов.

ЛИТЕРАТУРА

1. **Влах И., Сингхал К.** Машинные методы анализа и проектирования электронных схем / Пер. с англ. М.: Радио и связь, 1988. 560 с.
2. **Чуа Л. О., Лин Пен-Мин.** Машинный анализ электронных схем. Алгоритмы и вычислительные методы / Пер. с англ. М.: Энергия, 1980. 640 с.
3. **Деннис Дж., Шнабель Р.** Численные методы безусловной оптимизации и решения нелинейных уравнений / Пер. с англ. М.: Мир, 1988. 440 с.
4. **Кучумов А. А., Кучумов А. И.** Электроника и схемотехника. М.: Гелиос АРВ, 2017. 368 с.
5. **Петров М. Н., Гудков Г. В.** Моделирование компонентов и элементов интегральных схем: Уч. пос. СПб: Издательство «Лань», 2011. 464 с.
6. https://www.allaboutcircuits.com/textbook/semiconductors/chpt-3/spice-models/#.