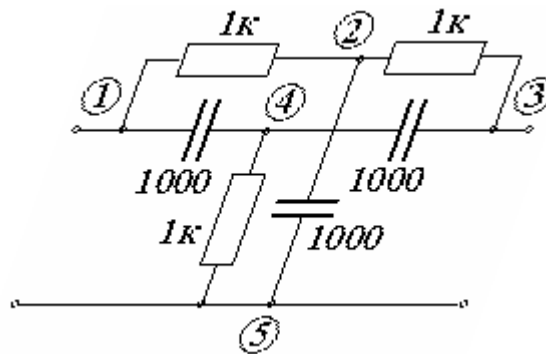


Романовский М.Н.

МЕТОДЫ И АЛГОРИТМЫ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ В РЭС

Руководство к практическим занятиям
и самостоятельной работе по дисциплине
«Компьютерное моделирование
процессов в РЭС»



Рецензент: Еханин С.Г., профессор кафедры КУДР, доктор физико–математических наук

Романовский М.Н.

Методы и алгоритмы моделирования процессов в РЭС. Руководство к практическим занятиям и самостоятельной работе по дисциплине «Компьютерное моделирование процессов в РЭС». – Томск: ТУСУР, 2016. – 66 с.

Рассмотрены основные методы и алгоритмы компьютерного моделирования процессов в РЭС.

Для студентов направления подготовки 211000.62 «Конструирование и технология электронных средств».

Содержание

ПРЕДИСЛОВИЕ	4
1 АНАЛИЗ ЛИНЕЙНЫХ СХЕМ.....	6
1.1 Цель работы	6
1.2 Формирование модели.....	6
1.2 Прямые методы	7
1.3 Итерационные методы	9
1.4 Задание	11
1.5 Контрольные вопросы	11
2 МОДЕЛИРОВАНИЕ ЭЛЕМЕНТОВ	11
2.1 Цель работы	11
2.2 Методика формирования моделей элементов.....	11
2.3 Моделирование резисторов	12
2.4 Вычисление коэффициента парной корреляции	13
2.5 Метод наименьших квадратов	15
2.6 Точность математической модели.....	19
2.7 Задание	21
2.8 Контрольные вопросы	21
3 АНАЛИЗ НЕЛИНЕЙНЫХ СХЕМ	21
3.1 Цель работы	21
3.2 Модели диода	21
3.3 Определение параметров модели диода.....	23
3.4 Метод Ньютона – Рафсона.....	25
3.5 Моделирование нелинейной схемы	25
3.6 Задание	28
3.7 Контрольные вопросы	28
4 МОДЕЛИРОВАНИЕ СТАТИЧЕСКИХ РЕЖИМОВ.....	28
4.1 Цель работы	28
4.2 Модели биполярного транзистора	29
4.3 Формирование модели схемы	31
4.4 Метод Ньютона для системы уравнений.....	32
4.5 Задание	37
4.6 Контрольные вопросы	37
5 МОДЕЛИРОВАНИЕ В ЧАСТОТНОЙ ОБЛАСТИ.....	37
5.1 Цель работы	37
5.2 Формирование модели.....	37
5.3 Алгоритм анализа.....	40
5.4 Задание	43
5.5 Контрольные вопросы	44
6 МЕТОДЫ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ	44
6.1 Цель работы	44
6.2 Явные и неявные методы.....	44
6.3 Точность и устойчивость методов интегрирования.....	46
6.4 Метод переменных состояния.....	48
6.5 Задание	53
6.6 Контрольные вопросы	54
7 КЛЮЧ НА ПОЛЕВОМ ТРАНЗИСТОРЕ	54
7.1 Цель работы	54

7.1 Метод трех зон	54
7.2 Модель ключа на полевом транзисторе.....	55
7.3 Задание	61
7.4 Контрольные вопросы	61
8 ОСНОВЫ МОДЕЛИРОВАНИЯ НА МИКРОУРОВНЕ.....	61
8.1 Введение	61
8.2 Формирование и реализация модели	62
8.3 Задание	65
8.4 Контрольные вопросы	66
ЛИТЕРАТУРА.....	66

Предисловие

Модели технических объектов в зависимости от места в иерархии описаний подразделяются на модели микро–, макро– и мета–уровня. Для **микро–уровня** характерно рассмотрение процессов в сплошных средах и непрерывном времени. Типичными моделями на микро–уровне являются дифференциальные уравнения в частных производных с заданными краевыми условиями. На **макро–уровне** используют укрупненную дискретизацию пространства по функциональному признаку, что приводит к моделям в виде систем обыкновенных дифференциальных уравнений. **Мета–уровень** отличает большое разнообразие типов используемых моделей, применение в качестве элементов достаточно сложных совокупностей деталей.

В большинстве технических объектов – систем можно выделить **три типа простейших элементов**: R, L и C. Элемент типа R – элемент рассеяния энергии. На этом элементе, как правило, происходит выделение тепла. На элементах типа L и C происходит накопление потенциальной или кинетической энергии. Для каждой физической подсистемы (электрической, механической, гидравлической и др.) характерны свои законы, но для простейших элементов R, L и C форма выражающих эти законы уравнений оказывается одинаковой.

Количественные представления свойств объектов называются параметрами. Различают **выходные, внутренние и внешние параметры**. Выходные параметры характеризуют свойства системы, внутренние – элементов системы, внешние – свойства внешней среды.

Физическое или информационное состояние объектов обычно характеризуют **фазовыми переменными** (ФП). Различают ФП типа потока и типа потенциала. Примерами ФП типа потока могут служить электрический ток, сила, тепловой поток, расход, а типа потенциала – электрическое напряжение, скорость, температура, давление.

Модели объектов – систем получают объединением компонентных и топологических уравнений. **Топологические уравнения** связывают однотипные ФП, относящиеся к разным элементам системы, а **компонентные уравнения** – разнотипные ФП для каждого из этих элементов. Иными словами, компонентные уравнения представляют собой модели элементов системы, а топологические – отражают ее структуру.

В уравнениях модели обычно фигурируют не все ФП, а только часть из них, достаточная для однозначной идентификации состояния объекта. Такие переменные называют *базисом* или базисными координатами. Через базис могут быть определены выходные параметры модели и все остальные ФП.

В настоящем руководстве основное внимание уделено моделированию РЭС на макро–уровне на примере электронных схем. Рассмотренные методы и алгоритмы, ввиду аналогий ФП, компонентных и топологических уравнений в подсистемах различной физической природы, применимы для анализа самых различных процессов в РЭС.

1 АНЛИЗ ЛИНЕЙНЫХ СХЕМ

1.1 Цель работы

Изучение алгоритмов формирования моделей линейных схем и решения систем линейных алгебраических уравнений.

1.2 Формирование модели

Модель схемы получают объединением компонентных и топологических уравнений. Компонентные уравнения связывают токи *и* напряжения для каждого из элементов, а топологические – токи *или* напряжения, относящиеся к разным элементам.

На рис. 1.1 приведена линейная схема, составленная из источника постоянного тока и резисторов. Компонентные уравнения элементов схемы:

$$\begin{aligned} J &= \text{const}, \\ I &= V / R, \end{aligned}$$

где J – ток источника, I – ток через резистор, V – напряжение на элементе; R – номинал резистора.

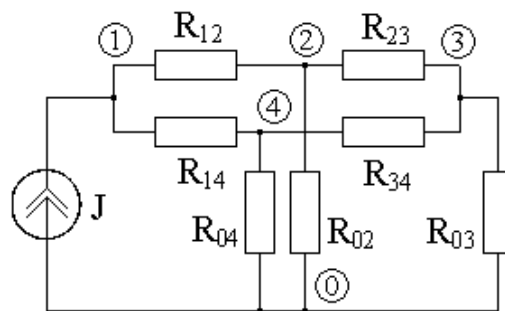


Рисунок 1.1 – Линейная схема для анализа

Для формирования топологических уравнений схемы удобно использовать метод узловых потенциалов (МУП). Искомые переменные – потенциалы всех узлов схемы, кроме опорного (нулевого) узла, потенциал которого принят равным нулю (земля). Основан МУП на первом законе Кирхгофа, который утверждает, что для любого узла схемы алгебраическая сумма токов инцидентных узлу ветвей равна нулю. **Вытекающие из узла токи ветвей берутся со знаком плюс, а втекающие – со знаком минус.**

Уравнения для узлов схемы рис. 1.1 имеют вид

$$\begin{aligned} (v_1 - v_2) / R_{12} + (v_1 - v_4) / R_{14} &= J, \\ (v_2 - v_1) / R_{12} + (v_2 - v_3) / R_{23} + (v_2 - 0) / R_{02} &= 0, \\ (v_3 - v_2) / R_{23} + (v_3 - v_4) / R_{34} + (v_3 - 0) / R_{03} &= 0, \\ (v_4 - v_1) / R_{14} + (v_4 - v_3) / R_{34} + (v_4 - 0) / R_{04} &= 0. \end{aligned}$$

Обозначив проводимости резисторов $1 / R = G$, получим систему линейных алгебраических уравнений (СЛАУ) в виде

$$(G_{12} + G_{14}) v_1 - G_{12} v_2 - G_{14} v_4 = J,$$

$$\begin{aligned}
 -G_{12} v_1 + (G_{12} + G_{23} + G_{02}) v_2 - G_{23} v_3 &= 0, \\
 -G_{23} v_2 + (G_{23} + G_{34} + G_{03}) v_3 - G_{34} v_4 &= 0, \\
 -G_{14} v_1 - G_{34} v_3 + (G_{14} + G_{34} + G_{04}) v_4 &= 0.
 \end{aligned}$$

В матричной форме СЛАУ имеет вид

$$Y V = I, \quad (1.1)$$

где Y – матрица узловых проводимостей, V – искомый вектор узловых потенциалов, I – вектор узловых токов.

Для схемы рис. 1.1

$$\begin{vmatrix}
 G_{12} + G_{14} & -G_{12} & 0 & -G_{14} \\
 -G_{12} & G_{12} + G_{23} + G_{02} & -G_{23} & 0 \\
 0 & -G_{23} & G_{23} + G_{34} + G_{03} & -G_{34} \\
 -G_{14} & 0 & -G_{34} & G_{14} + G_{34} + G_{04}
 \end{vmatrix} = Y,$$

а компоненты вектора узловых токов I равны нулю все, кроме первой: компонента i_1 равна току независимого источника J .

Таким образом, проводимость инцидентной узлам m и n ветви аддитивно (как слагаемое) включается в диагональные элементы Y_{mm} , Y_{nn} матрицы узловых проводимостей со знаком плюс и в недиагональные элементы Y_{mn} и Y_{nm} со знаком минус.

Заметим, что если один из узлов ветви – опорный, то проводимость этой ветви учитывается в одном элементе матрицы Y – собственной проводимости Y_{kk} , где k – номер незаземленного узла ветви.

Приведенный пример иллюстрирует свойство симметричности матрицы узловых проводимостей для пассивной схемы. Оно может быть использовано при записи и организации хранения матрицы в ЭВМ.

Для решения СЛАУ используют прямые и итерационные методы. Прямые (или точные) методы позволяют найти решение за определенное количество шагов. Итерационные методы основаны на использовании повторяющегося процесса и позволяют получить решение в результате последовательных приближений.

1.2 Прямые методы

Пусть СЛАУ имеет вид

$$A X = B, \quad (1.2)$$

где X – вектор искомых неизвестных; A – матрица коэффициентов при неизвестных – размерностью (N, N) ; B – вектор правых частей (столбец свободных членов). Если справа к матрице A приписать вектор B , то получится так называемая расширенная матрица.

К прямым методам решения СЛАУ относятся метод Гаусса и его модификации. Алгоритм метода Гаусса включает прямой и обратный ход.

Прямой ход заключается в последовательном исключении неизвестных из (1.2). Сначала первое уравнение представляется в явном относительно x_1 виде:

$$x_1 = - (a_{12}/a_{11}) x_2 - \dots - (a_{1N}/a_{11}) x_N + b_1/a_{11}. \quad (1.2)$$

Затем x_1 подставляется во все оставшиеся уравнения. Эта подстановка приводит к исключению из уравнений со второго по N -е переменной x_1 и к изменению коэффициентов a_{ij} расширенной матрицы в соответствии с выражением

$$a_{ij} = a_{ij} - a_{i1} a_{1j}/a_{11},$$

Далее второе уравнение представляется в явном относительно x_2 виде, после чего x_2 исключается из уравнений с третьего по N -е, и т. д. Другими словами, исключение x_k сводится к пересчету коэффициентов по формуле

$$a_{ij} = a_{ij} - a_{ik} a_{kj}/a_{kk} \quad (1.3)$$

для $i = (k+1), (k+2), \dots, (N+1)$. Процедура исключения выполняется для всех x_i , кроме последнего x_N . В результате прямого хода преобразованная матрица A принимает треугольный вид.

Обратный ход заключается в вычислении вектора искомых неизвестных X . Он начинается с последнего уравнения. В нем после прямого хода осталась единственная переменная x_N , которая вычисляется делением b_N на a_{NN} (величины b_N , a_{NN} , как и большинство других коэффициентов уравнений, изменили свои значения при прямом ходе по сравнению с исходными). Далее из $(N-1)$ -го уравнения определяется x_{N-1} и т. д. Другими словами, обратный ход выполняется по формуле

$$x_k = (b_k - \sum_{j=k+1}^N a_{kj} x_j) / a_{kk}. \quad (1.4)$$

Рассмотренный алгоритм метода Гаусса представлен ниже в виде подпрограммы на языке Паскаль.

...

```
type vec=array [1..N] of real; mat=array [1..N,1..N+1] of real;
```

...

```
PROCEDURE Gauss(N: byte; var A: mat; var X: vec); {Метод Гаусса}
```

```
  var I, J, K: byte; S, P: real;
```

```
  begin
```

```
    for K:=1 to N do begin S:=A[K,K]; {Прямой ход}
```

```
      for I:=K+1 to N do begin P:=A[I,K]/S;
```

```
        for J:=K+1 to N+1 do A[I,J]:=A[I,J]-A[K,J]*P
```

```
      end
```

```
    end;
```



```

for I:=N downto 1 do begin                                { Обратный ход }
  Pm:=A[I,N+1]; S:=A[I,I];
  for J:=I+1 to N do P:=P-A[I,J]*X[J];
  X[I]:=P/S
end
end;

```

Как следует из (1.3), все элементы на диагонали a_{kk} должны быть отличны от нуля, иначе численный процесс остановится из-за деления на нуль. Если какой-либо диагональный элемент a_{kk} будет иметь очень маленькое значение, то при вычислении по (1.3) возникнут большие ошибки округления, распространяемые на весь дальнейший процесс исключения.

В методе Гаусса эта проблема может быть решена выбором в качестве a_{kk} на каждой стадии исключений максимального по модулю из оставшихся коэффициентов с соответствующей перестановкой строк и столбцов матрицы A (способ выбора главного элемента).

В приведенной ниже подпрограмме представлен метод исключения Гаусса с частичным выбором главного элемента, – в пределах столбца.

```

PROCEDURE Gauss(A: mat; var X: vec);                    { Метод Гаусса }
var I,J,K:byte; R,S:real;
begin
  for K:=1 to N-1 do begin
    S:=A[K,K]; J:=K;                                       { Выбор главного элемента }
    for I:=K+1 to N do begin R:=A[I,K];
      if Abs(R)>Abs(S) then begin S:=R; J:=I end end;
    if J<>K then for I:=K to N+1 do                       { Перестановка строк }
      begin R:=A[K,I]; A[K,I]:=A[J,I]; A[J,I]:=R end;
    for I:=K+1 to N do begin R:=A[I,K]/S;                   { Прямой ход }
      for J:=K+1 to N+1 do A[I,J]:=A[I,J]-A[K,J]*R end
    end;
    if S<>0 then for I:=N downto 1 do begin                { Обратный ход }
      S:=A[I,N+1]; for J:=I+1 to N do S:=S-A[I,J]*X[J];
      X[I]:=S/A[I,I] end
    end;

```

1.3 Итерационные методы

Итерационные методы решения СЛАУ устанавливают процедуру уточнения определенного начального приближения к решению. При

выполнении условий сходимости они позволяют достичь любой точности простым повторением итераций. Часто они позволяют достичь решения с заранее заданной точностью быстрее, чем прямые методы, а также позволяют решать большие системы уравнений.

Для решения (1.2) итерационными методами ее преобразуют к виду

$$\begin{aligned}x_1 &= (b_1 - a_{11}x_1 - a_{12}x_2 - \dots - a_{1N}x_N)/a_{11} + x_1, \\x_2 &= (b_2 - a_{21}x_1 - a_{22}x_2 - \dots - a_{2N}x_N)/a_{22} + x_2, \\&\dots \\x_N &= (b_N - a_{N1}x_1 - a_{N2}x_2 - \dots - a_{NN}x_N)/a_{NN} + x_1.\end{aligned}$$

Задав вектор начальных приближений $x_1(0)$, $x_2(0)$, ..., $x_N(0)$, итерационный процесс можно организовать следующим образом:

$$\begin{aligned}x_1(m+1) &= f_1[x_1(m), x_2(m), \dots, x_N(m)], \\x_2(m+1) &= f_2[x_1(m), x_2(m), \dots, x_N(m)], \\&\dots \\x_N(m+1) &= f_N[x_1(m), x_2(m), \dots, x_N(m)],\end{aligned}\tag{1.5}$$

где $m+1$ – порядковый номер итерации.

Процесс (1.5) соответствует *методу простых итераций* (или Якоби) для системы уравнений. Его можно изменить, если использовать приближения к решениям, найденные при выполнении текущей итерации:

$$\begin{aligned}x_1(m+1) &= f_1[x_1(m), x_2(m), \dots, x_N(m)], \\x_2(m+1) &= f_2[x_1(m+1), x_2(m), \dots, x_N(m)], \\&\dots \\x_N(m+1) &= f_n[x_1(m+1), x_2(m+1), \dots, x_N(m)],\end{aligned}\tag{1.6}$$

Такое изменение носит название *метода Зейделя* и приводит, как правило, к ускорению сходимости. Заканчивается итерационный процесс, когда выполняется условие

$$|x_k(m+1) - x_k(m)| < \delta,$$

где $k = 1, 2, \dots, N$, δ – заданная погрешность.

Для сходимости итерационных методов необходимо, чтобы значения диагональных элементов матрицы A были преобладающими по абсолютной величине по сравнению с др. элементами. Выполнение этого условия можно обеспечить путем перестановки в исходной матрице уравнений и неизвестных.

Методы простых итераций и Зейделя имеют разные области сходимости. Их можно применять и к решению систем нелинейных уравнений.

1.4 Задание

1.4.1 Ознакомиться с подразделами 1.1 – 1.3 и ответить на контрольные вопросы.

1.4.2 Ознакомиться с программой анализа линейных схем (файл ...) и составить графические схемы алгоритмов методов решения СЛАУ.

1.4.3 Сформировать модель линейной схемы, предложенной преподавателем.

1.4.3 С использованием программы п. 1.4.2 определить потенциалы узлов схемы представленными в программе методами.

1.4.4 Проанализировать результаты и оформить отчет.

1.5 Контрольные вопросы

1.5.1 Какова цель работы?

1.5.2 Что называют компонентными и топологическими уравнениями?

1.5.2 В чем суть метода узловых потенциалов?

1.5.2 Что представляет собой модель линейной электрической цепи?

1.5.1 Чем отличаются прямые и итерационные методы решения СЛАУ?

1.5.2 Что называют расширенной матрицей коэффициентов при неизвестных?

1.5.3 В чем суть метода Гаусса?

1.5.4 Как будет выглядеть расширенная матрица на каждом шаге алгоритма метода Гаусса при решении системы уравнений

$$\begin{aligned}x_1 + 3x_2 - x_3 &= 3, \\ -x_1 - 2x_2 + 4x_3 &= 1, \\ x_1 + 3x_2 + 2x_3 &= 6.\end{aligned}$$

1.5.5 Что такое вектор начальных приближений?

1.5.6 Что называют областью сходимости итерационных методов?

1.5.7 Чем отличаются методы Якоби и Зейделя?

2 МОДЕЛИРОВАНИЕ ЭЛЕМЕНТОВ

2.1 Цель работы

Изучение алгоритмов формирования моделей элементов и аппроксимации характеристик элементов методом наименьших квадратов.

2.2 Методика формирования моделей элементов

Моделирование элементов обычно включает следующие операции:

1) выбор свойств элемента, подлежащих отражению в модели;

2) сбор информации, о выбранных свойствах;

3) синтез структуры модели;

4) расчет числовых значений параметров;

5) оценка точности и адекватности модели.

Операции 2 – 5 могут выполняться многократно в процессе последовательного приближения к желаемому результату.

Выбор свойств элемента, подлежащих отражению в модели, основан на анализе возможных применений модели и определяет степень ее

универсальности. Источниками сведений о выбранных свойствах могут быть опыт и знания разработчика модели, научно–техническая литература (прежде всего справочная), описания прототипов – имеющихся моделей элементов, близких по своим свойствам исследуемому, результаты измерения параметров и т. п.

Структура модели – общий вид математических соотношений модели без конкретизации числовых значений фигурирующих в них параметров. Структура модели может быть представлена также в графической форме, например в виде эквивалентной схемы. Синтез структуры – наиболее ответственная и трудно формализуемая операция.

Числовые значения параметров определяются из условия минимизации погрешности модели заданной структуры: $\min E_M(X)$, где X – вектор параметров модели, принадлежащий области варьирования параметров X_A ; E_M – определяется в соответствии с выражениями:

$$e_i = (y_{i\bar{n}} - y_{i\bar{e}}) / y_{i\bar{e}}, \quad i = 1, 2, \dots, m; \quad (2.1)$$

$$E_M = \max_{i \in \{1:m\}} e_i. \quad (2.2)$$

Здесь Y_M – функция X , Y_E определяется по результатам экспериментов либо физических, либо вычислительных (с использованием более точных моделей, если таковые имеются в иерархическом ряду).

Для оценки точности должны использоваться значения Y_E , которые не фигурировали в решении предыдущей задачи.

Сведения об области адекватности (ОА) модели могут представляться в виде диапазонов изменения внешних параметров, в которых модель адекватна (т. е. погрешность модели не превышает заданной).

2.3 Моделирование резисторов

Резисторы служат для диссипации (рассеяния) энергии и подразделяются на линейные и нелинейные. При известном падении напряжения V на резисторе и токе J через него сопротивление

$$R = \frac{V}{J}. \quad (2.3)$$

Для линейного резистора сопротивление R постоянно, в случае нелинейного – зависит от напряжения (или тока).

Пусть падение напряжения на резисторе и ток заданы в виде последовательностей V_i и J_i , где $i = 0, 1, \dots, n$; $n+1$ – количество экспериментальных точек. Сопротивление линейного резистора можно вычислить по закону Ома (2.3) с оценкой точности. Для нелинейного резистора следует определить аналитически вид функциональной зависимости $R(J)$. Простой прием решения такой задачи – использование полиномов.

Зададим функциональную связь между последовательностями V_i и J_i в виде полинома степени $(m + 1)$

$$V = a_0 \cdot J + a_1 \cdot J^2 + \dots + a_m \cdot J^{m+1}. \quad (2.4)$$

Разделив левую и правую часть (2.4) на J , получим аналитическую зависимость сопротивления от тока

$$R = a_0 + a_1 \cdot J + \dots + a_m \cdot J^m = \sum_{i=0}^m a_i \cdot J^i. \quad (2.5)$$

Исходя из требуемой точности и удобства использования формулы (2.5) в последующих расчетах, разработчик должен решить, каким количеством членов ограничиться. Если рассматривается достаточно малая область экспериментальной зависимости, то можно воспользоваться, например, квадратичной функцией ($m = 2$)

$$R = a_0 + a_1 \cdot J + a_2 \cdot J^2. \quad (2.6)$$

Для определения коэффициентов a_0 , a_1 , a_2 достаточно трех близлежащих значений V_i и J_i , чтобы составить систему линейных уравнений относительно неизвестных

$$\begin{aligned} \frac{V_1}{J_1} &= a_0 + a_1 \cdot J_1 + a_2 \cdot J_1^2, \\ \frac{V_2}{J_2} &= a_0 + a_1 \cdot J_2 + a_2 \cdot J_2^2, \\ \frac{V_3}{J_3} &= a_0 + a_1 \cdot J_3 + a_2 \cdot J_3^2. \end{aligned} \quad (2.7)$$

При $n > m$ можно проводить полиномиальную аппроксимацию $R(J)$ по методу наименьших квадратов. Если $n = m$, то имеет место обычная интерполяция.

2.4 Вычисление коэффициента парной корреляции

Корреляция – признак, указывающий на взаимосвязь ряда численных последовательностей. Парная корреляция характеризует взаимосвязь двух последовательностей Y_i и X_i , $i = 1..n$.

Коэффициент парной корреляции

$$K = \frac{\sum_{i=0}^n X_i \cdot Y_i - \frac{\left(\sum_{i=0}^n X_i\right) \cdot \left(\sum_{i=0}^n Y_i\right)}{(n+1)}}{\left[\sum_{i=0}^n X_i^2 - \frac{\left(\sum_{i=0}^n X_i\right)^2}{(n+1)}\right]^{\frac{1}{2}} \cdot \left[\sum_{i=0}^n Y_i^2 - \frac{\left(\sum_{i=0}^n Y_i\right)^2}{(n+1)}\right]^{\frac{1}{2}}} \quad (2.8)$$

характеризует степень отклонения связи X_i и Y_i от линейной. Если $|K|$ близок единице, то эта связь линейна, т. е. $Y = A \cdot X + B$. Знак K определяет знак коэффициента A : если $K > 0$, то $A > 0$, и напротив, при $K < 0$ $A < 0$.

Ниже приведена Паскаль – программа вычисления коэффициента K .

```
PROGRAM KOR;      {Вычисление коэффициента парной корреляции}
{$R-} uses Crt;
```

```
type vec=array [0..4] of real; point=^vec;
```

```
var N:byte; K:real; J,V:point; Size:word;
```

```
PROCEDURE TabXY(var N:byte;var X,Y:point);
```

```
  var I:byte; Fd:text; NmFd:string[7];
```

```
  begin
```

```
    Write('File Number:'); Readln(NmFd); NmFd:='r'+NmFd+'.dat';
```

```
    Assign(Fd,NmFd); Reset(Fd); Readln(Fd,N);
```

```
    Size:=SizeOf(real)*(N+1); GetMem(X,Size); GetMem(Y,Size);
```

```
    for I:=0 to N do Readln(Fd,X^[I],Y^[I]); Close(Fd)
```

```
  end;
```

```
PROCEDURE KorXY(N:byte; X,Y:point;var K:real);
```

```
  var I:byte; S:vec;
```

```
  begin
```

```
    for I:=0 to 4 do S[I]:=0.0;
```

```
    for I:=0 to N do begin
```

```
      S[0]:=S[0]+X^[I]; S[1]:=S[1]+Y^[I];
```

```
      S[2]:=S[2]+Sqr(X^[I]); S[3]:=S[3]+Sqr(Y^[I]);
```

```
      S[4]:=S[4]+X^[I]*Y^[I];
```

```
    end;
```

```
    K:=S[4]-S[0]*S[1]/(N+1); K:=K/Sqrt(S[2]-Sqr(S[0])/(N+1));
```

```
    K:=K/Sqrt(S[3]-Sqr(S[1])/(N+1));
```

```

end;
BEGIN
  repeat
    TabXY(N,J,V); KorXY(N,J,V,K); Writeln('K=',K:5:2)
  until ReadKey=#27;
  FreeMem(J,Size); FreeMem(V,Size)
  {$R-}
END.

```

2.5 Метод наименьших квадратов

Пусть структура модели описывается формулой вида

$$R = f(J, a_0, a_1, \dots, a_m), \quad (2.9)$$

где f – известная функция; a_0, a_1, \dots, a_m – независимые постоянные коэффициенты.

Задача нахождения наилучших значений этих коэффициентов сводится к некоторой минимизации отклонений

$$E_i = f(J_i, a_0, a_1, \dots, a_m) - R_i, \quad (2.10)$$

где R_i – значения экспериментальной зависимости $R(J)$ в точках J_i .

Одним из методов решения такой задачи является метод наименьших квадратов (МНК), в котором коэффициенты a_0, a_1, \dots, a_m формулы (2.10) находят из условия минимума суммы квадратов отклонений для всех точек

$$S = \sum_{i=0}^n E_i^2. \quad (2.11)$$

В этом заключается суть МНК. Математически условия минимума суммы квадратов отклонений можно получить, приравняв нулю частные производные от S по коэффициентам a_0, a_1, \dots, a_m .

Наиболее распространен способ выбора функции f в виде линейной комбинации базисных функций

$$R = a_0 \cdot f_0(J) + a_1 \cdot f_1(J) + \dots + a_m \cdot f_m(J). \quad (2.12)$$

В этом случае

$$\begin{aligned} \frac{dS}{da_0} &= 2 \cdot \sum_{i=0}^n [a_0 \cdot f_0(J_i) + a_1 \cdot f_1(J_i) + \dots + a_m \cdot f_m(J_i) - R_i] \cdot f_0(J_i) = 0, \\ \frac{dS}{da_1} &= 2 \cdot \sum_{i=0}^n [a_0 \cdot f_0(J_i) + a_1 \cdot f_1(J_i) + \dots + a_m \cdot f_m(J_i) - R_i] \cdot f_1(J_i) = 0, \\ &\dots \\ \frac{dS}{da_m} &= 2 \cdot \sum_{i=0}^n [a_0 \cdot f_0(J_i) + a_1 \cdot f_1(J_i) + \dots + a_m \cdot f_m(J_i) - R_i] \cdot f_m(J_i) = 0. \end{aligned} \quad (2.13)$$

Раскрывая скобки и выполняя суммирование, получим систему линейных уравнений вида

$$\begin{vmatrix} (f_0, f_0) & (f_0, f_1) & \dots & (f_0, f_m) \\ (f_0, f_1) & (f_1, f_1) & \dots & (f_1, f_m) \\ \dots & \dots & \dots & \dots \\ (f_0, f_m) & (f_1, f_m) & \dots & (f_m, f_m) \end{vmatrix} \cdot \begin{vmatrix} a_0 \\ a_1 \\ \dots \\ a_m \end{vmatrix} = \begin{vmatrix} (f_0, f) \\ (f_1, f) \\ \dots \\ (f_m, f) \end{vmatrix}. \quad (2.14)$$

Матрица коэффициентов при неизвестных системы уравнений (2.14) называется матрицей Грама. Элементы матрицы Грама являются скалярными произведениями базисных функций

$$(f_j, f_k) = \sum_{i=0}^n f_j(J_i) \cdot f_k(J_i). \quad (2.15)$$

Расширенная матрица системы уравнений (1.18) получается добавлением справа к матрице Грама столбца свободных членов. Скалярные произведения, являющиеся элементами столбца, определяются аналогично (2.15)

$$(f_j, f) = \sum_{i=0}^n f_j(J_i) \cdot R(J_i). \quad (2.16)$$

Выбор конкретных базисных функций f_0, f_1, \dots, f_m зависит от свойств аппроксимируемой зависимости $R(J)$, таких как периодичность, экспоненциальный или логарифмический характер, свойства симметрии, наличие асимптотики и т. д.

Ниже приведена Паскаль – программа метода наименьших квадратов. Вид базисных функций задается в процедуре Func. Формирование матрицы Грама производится в процедуре Gram. Предварительно значения базисных функций $F_i[J]$ ($J = 0..M$) в каждом узле $X[I]$ ($I = 0..N$) заносятся в элементы двумерного массива $F[J,I]$ (матрица плана); затем по формулам (2.15) и (2.16) вычисляются скалярные произведения S, R и заполняется расширенная матрица $MatG$. В процедуре GrQ производится вычисление среднеквадратичного отклонения (см. следующий раздел).

```
PROGRAM MNQ;                                {Метод наименьших квадратов}
uses Crt;
```



```

const Mmax=6; Nmax=255;
type vecN=array [0..Nmax] of real;
    vecM=array [0..Mmax] of real;
    mat=array [0..Mmax,0..Mmax+1] of real;
var I,N,M:byte; S:real;
    A:vecM; J,R:vecN; MatG:mat;
PROCEDURE TabXY(var N:byte;var X,Y:vecN);
                                {N пар значений R(J) - в массивы Y, X}
var I:byte; Fd:text; NmF:string[7];
begin
    Write('File Number:');
    Read(NmF); NmF:='r'+NmF+'.dat';
    Assign(Fd,NmF); Reset(Fd); Readln(Fd,N);
    for I:=0 to N do begin
        Readln(Fd,X[I],Y[I]); Y[I]:=Y[I]/X[I] end;
    Close(Fd);
end;
PROCEDURE Func(M:byte; X:real;var Fi:vecM);
                                {Вычисление базисных функций}
var I:byte;
begin
    Fi[0]:=1.0;
    for I:=1 to M do Fi[I]:=Fi[I-1]*X
end;
PROCEDURE Gauss(M:byte;var A:mat;var X:vecM); {Метод Гаусса}
var I,J,K:byte; Piv,Pm:real;
begin
    for K:=0 to M do begin Piv:=A[K,K]; {Прямой ход}
        for I:=K+1 to M do begin Pm:=A[I,K]/Piv;
            for J:=K+1 to M+1 do A[I,J]:=A[I,J]-A[K,J]*Pm
        end
    end;

```

```

for I:=M downto 0 do begin                                     { Обратный ход}
  Pm:=A[I,M+1]; Piv:=A[I,I];
  for J:=I+1 to M do Pm:=Pm-A[I,J]*X[J];
  X[I]:=Pm/Piv
end
end;
PROCEDURE Gram(X,Y:vecN; M,N:byte;var MatG:mat);
                                     { Формирование расширенной матрицы Грама}
var P:array [0..Mmax,0..Nmax] of real;
    I,J,K:byte; S,R:real; Fi:vecM;
begin
  for I:=0 to N do begin
    Func(M,X[I],Fi);
    for J:=0 to M do P[J,I]:=Fi[J]
  end;
  for K:=0 to M do begin
    for J:=K to M do begin
      S:=0.0; R:=0.0;
      for I:=0 to N do begin
        S:=S+P[K,I]*P[J,I];
        if J=M then R:=R+P[K,I]*Y[I]
      end;
      MatG[K,J]:=S; MatG[J,K]:=S
    end;
    MatG[K,M+1]:=R
  end
end;
PROCEDURE GrQ(X,Y:vecN; M,N:byte; A:vecM;var S:real);
                                     { Вычисление среднеквадратичного отклонения}
var I,K:byte; Q:real;
begin
  S:=0.0;

```

```

for I:=0 to N do begin Q:=A[M];
  for K:=1 to M do Q:=Q*X[I]+A[M-K];
  S:=S+Sqr(Q-Y[I])
end;
S:=Sqrt(S/(N*(N+1)))
end;
BEGIN                                     {Головная программа}
TabXY(N,J,R);
repeat
  Write('M:'); Read(M);
  Gram(J,R,M,N,MatG); Gauss(M,MatG,A);
  for I:=0 to M do Writeln('A(',I:1,')=',A[I]);
  GrQ(J,R,M,N,A,S); Writeln('Ескв=',S);
until ReadKey=#27
END.

```

2.6 Точность математической модели

Точность модели резистора можно оценить путем определения разности ординат реальной и аппроксимирующей (идеальной) зависимостей $R(J)$ в определенном множестве точек. Для этого требуется каким-то образом из множества разностей ординат \dot{a}_i (вектора) получить одно значение (скалярное), служащее в определенном смысле мерой точности и зависящее от ошибок \dot{a}_i . Такая функция ошибок должна уменьшаться с уменьшением расхождения реальной и идеальной зависимостей $R(J)$ и обращаться в нуль, если эти зависимости совпали.

Указанным требованиям удовлетворяет функция

$$E(2 \cdot p) = \left(\sum_{i=0}^n \dot{a}_i^{2p} \right)^{\frac{1}{2p}}, \quad (2.17)$$

где показатель степени p больше или равен единице.

При $p = 1$ функция ошибок

$$E(2) = \left(\sum_{i=0}^n \dot{a}_i^2 \right)^{\frac{1}{2}} \quad (2.18)$$

часто трактуется как евклидова норма, являющаяся основной мерой расстояния от начала координат до точки в $(n + 1)$ -мерном пространстве.

Рост показателя степени функции ошибок автоматически увеличивает вес больших ошибок. При p , стремящемся к бесконечности, функция ошибок пропорциональна максимальной:

$$E_{\max} = \max |\dot{a}_i|. \quad (2.19)$$

На практике для сведения векторной оценки погрешности к скалярной, преимущественно используют (2.19) или среднеквадратичное отклонение:

$$E_{\text{н\acute{e}а}} = \left\{ \sum_{i=0}^n \frac{\dot{a}_i^2}{[n \cdot (n+1)]} \right\}^{\frac{1}{2}}. \quad (2.20)$$

При реализации численных методов необходимо учитывать:

- δ_1 – алгоритмические погрешности – погрешности описания моделей;
- δ_2 – трансформированные погрешности – погрешности, которые накапливаются в ходе решения задачи и обуславливаются погрешностями исходных данных;
- δ_3 – вычислительные погрешности – обуславливающиеся погрешностями выполнения арифметических действий.

Пусть имеем задачу

$$Y = F(X). \quad (2.21)$$

Обозначим: $F^*(X)$ - приближенный алгоритм; X^* – приближенные значения исходных данных ($X^* = X \pm \Delta X$); $F^{**}(X^*)$ – приближенный алгоритм с учетом вычислительных погрешностей и погрешностей исходных данных.

Тогда

$$\begin{aligned} \delta_1 &= F(X) - F^*(X); \\ \delta_2 &= F^*(X) - F^*(X^*); \\ \delta_3 &= F^*(X^*) - F^{**}(X^*). \end{aligned} \quad (2.22)$$

Полная погрешность

$$\delta = F(X) - F^{**}(X^*), \quad (2.23)$$

отсюда

$$|\delta| \leq |\delta_1| + |\delta_2| + |\delta_3|. \quad (2.24)$$

Для уменьшения вычислительных погрешностей при сложении или вычитании длинной последовательности чисел целесообразно работать сначала с наименьшими числами. Следует избегать вычитания двух почти равных чисел (формулы можно преобразовать так, чтобы в этом не было необходимости). В любом случае следует сводить к минимуму число необходимых арифметических действий.

2.7 Задание

2.7.1 Ознакомиться с подразделами 2.1 – 2.6 и ответить на контрольные вопросы.

2.7.2 Файлы Rwn.dat содержат вольтамперные характеристики резисторов в виде последовательностей пар значений V_i и J_i ; $n = 1..4$ обозначает номер резистора, w – номер варианта. Вычислить коэффициенты парной корреляции и определить какие файлы соответствуют линейным резисторам, а какие – нелинейным. Вариант w задается преподавателем.

2.7.3 Ознакомиться с программой МНК и составить схемы алгоритмов головной программы и процедуры формирования матрицы Грама.

2.7.4 Определить сопротивление линейных и параметры полиномиальной аппроксимации нелинейных резисторов с оценкой точности (исходными являются файлы Rwn.dat). Принять во внимание, что вольтамперные характеристики всех четырех резисторов измерены с одинаковой точностью.

2.7.5 Проанализировать результаты и оформить отчет.

2.8 Контрольные вопросы

2.8.1 Какова цель работы?

2.8.4 Что такое корреляция, аппроксимация, интерполяция?

2.8.5 Что представляют собой математические модели элементов системы? Как они формируются?

2.8.6 Чем определяется точность модели?

2.8.7 Что такое область адекватности?

2.8.8 Точное описание некоторого свойства технического объекта дается выражением $Y = X^2$. Какова область адекватности математической модели $Y = 2X - 1$ (в одномерном пространстве переменной X), если точность модели должна быть не хуже 20 %?

2.8.9 В чем суть метода наименьших квадратов?

3 АНАЛИЗ НЕЛИНЕЙНЫХ СХЕМ

3.1 Цель работы

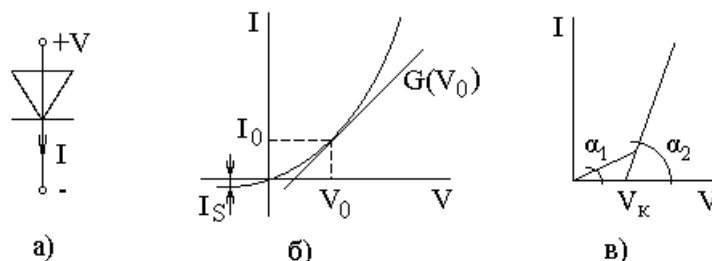
Изучение моделей диода и алгоритма анализа нелинейных схем.

3.2 Модели диода

По степени универсальности модели полупроводниковых приборов подразделяются:

- 1) на динамические и статические;
- 2) линейные и нелинейные;
- 3) для большого и малого сигналов;
- 4) высокочастотные и низкочастотные.

Данная классификация позволяет организовать иерархический ряд моделей,



отличающихся вычислительными затратами и допускающими переход от одной модели к другой в процессе моделирования. Очевидно, что наиболее точной и универсальной в таком ряду является динамическая высокочастотная модель большого сигнала.

Рисунок 3.1 – Графическое обозначение (а), ВАХ (б) диода и кусочно-линейная аппроксимация прямой ветви ВАХ (в)

Вольтамперная характеристика (ВАХ) полупроводникового диода (см. рис. 3.1 описывается выражением

$$I = I_s [\exp(V / U_T) - 1], \quad (3.1)$$

где q – заряд электрона; M – коэффициент неидеальности, учитывающий отклонение реальной характеристики от теоретической характеристики p - n перехода; k – постоянная Больцмана; T – температура в Кельвинах; V – напряжение, приложенное к диоду.

Идеальный диод можно представить как нелинейный резистор. В эквивалентных схемах для такого элемента иногда используют обозначение рис. 3.1, а, имея в виду безинерционный нелинейный двухполюсник с диодной ВАХ. Чаще статическую модель диода представляют в виде источника тока, управляемого напряжением. В случае кусочно-линейной аппроксимации вольтамперной характеристики (рис. 3.1, в) управляемый напряжением источник тока I_{Π} можно рассматривать как упрощенную статическую модель диода, где V_K , α_1 , α_2 – ее параметры.

Когда к диоду приложено постоянное напряжение V_0 , через него протекает постоянный ток I_0 . Эти величины определяют рабочую точку диода. На V_0 могут накладываться малые приращения напряжения, которые связаны с изменениями тока дифференциальной проводимостью

$$dI / dV \Big|_{V=V_0} = (I_s / U_T) \exp(V_0 / U_T), \quad (3.3)$$

называемой динамической проводимостью диода. Величина $G(V_0)$, равная крутизне ВАХ в точке V_0 (см. рис. 3.1, б), представляет собой низкочастотную модель диода для малого сигнала.

Низкочастотная малосигнальная модель диода не содержит инерционных элементов. Модель диода для высоких частот (рис. 3.2, а) включает динамическую проводимость $G(V_0)$ и емкость $C(V_0)$, зависящие от рабочей точки.

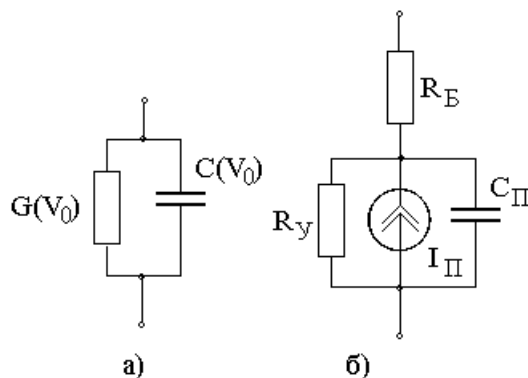


Рисунок 3.2 – Эквивалентная схема малосигнальной (а) и нелинейной динамической (б) модели диода

Эквивалентная схема более общей нелинейной динамической модели диода (рис. 3.2, б) содержит управляемый напряжением источник тока I_n , сопротивление диода R_δ при прямом смещении (сопротивление базы), сопротивление утечки R_y , емкость C_n р – n перехода. Модель основана на уравнении Эберса – Молла для управляемого источника тока и учитывает зависимость барьерной C_δ и диффузионной C_d составляющих емкости C_n от напряжения:

$$\begin{aligned} V_d &= I_d R_\delta + V_n, \\ I_d &= I_n + V_n / R_y + C_n dV_n / dt, \\ I_n &= I_s [\exp(V_n / U_T) - 1], \\ C_n &= T_d I_n / U_T + C_{\delta 0} (1 + Q V_n / U_k), \text{ при } V_n > 0, \\ C_n &= T_d I_n / U_T + C_{\delta 0} (1 - V_n / U_k)^{-Q}, \text{ при } V_n \leq 0, \\ U_T &= kTM / q, \end{aligned} \quad (3.4)$$

где U_k – контактная разность потенциалов; Q – коэффициент, учитывающий распределение примесей в р – n переходе (принимает значения от 1/2 до 1/3); T_d – постоянная времени диода; $C_{\delta 0}$ – барьерная емкость р-п перехода при напряжении $V_n = 0$. При $M = 1$ и $T = 300$ К тепловой потенциал $U_T = 26$ мВ.

Если диод открыт напряжением $V_n > 4U_T$ выполняются условия $\exp(V_n/U_T) \gg 1$, $I_y = V_n/R_y \ll I_n$, т. е. уравнение прямой ветви ВАХ может быть представлено в виде:

$$I = I_s \exp(V / U_T) \quad (3.5)$$

Для малого прямого напряжения $I = -I_s$. При обратном смещении выполняется условие $I_y \gg I_n$ и уравнение обратной ветви ВАХ диода имеет вид

$$I_d = I_y = V_n / R_y. \quad (3.6)$$

Диффузионная емкость $C_d = T_d I_n / U_T$ может достигать больших значений, чем барьерная емкость C_δ . Барьерная доминирует при обратном, диффузионная – при прямом смещении.

3.3 Определение параметров модели диода

Статические параметры I_s , R_δ , U_T модели диода определяются из условия

$$\sum_{i=1}^N (V_{yi} - V_{pi}) = \min, \quad (3.7)$$

где V_{yi} , V_{pi} – экспериментальное и расчетное значения падения напряжения на диоде при токе I_i ; N – число экспериментальных точек ВАХ.

При обработке точек прямой ветви ВАХ по методу наименьших квадратов условие (3.7) с учетом (3.5) сводится к системе уравнений

$$\begin{aligned} A_{11}G + A_{12}U_T + A_{13}R_{\dot{a}} &= A_{14}, \\ A_{21}G + A_{22}U_T + A_{23}R_{\dot{a}} &= A_{24}, \\ A_{31}G + A_{32}U_T + A_{33}R_{\dot{a}} &= A_{34}, \end{aligned} \quad (3.8)$$

где $G = -U_T \ln I_s$, $I_s = \exp(-G/U_T)$, $A_{11} = N$, $A_{12} = A_{21} = \sum_{i=1}^N \ln I_i$,

$$A_{13} = A_{31} = \sum_{i=1}^N I_i, \quad A_{14} = \sum_{i=1}^N I_i, \quad A_{22} = \sum_{i=1}^N (\ln I_i)^2, \quad A_{24} = \sum_{i=1}^N (\ln I_i) V_i, \quad A_{33} = \sum_{i=1}^N (\ln I_i)^2,$$

$$A_{32} = A_{23} = \sum_{i=1}^N (\ln I_i) I_i, \quad A_{34} = A_{43} = \sum_{i=1}^N \ln I_i.$$

Неизвестные I_s , U_T , $R_{\dot{a}}$ уравнений (3.8) определяют с помощью обычных методов решения систем линейных уравнений. Значение параметра R_y , моделирующего сопротивление утечки при обратном включении диода, определяется по формуле

$$R_y = V_{\dot{a}0} / I_{\dot{a}0} \quad (3.9)$$

где $I_{\text{обр}}$ – заданное значение обратного тока при обратном напряжении $V_{\text{обр}}$.

Ниже представлен фрагмент Паскаль – программы определения параметров I_s , U_T , $R_{\dot{a}}$ из прямой ветви ВАХ диода. От программы, рассмотренной в предыдущей работе, она отличается набором базисных функций (процедура Func).

```
PROGRAM MNQDIOD;
```

```
...
```

```
PROCEDURE Func(M:byte; X:real; var Fi:vecM);
```

```
  begin
```

```
    Fi[0]:=1.0; Fi[1]:=Ln(X); Fi[2]:=X
```

```
  end;
```

```
...
```

```
BEGIN
```

```
  TabXY(N,J,V); Gram(J,V,N,MatG); Gauss(M,MatG,A);
```

```
  Is:=Exp(-A[0]/A[1]);
```

```
  Writeln('Is=',Is:10,'; Ut=',A[1]:5:3,'; Rb=',A[2]:5:3);
```

```
  for I:=0 to M do Write(' A(',I:1,')=',A[I]:10); Writeln;
```

```
  GrQ(J,V,M,N,A,S,E); Writeln('S=',S:10,'; E=',E:10);
```

```
  repeat until ReadKey=#27
```


END.

3.4 Метод Ньютона – Рафсона

Пусть $x = r$ – решение скалярного уравнения

$$f(x) = 0,$$

$x = x(n - 1)$ – некоторое приближенное значение r . Тогда, по теореме Тейлора,

$$f(r) = f[x(n - 1)] + [r - x(n - 1)]f'[x(n - 1)] + \dots,$$

где штрих обозначает производную функции по x .

Если оставить только линейный член в разложении и обозначить $x(n)$ как новое приближение к r , то с учетом $f(r) = 0$ можно записать

$$0 = f[x(n - 1)] + [x(n) - X(n - 1)]f'[x(n - 1)],$$

откуда следует формула для ньютоновских итераций

$$x(n) = x(n - 1) - f[x(n - 1)]/f'[x(n - 1)], \quad (3.10)$$

3.5 Моделирование нелинейной схемы

На постоянном токе простая схема, представленная на рис. 3.3, а, преобразуется к виду, показанному на рис. 3.3, б, поскольку катушка индуктивности представляет собой короткозамкнутый отрезок, а конденсатор – разрыв цепи.

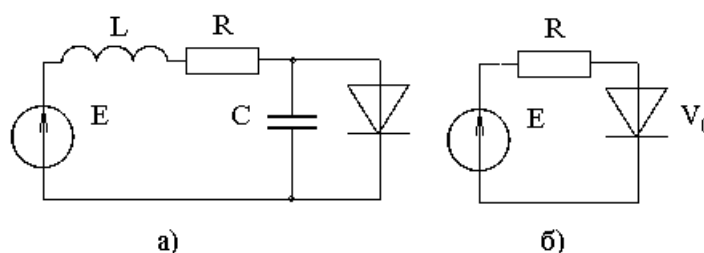


Рисунок 3.3 – Схема электрическая принципиальная (а) и эквивалентная схема (б) на постоянном токе

По законам Кирхгофа и Ома,

$$E = I R + V_0.$$

Падение напряжения на диоде

$$V_0 = I R_A + U_0 \ln(I/I_s + 1).$$

Отсюда

$$E = I(R + R_A) + U_T \ln(I/I_s + 1). \quad (3.11)$$

При заданном E ток I можно найти из уравнения (3.11) численными методами. Используя метод Ньютона – Рафсона, получим

$$I(n) = I(n - 1) - \frac{I(n - 1)(R + R_A) + U_T \ln[I(n - 1)/I_s + 1] - E}{R + R_A + U_T/[I(n - 1) + I_s]}.$$

При использовании метода простых итераций

$$I(n) = \{E - U_T \ln[I(n-1)/I_s + 1]\} / (R + R_a).$$

Последовательность $x(n)$ может сходиться к точному решению r , а может и не сходиться. Скорость сходимости исследуют, вычисляя ошибку

$$e(n) = r - x(n)$$

на каждой итерации. Эта ошибка обусловлена усечением ряда Тейлора при выводе формулы (3.12) и связана с ошибкой на предыдущем шаге соотношением

$$e(n) = -0.5[f''(r) / f'(r)]e^2(n-1).$$

Таким образом, метод Ньютона – Рафсона имеет квадратичную сходимость: ошибка на каждом последующем шаге итераций пропорциональна квадрату ошибки на предыдущем шаге.

Условия, при которых метод Ньютона – Рафсона будет сходиться, удобно проиллюстрировать графически. На рис. 3.4, а приведен график монотонной функции $f(x)$ и проведена касательная в точке $x(n-1)$.

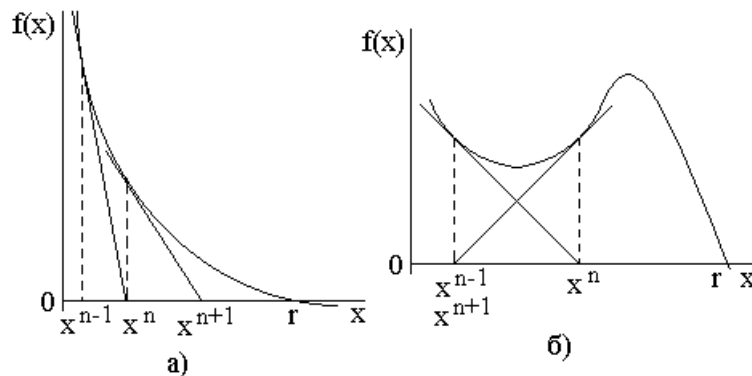


Рисунок 3.4 – Условия сходимости метода Ньютона-Рафсона

Новое приближенное значение решения $x(n)$ определяется выражением

$$f'[x(n-1)] = \frac{f[x(n-1)] - 0}{x(n-1) - x(n)}.$$

Это выражение может быть преобразовано в формулу для итераций. На рис. 3.4, а видно, что каждое последующее значение $x(n)$ все больше приближается к точному решению r уравнения $f(x) = 0$, т. е. метод сходится.

Рассмотрим функцию другого вида (рис. 3.4, б). Здесь значение $x(n+1)$ находится дальше от r , чем значение $x(n-1)$. Происходят колебания около локального минимума функции $f(x)$, т. е. при таком выборе начальных условий метод не сходится. Обеспечить сходимость можно, если изменить начальные условия так, чтобы новое приближенное значение решения оказалось справа от минимума функции $f(x)$.

Ниже приведена программа определения рабочей точки диода методами Ньютона – Рафсона и простых итераций для схемы, представленной на рис. 3.3. При анализе других нелинейных схем следует модифицировать ввод и подпрограммы вычисления функций F и PF .

```

PROGRAM IterNR;
const E=1E-5; Kmax=10;
type func=FUNCTION (X:real): real;
var Is,Rb,Ut,Vc,R,Inp,I,V,dI,dV:real; K:byte;
{$F+}
FUNCTION F(I:real):real;
  begin F:=(Vc-Ut*Ln(I/Is+1))/(R+Rb) end;
FUNCTION FP(I:real):real;
  var F,P:real;
  begin
    F:=(I*(R+Rb)+Ut*Ln(I/Is+1)-Vc);
    P:=R+Rb+Ut/(I+Is); FP:=F/P
  end;
{$F-}
FUNCTION Vn(I:real):real;
  begin Vn:=I*Rb+Ut*Ln(I/Is+1) end;
PROCEDURE Iter(var K:byte; var X:real; F:func);
  var X1:real;
  begin K:=0;
    repeat
      X1:=X; X:=F(X); K:=K+1;
      if K>Kmax then begin Writeln('It: K>Kmax'); Exit end
    until Abs(X1-X)<E;
  end;
PROCEDURE NR(var K:byte; var X:real; FP:func);
  var F:real;
  begin K:=0;
    repeat
      F:=FP(X); X:=X-F; K:=K+1;
      if K>Kmax then begin Writeln('NR: K>Kmax'); Exit end
    until Abs(F)<E;
  end;

```

BEGIN

```

Write('Is='); Readln(Is); Write('Ut='); Readln(Ut);
Write('Rb='); Readln(Rb); Write('Vc='); Readln(Vc);
Write('R ='); Readln(R); Write('Inp='); Readln(Inp);
I:=Inp; Iter(K,I,F);
Writeln('It: K=',K:2,'; Io=',I:6:4,'; Vo=',Vn(I):6:4);
I:=Inp; NR(K,I,FP); V:=Vn(I);
Writeln('NR: K=',K:2,'; Io=',I:6:4,'; Vo=',V:6:4);
Inp:=I*(E+1); Writeln('G(Vo)=',(Inp-I)/(Vn(Inp)-V):6:4)

```

END.**3.6 Задание**

3.6.1 Ознакомиться с подразделами 3.1 – 3.5 и ответить на контрольные вопросы.

3.6.2 Файл D*.DAT содержит последовательность точек V_i , I_i прямой ветви ВАХ диода D* (номер диода, обозначенный знаком *, выдается преподавателем). С использованием программы MNQDIOD определить параметры I_s , U_T , R_σ нелинейной модели диода с оценкой точности.

3.6.4 Ознакомиться с программой определения рабочей точки диода методами простых итераций и Ньютона – Рафсона. Составить схемы алгоритмов анализа.

3.6.3 Сформировать модель диодной схемы на постоянном токе. Схема задается преподавателем.

3.6.4 Определить рабочую точку диода и параметр $G(V_o)$ малосигнальной модели.

3.6.5. Проанализировать результаты и оформить отчет.

3.7 Контрольные вопросы

3.7.1 Какова цель работы?

3.7.2 Что представляют собой модели диодов и каков смысл их параметров?

3.7.3 Какова природа диффузионной и барьерной емкости?

3.7.4 Чем определяется коэффициент Q в модели (3.4)?

3.7.5 Что представляет собой модель диодной схемы на постоянном токе?

3.7.6 Как определить рабочую точку диода?

3.7.7 В чем суть методов простых итераций и Ньютона - Рафсона?

4 МОДЕЛИРОВАНИЕ СТАТИЧЕСКИХ РЕЖИМОВ**4.1 Цель работы**

Изучение нелинейных моделей биполярных транзисторов и алгоритмов моделирования статических режимов нелинейных схем.

4.2 Модели биполярного транзистора

В качестве нелинейных моделей биполярного транзистора наибольшее распространение получили различные модификации модели Эберса – Молла, в основе которой лежит представление транзистора как системы двух взаимодействующих диодов.

Инжекционный вариант модели Эберса – Молла, соответствующий эквивалентной схеме рис. 4.1, а, представляют уравнения

$$\begin{aligned} I_K &= \alpha_n I_{ДЭ} - I_{ДК}; \\ I_Э &= \alpha_i I_{ДК} - I_{ДЭ} \end{aligned} \quad (4.1)$$

при

$$\begin{aligned} I_{ДЭ} &= I_{ЭН} [\exp(V_Э/U_T) - 1], \\ I_{ДК} &= I_{КН} [\exp(V_К/U_T) - 1], \end{aligned} \quad (4.2)$$

где $I_{ЭН}$, $I_{КН}$ – токи насыщения переходов база – эмиттер и база – коллектор; $V_Э$, $V_К$ – напряжения на переходах; α_n , α_i – коэффициенты прямой и обратной передачи тока в схеме с общей базой; $U_T = kT/q$, k – постоянная Больцмана, T – температура, q – заряд электрона.

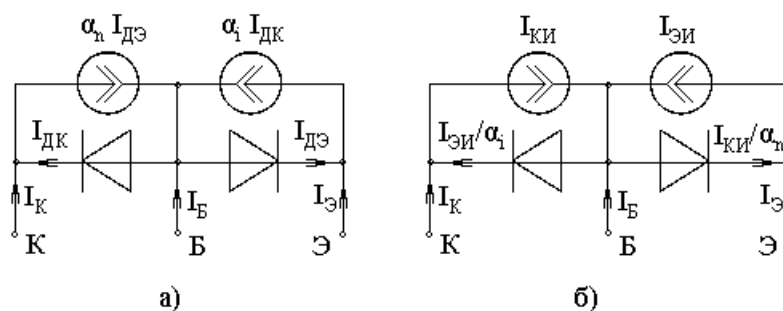


Рисунок 4.1 – Инжекционная (а) и передаточная (б) модели Эберса – Молла

Физические свойства транзистора требуют, чтобы выполнялось условие

$$\alpha_n I_{ЭН} = \alpha_i I_{КН} = I_H. \quad (4.3)$$

Используя I_H , можно записать уравнение (4.1) в форме наиболее подходящей для ЭВМ, и уменьшить число параметров с четырех до трех:

$$\begin{aligned} I_K &= I_{КИ} - I_{ЭИ}/\alpha_i, \\ I_Э &= I_{ЭИ} - I_{КИ}/\alpha_n, \end{aligned} \quad (4.4)$$

где $I_{КИ,ЭИ} = I_H [\exp(V_{Э,К}/U_T)]$ – коллекторный и эмиттерный источники тока.

Уравнения (4.4) представляют передаточный вариант модели Эберса – Молла (рис. 4.1, б). Еще одну модель можно получить, воспользовавшись связью между коэффициентами передачи тока в схемах с общей базой (α_n , α_i) и общим эмиттером (β_n , β_i):

$$\alpha_{n,i} = \beta_{n,i}/(1 + \beta_{n,i}). \quad (4.5)$$

Подставив (4. 5) в (4. 4), получим:

$$\begin{aligned} I_K &= (I_{KI} - I_{ЭИ}) - I_{ЭИ}/\beta_i, \\ I_{Э} &= -I_{KI}/\beta_n - (I_{KI} - I_{ЭИ}). \end{aligned} \quad (4. 6)$$

Введя обозначение $I_{КТ} = (I_{KI} - I_{ЭИ})$, смоделируем эти уравнения схемой с одним источником тока (рис. 4.2, а). Эту модель называют нелинейной гибридной П – моделью.

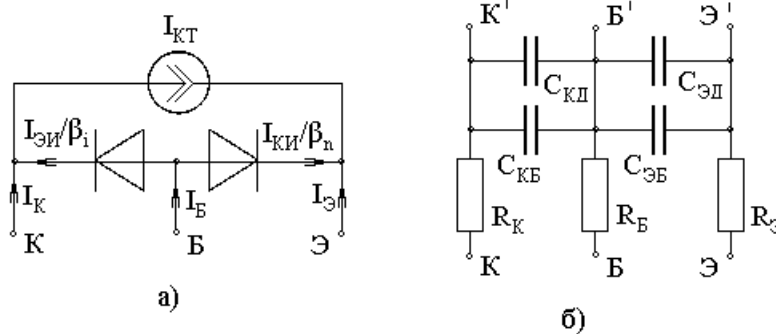


Рисунок 4.2 – Гибридная (а) и более общая (б) модели Эберса – Молла

Рассмотренные модели пригодны только для низких частот. Эквивалентная схема более общей нелинейной модели (рис. 4. 2, б) содержит: $C_{КД}$, $C_{ЭД}$, $C_{КБ}$, $C_{ЭБ}$ – диффузионные и барьерные емкости переходов; $R_{Э}$, $R_{Б}$, $R_{К}$ – сопротивления объема областей эмиттера, базы и коллектора, а также их внешних выводов; низкочастотные модели, рассмотренные ранее, могут быть подключены к точкам Б', К' и Э'.

Обычно сопротивления считаются постоянными. Типичные значения $R_{К}$ составляют от нескольких Ом до нескольких сотен Ом; $R_{Э}$ – обычно 1 Ом, и им часто пренебрегают; $R_{Б}$ – около 10 Ом для СВЧ и до нескольких кОм для низкочастотных транзисторов. Емкости переходов зависят от напряжений и описываются уравнениями, подобными тем, которые рассматривались для диода.

Для учета эффекта Эрли в модель Эберса – Молла можно добавить источник тока $(V_K/V_A) I_{Э}$ между внутренним эмиттером и коллектором, где V_A – напряжение Эрли. В результате число параметров модели увеличится. Дополнительно в основную модель могут быть введены параметры, учитывающие зависимость α_n и α_i от плотности тока и рабочей частоты. Можно дополнительно подсоединить к базе диоды, учитывающие двумерный эффект концентрации тока вдоль перехода база – эмиттер. Очевидно, что уточнение модели неизбежно приводит к увеличению числа параметров и модель транзистора становится более сложной.

Еще одна нелинейная динамическая модель биполярного транзистора – модель Гуммеля – Пуна основана на интегральных соотношениях для зарядов и связывает внешние электрические характеристики транзистора с зарядом в базе. Это очень точная модель, объясняющая многие физические эффекты, соответственно для ее описания требуется много параметров.

Последовательное упрощение модели Гуммеля – Пуна в конце концов приводит к простейшей модели Эберса – Молла.

При анализе функционирования транзисторов на малом переменном сигнале используют линейные модели.

4.3 Формирование модели схемы

Для моделирования статических режимов нелинейных схем обычно используется метод узловых потенциалов, как наиболее подходящий для расчетов на ЭВМ. Для каждого узла схемы составляют уравнение баланса токов, причем ток, втекающий в узел, берется со знаком минус, а вытекающий – со знаком плюс. Полученная система уравнений решается методом Ньютона – Рафсона или др. численным методом, пригодным для решения системы нелинейных уравнений. Предварительно из моделируемой схемы исключают реактивные элементы, а источники напряжения, если они есть, преобразуют в эквивалентные источники тока.

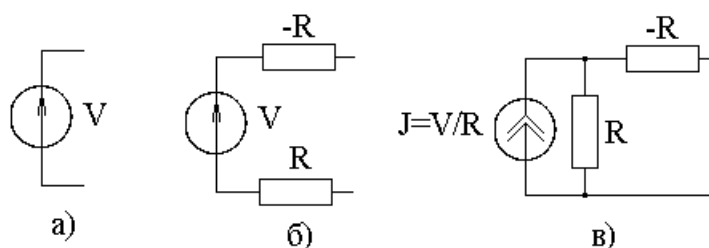


Рисунок 4.3 – Преобразование источника напряжения в эквивалентный источник тока

Простой способ преобразования источника напряжения в источник тока основан на теореме Нортна. На рис. 4.3, а показан идеальный источник напряжения, на рис. 4.3, б представлен тот же источник с двумя включенными последовательно с ним резисторами. Сопротивления резисторов имеют одинаковые по модулю значения и разные знаки, что не изменяет процессов в схеме. На рис. 4.3, в приведена эквивалентная схема с источником тока и параллельно ему включенным резистором. Эта эквивалентная схема может быть использована при записи уравнений. Значение сопротивления R может быть выбрано с учетом удобства проведения вычислений. Таким образом, источники напряжения любых видов, даже с нелинейной зависимостью от переменных в других ветвях, могут быть учтены при составлении уравнений.

В общем случае модель нелинейной схемы представляет собой систему нелинейных алгебраических уравнений

$$F(X) = 0, \quad (4.7)$$

где F – вектор–функция, X – вектор базисных координат.

Рассмотрим формирование модели методом узловых потенциалов на примере транзисторного усилителя (рис. 4.4, а). Используем инжекционную модель транзистора (см. рис. 4.1, а). Преобразуя источник напряжения в эквивалентный источник тока, в соответствии с теоремой Нортна, и

учитывая сопротивление базы транзистора R_B , получим эквивалентную схему усилителя, показанную на рис. 4.4, б. Уравнения для узлов этой схемы можно представить в виде

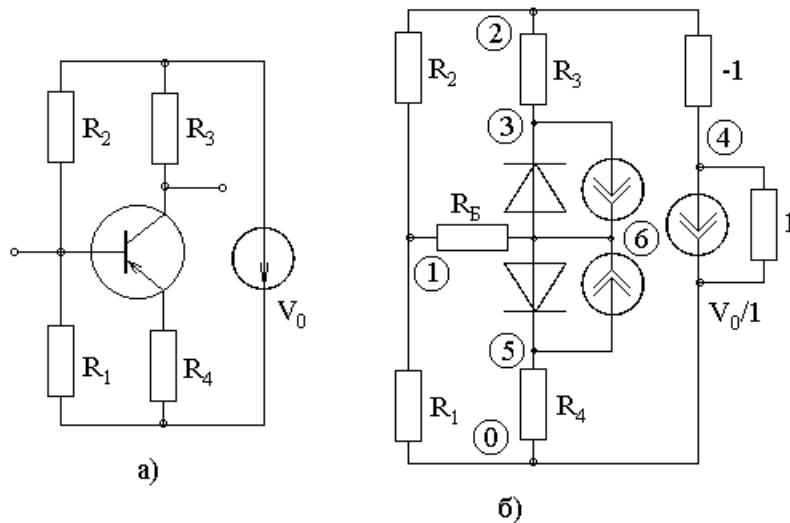


Рисунок 4.4 – Принципиальная (а) и эквивалентная (б) схемы транзисторного усилителя

$$\begin{aligned}
 i_1 &= v_1/R_1 + (v_1-v_2)/R_2 + (v_1-v_6)/R_B, \\
 i_2 &= (v_2-v_1)/R_2 + (v_2-v_3)/R_3 + (v_2-v_4)/(-1), \\
 i_3 &= (v_3-v_2)/R_3 - I_{36}, \\
 i_4 &= v_4/1 + (v_4-v_2)/(-1), \\
 i_5 &= v_5/R_5 + I_{56}, \\
 i_6 &= (v_6-v_1)/R_B + I_{36} - I_{56}, \\
 i_{36} &= I_{KH} \{ \exp[(v_6-v_3)/U_T] - 1 \} - \alpha_n I_{ЭН} \{ \exp[(v_6-v_5)/U_T] - 1 \}, \\
 i_{56} &= \alpha_i I_{KH} \{ \exp[(v_6-v_3)/U_T] - 1 \} - I_{ЭН} \{ \exp[(v_6-v_5)/U_T] - 1 \}.
 \end{aligned} \tag{4.8}$$

В векторной форме (4.8) имеет вид

$$I = W(V), \tag{4.9}$$

где компоненты вектора $W(V)$ являются функциями v_1, v_2, \dots, v_6 , а вектора I – равны нулю все, кроме четвертой: компонента i_4 равна току независимого источника $I_0 = V_0/1$.

Перепишем (4.9) в форме

$$F(V) = W(V) - I = 0. \tag{4.10}$$

Для решения (4.10) начальное приближение можно выбрать из физических соображений, а далее применять метод Ньютона – Рафсона.

4.4 Метод Ньютона для системы уравнений

Предположим, что система нелинейных алгебраических уравнений (4.7) имеет решение. Обозначим его через X^* и разложим компоненты вектор–

функции F в ряд Тейлора в окрестности решения. Предположив далее, что X близко к X^* , пренебрежем членами высокого порядка и запишем систему уравнений в линеаризованной форме

$$F(X^*) = F(X) + A (X^* - X), \quad (4.12)$$

где

$$A_{|x} = \begin{vmatrix} \partial f_1 / \partial x_1 & \partial f_1 / \partial x_2 & \dots & \partial f_1 / \partial x_N \\ \partial f_2 / \partial x_1 & \partial f_2 / \partial x_2 & \dots & \partial f_2 / \partial x_N \\ \dots & \dots & \dots & \dots \\ \partial f_N / \partial x_1 & \partial f_N / \partial x_2 & \dots & \partial f_N / \partial x_N \end{vmatrix}_x$$

– матрица Якоби функции F .

Если приравнять нулю правую часть уравнения (4.12), то решение не будет равно вектору X^* , т. к. пренебрегли членами высокого порядка. В результате получим некоторое новое значение X . Используя индексы в круглых скобках для обозначения последовательности итераций, получаем

$$F[X(k)] + M[X(k+1) - X(k)] = 0. \quad (4.13)$$

Формально решение (4.13) определяется как

$$X(k+1) = X(k) - F[X(k)]/A. \quad (4.14)$$

На практике же матрица Якоби не обращается. Обозначив

$$dX(k) = X(k+1) - X(k),$$

перепишем (4.14)

$$A dX(k) = -F[X(k)]. \quad (4.15)$$

Решение находим методом Гаусса (или др. методом решения систем линейных уравнений), и значение $X(k+1)$ определяем из выражения

$$X(k+1) = X(k) + dX(k). \quad (4.16)$$

Формулы (4.15), (4.16) и представляют метод Ньютона – Рафсона для систем уравнений.

Программная реализация метода Ньютона – Рафсона возможна по следующей схеме.

1. Задаем относительную погрешность – E , число уравнений – N , максимальное число итераций – M и вектор начальных приближений – X (с компонентами x_i , где i принимает значения от 1 до N).

2. Формируем матрицу Якоби – A . Поскольку аналитическое дифференцирование $F(X)$ нежелательно, частные производные $\partial f/\partial x$ заменяем их приближенными конечно - разностными значениями

$$a_{ji} = \partial f_j / \partial x_i = [f_j(x_i+h_i) - f_j(x_i)]/h_i,$$

где i и j принимают значения от 1 до N ; $h_i = E|x_i|$ – малое приращение x_i .

3. Составляем и решаем методом Гаусса систему линейных уравнений для малых приращений dx_i .

4. Вычисляем уточненное значение x_i .

5. Для всех i проверяем условие $|dx_i/x_i| > E$. Если оно выполняется, идем к пункту 2, т. е. выполняем новую итерацию. Иначе считаем вектор X найденным решением.

Главный недостаток метода Ньютона – Рафсона заключается в том, что сходимость к точному решению имеется не всегда, причем заранее предсказать наличие или отсутствие сходимости не удастся, т. к. условия сходимости метода выражаются довольно сложно и для их использования требуется чрезмерно большой объем вычислений. Стремление повысить надежность метода Ньютона – Рафсона привело к появлению ряда его модификаций. Повышение надежности достигается за счет роста объема вычислений по сравнению с основным методом.

В приведенной ниже программе использован метод изменения параметров (движущейся области сходимости), имеющий простую физическую интерпретацию. Пусть решаемая система уравнений отражает состояние физической системы при наличии определенных внешних воздействий. В большинстве случаев выполняется условие $X = 0$, если все внешние воздействия также нулевые, причем с ростом внешних воздействий, как правило, наблюдается рост и большинства компонент вектора X . Поэтому если выбрать в качестве исходного приближения вектор $X \rightarrow 0$, то вероятность сходимости будет тем выше, чем меньше уровень внешних воздействий. Это обстоятельство и используется в алгоритме, в котором внешнее воздействие увеличивается сравнительно небольшими скачками, после каждого скачка выполняются ньютоновские итерации и получается некоторое промежуточное решение X , принимаемое за исходную точку для итераций при следующем уровне внешних воздействий. Здесь решение системы уравнений производится столько раз, сколько было скачков при изменении внешних воздействий от нулевого уровня до уровня, заданного по условиям задачи.

```

PROGRAM MSRTS;           {Моделирование статических режимов}
{$N+}{$E+}
const E=1E-6; N=6; Kmax=10;           {N-число уравнений}
      Ut=0.026; An=0.993; Ak=0.398;     {Параметры КТ316D}
      Inn=12.8E-12; Ikn=15.5E-12; Rb=70.6;
      R1=30000; R2=91000; R3=820; R5=1001; Vo=12;
type vec=array [1..N] of real; mat=array [1..N,1..N+1] of real;
var Vtek,dV:real; I,K:byte; V,F:vec; A:mat;
PROCEDURE FuncNR(N:byte; V:vec;var F:vec); {Вычисление F(V)}

```

```
var Jk,Jn:double;
```

```
begin
```

```
  Jk:=Ikn*(Exp((V[6]-V[3])/Ut)-1);
```

```
  Jn:=Inn*(Exp((V[6]-V[5])/Ut)-1);
```

```
  F[1]:=V[1]/R1+(V[1]-V[2])/R2+(V[1]-V[6])/Rb;
```

```
  F[2]:=V[2]/R2+(V[2]-V[3])/R3-V[2]+V[4];
```

```
  F[3]:=V[3]/R3-Jk+An*Jn;
```

```
  F[4]:=V[2]-Vtek;
```

```
  F[5]:=V[5]/R5-Jn+Ak*Jk;
```

```
  F[6]:=V[6]/Rb+Jk*(1-Ak)+Jn*(1-An);
```

```
end;
```

```
PROCEDURE Gauss(A: mat; var X: vec);           {Метод Гаусса}
```

```
var I,J,K:byte; R,S:real;
```

```
begin
```

```
  for K:=1 to N-1 do begin
```

```
    S:=A[K,K]; J:=K;                               {Выбор главного элемента}
```

```
    for I:=K+1 to N do begin R:=A[I,K];
```

```
      if Abs(R)>Abs(S) then begin S:=R; J:=I end end;
```

```
    if J<>K then for I:=K to N+1 do           {Перестановка строк}
```

```
      begin R:=A[K,I]; A[K,I]:=A[J,I]; A[J,I]:=R end;
```

```
    for I:=K+1 to N do begin R:=A[I,K]/S;       {Прямой ход}
```

```
      for J:=K+1 to N+1 do A[I,J]:=A[I,J]-A[K,J]*R end
```

```
  end;
```

```
  if S<>0 then for I:=N downto 1 do begin      {Обратный ход}
```

```
    S:=A[I,N+1]; for J:=I+1 to N do S:=S-A[I,J]*X[J];
```

```
    X[I]:=S/A[I,I] end
```

```
end;
```

```
PROCEDURE NR(Vtek:real; var V:vec; var K:byte); {Метод Ньютона}
```

```
var I,J:byte; Vsw,dV:real; Flag:boolean;
```

```
begin K:=0;
```

```
  repeat
```

```
    Inc(K); if K>Kmax then Exit;
```

```

FuncNR(N,V,F);
for I:=1 to N do A[I,N+1]:=-F[I];
for J:=1 to N do begin                                {Расчет элементов матрицы Якоби}
  Vsw:=V[J];
  if Vsw<>0.0 then dV:=E*Abs(Vsw) else dV:=E;
  V[J]:=Vsw+dV; FuncNR(N,V,F);
  for I:=1 to N do A[I,J]:=(F[I]+A[I,N+1])/dV;
  V[J]:=Vsw
end;
Gauss(A,F); Flag:=true;
for I:=1 to N do begin
  V[I]:=V[I]+F[I];
  if Abs(F[I]/V[I])>E then Flag:=false
end
until Flag
end;
BEGIN                                                {Основная программа}
  Write('dV='); Readln(dV);      {Шаг по напряжению питания}
  for I:=1 to N do begin
    Write('V(',I:1,')='); Readln(V[I])  {Нулевое приближение}
  end;
  Vtek:=dV;                                {Текущее напряжение питания}
  Write('K:');
  repeat
    NR(Vtek,V,K); Write(K:2);
    if K>Kmax then begin Writeln; Exit end;
    Vtek:=Vtek+dV;
  until Vtek>(Vo-0.5*dV);
  NR(Vo,V,K); Writeln(K:2); if K>Kmax then Exit;
  for I:=1 to N do Writeln('V(',I:1,')=',V[I]:6:3)
END.

```

4.5 Задание

4.5.1 Ознакомиться с подразделами 4.1 – 4.4 и ответить на контрольные вопросы.

4.5.2 Составить и закодировать эквивалентную схему модели транзисторного усилителя. Принципиальная электрическая схема для анализа выдается преподавателем.

4.5.3 Ознакомиться с программой анализа электронных схем на постоянном токе MSRTS. Составить графическую схему алгоритма анализа.

4.5.4 Сформировать узловые уравнения и рассчитать рабочую точку транзистора.

4.5.5 Проанализировать результаты и оформить отчет.

4.6 Контрольные вопросы

4.6.1 Какова цель работы?

4.6.2 Что представляют собой статические модели биполярного транзистора? Каков смысл их параметров?

4.6.3 В чем суть метода узловых потенциалов?

4.6.4 Как формируются узловые уравнения?

4.6.5 Как источник напряжения заменить эквивалентным источником тока?

4.6.7 В чем заключается метод Ньютона – Рафсона для системы уравнений?

4.6.6 Чем примечателен метод движущейся области сходимости?

5 МОДЕЛИРОВАНИЕ В ЧАСТОТНОЙ ОБЛАСТИ

5.1 Цель работы

Изучение методики формирования модели и алгоритмов анализа электронных схем в частотной области.

5.2 Формирование модели

Комплексная частотная характеристика определяется как отношение

$$F(j\omega) = U_{\text{АУО}}(j\omega) / U_{\text{АО}}(j\omega), \quad (5.1)$$

где $U_{\text{ВХ}}$, $U_{\text{ВЫХ}}$ – входное и выходное напряжения; j – мнимая единица; ω – угловая частота.

Для схемы, содержащей 10 – 20 компонентов, комплексную частотную характеристику можно описать аналитически. В моделировании больших схем широко используется численный подход, когда $F(j\omega)$ вычисляется по точкам (дискретно–частотный метод).

В качестве входного сигнала используется источник напряжения $E_{\text{ВХ}}$ с единичной комплексной амплитудой, нулевой начальной фазой и малым внутренним сопротивлением. При указанном входном сигнале будет верно соотношение

$$F(j\omega) = U_{\text{АУО}}(j\omega) / E_{\text{АО}}(j\omega) = U_{\text{АУО}}(j\omega). \quad (5.2)$$

По известному на некоторой частоте w_i комплексному значению выходного напряжения $U_{\text{АВВ}}(jw_i) = A(w_i) + jB(w_i)$ можно вычислить точку амплитудно–частотной (АЧХ) – $K(w_i)$ и фазо–частотной (ФЧХ) – $\Phi(w_i)$ характеристик электронной схемы:

$$K(W_i) = [A^2(W_i) + B^2(W_i)]^{\frac{1}{2}}, \quad (5.3, \text{ а})$$

$$\Phi(W_i) = \text{arctg}[B(W_i) / A(W_i)]. \quad (5.3, \text{ б})$$

На начальном этапе моделирования электронной схемы в частотной области формируется эквивалентная схема для малого переменного сигнала. При этом на принципиальной схеме **постоянные источники напряжения замыкаются накоротко, а постоянные источники тока размыкаются**. Единичный источник переменного напряжения на входе схемы преобразуется в единичный источник тока $I_{\text{ВХ}}$.

Базис метода узловых потенциалов составляют потенциалы всех узлов эквивалентной схемы, кроме опорного узла, потенциал которого принят равным нулю. Основан метод узловых потенциалов на первом законе Кирхгофа, который утверждает, что для любого узла схемы алгебраическая сумма токов, вытекающих через пассивные компоненты и втекающих через независимые источники тока, равна нулю.

Для конкретной частоты w входного сигнала узловые уравнения могут быть записаны в виде системы линейных алгебраических уравнений

$$Y(jw) V(jw) = I(jw), \quad (5.4)$$

где $Y(jw)$ – матрица узловых проводимостей; $V(jw)$ – искомый вектор узловых потенциалов; $I(jw)$ – вектор узловых токов.

Первый закон Кирхгофа непосредственно показывает, как формируется вектор узловых токов при составлении уравнений: произвольный элемент вектора i_j равен сумме токов независимых источников, втекающих в j -й узел. Для модели схемы в частотной области составляющая вектора I , соответствующая входной ветви, будет равна $I_{\text{ВХ}}$, а остальные – нулю.

Компонентные уравнения пассивных элементов эквивалентной схемы на частоте W имеют следующий вид:

$$\begin{aligned} I_R &= V/R, \\ I_C &= jwCV, \\ I_L &= -jV/wL, \end{aligned} \quad (5.5)$$

где I_R , I_C , I_L – переменные токи через резистор, конденсатор и катушку индуктивности; V – напряжение на элементе; R , C , L – номиналы элементов.

Соответственно проводимости ветвей равны

$$\begin{aligned} y_R &= 1/R, \\ y_C &= jwC, \end{aligned} \quad (5.6)$$

$$y_L = -j/\omega L.$$

В алгоритме формирования матрицы узловых проводимостей Y последовательно просматриваются все ветви эквивалентной схемы. Пусть очередной ветвью является ветвь с номером k , включенная между узлами с номерами i и j . Проводимость этой ветви $y_k = dI_k/dV_k$, где I_k и V_k соответственно ток и напряжение ветви. Т. к. $V_k = V_i - V_j$, где V_i и V_j – потенциалы узлов i и j , то $dI_k/dV_i = y_k$, и $dI_k/dV_j = -y_k$. Поэтому проводимость y_k аддитивно (как слагаемое) включается в диагональные элементы y_{ii} , y_{jj} матрицы узловых проводимостей со знаком плюс и в недиагональные элементы y_{ij} и y_{ji} со знаком минус.

Заметим, что если один из узлов ветви – опорный, то проводимость этой ветви учитывается лишь в одном элементе матрицы Y – в собственной проводимости y_{ii} , где i – номер незаземленного узла ветви.

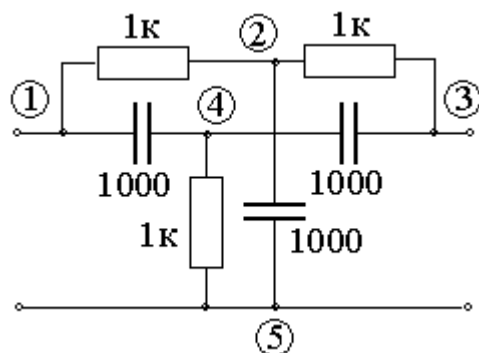


Рисунок 5.4 – Принципиальная электрическая схема двойного Т – образного моста

Для схемы рис. 5.4 матрицу узловых проводимостей при частоте 10^6 рад/с можно представить в виде

$$\begin{vmatrix} 0.001 + j0.001 & -0.001 & 0 & -j0.001 & 0 \\ -0.001 & 0.002 + j0.001 & -0.001 & 0 & -j0.001 \\ 0 & -0.001 & 0.001 + j0.001 & -j0.001 & 0 \\ -j0.001 & 0 & -j0.001 & 0.002 + j0.001 & -0.001 \\ 0 & -j0.001 & 0 & 0 & 0.001 + j0.001 \end{vmatrix}.$$

Такая матрица является вырожденной и называется неопределенной (ее определитель равен нулю). Определенная матрица может быть просто получена из неопределенной путем исключения из нее строки и столбца, соответствующих опорному узлу.

Многие нелинейные элементы РЭС моделируются с помощью управляемых напряжением источников тока. При малом сигнале такие элементы могут быть линеаризованы в окрестности рабочей точки и описаны линейной моделью.

Управляемый напряжением источник тока можно легко включить в уравнения, записанные по методу узловых потенциалов. Обозначим (см. рис. 5.5) его входные клеммы j, j' , а выходные – k, k' . Токи и напряжения элемента связаны соотношениями:

$$I_j = 0;$$

$$I_{j'} = 0;$$

$$I_k = g(V_j - V_{j'});$$

$$I_{k'} = -g(V_j - V_{j'}).$$

Проведя необходимые преобразования, получим матрицу узловых проводимостей для управляемого напряжением источника тока

$$\begin{array}{c|cccc} & j & j' & k & k' \\ \hline j & 0 & 0 & 0 & 0 \\ j' & 0 & 0 & 0 & 0 \\ k & g & -g & 0 & 0 \\ k' & -g & g & 0 & 0 \end{array}.$$

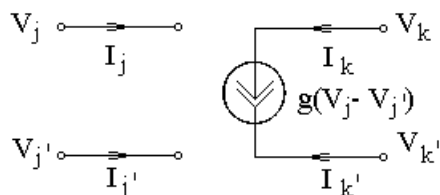


Рисунок 5.5 – Управляемый напряжением источник

5.3 Алгоритм анализа

Решение уравнения (5.4) можно выполнить, используя стандартную программу решения системы линейных уравнений с комплексными коэффициентами. Подставляя в формулу (5.4) разные значения частоты ω_i и вычисляя вектор $v(j\omega_i)$ на каждой частоте, получим комплексную частотную характеристику моделируемого устройства.

В приведенной ниже программе MLS метод исключения Гаусса с частичным (в пределах столбца) выбором главного элемента использован для решения системы уравнений с комплексными коэффициентами. В Паскале нет встроенного комплексного типа, поэтому разработан модуль СХ для работы с комплексными переменными. Комплексная матрица узловых проводимостей Y представлена в виде:

$$Y = Y_C + Y_L + Y_R,$$

поскольку проводимость активных ветвей схемы не зависит от частоты, емкостных – увеличивается с частотой, а индуктивных – уменьшается.

PROGRAM MLS;

uses Сх;


```

const N=4;
type cvec=array [1..N] of cmplx;
    cmat=array [1..N,1..N+1] of cmplx;
    vec=array [1..N] of real; mat=array [1..N,1..N] of real;
var V:cvec; Y:cmat; Kv,Fi:vec; Wn,Wk,W:real; J,K:byte;
PROCEDURE MatY(W:real; var Y:cmat);    {Матрица проводимости}
    const Ire:vec=(100.0,0.0,0.0,0.0); Iim:vec=(0.0,0.0,0.0,0.0);
        Yl:mat=((0.0,0.0,0.0,0.0),(0.0,0.0,0.0,0.0),
                (0.0,0.0,0.0,0.0),(0.0,0.0,0.0,0.0));
        Yr:mat=((100.001,-0.001, 0.0, 0.0 ),
                (-0.001, 0.002,-0.001, 0.0 ),
                ( 0.0, -0.001, 0.002, 0.0 ),
                ( 0.0, 0.0, 0.0, 0.002));
        Yc:mat=(( 0.001, 0.0, 0.0, -0.001),
                ( 0.0, 0.002, 0.0, 0.0 ),
                ( 0.0, 0.0, 0.001,-0.001),
                (-0.001, 0.0, -0.001, 0.002));
var Yim:mat;
begin
    for J:=1 to N do begin
        for K:=1 to N do begin
            Yim[J,K]:=Yc[J,K]*W+Yl[J,K]/W;
            InitC(Y[J,K],Yr[J,K],Yim[J,K]) end;
            InitC(Y[J,N+1],Ire[J],Iim[J]) end;
        end;
    end;
PROCEDURE GaussC(A:cmat; var X:cvec);    {Метод Гаусса}
var I,J,K:byte; Xk,Xi:real; P,R,S:cmplx;
begin
    for K:=1 to N-1 do begin
        AbsC(A[K,K],Xk); J:=K;    {Выбор главного элемента}
        for I:=K+1 to N do begin AbsC(A[I,K],Xi);
            if Xi>Xk then begin Xk:=Xi; J:=I end end;
    end;

```

```

if J<>K then for I:=K to N+1 do                                {Перестановка строк}
  begin R:=A[K,I]; A[K,I]:=A[J,I]; A[J,I]:=R end;  

  S:=A[K,K];                                                    {Прямой ход}
for I:=K+1 to N do begin DivC(A[I,K],S,R);
  for J:=K+1 to N+1 do begin
    MultC(A[K,J],R,P);
    SubtrC(A[I,J],P,A[I,J]) end end
end;  

for I:=N downto 1 do begin S:=A[I,N+1];                        {Обратный ход}
  for J:=I+1 to N do begin
    MultC(A[I,J],X[J],R); SubtrC(S,R,S) end;  

    DivC(S,A[I,I],X[I]) end
end;  

BEGIN  

  Writeln('Начальное и конечное значения частоты (МГц):');  

  Readln(Wn,Wk); W:=Wn;  

repeat  

  MatY(W,Y); GaussC(Y,V);  

  for J:=1 to N do begin  

    AbsC(V[J],Kv[J]);  

    Fi[J]:=Arctan(V[J].Im/V[J].Re)*180.0/3.14 end;  

  Writeln(W:9:2,Kv[1]:9:2,Fi[1]:9:2,Kv[3]:9:3,Fi[3]:9:3);  

  W:=W*1.2  

until W>=Wk  

END.  
  

UNIT Cx;  

interface  

  type cmplx=record Re,Im:real end;  

  procedure InitC(var C:cmplx; R,I:real);  

  procedure AbsC(C:cmplx;var R:real);  

  procedure DivC(C1,C2:cmplx;var R:cmplx);

```

```

procedure MultC(C1,C2:cmplx;var R:cmplx);
procedure AddC(C1,C2:cmplx;var R:cmplx);
procedure SubtrC(C1,C2:cmplx;var R:cmplx);

```

implementation

```

PROCEDURE InitC(var C:cmplx; R,I:real);
  begin C.Re:=R; C.Im:=I end;
PROCEDURE AbsC(C:cmplx;var R:real);
  begin R:=Sqrt(C.Re*C.Re+C.Im*C.Im) end;
PROCEDURE DivC(C1,C2:cmplx;var R:cmplx);
  var T:real;
  begin
    T:=C2.Re*C2.Re+C2.Im*C2.Im;
    R.Re:=(C1.Re*C2.Re+C1.Im*C2.Im)/T;
    R.Im:=(C2.Re*C1.Im-C1.Re*C2.Im)/T
  end;
PROCEDURE MultC(C1,C2:cmplx; var R:cmplx);
  begin
    R.Re:=C1.Re*C2.Re-C1.Im*C2.Im;
    R.Im:=C1.Re*C2.Im+C2.Re*C1.Im
  end;
PROCEDURE AddC(C1,C2:cmplx; var R:cmplx);
  begin R.Re:=C1.Re+C2.Re; R.Im:=C1.Im+C2.Im end;
PROCEDURE SubtrC(C1,C2:cmplx; var R:cmplx);
  begin R.Re:=C1.Re-C2.Re; R.Im:=C1.Im-C2.Im end;
END.

```

5.4 Задание

5.4.1 Ознакомиться с подразделами 5.1 – 5.3 и ответить на контрольные вопросы.

5.4.2 Ознакомиться с программой анализа электронных схем в частотной области. Составить схему алгоритма анализа.

5.4.3 Сформировать эквивалентную схему модели. Принципиальная электрическая схема для анализа выдается преподавателем.

5.4.4 Сформировать матрицу узловых проводимостей и вектор узловых токов.

5.4.5 Рассчитать АЧХ и ФЧХ схемы; частоту изменять по логарифмическому закону.

5.4.6 Проанализировать результаты и оформить отчет.

5.5 Контрольные вопросы

5.4.1 Что называют комплексной частотной характеристикой?

5.4.2 Почему при формировании эквивалентной схемы модели постоянные источники напряжения замыкаются накоротко, а постоянные источники тока размыкаются?

5.4.3 Зачем единичный источник напряжения на входе схемы преобразуется в единичный источник тока?

5.4.4 Как формируются вектор узловых токов и матрица узловых проводимостей?

5.4.5 В чем отличие неопределенной матрицы узловых проводимостей от определенной?

5.4.6 Почему матрица узловых проводимостей представлена как сумма трех матриц – Y_C , Y_L и Y_R ?

6 МЕТОДЫ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ

6.1 Цель работы

Изучение численных методов интегрирования обыкновенных дифференциальных уравнений (ОДУ) и алгоритма формирования моделей переходных процессов методом переменных состояния.

6.2 Явные и неявные методы

Анализ переходных процессов в электронных схемах сводится к решению ОДУ. Для решения систем ОДУ на ЭВМ применяются явные и неявные методы численного интегрирования. В обоих случаях решение системы

$$V' = F(V, t) \quad (6.1)$$

при известных начальных условиях $V(0) = V_0$ заключается в последовательном вычислении значений вектора базисных координат $V(k)$ на конечном множестве k точек временной оси

$$t_k = \sum_{i=1}^k h_i,$$

где $h_i = t_i - t_{i-1}$ – величина i -го шага интегрирования. Результаты решения – зависимость $V(k)$ в табличной форме.

Различия между методами численного интегрирования заключается в способах предсказания $V(k)$ по известному значению $V(k-1)$, а в более общем случае и по другим уже известным значениям V . Большинство этих способов основано на аппроксимации производных конечно – разностными (алгебраическими) выражениями. Вид аппроксимирующего выражения существенно влияет на точность, устойчивость и скорость вычислений.

Для решения систем ОДУ на ЭВМ применяются явные и неявные методы численного интегрирования. Различия между методами численного интегрирования заключается в способах предсказания $V(k)$ по известному значению $V(k-1)$, а в общем случае и по др. уже известным значениям V . Большинство этих способов основано на аппроксимации производных конечноразностными (алгебраическими) выражениями. Вид аппроксимирующего выражения существенно влияет на точность, устойчивость и скорость вычислений.

Представление об **явных** методах численного интегрирования ОДУ можно получить, если воспользоваться разложением элементов вектора $V(t)$ в ряд Тейлора в окрестностях точки $t(k)$:

$$v_j(k+1) = v_j(k) + v_j'(k) \cdot h + v_j''(k) \cdot h^2 / 2! + \dots$$

Ограничим разложение членами до p -й производной включительно и, считая $h = \text{const}$, заменим оставшиеся производные отношениями конечных разностей. С учетом (6.1) находим:

$$V(k+1) = V(k) + h \cdot \sum_{i=1}^p A_i \cdot F[V(k+1-i)]. \quad (6.2)$$

Величину p называют порядком метода интегрирования. Коэффициенты A_i зависят от способа аппроксимации элементов вектора $V(t)$ (применение разложения в ряд Тейлора – не единственно возможный способ).

Формула (6.2) характеризует большую группу явных разностных методов численного интегрирования. Ее характерная особенность – искомое на k -м шаге значение $V(k)$ явно выражено через значения переменных, определенных на предыдущих шагах.

Простейший из явных методов – метод Эйлера ($p = 1$), его формула

$$V(k+1) = V(k) + h F[V(k)] \quad (6.3)$$

получается непосредственно из разложения $V(t)$ в ряд Тейлора при пренебрежении членами со второй и более высокими степенями h .

Из методов с $p > 1$ наиболее известны методы Адамса ($p = 5$) и Рунге – Кутта. Вариант метода Рунге – Кутта с $p = 4$ характеризуется вычислениями по следующим формулам:

$$V(k+1) = V(k) + h (F_1 + 2 F_2 + 2 F_3 + F_4) / 6,$$

где $F_1 = F[V(k), t_k]$; $F_2 = F[V(k) + F_1/2, t_k + h/2]$; $F_3 = F[V(k) + F_2/2, t_k + h/2]$; $F_4 = F[V(k) + F_3, t_k + h]$.

Явные формулы численного интегрирования удобно применять в тех случаях, когда модель схемы удастся представить в нормальной форме Коши (6.1) с известными начальными условиями V_0 .

Обобщение группы **неявных** разностных формул интегрирования, полученное аналогично (6.2), имеет вид

$$V(k+1) = V(k) + h \sum_{i=1}^p B_i F[V(k+2-i)]. \quad (6.4)$$

В этой формуле, в отличие от (6.3), вектор базисных координат для новой точки $V(k+1)$ не выражен явно через уже известные величины. Действительно, при $i = 1$ в правой части (6.4) фигурирует $V(k+1)$. Следовательно, (6.4) должна рассматриваться как запись системы нелинейных алгебраических и трансцендентных уравнений с неизвестным вектором $V(k+1)$.

В методе первого порядка – неявном методе Эйлера – (6.4) имеет вид

$$V(k+1) = V(k) + h F[V(k+1), t(k+1)]. \quad (6.5)$$

Неявная формула второго порядка – формула Шихмана – получается из (6.4) при $B_1=2/3$ и $B_2=1/3$:

$$V(k+1) = -1/3 V(k-1) + 4/3 V(k) + 2/3 h F[V(k+1), t(k+1)].$$

Важным свойством неявных методов является удобство их применения к решению систем алгебродифференциальных уравнений вида

$$Q(V', V, t) = 0. \quad (6.6)$$

Эта возможность реализуется путем подстановки в (6.6) вместо вектора производных V' выражения, вытекающего из формулы численного интегрирования (6.4). В самом деле, (6.4) можно записать в явном виде относительно $F[V(k+1)]$:

$$F[V(k+1)] = V'(k+1) = 1/B_1 \{ [V(k+1) - V(k)]/h - \sum_{i=2}^p B_i F[V(k+2-i)] \}. \quad (6.7)$$

В правой части (6.7) неизвестен только вектор $V(k+1)$, все остальные величины определены при выполнении предыдущих шагов. Поэтому после подстановки $V'(k+1)$ в (6.6) получаем систему алгебраических и трансцендентных уравнений

$$Q[V'(k+1), V(k+1), t_{k+1}] = 0, \quad (6.8)$$

подлежащую решению на каждом шаге интегрирования.

Чаще всего решение систем алгебраических и трансцендентных уравнений в процессе неявного интегрирования выполняется методом Ньютона – Рафсона.

6.3 Точность и устойчивость методов интегрирования

Полная ошибка интегрирования ОДУ на k -м шаге зависит от следующих составляющих:

- 1) ошибки аппроксимации $E_A(k)$, связанной с заменой производных конечноразностными выражениями;
- 2) ошибки вычислений $E_B(k)$, обусловленной погрешностями выполнения арифметических действий;

3) ошибки накопления $E_H(k)$, равной полной ошибке на предыдущем шаге.

Метод численного интегрирования называется устойчивым, если накопленная погрешность $E_H(k)$ остается ограниченной при k , стремящемся к бесконечности.

Устойчивость методов обычно исследуют на модельной задаче:

$$V' = A V, \quad (6.9)$$

где A – постоянная действительная матрица (N, N); шаг интегрирования h полагается постоянным.

Если метод неустойчив для системы (6.9), то он, как правило, будет неустойчив и для нелинейной системы (6.1). В последнем случае матрице A будет соответствовать матрица Якоби в точке t_k , т. к. в пределах шага интегрирования $h(k)$ матрицу Якоби можно считать постоянной.

Устойчивость метода численного интегрирования зависит не только от h , но и от собственных значений матрицы A (или матрицы Якоби). Собственными значениями матрицы A называют корни характеристического уравнения

$$\det(A - L \cdot B) = 0,$$

где B – единичная матрица, \det – определитель. Спектр собственных значений – L_i , $i = 1, 2, \dots, n$. В общем случае L_i – комплексные величины, т. е. $L_i = \text{Re}(L_i) + j \text{Im}(L_i)$, где j – мнимая единица. Отношение $g = \max|L_i| / \min|L_i|$ называют числом обусловленности матрицы A .

Аналитическое решение системы (6.9) можно представить в виде суммы отдельных фундаментальных решений, т. е.

$$V(t) = \sum_{i=1}^N C_i \cdot X_i \cdot \exp(L_i \cdot t), \quad (6.10)$$

где C_i – постоянные; X_i – собственный вектор матрицы A , соответствующий собственному значению L_i .

Каждое фундаментальное решение характеризует отклик системы на внешнее воздействие.

Если $\text{Re}(L_i) < 0$ для всех i , то все фундаментальные решения будут иметь затухающий характер, а система ОДУ будет устойчивой. Если хотя бы одно $\text{Re}(L_i) > 0$ для некоторого i , то i -я компонента для каждой переменной в (6.10) неограниченно возрастает, и система ОДУ будет неустойчивой.

Если $\text{Im}(L_i) = 0$ для всех i , то переходные процессы носят аperiodический характер. Если $\text{Im}(L_i)$ отлична от нуля для некоторых i , то каждая переменная $v_i(t)$ в (6.10) содержит осциллирующую синусоидальную компоненту

$$C_i \sin(\text{Im}(L_i) t + \varphi_i) \exp(\text{Re}(L_i) t),$$

где C_i, φ_i – постоянные.

Таким образом, собственные значения матрицы A (или матрицы Якоби) характеризуют переходные процессы в моделируемом объекте: устойчивость, наличие осцилляций и т. п. В то же время они определяют и математические свойства самой модели. В частности, система ОДУ будет плохо обусловленной (или жесткой), если число обусловленности $g \gg 1$.

Для большинства задач анализа наиболее важен случай, когда $\text{Re}(L_i) < 0$ для всех i , т. е. все фундаментальные решения затухают при увеличении t со скоростью, пропорциональной $T_i = -1/\text{Re}(L_i)$, называемой постоянной времени системы.

Явный метод Эйлера устойчив при $h < 2 \min|T_i|$, а неявный – при любых значениях шага h .

6.4 Метод переменных состояния

Исходная математическая модель схемы для анализа переходных процессов может быть представлена в явной (6.1) или неявной (6.6) форме.

Методика формирования модели в явной форме (нормальной форме Коши) включает два этапа:

1) на основе законов Кирхгофа емкостные токи J_C и индуктивные напряжения U_L выражаются через токи и напряжения остальных ветвей;

2) J_C и U_L заменяются производными в соответствии с компонентными уравнениями, а остальные переменные выражаются через J_C и U_L .

Проиллюстрируем эту методику простым примером формирования модели для схемы рис. 6.1.

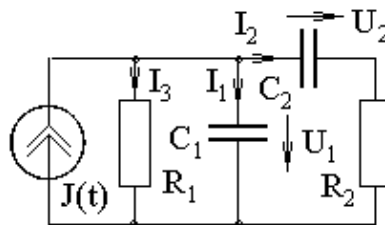


Рисунок 6.1 – Электрическая цепь

Система топологических уравнений имеет вид

$$I_1 = J(t) - I_2 - I_3,$$

$$I_2 R_2 = (U_1 - U_2),$$

$$I_3 = U_1/R_1.$$

Используя компонентные уравнения $I_1 = C_1 dU_1/dt$, $I_2 = C_2 dU_2/dt$, получаем нормальную систему ОДУ:

$$dU_1/dt = [J(t) - (U_1 - U_2)/R_2 - U_1/R_1]/C_1,$$

$$dU_2/dt = (U_1 - U_2)/(R_2 C_2). \quad (6.11)$$

Для номиналов элементов схемы рис. 6.1: $R_1 = 2/3$ МОм; $R_2 = 1$ МОм; $C_1 = 3$ мкФ; $C_2 = 3$ мкФ; $J(t) = 0$, при $t < 0$, $J(t) = 6$ мкА, при $t \geq 0$

$$\begin{aligned} U_1' &= -5 \cdot U_1 / 6 + U_2 / 3 + 2.5, \\ U_2' &= U_1 / 3 - U_2 / 3, \end{aligned} \quad (6.12)$$

при начальных условиях $U_1(0) = U_2(0) = 0$.

Матрица Якоби для системы уравнений (6.8) имеет собственные значения $L_1 = -1$, $L_2 = -1/6$, которым соответствуют постоянные времени $T_1 = 1$ и $T_2 = 6$.

Ниже приведена программа численного интегрирования системы (6.11) явным и неявным методами Эйлера. Для оценки точности в программе использовано аналитическое решение (6.12).

```

PROGRAM MPP8;                                {Методы Эйлера}
uses Crt,Graph;
const N=2;                                    {N-число уравнений}
        Tk=18.0;                               {(Tk-0.0)-временной диапазон}
type vec=array [1..N] of real;
var Nt:integer; W:byte; Van:vec;
        Vmax,Size:word; P:pointer;
PROCEDURE Rp(V:vec; var P:vec); {Расчет производных dV/dT=P}
begin
        P[1]:=2.5-5*V[1]/6+V[2]/3;
        P[2]:=V[1]/3-V[2]/3;
end;
PROCEDURE An(T:real; var V:vec); {Аналитическое решение}
begin
        V[1]:=5.0-2.0*Exp(-T)-3.0*Exp(-T/6.0);
        V[2]:=5.0+Exp(-T)-6.0*Exp(-T/6.0)
end;
PROCEDURE GrAn; {Вывод аналитического решения и сетки}
const Nx=5; Ny=5;
        dX=250 div Nx; dY=200 div Ny; {Шаг сетки по X и Y}
var Gd,Gm:integer; Sy,Sx:string[4];
        I,J:byte; Vm,T,dT,dV:real;
begin
        Gd:=Detect; InitGraph(Gd,Gm,'c:\tp\bgi'); ClearDevice;
        SetBkColor(0); SetViewPort(0,0,319,239,True); SetColor(5);

```

```

Line(40,25,40,225); Line(40,225,290,225);    {Оси Y, X}
An(Tk,Van); Vm:=Van[1];
for I:=2 to N do if Vm<Van[I] then Vm:=Van[I];
Vmax:=Round(Vm+0.5);
dT:=Tk/250; dV:=Vmax/200;    {Вывод аналитического решения}
for J:=1 to 250 do begin T:=J*dT; An(T,Van);
  for I:=1 to N do begin
    if Odd(I) then SetColor(2) else SetColor(3);
    Line(J+40,224,J+40,Round(225-Van[I]/dV))
  end
end;
dV:=Vmax/Ny; SetColor(1);
for I:=0 to Ny do begin    {Горизонтальные линии сетки}
  if I<>0 then Line(40,225-I*dY,290,225-I*dY);
  Str(I*dV:4:1,Sy);
  OutTextXY(3,(225-I*dY),Sy)    {Подписи по Y}
end;
dT:=Tk/Nx;
for I:=1 to Nx do begin    {Вертикальные линии сетки}
  Line(40+I*dX,25,40+I*dX,225);
  Str(I*dT:4:1,Sx);
  OutTextXY(15+I*dX,230,Sx)    {Подписи по X}
end;
Size:=ImageSize(10,25,310,239); GetMem(P,Size);
GetImage(10,25,310,239,P^);    {Область экрана - в буфер}
end;
PROCEDURE Eul(H:real; var V:vec);    {Явный метод Эйлера}
var P:vec; I:byte;
begin
  Rp(V,P); for I:=1 to N do V[I]:=V[I]+H*P[I]
end;
PROCEDURE EulNe(H:real; var Vh:vec);    {Неявный метод Эйлера}

```

PROCEDURE NR(V:vec; var Vh:vec); {Метод Ньютона-Рафсона}

const Eps=1E-6;

type mat=array [1..N,1..N+1] of real;

var I,J,R,Iter:byte; Vsw,dV:real; F:vec; A:mat;

PROCEDURE FuncNR(V,Vh:vec;var F:vec); {Вычисление F(V)}

var I:byte; P:vec;

begin

Rp(Vh,P); for I:=1 to N do F[I]:=Vh[I]-V[I]-H*P[I]

end;

PROCEDURE Gauss(var A:mat; var X:vec); {Метод Гаусса}

var I,J,K:byte; R,S:real;

begin

for K:=1 to N-1 do **begin**

S:=A[K,K]; J:=K; {Выбор главного элемента}

for I:=K+1 to N do **begin** R:=A[I,K];

if Abs(R)>Abs(S) then **begin** S:=R; J:=I **end end;**

if J<>K then for I:=K to N+1 do {Перестановка строк}

begin R:=A[K,I]; A[K,I]:=A[J,I]; A[J,I]:=R **end;**

for I:=K+1 to N do **begin** R:=A[I,K]/S; {Прямой ход}

for J:=K to N+1 do A[I,J]:=A[I,J]-A[K,J]*R **end**

end;

if S<>0 then for I:=N downto 1 do **begin** {Обратный ход}

S:=A[I,N+1]; for J:=I+1 to N do S:=S-A[I,J]*X[J];

X[I]:=S/A[I,I] **end**

end;

begin Iter:=0;

repeat Inc(Iter);

FuncNR(V,Vh,F);

for I:=1 to N do A[I,N+1]:=-F[I];

for J:=1 to N do **begin**

Vsw:=Vh[J];

if Vsw<>0.0 then dV:=Eps*Abs(Vsw) else dV:=Eps;

```

    Vh[J]:=Vsw+dV; FuncNR(V,Vh,F);
    for I:=1 to N do A[I,J]:=(F[I]+A[I,N+1])/dV;
    Vh[J]:=Vsw
end;
Gauss(A,F); R:=0;
for I:=1 to N do begin
    Vh[I]:=Vh[I]+F[I];
    if Abs(F[I]/Vh[I])>Eps then R:=1
end
until R<>1;
end;
begin
    NR(Vh,Vh)
end;
PROCEDURE GrEul;          {Вывод результатов и погрешности}
var Sh,Seul:String[12]; V:vec;      {численного решения}
    X1,X2:integer; I,J:word; H,dX,dY:real;
    Y1,Y2:array [1..N] of integer;
begin
    for I:=1 to N do begin V[I]:=0.0; Y1[I]:=225 end;
    X1:=40; dX:=250/Nt; dY:=200/Vmax;
    H:=Tk/Nt; Str(H:5:3,Sh); Sh:=' H'+Sh;
    for J:=1 to Nt do begin
        if Odd(W) then begin SetColor(14); Eul(H,V);
            OutTextXY(20,10,Sh) end
            else begin SetColor(15); EulNe(H,V) end;
        X2:=Round(J*dX+40);
        for I:=1 to N do begin Y2[I]:=Round(225-V[I]*dY);
            Line(X1,Y1[I],X2,Y2[I]); Y1[I]:=Y2[I];
        end; X1:=X2;
    end;
end;
for I:=1 to N do begin {Расчет относительной погрешности}

```

```

    Str(100*(V[I]-Van[I])/Van[I]:7:3,Seul); Seul:=Seul+'%';
    if Odd(W) then OutTextXY(150,150+10*I,Seul)
        else OutTextXY(150,180+10*I,Seul)
    end
end;

BEGIN                                { Основная программа }
GrAn;
repeat SetViewPort(0,0,319,239,True);
SetColor(4); OutTextXY(133,149,'Введите: Nt=');
Window(30,10,35,10); read(Nt);
if Nt>0 then begin W:=0; PutImage(10,25,P^,0) end;
while (Nt>0) and (W<8) do begin Inc(W);
    case W of
        3: SetViewPort(320,0,639,239,True);
        5: SetViewPort(0,240,319,479,True);
        7: SetViewPort(320,240,639,479,True);
    end;
    if Odd(W) then begin ClearViewPort;
        PutImage(10,25,P^,0); GrEul end
        else begin GrEul; Inc(Nt,2) end
    end
until Nt<=0;
FreeMem(P,Size); CloseGraph
END.

```

6.5 Задание

6.5.1 Ознакомиться с разделами 6.1 – 6.4 и ответить на контрольные вопросы.

6.5.2 Ознакомиться с программой анализа переходных процессов. Составить схему алгоритма явного и неявного методов Эйлера.

6.5.3 Сформировать эквивалентную схему и уравнения модели переходного процесса. Принципиальная электрическая схема для анализа выдается преподавателем.

6.5.4 Исследовать точность и устойчивость явного и неявного методов Эйлера в зависимости от величины шага интегрирования h .

6.5.5 Проанализировать результаты и оформить отчет.

6.6 Контрольные вопросы

6.6.1 Какова цель работы?

6.6.2 В чем суть методов численного интегрирования?

6.6.3 Что отличает явные и неявные методы интегрирования?

6.6.4 Чем определяется точность численного интегрирования?

6.6.5 Что отличает устойчивости системы ОДУ и метода численного интегрирования ОДУ?

6.6.6 Как найти собственные значения матрицы Якоби?

6.6.7 Что называют постоянными времени системы?

6.6.8 Какие системы называют жесткими?

7 КЛЮЧ НА ПОЛЕВОМ ТРАНЗИСТОРЕ

7.1 Цель работы

Изучение алгоритма автоматического выбора шага численного интегрирования и модели ключа на полевом транзисторе.

7.1 Метод трех зон

Анализ переходных процессов сводится к решению обыкновенных дифференциальных уравнений (ОДУ). Шаг численного интегрирования ОДУ должен быть переменным и выбираться в процессе анализа автоматически. Автоматический выбор шага производится с целью интегрирования с величинами шагов, максимально возможными при условии сохранения требуемой точности. При применении неустойчивых методов автоматический выбор шага должен гарантировать также выполнение условий устойчивости.

Как правило, автоматический выбор шага основан на сопоставлении заданной $E_{\text{зад}}$ и получающейся E погрешностей. Способы оценки общей погрешности интегрирования, накопленной за все предыдущие шаги, отсутствуют. Имеется возможность контролировать лишь локальную погрешность, т. е. погрешность E , допущенную на одном шаге. В связи с этим возникают определенные трудности в назначении величин $E_{\text{зад}}$, т. к. эти величины непосредственно не характеризуют погрешность численного интегрирования. Однако, приобретя некоторый опыт применения программ анализа переходных процессов, инженер может для класса интересующих его задач установить соответствие между $E_{\text{зад}}$ и получающейся погрешностью численного решения.

При разработке алгоритмов автоматического выбора шага приходится выбирать способ оценки E и способ изменения шага по результатам сравнения E с $E_{\text{зад}}$. Для оценки E наиболее часто используют приближенное вычисление погрешностей усечения или сопоставление результатов прогноза и коррекции. Одним из наиболее популярных способов выбора величины шага h является способ трех зон, в котором в качестве исходных данных задаются два критических значения допустимых погрешностей: $E_{\text{зад}1}$ – граница между зонами недопустимо большой погрешности и оптимальной, в

которой при приемлемой погрешности обеспечиваются большие величины шагов; $E_{зад2}$ – граница между оптимальной зоной и зоной, где малая погрешность достигается за счет неоправданно малых шагов. Алгоритм трех зон можно выразить следующим образом:

$$h_{k+1} = \begin{cases} m_1 h_k, & \text{если } E > E_{зад1}, \\ h_k, & \text{если } E_{зад1} > E > E_{зад2}, \\ m_2 h_k, & \text{если } E < E_{зад2}, \end{cases}$$

где $m_1 < 1$, $m_2 > 1$.

Константы $E_{зад1}$, $E_{зад2}$, m_1 , m_2 обычно выбираются на основе накопленного опыта; часто $E_{зад1}/E_{зад2} = 8$; $m_1 = 0,5$; $m_2 = 2$.

7.2 Модель ключа на полевом транзисторе

Принципиальная схема ключа приведена на рис. 7.1, а, эквивалентная схема – на рис. 7.2.

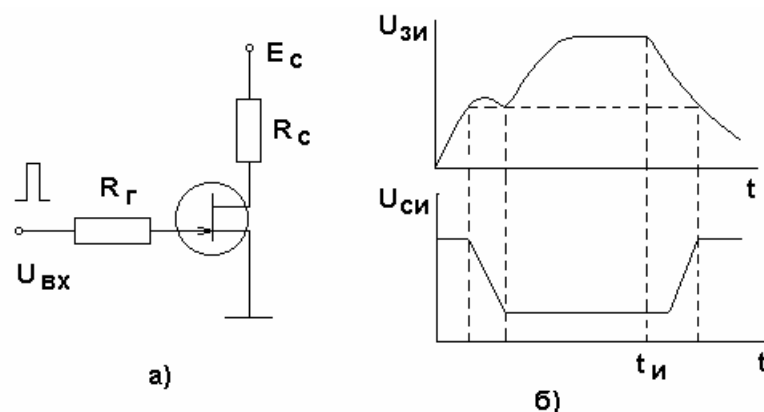


Рисунок 7.1 – Ключ на мощном МДП транзисторе (а) и временные диаграммы его работы (б)

Приняты обозначения: C_{11} – входная, C_{22} – выходная и C_{12} – проходная емкости, $I_c(U_{зи}, U_{си})$ – источник тока, описываемый семейством выходных вольт – амперных характеристик транзистора:

$$I_c(U_{зи}, U_{си}) = M(U_{си}) \cdot \{1 - \exp[-k \cdot U_{зи} \cdot S / M(U_{си})]\},$$

$$M(U_{си}) = I_s \cdot \{1 + \text{th}[S \cdot (U_{си} - U_s) / I_s]\},$$

где U_s и I_s – рабочая точка передаточной характеристики; k – коэффициент аппроксимации; $S = dI_N/dU_{се}$ – малосигнальная крутизна.

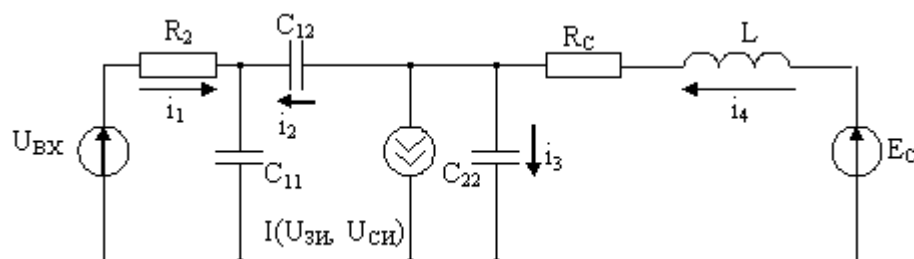


Рисунок 7.2 – Эквивалентная схема ключа на МДП транзисторе

Система дифференциальных уравнений для эквивалентной схемы рис. 7.2 имеет вид:

$$\begin{aligned}dU_{\text{CЭ}}/dt &= (i_1 + i_2)/C_{11}, \\dU_{\text{НЭ}}/dt &= i_3/C_{22}, \\di_4/dt &= (E_c - i_4 R_c - U_{\text{CЭ}})/L,,\end{aligned}$$

где

$$\begin{aligned}i_1 &= [U_{\text{АО}}(t) - U_{\text{CЭ}}]/R_a, i_2 = \{[i_4 - I_c(U_{3Э}, U_{\text{НЭ}})]/C_{22} - i_1/C_{11}\}/(1/C_{11} + 1/C_{22} + 1/C_{12}), \\i_3 &= i_4 - I_{\text{Н}}(U_{\text{CЭ}}, U_{\text{НЭ}}) - i_2.\end{aligned}$$

C_{11} слабо зависит от напряжения $U_{3И}$, поэтому можно положить $C_{11} = \text{const}$. Однако C_{12} и C_{22} сильно зависят от $U_{СИ}$ и аппроксимируются выражениями:

$$\begin{aligned}\tilde{N}_{12} &= C_{12\text{min}} + A \exp(-B U_{\text{НЭ}}), \\ \tilde{N}_{22} &= C_{22\text{min}} + C \exp(-D U_{\text{НЭ}}),\end{aligned}$$

где величины $C_{12\text{min}}$, A , B , $C_{22\text{min}}$, C , D определяются с помощью экспоненциальной регрессии по экспериментальным зависимостям C_{12} и C_{22} от $U_{СИ}$.

В настоящей работе $U_{\text{ВХ}}(t)$ задается в виде импульса с экспоненциальным нарастанием и спадом

$$\begin{aligned}U_{\text{АО}}(t) &= U_{\text{IАО}} [1 - \exp(-t/t_{\text{АО}})], \text{ при } 0 \leq t \leq t_{И}, \\ U_{\text{ВО}}(t) &= U_{\text{IАО}} [1 - \exp(-t_{\text{Э}}/t_{\text{ВО}})] \exp[(t_{\text{Э}} - t)/t_{\text{АО}}], \text{ при } t > t_{И},\end{aligned}$$

где $U_{\text{МВХ}}$ – амплитуда; $t_{И}$ – длительность; $t_{\text{ВХ}}$ – постоянная времени нарастания и спада входного импульса.

В приведенной ниже программе решение системы ОДУ выполняется методом Рунге – Кутта – Мерсона при заданных начальных условиях $U_{3И}(t_0)$, $U_{СИ}(t_0)$, $i_4(t_0)$, t_0 , а также $U_{\text{ВХ}}(t)$.

Мерсон предложил модификацию метода Рунге – Кутта четвертого порядка, позволяющую оценивать погрешность на каждом шаге и принимать решение об изменении шага. Схема Мерсона требует на каждом шаге вычислять правую часть ОДУ в пяти точках:

$$y(x_0 + h) = y_0 + (k_4 + k_5)/2 + O(h^5),$$

где

$$\begin{aligned}k_1 &= h_3 f(x_0, y_0), \quad h_3 = h/3, \\ k_2 &= h_3 f(x_0 + h_3, y_0 + k_1), \\ k_3 &= hf[x_0 + h_3, y_0 + (k_1 + k_2)/2], \\ k_4 &= k_2 + 4h_3 f[x_0 + h/2, y_0 + 0.375(k_1 + k_3)], \\ k_5 &= h_3 f[x_0 + h, y_0 + 1.5(k_4 - k_3)].\end{aligned}$$

За счет только одного дополнительного коэффициента k_1 по сравнению с классической схемой Рунге – Кутта на каждом шаге можно определить погрешность решения R по формуле

$$10R = 2k_4 - 3k_3 - k_5.$$

```

PROGRAM MK9;      { Модель ключа на мощном МДП транзисторе }
{$N+}{$E+}
const N=3;      { Число уравнений }
  Is=11.0; Us=12.6; S=2.72; K=0.7; C11=3.5E-10;
  C12min=8E-12; A=3.5E-11; B=0.088;
  C22min=1.3E-10; C=6.59E-10; D=0.33;      { Параметры КП913А }
  Ec=45.0; Rc=5.0; Rg=10.0; L=1E-8;
  Umw=20.0; Twx=3E-9; Ti=14E-9;
  Hn=1E-9; Eps=1E-3; Tn=0.0; Tk=50E-9;
type vec=array [1..N] of real;
var I:byte; U:vec; Uw,Ep,H,T:real; fd:text; F:string[2];
PROCEDURE Pr(T:real;var U,F:vec);      { Расчет производных }
var C12,C22,M,Ic,I1,I2,I3:real;
FUNCTION Th(X:real):real;      { Тангенс гиперболический }
  begin Th:=(Exp(X)-Exp(-X))/(Exp(X)+Exp(-X)) end;
FUNCTION Uwx(T:real):real;      { Входное напряжение }
  begin
    if T<Ti then Uwx:=Umw*(1-Exp(-T/Twx))
      else Uwx:=Umw*(1-Exp(-Ti/Twx))*Exp((Ti-T)/Twx)
  end;
begin
  C12:=C12min+A*Exp(-B*U[2]); C22:=C22min+C*Exp(-D*U[2]);
  M:=Is*(1+Th(S*(U[1]-Us)/Is)); Ic:=M*(1-Exp(-K*S*U[2]/M));
  Uw:=Uwx(T); I1:=(Uw-U[1])/Rg;
  I2:=((U[3]-Ic)/C22-I1/C11)/(1/C11+1/C22+1/C12);
  I3:=U[3]-Ic-I2;
  F[1]:=(I1+I2)/C11; F[2]:=I3/C22; F[3]:=(Ec-U[3]*Rc-U[2])/L
end;
PROCEDURE RKM(N:byte;E:real;var X,H:real;var Y:vec);

```

```

var I:byte; H3,H4,R,A:real; Z,F0,F,K1,K3:vec;
begin
  Pr(X,Y,F0);
  for I:=1 to N do Z[I]:=Y[I];
  repeat H3:=H/3; H4:=4*H3;
    for I:=1 to N do begin K1[I]:=H3*F0[I];
      Y[I]:=Z[I]+K1[I] end;
    Pr(X+H3,Y,F);
    for I:=1 to N do Y[I]:=Z[I]+(K1[I]+H3*F[I])/2;
    Pr(X+H3,Y,F);
    for I:=1 to N do begin K3[I]:=H*F[I];
      Y[I]:=Z[I]+0.375*(K1[I]+K3[I]) end;
    Pr(X+H/2,Y,F);
    for I:=1 to N do begin K1[I]:=K1[I]+H4*F[I];
      Y[I]:=Z[I]+1.5*(K1[I]-K3[I]) end;
    Pr(X+H,Y,F); R:=0;
    for I:=1 to N do begin A:=H3*F[I];
      Y[I]:=Z[I]+(K1[I]+A)/2; A:=2*K1[I]-3*K3[I]-A;
      if Y[I]<>0.0 then A:=A/Y[I];
      if Abs(A)>R then R:=Abs(A) end;
    H:=H/2
  until R<E;
  H:=2*H; X:=X+H; if 32*R<E then H:=2*H
end;
BEGIN {Основная программа}
  Write('Номер файла данных: '); Readln(F);
  Assign(Fd,'fd'+F); Rewrite(Fd);
  H:=Hn; T:=Tn; U[1]:=0; U[2]:=Ec; U[3]:=0;
  Writeln(Fd,' T,nc 50*H,nc Uвх,B U1,B U2,B U3,B');
  Write('Ждите... ');
  repeat
    RKM(N,Eps,T,H,U);

```

```

Write(Fd,1E9*T:7:2,50E9*H:7:3,Uw:6:2);
for I:=1 to N do Write(Fd,U[I]:8:3); Writeln(Fd);
until T>=Tk;
Close(Fd); Writeln('Результаты - в файле ',fd'+F)
END.

```

```

program PGr1;
uses Crt,Graph,Gr;
const M=3;           { Число файлов fd1..fdM}
var F:text; C:string[2]; S:char; Flag:boolean;
    I,J,K,Gd,Gm,Conter:integer;
begin
  Write('Начальный участок? '); Readln(S);
  if S in ['Y','y','Д','д'] then begin
    Write('Сколько точек? '); Readln(Conter) end;
  Clrscr; Gd:=0; {Gm:=VgaHi;} InitGraph(Gd,Gm,'C:\Tp\Bgi');
  SetBkColor(0); SetColor(11);
  for K:=M downto 1 do begin
    Str(K:1,C);
    Assign(F,'fd'+C); Reset(F);
    Readln(F); I:=0; Flag:=true;
    while not Eof(F) and Flag do begin
      if S in ['Y','y','Д','д'] then Flag:=I<=Conter;
      Read(F,X[I]);
      for J:=1 to 5 do Read(F,Y[J,I]);
      Readln(F); Inc(I) end;
    Conter:=I-1; Close(F);
    if K=M then begin G1(Conter); G2(Conter,K) end
      else G2(Conter,K) end;
    Readln; CloseGraph
  end.

```

```

unit Gr;
interface
  uses Graph;
  const Xl=99; Xr=500; Yu=39; Yd=340; Xlr=Xr-Xl; Yud=Yd-Yu;
    Ny=5; Nx=5; dT=1; Max=50.0; Min=0.0;
  var X:array [0..Xlr] of real; Dx,Dy:real;
    Y:array [1..5,0..Xlr] of real;
  procedure G1(Cnt:integer); procedure G2(Cnt,K:integer);
implementation
  var I,J,tX,tY:integer;
  procedure G1(Cnt:integer);
    var St:string[7]; Sr:string[4];
      DxS,DyS:real; Idy,Idx:integer;
    begin
      { Определение размещения графика и вывод осей координат }
      Dy:=(Max-Min)/Yud; Dx:=X[Cnt]/Xlr;
      Line(Xl,Yu,Xl,Yd); Line(Xl,Yd,Xr,Yd);
      { Вывод координатной сетки и надписей по осям координат }
      DyS:=(Max-Min)/Ny; Idy:=Yud div Ny;
      for I:=0 to Ny do { Горизонтальные линии } begin
        tY:=Yd-Idy*I; if tY<>Yd then Line(Xl,tY,Xr,tY);
        Str(I*DyS:7:2,St); OutTextXY(Xl-62,tY-3,St) end;
      DxS:=X[Cnt]*dT/Nx; Idx:=Xlr div Nx;
      for I:=1 to Nx do { Вертикальные линии } begin
        tX:=Xl+Idx*I; Line(tX,Yu,tX,Yd);
        Str(I*DxS:7:2,Sr); OutTextXY(tX-15,Yd+12,Sr) end
      end;
  procedure G2(Cnt,K:integer);
    var R:1..3;
  begin
    Readln;
    case K of 1:SetColor(13);

```

```

2:SetColor(11);
else SetColor(10) end;
for J:=1 to 5 do begin
  if J in [1] then R:=2*K else R:=1;
  for I:=0 to Cnt do begin
    tY:=Yd-Round(Y[J,I]/Dy); tX:=Xl+Round(X[I]/Dx);
    Circle(tX,tY,R) end end;
  end;
end.

```

7.3 Задание

7.3.1 Ответить на контрольные вопросы.

7.3.2 Ознакомиться с программой и составить алгоритм анализа модели ключа на полевом транзисторе.

7.3.3 Рассчитать зависимости $U_{ВХ}(t)$, $U_{ЗИ}(t)$, $U_{СИ}(t)$, $h(t)$ и построить графики (с использованием программы PGr1).

7.3.4 Оценить влияние на указанные зависимости одного из параметров эквивалентной схемы (по указанию преподавателя).

7.3.5 Проанализировать результаты и оформить отчет.

Исходные данные: $I_c = 11$ А, $U_s = 12.6$ В, $S = 2.72$ А/В, $k = 0.7$, $C_{11} = 3.5E-10$ Ф, $C_{12min} = 8E-12$ Ф, $A = 3.5E-11$ Ф, $S = 0.088$ 1/В, $C_{22min} = 1.3E-10$ Ф, $C = 6.59E-10$ Ф, $D = 0.33$ 1/В (параметры транзистора КП913А), $E_c = 45$ В, $R_c = 5$ Ом, $L = 1E-8$ Гн, $U_{max} = 20$ В, $t_{вх} = 3E-9$ с, $t_{и} = 14E-9$ с, начальный шаг $h_0 = 1E-9$ с, погрешность вычислений $Q = 0.001$, $t_0 = 0$, $R_r = 10$ Ом.

7.4 Контрольные вопросы

7.4.1 Чем обусловлена необходимость автоматического выбора шага численного интегрирования ОДУ при анализе переходных процессов?

7.4.2 В чем суть метода трех зон?

7.4.3 Как работает ключ на МДП транзисторе?

7.4.4 Каков смысл параметров эквивалентной схемы ключа?

7.4.5 В чем заключается метод Рунге – Кутта – Мерсона?

8 ОСНОВЫ МОДЕЛИРОВАНИЯ НА МИКРОУРОВНЕ

8.1 Цель работы

Ознакомление с моделированием объектов на микро–уровне на примере анализа стационарных электрических полей в печатной плате методами конечных разностей.

8.2 Метод сеток

Микро–уровень – иерархический уровень в описаниях сложных технических объектов, характерной особенностью которого является

рассмотрение физических процессов в сплошных средах и непрерывном времени. Типичными математическими моделями на микро–уровне являются дифференциальные уравнения в частных производных с заданными краевыми условиями. Основным методом решения таких уравнений является метод сеток. Результатами решения могут быть поля напряжений и деформаций в деталях механических конструкций, электрического потенциала в электронных приборах и т. п.

Сущность метода сеток состоит в аппроксимации искомой непрерывной функции совокупностью приближенных значений, рассчитанных, в некоторых точках заданной области - узлах. Совокупность узлов, соединенных определенным образом, и образует сетку. Сетка, в свою очередь, является дискретной моделью непрерывной области определения искомой функции.

Применение метода сеток позволяет свести дифференциальную краевую задачу к системе алгебраических уравнений относительно неизвестных узловых значений функции. В общем случае алгоритм метода сеток состоит из трех этапов:

- 1) построение сетки в заданной области (дискретизация задачи);
- 2) получение системы алгебраических уравнений относительно узловых значений (алгебраизация задачи);
- 3) решение полученной системы алгебраических уравнений.

Наиболее часто используются два варианта метода сеток: метод конечных элементов и метод конечных разностей. Эти методы отличаются друг от друга на первых двух этапах алгоритма и практически идентичны на этапе 3).

8.2 Формирование и реализация модели

Необходимо исследовать электрическое поле в ограниченном участке сечения многослойной печатной платы. Сечение параллельно плоскости ХУ (рис. 8.1, а). Сигнальные проводники располагаются на внутренних слоях платы и проходят перпендикулярно плоскости сечения (на рисунке не показаны, поскольку их геометрические размеры и взаимное расположение, пока не имеют значения). Стационарное распределение потенциала V внутри рассматриваемого участка описывается двумерным уравнением Лапласа

$$\partial^2 V / \partial X^2 + \partial^2 V / \partial Y^2 = 0. \quad (8.1)$$

Краевые условия в направлении оси X (на боковых границах участка) для упрощения задачи определяются из условия симметрии, краевыми условиями в направлении оси Y являются потенциалы на слоях питания (А) и "земли" (В).

Рассмотрим этапы решения этой задачи методом конечных разностей.

Этап 1) – дискретизация. На заданном участке сечения строится равномерная прямоугольная сетка с шагом, равным одной десятой толщины слоя диэлектрика, т. е. $h_x = h_y = h = 0.04$ мм (рис. 8.1, б).

Этап 2) – алгебраизация. Для аппроксимации частных производных (8.1) в узлах с координатами (i, j) , где $i = 1..59$, $j = 1..109$, воспользуемся отношением конечных разностей:

$$(V_{i+1,j} - 2V_{i,j} + V_{i-1,j})/h^2 + (V_{i,j+1} - 2V_{i,j} + V_{i,j-1})/h^2 = 0. \quad (8.2)$$

Умножив (8.2) на h^2 , получим для каждого из этих узлов уравнение Лапласа в виде

$$V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1} - 4V_{i,j} = 0. \quad (8.3)$$

На боковых границах, т. е. при $j = 0, 110$ уравнение Лапласа записывается с учетом симметрии распределения потенциала слева и справа от границы:

$$\begin{aligned} V_{i+1,0} + V_{i-1,0} + 2V_{i,1} - 4V_{i,0} &= 0, \\ V_{i+1,110} + V_{i-1,110} + 2V_{i,109} - 4V_{i,110} &= 0. \end{aligned} \quad (8.4)$$

Этап 3) – расчет потенциалов узлов сетки. Перепишем (8.3) и (8.4) в явном относительно $V_{i,j}$ виде. Полученную систему уравнений можно решать методом итераций или др. методами. В приведенной ниже программе использован метод итераций.

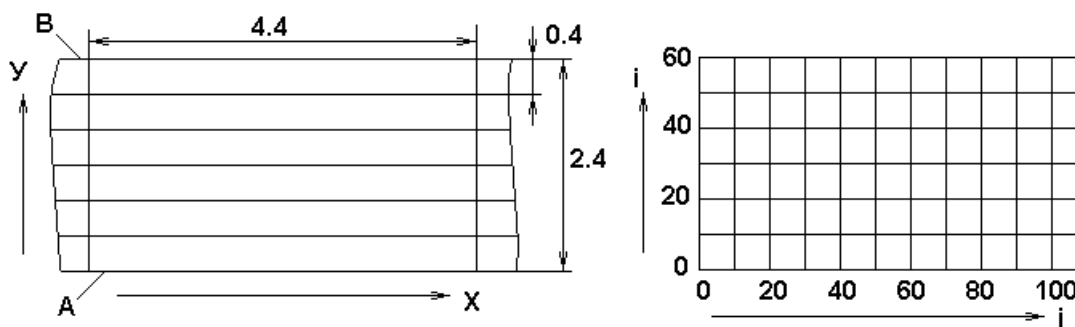


Рисунок 8.1 – Участок сечения многослойной печатной платы (а) и соответствующая ему сетка (б)

Приведенные на рис. 8.1, а, б сечение и сетка не содержат информации о сигнальных проводниках, располагаемых на внутренних слоях печатной платы. Чтобы смоделировать любой сигнальный проводник, необходимо:

- 1) определить узлы сетки, соответствующие координатам местоположения проводника (учитывается только его ширина);
- 2) включить в программу операторы, обеспечивающие необходимый потенциал на проводнике (не изменяющийся в процессе итераций).

Например, необходимо смоделировать сигнальный проводник шириной 1.2 мм и потенциалом 4 В, расположенный точно в середине рассматриваемого участка. Такому расположению проводника соответствует узлы сетки с $i = 30$, $j = 40..70$. В приведенной программе постоянство потенциала таких узлов обеспечивает выделенный оператор условного перехода.

PROGRAM MP10; {Расчет распределения потенциала в сечении}

```

uses Crt,Graph;           { многослойной печатной платы }
const Lm=7; Ld=Lm-1;     { Число слоев металлизации и диэлектрика }
    Vpit=24.0;           { Напряжения на шине питания, В }
    Vpr1=24.0; Vpr2=8.0;           { На проводниках 1 и 2, В }
    Imax=Ld*10; Jmax=110; { Максимальные номера узлов сетки }
    I1=Imax-1; J1=Jmax-1; I3=3*Imax; J3=3*Jmax;
    Ipr1=10; JprMin1=50; JprMax1=70;     { Проводник 1 }
    Jpr1=[JprMin1..JprMax1];
    Ipr2=50; JprMin2=50; JprMax2=70;     { Проводник 2 }
    Jpr2=[JprMin2..JprMax2];
    dVdop=0.01;           { Допустимая ошибка, В }
var V:array[0..Imax,0..Jmax] of real;
    I,J,Iter:byte; Vtek,dVtek,dV:real;
PROCEDURE GraphV;           { Вывод линий равного потенциала }
var Gd,Gm,R:integer; I,J,L:byte;
    Vgr:array[0..Ld] of integer; St:string[7];
begin
    Gd:=Detect; InitGraph(Gd,Gm,'c:\tp\bgi'); ClearDevice;
    SetViewPort(120,100,J3+240,I3+180,ClipOn); ClearViewPort;
    SetBkColor(0); SetColor(7);
    SetFillStyle(SolidFill,0); Bar3D(135,5,258,21,3,True);
    OutTextXY(141,10,'ЭКВИПОТЕНЦИАЛИ');
    Bar3D(0,50,J3,I3+50,3,True);
    Line(3*JprMin1,3*Ipr1+50,3*JprMax1,3*Ipr1+50);
    Line(3*JprMin2,3*Ipr2+50,3*JprMax2,3*Ipr2+50);
for L:=0 to Ld do begin
    I:=I3*L div Ld+50;
    Vgr[L]:=Round(Vpit*L) div Ld;
    Str(Vgr[L]:4,St); St:=St+' B';
    OutTextXY(J3,(I-4),St) end;
for I:=0 to Imax do

```



```

for J:=0 to Jmax do begin
  R:=Round(V[I,J]);
  for L:=0 to Ld do
    if R=Vgr[L] then PutPixel(3*J,3*I+50,L+2)
  end;
repeat until KeyPressed;
  CloseGraph
end;
BEGIN
for I:=0 to Imax do      { Начальные и граничные условия }
  for J:=0 to Jmax do V[I,J]:=Vpit*I/Imax;
  Iter:=0;
repeat Inc(Iter); dVtek:=0.0;
  for I:=1 to I1 do begin
    for J:=1 to J1 do begin
      if (I=Ipr1) and (J in Jpr1) then Vtek:=Vpr1
      else if (I=Ipr2) and (J in Jpr2) then Vtek:=Vpr2
      else begin
        Vtek:=(V[I+1,J]+V[I-1,J]+V[I,J+1]+V[I,J-1])/4.0;
        dV:=abs(Vtek-V[I,J]);
        if dV>dVtek then dVtek:=dV
      end; V[I,J]:=Vtek
    end;
    V[I,0]:=V[I,2];      { Из условия симметрии на левой }
    V[I,Jmax]:=V[I,Jmax-2]      { и правой границах }
  end;
  writeln(Iter:3,dVtek:11:5)
until dVtek<dVdop;
  GraphV
END.

```

8.3 Задание

8.3.1 Ознакомиться с описанием работы и ответить на контрольные вопросы.

8.3.2 Начертить сетку для исследуемого участка печатной платы, нанести на чертеж выбранную комбинацию сигнальных проводников.

8.3.3 Рассчитать электрические потенциалы в узлах сетки, нанести на чертеж линии эквипотенциального уровня с шагом 0.5 В.

8.3.4 Выявить области с наибольшей напряженностью электрического поля, изменить взаимное расположение проводников таким образом, чтобы уменьшить максимальную напряженность поля; повторить п. 2, оценить результаты перекомпоновки.

8.3.5 Проанализировать результаты и оформить отчет.

8.4 Контрольные вопросы

8.4.1 Какие основные процедуры включает в себя метод конечных разностей?

8.4.2 Какие важные допущения приняты в лабораторной работе? Как они повлияли на точность и сложность расчетов?

8.4.3 Каким образом можно повысить точность анализа полей в печатной плате?

8.4.4 Чем определяется величина шага сетки?

8.4.5 Возможны ли другие краевые условия? Какие?

Литература

1. Фидлер Дж. К., Найтингейл К. Машинное проектирование электронных схем/ Пер. с англ. и предисл. Д. И. Панфилова, А. Г. Соколова; Под ред. Г.Г. Казеннова. – М.: Высш. шк., 1985. – С. 143.

2. Норенков И.П., Маничев В.Б. Системы автоматизированного проектирования электронной и вычислительной аппаратуры: – М.: Высш. шк., 1933. – С. 165.

3. Системы автоматизированного проектирования. Кн. 5. Автоматизация функционального проектирования: Учеб. пособие для втузов/ П.К. Кузьмин, В.Б. Маничев; Под ред И.П. Норенкова. – Минск: Выш. шк., 1988. – С. 7.

4. Дьяконов В.П. Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ. – М.: Наука, 1967. – С. 207.

5. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран, Паскаль. – Томск: МП «РАСКО», 1991. – 272 с.

6. Системы автоматизированного проектирования. Кн. 4. Математические модели технических объектов: Учебное пособие для втузов / В. А. Трудоношин, Н. В. Пивоварова; Под ред.: И. П. Норенкова. – Минск: Выш. шк., 1988, – 159 с.

7. Системы автоматизированного проектирования. Кн. 7. Лабораторный практикум: Учеб. пособие для втузов/ Т. И. Вулдакова, Д. М. Жук, С. С. Комалов и др.; Под ред. И. П. Норенкова. – Минск: Выш. шк., 1988. – 143 с.