**ОПИСАНИЕ ПРИМЕНЕНИЯ**

Программа по расчету режимов *Regim\_Mlab* предназачена для расчета режимов распределительных электрических сетей произвольной конфигурации на основе контурных уравнений. Програма выполняет следующие основные функции: создание информационной модели с данными по сети, создание графа, представляющего топологическую информацию по схеме сети, собственно расчет режима, а также графическое отображения топологии схемы, ввод/вывод информации из Excel.

**1 Установка программы**

Программа предназначена для выполнения на персональных компьютерах общего назначения. Для запуска программы необходимо выполнить следующие требования к аппартному и программному окружению:

1. Требования к аппаратной части:
   1. Процессор *Intel* или *AMD* x86 с поддержкой *SSE2*;
   2. Оперативная память – не менее 2 Гб;
   3. Пространство на жестком диске – 4 Гб.
   4. Графический адаптер должен поддерживать технологию OpenGL3.3;
2. Требования к программному окружению:
   1. ОС MS Windows *XP* *SP*3 или выше (разрядность операционной системы может быть как 32 бита, так и 64);
   2. Установленный пакет *MS Offise* версии 2003 или выше, включающий *MS Word* для работы с документацией и *MS Excel* для возможности работы Matlab с *Excel* файлами;
   3. Программный комплекс *MATLAB* версии 2015*b* (установка модуля *Simulink* не требуется).

Установка программы производится путем копирования раздела под одноименным названием *Regim\_Mlab* целиком(ВАЖНО!!! Следует копировать раздел не изменяя его названия, а также внутренней структуры, названий подразделов и файлов)с *CD*-диска с ПО на жесткий компьютера на любой том, в любой его раздел или подраздел. Автором программа устанавливалась в корень тома *E:*\. Путь в таком случае выглядит следующим образом: *E:*\*Regim\_Mlab*

После установки папка с программой должна иметь вид представленный на рисунке 1.

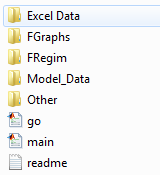


Рисунок 1 – Вид раздела с программой *Regim\_Mlab*

**2 Запуск программы**

Для запуска программы следует запустить *m*-файл *main.m*. Открывшееся окно с программой должно иметь вид представленный на рисунке 2.

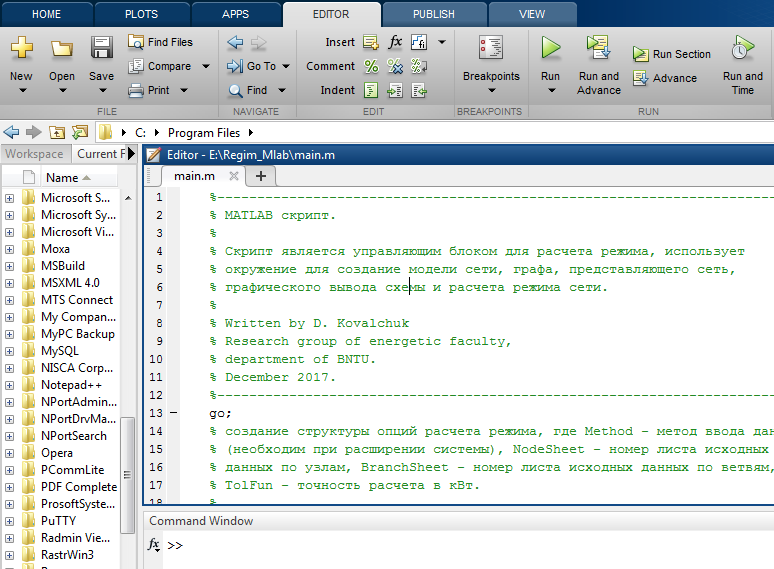


Рисунок 2 – Окно с *m*-файлом *main.m*

После открытия окна с программой для запуска программы следут открыть вкладку *EDITOR*, и нажать кнопку *Run* или нажать «горячую» клавишу *F*5.

При первом запуске программы в текущей сесии работы с Windows есть возможность, того что путь к рабочей папке будет указан неверно. В таком случае *MATLAB* предложит сменить текущий каталог на каталог в котором размещен запускаемый файл *main.m*. Следует нажать кнопку «*Change Folder*» или нажать клавишу *Enter*. Вид окна приведен на рисунке 3.

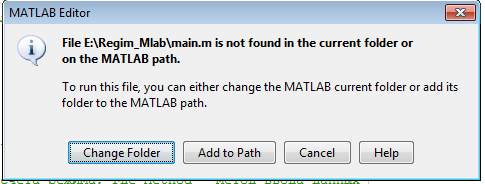


Рисунок 3 – Вид диалогового окна о смене рабочего каталога *MATLAB*

При успешном выполнении программы командное окно будет иметь вид, представленный на рисунке 4.

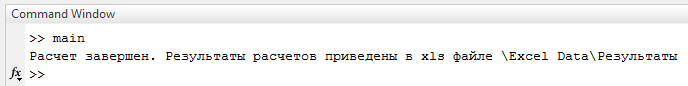


Рисунок 4 – Вид командного окна при успешном выполнении программы

**3 Ввод исходных данных**

Для ввода данных используется *Excel* файл с названием «Исходные\_данные.xls» на листах «Узлы», «Ветви». В данном файле на 2 листах расположена информация об узлах и ветвях схемы. Excel-файлы размещаются в подкаталоге   
*Regim\_Mlab\Excel Data*. В папке размещаются несколько тестовых Excel-файлов исходных данных открывается файл с названием «Исходные\_данные.xls» для изменения выбранного файла следует изменить название, например, «Исходные\_данные\_1.xls» на «Исходные\_данные.xls» и наоборот. Формат ввода исходных данных по узлам приведен в таблице 1.

Таблица 1 – Исходные данные по узлам для работы программы *Regim\_Mlab* контрольной схемы №1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Uном, кВ | 10,5 |  |  |  |  |  |  |  |  |  |
| Пользовательский номер | Пользовательское  наименование | Тип | Pнагр,  кВт | Qнагр,  квар | Pген,  кВт | Qген,  квар | Qmin,  квар | Qmax, квар | Uуст, кВ | δU, ° |
| 1 | ЦП1 | ЦП |  |  |  |  |  |  | 10,5 | 0 |
| 2 | ТП1 | Нагр | 400 | 200 |  |  |  |  |  |  |
| 3 | ТП2 | Нагр | 100 | 50 |  |  |  |  |  |  |
| 4 | Г | Ген |  |  | 300 | 150 | 200 | 1000 | 10,5 |  |
| 5 | ТП3 | Нагр | 300 | 100 |  |  |  |  |  |  |
| 6 | ТП4 | Нагр | 200 | 150 |  |  |  |  |  |  |
| 7 | ЦП2 | ЦП |  |  |  |  |  |  | 10,5 |  |
| 8 | ТП5 | Нагр | 200 | 150 |  |  |  |  |  | 0 |

В первой строке указано номинальное напряжения рассчитываемой сети (программа поддерживает расчет сети одного класса напряжения).

В столбце 1 приведен пользовательский номер узла – данное поле таблицы используется при построении графа схемы для определения начал и концов участков.

В столбце приведены 2 приведены пользовательские наименования узлов – носят чисто информационное значение.

В столбце 3 приведены типы узлов. Узлы бывают трех типов:

Первый, моделируемый моделью *PQ*, обозначается как «Нагр».

Второй, моделируемый моделью баланисирующего узла (БУ), обозначается как «ЦП».

Третий, моделируемый моделью *PU*, обозначается как «Ген».

В столбце 4 приведены активные мощности нагрузок узлов в кВт.

В столбце 5 приведены реактивные мощности нагрузок узлов в квар.

В столбце 6 приведены величины активной мощности генерации в кВт.

В столбце 7 приведены величины реактивной мощности генерации, для *PU* узлов данное значение принимается за начальное приближение.

В столбце 8 приведены величины минимальных пределов реактивной мощности генераторов в квар.

В столбце 9 приведены величины максимальных пределов реактивной мощности генераторов в квар.

В столбце 10 приведены величины уставок модулей напряжений для балансирующих и генераторных узлов,для нагрузочных узлов указанное напряжение не применяется в кВ.

В столбце 11 приведены величины уставок фаз напряжений для балансирующих узлов, для нагрузочных и генераторных узлов указанное параметр применяется в качестве первой итерации в градусах.

Таблица 2 – Исходные данные по ветвям для работы программы *Regim\_Mlab* контрольной схемы №1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Пользовательский номер | Пользовательское наименование | Номер  начала | Номер  конца | Комутационные  аппараты (Р1Р1) | Тип ветви | Ri-j, Ом | Xi-j, Ом | Pх-х, кВт | Qх-х(емк), квар | kт |
| 1 |  | 1 | 2 | Р2Р2 | Линия | 2,00 | 2,00 |  | -0,02 |  |
| 2 |  | 2 | 3 | Р2Р2 | Линия | 1,50 | 1,00 |  | -0,03 |  |
| 3 |  | 3 | 4 | Р2Р2 | Линия | 1,20 | 1,00 |  | -0,03 |  |
| 4 |  | 4 | 5 | Р2Р2 | Линия | 1,00 | 0,80 |  | -0,03 |  |
| 5 |  | 5 | 6 | Р2Р2 | Линия | 0,80 | 1,00 |  | -0,03 |  |
| 6 |  | 6 | 7 | Р2Р2 | Линия | 0,90 | 0,70 |  |  |  |
| 7 |  | 3 | 8 | Р2Р2 | Транс | 0,90 | 0,70 |  |  | 25 |

В столбце 1 приведен пользовательский номер ветви – носят чисто информационное значение.

В столбце 2 приведены приведены пользовательские наименования ветвей – носят чисто информационное значение.

В столбце 3 приведены приведены номера начал ветвей – используются для построения графа.

В столбце 4 приведены приведены номера концов ветвей – используются для построения графа.

В столбце 5 приведена запись типов и положения коммутационной аппаратуры по началам и концам участков. Запись должна состоять из 4 – ех символов. 1 и 3 – ий символ означает тип коммутационного аппарата. 2 и 4-ый символы означают его положение.

В программе применяется 4 типа коммутационных аппаратов:

Первый, разъединитель, обозначается «Р».

Второй, выключатель, обозначается «В».

Третий, выключатель нагрузки, обозначается «Н»

Четвертый, коммутационный аппарат отсутствует, обозначется « ».

В программе применяется 4 состояния для коммутационных аппаратов типа «Р», «В», «Н»:

Первый, аппарат отключен с запретом включения, обозначается «0».

Второй, аппарат отключен, обозначается «1».

Третий, аппарат включен, обозначается «2»

Четвертый, аппарат включен с запретом отключения, обозначется «3».

При отсутствии коммутационного аппарата для него возможно только состояние «3»

В столбце 6 приведены типы ветвей. Ветви бывают двух типов:

Первый, тип линия, обозначается «Линия».

Второй, тип трансформатор, обозначается «Транс».

В столбце 7 приведены активные сопротивления ветвей в Ом.

В столбце 8 приведены реактивные сопротивления ветвей в Ом.

В столбце 9 приведены величины потерь активной мощности холостого хода линий и трансформаторов в кВт.

В столбце 10 приведены величины потерь реактивной мощности холостого хода линий и трансформаторов в квар.

В столбце 11 приведены величины коэффициента трансформации трансформаторов в квар.

**4 Вывод результатов расчета**

Для ввода данных используется *Excel* файл с названием «Результаты.xls» на листах «Узлы», «Ветви», «Общие». В данном файле на 2 листах расположены результаты расчетов по узлам и ветвям схемы. Excel-файлы размещаются в подкаталоге   
*Regim\_Mlab\Excel Data*. При необходимости сохранения результатов расчет необходимо скопировать файл «Результаты.xls» с другим именем, например, «Результаты\_1.xls». Пример результатов расчетов по узлам приведен в таблице 3.

Таблица 3 – Результаты работы программы *Regim\_Mlab* по узлам контрольной схемы №1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Пользовательский номер | Тип | Pнагр,  кВт | Qнагр,  квар | Pген,  кВт | Qген,  квар | Qmin,  квар | Qmax, квар | U, кВ | δU, ° |
| 1 | ЦП | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 10,500 | 0,000 |
| 2 | Нагр | 400,00 | 200,00 | 0,00 | 0,00 | 0,00 | 0,00 | 10,416 | -0,426 |
| 3 | Нагр | 100,00 | 50,00 | 0,00 | 0,00 | 0,00 | 0,00 | 10,431 | -0,583 |
| 4 | Ген | 0,00 | 0,00 | 300,00 | 758,84 | 200,00 | 1000,00 | 10,500 | -0,679 |
| 5 | Нагр | 300,00 | 100,00 | 0,00 | 0,00 | 0,00 | 0,00 | 10,470 | -0,494 |
| 6 | Нагр | 200,00 | 150,00 | 0,00 | 0,00 | 0,00 | 0,00 | 10,466 | -0,232 |
| 7 | ЦП | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 10,500 | 0,000 |
| 8 | Нагр | 200,00 | 150,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,416 | -0,586 |

Все обозначения соответствуют формату исходных данных, за исключеним типов узлов. Добавляются типы узлов:

«Ген+» - обозначает генераторный узел, работающий на верхнем пределе реактивной мощности.

«Ген-» - обозначает генераторный узел, работающий на нижнем пределе реактивной мощности.

Пример результатов расчетов по ветвям приведен в таблице 4.

Таблица 4 – Результаты работы программы *Regim\_Mlab* по ветвям контрольной схемы №1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Пользовательский номер | Номер  начала | Номер  конца | Тип ветви | Pн, кВт | Qн, квар | Pк, кВт | Qк, квар | I, А | ∆U, кВ |
| 1 | 1 | 2 | Линия | 424,44 | 18,04 | 421,16 | 14,77 | 23,36 | 0,114 |
| 2 | 2 | 3 | Линия | 21,16 | -185,22 | 20,68 | -185,54 | 10,33 | 0,032 |
| 3 | 3 | 4 | Линия | -279,86 | -386,04 | -282,37 | -388,13 | 26,39 | 0,071 |
| 4 | 4 | 5 | Линия | 17,64 | 370,72 | 16,39 | 369,72 | 20,41 | 0,045 |
| 5 | 5 | 6 | Линия | -283,61 | 269,74 | -284,73 | 268,34 | 21,58 | 0,048 |
| 6 | 6 | 7 | Линия | -484,73 | 118,35 | -486,78 | 116,76 | 27,52 | 0,054 |
| 7 | 3 | 8 | Транс | 200,52 | 150,40 | 200,00 | 150,00 | 13,87 | 0,027 |

Все обозначения соответствуют формату исходных данных. Добавляются новые обозначения:

В столбце 5 приведены активные мощности начал ветвей в кВт;

В столбце 6 приведены реактивные мощности начал ветвей в квар;

В столбце 7 приведены активные мощности концов ветвей в кВт;

В столбце 8 приведены реактивные мощности концов ветвей в квар;

В столбце 9 приведены токи ветвей в А;

В столбце 10 приведены потери напряжений на ветвях в кВ.

В таблице 5 приведены интегральные расчетные параметры сети

Таблица 5 – Обобщенные результаты расчета *Regim\_Mlab* контрольной схемы №1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Активная генерация в сети, кВт | Активное потребление в сети, кВт | Реактивная генерация в сети, квар | Реактивное потребление в сети, квар | Суммарные потери активной мощности, кВт | Суммарные потери реактивной мощности, квар | Суммарные нагрузочные потери активной мощности, кВт | Суммарные потери холостого хода активной мощности, кВт | Суммарные потери активной мощности в линиях, кВт | Суммарные потери активной мощности в трансформаторах, кВт |
| 300,00 | 1200,00 | 758,84 | 650,00 | 11,2145 | 10,0260 | 11,1945 | 0,0200 | 10,6748 | 0,5397 |

**5. Контрольные примеры расчета**

Первым контрольным примером являются расчет сети, приведенной в файле «Исходные\_данные.xls», копия в «Исходные\_данные\_1.xls» (результаты в файле «Результаты\_1.xls»). Исходные данные и результаты приведены выше.

Исходные данные по 2 контрольной схеме хранятся в файле «Исходные данные\_2». Результаты расчета режима в файле «Результаты\_2.xls».

Исходные данные по 3 контрольной схеме хранятся в файле «Исходные данные\_3». Результаты расчета режима в файле «Результаты\_3.xls».

Исходные данные по 4 контрольной схеме хранятся в файле «Исходные данные\_4». Результаты расчета режима в файле «Результаты\_4.xls».

Исходные данные по 5 контрольной схеме хранятся в файле «Исходные данные\_4». Результаты расчета режима в файле «Результаты\_4.xls».

Ниже приведем таблицы с обобщенными результата расчета режима 5 контрольных сетей.

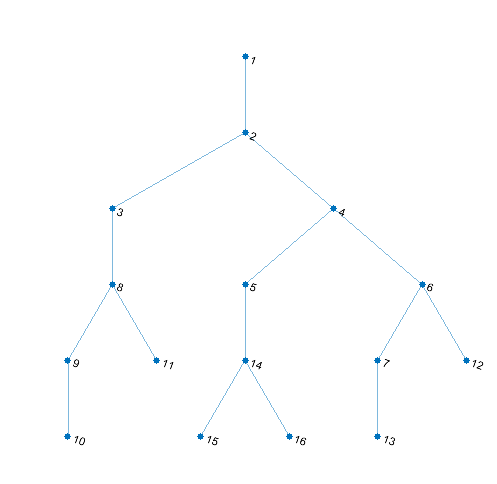


Рисунок 5 – Схема контрольной сети №2

Таблица 6 – Обобщенные результаты расчета *Regim\_Mlab* контрольной схемы №2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Активная генерация в сети, кВт | Активное потребление в сети, кВт | Реактивная генерация в сети, квар | Реактивное потребление в сети, квар | Суммарные потери активной мощности, кВт | Суммарные потери реактивной мощности, квар | Суммарные нагрузочные потери активной мощности, кВт | Суммарные потери холостого хода активной мощности, кВт | Суммарные потери активной мощности в линиях, кВт | Суммарные потери активной мощности в трансформаторах, кВт |
| 0,00 | 1536,00 | 0,00 | 843,00 | 352,3525 | 220,4699 | 352,3525 | 0,0000 | 352,3525 | 0,0000 |

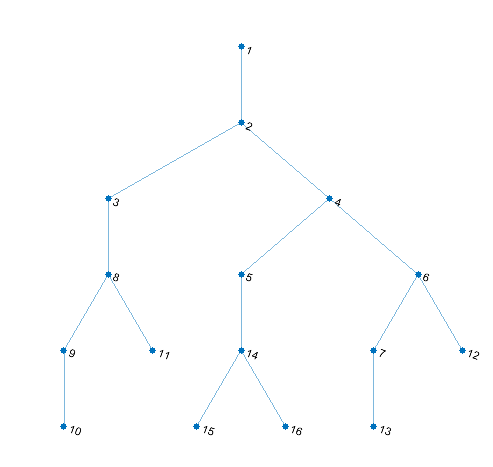


Рисунок 6 – Схема контрольной сети №3

Таблица 7 – Обобщенные результаты расчета *Regim\_Mlab* контрольной схемы №3

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Активная генерация в сети, кВт | Активное потребление в сети, кВт | Реактивная генерация в сети, квар | Реактивное потребление в сети, квар | Суммарные потери активной мощности, кВт | Суммарные потери реактивной мощности, квар | Суммарные нагрузочные потери активной мощности, кВт | Суммарные потери холостого хода активной мощности, кВт | Суммарные потери активной мощности в линиях, кВт | Суммарные потери активной мощности в трансформаторах, кВт |
| 2917,00 | 2339,00 | 1121,68 | 1274,00 | 303,3271 | 174,9378 | 303,3271 | 0,0000 | 303,3271 | 0,0000 |

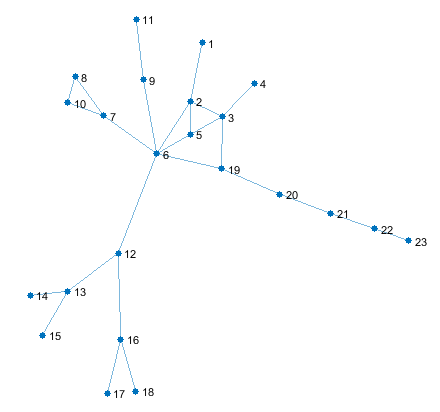


Рисунок 7 – Схема контрольной сети №4

Таблица 8 – Обобщенные результаты расчета *Regim\_Mlab* контрольной схемы №4

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Активная генерация в сети, кВт | Активное потребление в сети, кВт | Реактивная генерация в сети, квар | Реактивное потребление в сети, квар | Суммарные потери активной мощности, кВт | Суммарные потери реактивной мощности, квар | Суммарные нагрузочные потери активной мощности, кВт | Суммарные потери холостого хода активной мощности, кВт | Суммарные потери активной мощности в линиях, кВт | Суммарные потери активной мощности в трансформаторах, кВт |
| 1600,00 | 2970,00 | 1817,49 | 1530,00 | 228,9491 | 164,3580 | 228,9491 | 0,0000 | 228,9491 | 0,0000 |

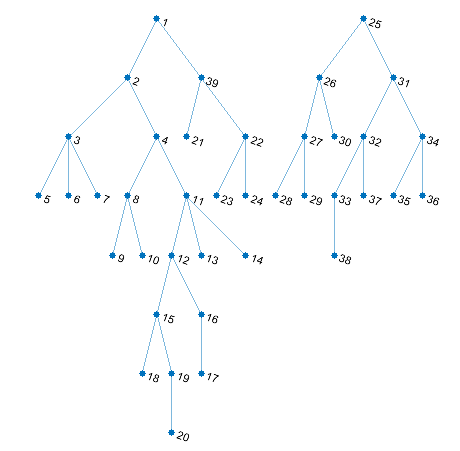


Рисунок 8 – Схема контрольной сети №5

Таблица 9 – Обобщенные результаты расчета *Regim\_Mlab* контрольной схемы №5

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Активная генерация в сети, кВт | Активное потребление в сети, кВт | Реактивная генерация в сети, квар | Реактивное потребление в сети, квар | Суммарные потери активной мощности, кВт | Суммарные потери реактивной мощности, квар | Суммарные нагрузочные потери активной мощности, кВт | Суммарные потери холостого хода активной мощности, кВт | Суммарные потери активной мощности в линиях, кВт | Суммарные потери активной мощности в трансформаторах, кВт |
| 1600,00 | 2970,00 | 1817,49 | 1530,00 | 228,9491 | 164,3580 | 228,9491 | 0,0000 | 228,9491 | 0,0000 |

**6. Описание применения программных средств разработчиком сложных систем**

Программа по расчету режимов *Regim\_Mlab* предназачена для расчета режимов распределительных электрических сетей произвольной конфигурации на основе контурных уравнений. Программа состоит из ряда функций, программ и скриптов, описанных ниже.

**6.1. Скрипт main**

Для демонстрации основных возможностей программы с вводом и выводом данных в файл Excel служит функция main. К таким возможностям относится создание модели и графа сети, графическое отображение сети, ввод и вывод результатов из файлов Excel и основная функция – расчет режима электрической сети. Функция main является скриптом – не имеет входных и выходных данных.

При проектировании программы не предусматривалось использование описываемого скрипта в составе других программных комплексов (за исключением копирования частей кода). Однако существует возможность использования глобального объекта hM полученного в результате работы данного скрипта в дальнейших расчетах в смежных программах.

**6.2 Класс CModelRS**

Класс CModelRS предназначен для хранения и предоставления доступа к данным о топологических и режимных параметрах распеделительной электрической сети, а также для дорасчета некоторых дополнительных и интегральных параметров на основании базовых.

Главной особенностью класса является то, что это handle класс. Это значит, что может существовать единественный объект данного класса, а при попытке копирования будет лишь создаваться ссылка на первый – единственный объект и все последующие изменения происходят с оригинальным экземпляром по ссылке.

Класс имеет следующие атрибуты:

NODE – структура, хранит данные по узлам;

BRAN – структура, хранит данные по ветвям;

COMM – структура, хранит интегральные параметры сети.

Стоит отметить, что прямой доступ к данным, хранящимся в атрибутах запрещен как на запись, так и на чтение. Доступ осуществляет при помощи ряда правил, описанных в функциях subsref и subsasgn, описанных ниже.

Ниже приведено описание всех публичных методов класса доступных пользователю класса.

**6.2.1 Конструктор CModelRS**

Конструктор класса CModelRS создает модель сети путем чтения из объекта класса CData (вместо объекта может быть использована структура с аналогичными полями) (класс работы с исходными данными).

Конструктор вызывается командой показаной ниже.

hM = CModelRS(DataNODE, DataBRAN, DataCOMM)

где, hM - handle созданного объекта класса;

DataNODE – поле объекта класса исходных данных CData;

DataBRAN - поле объекта класса CData, хранящий данные по ветвям

DataCOMM - поле объекта класса CData, хранящий общие данные;

Пример вызова команды:

hM = CModelRS(oDt.NODE, oDt.BRAN, oDt.COMM);

Примечание:

Часть полей класса является не данными, а handle-функциями осуществляющими связь расчетных параметров с основными параметрами режима.

Описание полей структур с данными о сети с примерами обращения к ним (примеры приведены для объекта класса CModelRS под названием hM) приведено ниже:

Данные по узлам:

Исходные данные:

hM.NODE.Nn – номера узлов в модели (double);

hM.NODE.Nn1 – пользовательские номера узлов (используются для построения графа схемы) (double);

hM.NODE.Type – исходные типы узлов (может принимать значение 1 – для PQ узла, 2 – для BU узла, 3 – для PU узла, 4 для PU узла на минимальном пределе, 5 – для PU узла на максимальном пределе);

hM.NODE.CurrType –типы узлов по результатам расчета режима (может принимать значение 1 – для PQ узла, 2 – для BU узла, 3 – для PU узла, 4 для PU узла на минимальном пределе, 5 – для PU узла на максимальном пределе);

hM.NODE.Pn – активная мощность нагрузки, кВт (double);

hM.NODE.PnX – активная мощность нагрузки с учетом потерь холостого хода линий и трансформаторов, квар (double) (расчетный параметр – рассчитывает при помощи метода CalcXX);

hM.NODE.Qn – реактивная мощность нагрузки, кВт (double);

hM.NODE.QnX – реактивная мощность нагрузки с учетом потерь холостого хода линий и трансформаторов, квар (double) (расчетный параметр – рассчитывает при помощи метода CalcXX);

hM.NODE.Pg – активная мощность генерации в узле, кВт (double);

hM.NODE.Qg – исходная реактивная мощность генерации в узле, квар (double);

hM.NODE.QgR – реактивная мощность генерации в узле после расчета режима, квар (double);

hM.NODE.Qmin – минимальный предел реактивной мощности (для PU-узла), квар (double);

hM.NODE.Qmax – минимальный предел реактивной мощности (для PU-узла), квар (double);

hM.NODE.Uu – уставка модуля напряжения в узле (для PU узлов, для прочих – нулевое приближение), кВ (double);

hM.NODE.dUu – уставка фазы напряжения в узле (для балансирующих узлов, для прочих – нулевое приближение), рад (double);

Рабочие данные:

hM.NODE.Pnb – активная мощность нагрузки после балансировки, кВт (double) (на момент написания программы значение поля равно значению значению нагрузки до балансировки);

hM.NODE.Qnb – реактивная мощность нагрузки после балансировки, квар (double) (на момент написания программы значение поля равно значению значению нагрузки до балансировки);

hM.NODE.U – модуль напряжения по результатам расчета режима (для узлов по низкой стороне трансформаторов – напряжение приведенное к высокой стороне трансформатора), кВ (double);

hM.NODE.Unn – модуль напряжения по низкой стороне трансформаторов по результатам расчета режима, кВ (double) (рассчитывается при помощи метода CalcUT);

hM.NODE.dU – фаза напряжения в узлах по результатам расчета режима, рад (double);

hM.NODE.U1 – действительная часть напряжения в узле, кВ (handle, значение звязано с основными параметрами через handle-функцию);

hM.NODE.U2 – мнимая часть напряжения в узле, кВ (handle, значение звязано с основными параметрами через handle-функцию);

Данные по ветвям:

Исходные данные:

hM.BRAN.Nb – номер ветви (double);

hM.BRAN.Nb1 – пользовательские номера ветви (double);

hM.BRAN.NbSt – номер узла начала (пользовательский) (double);

hM.BRAN.NbF – номер узла конца (пользовательский) (double);

hM.BRAN.NbStM – номер узла начала (порядковый) (double) (рассчитывается внутри конструктора);

hM.BRAN.NbFM – номер узла конца (порядковый) (double) (рассчитывается внутри конструктора);

hM.BRAN.CmTpS – тип коммутационного аппарата в начале ветви (double);

hM.BRAN.CmStS – состояние коммутационного аппарата в начале ветви (double);

hM.BRAN.CmTpS – тип коммутационного аппарата в конце ветви (double);

hM.BRAN.CmStS – состояние коммутационного аппарата в конце ветви (double);

hM.BRAN.Type – тип ветви (1-линия или 2-трансформатор) (double);

hM.BRAN.R – активное сопротивление участка, Ом (double);

hM.BRAN.X – реактивное сопротивление участка, Ом (double);

hM.BRAN.Pxx – активная мощность холостого хода участка, кВт (double);

hM.BRAN.Qxx – реактивная мощность холостого хода участка, квар (double);

hM.BRAN.Pn – активная мощность начала участка, кВт (double);

hM.BRAN.Pk – активная мощность конца участка, кВт (double);

hM.BRAN.Qn – реактивная мощность начала участка, квар (double);

hM.BRAN.Qk – реактивная мощность конца участка, квар (double);

hM.BRAN.I1 – действительная часть тока ветви, А (handle);

hM.BRAN.I2 – мнимая часть тока ветви, А (handle);

hM.BRAN.Is – модуль тока ветви, А (handle);

hM.BRAN.Sn – полная мощность (модуль) начала ветви, кВА (handle);

hM.BRAN.Sk – полная мощность (модуль) конца ветви, кВА (handle);

hM.BRAN.dP – потери активной мощности на участке, кВт (handle);

hM.BRAN.dQ – потери активной мощности на участке, квар (handle);

hM.BRAN.dU1 – падение напряжения (действительная часть) на участке, кВ (handle);

hM.BRAN.dU2 – падение напряжения (мнимая часть) на участке, кВ (handle);

hM.BRAN.dU2 – потери напряжения (действительная часть) на участке, кВ (handle);

Интегральные параметры:

Интегральные параметры рассчитваются в специальном методе класса CModelRS – CalcSum (за исключением номинального напряжения и точности расчетов).

Данные по линиям:

hM.COMM.LINE.dpn – суммарные нагрузочные потери активной мощности в линиях, кВт;

hM.COMM.LINE.dpx – суммарные потери холостого хода активной мощности в линиях, кВт;

hM.COMM.LINE.dp – суммарные потери активной мощности в линиях, кВт;

hM.COMM.LINE.dqn – суммарные нагрузочные потери реактивной мощности в линиях, квар;

hM.COMM.LINE.dqx – суммарные потери холостого хода реактивной мощности в линиях, квар;

hM.COMM.LINE.dq – суммарные потери реактивной мощности в линиях, квар;

Данные по трансфоматорам:

hM.COMM.TRANS.dpn – суммарные нагрузочные потери активной мощности в трансформаторах, кВт;

hM.COMM.TRANS.dpx – суммарные потери холостого хода активной мощности в трансформаторах, кВт;

hM.COMM.TRANS.dp – суммарные потери активной мощности в трансформаторах, кВт;

hM.COMM.TRANS.dqn – суммарные нагрузочные потери реактивной мощности в трансформаторах, квар;

hM.COMM.TRANS.dqx – суммарные потери холостого хода реактивной мощности в трансформаторах, квар;

hM.COMM.TRANS.dq – суммарные потери реактивной мощности в трансформаторах, квар;

Общие данные:

hM.COMM.Un – номинальное напряжение сети;

hM.COMM.TolFun – заданная точность расчетов;

hM.COMM.ppotr – суммарная активная мощность нагрузки;

hM.COMM.pgen – суммарная активная мощность генерации в сети (за исключением балансирующих узлов);

hM.COMM.qpotr – суммарная реактивная мощность нагрузки;

hM.COMM.qgen – суммарная реактивная мощность генерации в сети (за исключением балансирующих узлов);

hM.COMM.dpn – суммарные нагрузочные потери активной мощности, кВт;

hM.COMM.dpx – суммарные потери холостого хода активной мощности, кВт;

hM.COMM.dp – суммарные потери активной мощности, кВт.

**6.2.2 Метод идексного присваивания subsasgn**

Метод осуществляет запись данных в поля объекта класса. Метод вызывается неявно при попытке прямой записи к соответствующему полю. Метод позволяет производить запись в любое поле объекта, допускается как поэлементное присваивание, так и векторов целиком (частей векторов) (для полей-векторов). Примеры присваиваний приведены ниже. Особенностью является то, что реально данные в классе по ветвям и узлам хранятся в виде структур массивов, а обращение к ним происходит так, как будто хранение организовано в виде массивов структур (прямое обращение к данным по узлам внутри класса (без переформирования запроса выглядит: hM.BRAN.R(1), а с переформированием hM.BRAN(1).R).

hM.COMM.dp=1;

hM.BRAN(1).R=1;

hM.BRAN(1:2).X=[1,2];

hM.BRAN.Pn=[1;2;3] (при количестве ветвей равном трем трех ветвей).

Стоит отметить, что функция записи не имеет встроенных средств защиты элементов класса от безвозвратного изменения или удаления при записи неверного числа элементов или элементов неверных типов (например, запись в поле размерностью 6 четырех элементов изменит размерность поля и удалит лишние элементы). Такое решение принято для ускорения записи в класс, так как контроль может вызывать замедление работы неприемлемое для больших схем. Ответственность за правильное использование класса лежит на пользователе класса.

**6.2.3 Метод идексного присваивания subsref**

Метод чтение данных из поляй объекта класса. Метод вызывается неявно при попытке прямого чтения из соответствующего поля. Метод позволяет производить запись в любое поле объекта, допускается как поэлементное чтение, так и чтение векторов целиком (частей векторов) (для полей-векторов). Примеры чтений приведены ниже. Особенностью является то, что реально данные в классе по ветвям и узлам хранятся в виде структур массивов, а обращение к ним происходит так, как будто хранение организовано в виде массивов структур (прямое обращение к данным по узлам внутри класса (без переформирования запроса выглядит: hM.BRAN.R(1), а с переформированием hM.BRAN(1).R).

A=hM.COMM.dp;

A=hM.BRAN(1).R;

A=hM.BRAN(1:2).X;

A=hM.BRAN.Pn; (при количестве ветвей равном трем трех ветвей).

Существуют особые формы обращения к количеству ветвей и узлов (hM.NODE.n, hM.BRAN.n).

Разрешается обращение в соответствии с записями в описании конструктора.

**6.2.4 Метод расчета интегральных параметров CalcSum**

Метод осуществляет расчет интегральных параметров (hM.COMM) электрической сети на основании заполненных полей по потокораспределении и генерации в узлах. Методы необходимо вызывать уже после произведенного расчета режима.

Функция требует в качестве входного параметра объект класса, изменения производятся непосредственно в самом объекте.

hM = CalcSum(hM).

Результаты отображаются в полях модели hM.COMM.

**6.2.5 Метод учета потерь холостого хода CalcXX**

Метод производит добавление мощностей холостого хода линий и трансформаторов к мощностям нагрузок узлов (учет потерь холостого хода в виде).

Метод вызывается непосредственно в конструкторе CModelRS, однако является публичным и возможен пересчет потерь холостого хода по результатам каких-либо расчетов уже после формирования объекта CModelRS.

Функция требует в качестве входного параметра объект класса, изменения производятся непосредственно в самом объекте.

hM = CalcXX(hM).

Результаты отображаются в полях модели hM.COMM.

**6.2.6 Метод учета потерь холостого хода CalcUT**

Метод производит расчет напряжений по низкой стороне, приведенных к низшей, трансформаторов на основании величины напряжений по низкой стороне, приведенных к высшей стороне.

Метод вызывается непосредственно в конструкторе CModelRS, однако является публичным и возможен пересчет потерь холостого хода по результатам каких-либо расчетов уже после формирования объекта CModelRS.

Функция требует в качестве входного параметра объект класса, изменения производятся непосредственно в самом объекте.

hM = CalcUT(hM).

Результаты отображаются в поле модели hM.NODE.Unn.

**6.3 Класс CGraph**

Класс CGraph предназначен для построения графа схемы и предоставления интерфейсов для доступа к информации о графе. Класс имеет тип Value – это означает, что может существовать несколько экземпляров типа.

Особенностью класса является запрет на изменение его полей после создания объекта при помощи конструктора. При необходимости изменении элементов класса, необходимо его перестроение.

Класс имеет следующие атрибуты:

nod - массив длиной n индексов узлов в общем массиве данных по узлам схемы длиной N (n <= N)

ira - массив чисел смежных узлов длиной n, где n - число узлов

lra - массив начал расположения адресов смежных узлов и ветвей в массивах ka и kb длиной n, где n - число узлов

rib - массив длиной m индексов ветвей в общем массиве данных по ветвям схемы длиной M (m <= M)

ny1 - массив адресов узлов начала ветвей (в nod) длиной m, где m - число ветвей

ny2 - массив адресов узлов конца ветвей (в nod) длиной m, где m - число ветвей

ka - массив адресов смежных узлов (в nod) длиной 2m, где m - число ветвей

kb - массив адресов смежных ветвей (в rib) длиной 2m, где m - число ветвей

ny - количество узлов

nb - количество ветвей

Стоит отметить, что прямой доступ к данным, хранящимся в атрибутах запрещен. Доступ осуществляет при помощи ряда правил, описанных в функции subsref, описанных ниже.

Ниже приведено описание всех публичных методов класса доступных пользователю класса.

**6.3.1 Конструктор CGraph**

Конструктор класса CGraph создает объект класса CGraph. Существует три способа создания объекта описываемого класса:

1. g = CGraph – создает пустой объект;
2. g = CGraph(g1) – создает копию объекта g1 в объект g;
3. g = CGraph(hFN, I, hFV, J) – создает объект g, на основании данных о началах и концах ветвей

где, hFN – handle на индексы узлов схемы (пользовательские).

hFV – handle на индексы номеров начал и концов (пользовательские).

I – массив номеров (порядковые в модели) используемыех узлов для создания графа.

J – массив номеров (порядковые в модели) используемых ветвей для создания графа.

**6.3.2 Метод идексного присваивания subsref**

Метод чтение данных из полей объекта класса. Метод вызывается неявно при попытке прямого чтения из соответствующего поля. Метод позволяет производить запись в любое поле объекта, допускается как поэлементное чтение, так и чтение векторов целиком (частей векторов) (для полей-векторов). Примеры чтений приведены ниже. Особенностью является то, что реально данные в классе по ветвям и узлам хранятся в виде структур массивов, а обращение к ним происходит так, как будто хранение организовано в виде массивов структур (прямое обращение к данным по узлам внутри класса (без переформирования запроса выглядит: hM.BRAN.R(1), а с переформированием hM.BRAN(1).R).

A=g.nod(1);

A=g.rib(1:2);

A=g.nod.ny(:);

Существуют особые формы обращения к количеству ветвей и узлов (g.nod.n, g.rib.n).

Разрешаются следующие виды обращения:

g.nod или (g.nod(1), g.nod(1:5)) – вызывает перечень номеров узлов (в модели), разрешается обращение к единичному элементу, а также к частичной группе элементов, (double);

g.nod.n – количество узлов, (double);

g.nod(1).an – вектор узлов смежных с узлом i (1 в примере) (использует приватный метод getan), (функция);

g.nod(1).ar – вектор ветвей смежных с узлом i (1 в примере) (использует привватный метод getar), (функция);

g.rib или (g.rib(1), g.rib(1:5)) – вызывает перечень номеров ветвей (в модели), разрешается обращение к единичному элементу, а также к частичной группе элементов, (double);

g.rib.n – количество ветвей, (double);

g.rib(:).ny1 или (g.rib(1).ny1, g.rib(1:5).ny1) – вызывает номер начала ветви, (double);

g.rib(:).ny2 или (g.rib(1).ny2, g.rib(1:5).ny2) – вызывает номер конца ветви, (double);

g.rib(:).ny или (g.rib(1).ny, g.rib(1:5).ny) – вызывает вектор (двумерный массив) номеров начала и конца ветви, (double);

Стоит обратить внимание, что при каждом обращении g.nod(1).an, g.nod(1).ar осуществляется вычисление списка узлов (ветвей) смежных с данным, так как эта информация не хранится в памяти, поэтому для ускорения работы программ, использующих данный класс, имеет смысл запоминать данную информацию во внешних по отношению к классу переменных.

**6.3.3 Контроль пустоты графа isempty**

Данный метод осуществляет контроль пустоты (отсутствия узлов и ветвей) в графе.

Вызов метод необходимо производить по аналогии с примером, приведенным ниже.

TF = isempty(g)

g-объект класса CGraph;

TF – логический признак равный 1 для пустого графа, 0 – для непустого графа.

**6.4 Функция расчета режима fDriveRegim\_PI**

Функция расчета режима fDriveRegim\_PI является управляющей функцией для решения задачи расчета режима электрической сети одного напряжения произвольной конфигурации с возможностью учета PU-узлов (с учетом ограничений по реактивной мощности) на основе метода контурных уравнений (при этом наибольшая эффективность должна достигаться для схем распределительной сети 6-10 кВ при небольшом числе контуров).

Под расчетом режима понимается определение потокораспределения по элементам схемы, а также величин напряжений в узлах на основании известной информации о параметрах элементов сети (топологическая взаимосвязанность схемы, активные и реактивные сопротивления, величины нагрузок, активной мощности генерации), а также модулей и фах напряжений в балансирующих узлах и уставок модулей напряжений в PU –узлах.

Максимальное количество итераций расчета задано в виде константы и равно 100. Для его изменения – необходима правка кода функции fDriveRegim\_PI в строке №115 optionsSol.MaxIter=100. Изменение поддаются также параметры контроля пределов реактивной мощности генераторов, которые записаны в строках №52-55 функции fSolvReg\_PI. Параметры решения линейных системы уравнений не поддаются корректировке, параметры решения нелинейных системы уравнений (решатель fsolve) заданы в строке 49 функции fSolvReg\_PI.

Синтаксис вызова функции и описание входных данных приведен ниже.

varargout = fDriveRegim\_PI(g, Input)

varargout – cell массив выходных данных;

g – связанный граф расчетной подсхемы (образуется после работы функции fGraphSub) (объект типа CGraph);

Input – структура входных параметров рассчитываемой подсхемы.

varargout{1} - структура данных по узлам;

varargout{2} - структура данных по ветвям;

varargout{3} - структура диагностических данных о расчете режима;

varargout{1}.QgR - вектор результирующих реактивных мощностей генераторов;

varargout{1}.CurrType - вектор результирующих типов узлов (генераторных);

varargout{1}.U - вектор модулей напряжений в узлах;

varargout{1}.dU - вектор фаз напряжений в узлах;

varargout{2}.Pn - вектор потока активной мощности в начале ветви;

varargout{2}.Pk - вектор потока активной мощности в конце ветви;

varargout{2}.Qn - вектор потока реактивной мощности в начале ветви;

varargout{2}.Qk - вектор потока реактивной мощности в конце ветви;

varargout{3}.Flag - переменная означающая статус завершения расчета режима (1 - расчет завершен нормально, 2 - превышено максимальное число итераций);

varargout{3}.Iter - количество итераций;

varargout{3}.NbVal - вектор небалансов по узловым уравнениям в узлах.

В третьей ячейке массива выходных данных расположена диагностическая информация которую можно не выводить при отсутствии необходимости в таком случае массив выходных данных будет содержать лишь 2 ячейки первую и вторую.

Input – структура, которая получается в результате работы функции fInputCalcReg\_PI, состоящая из трех полей

Input.COMM - общие данные;

Input.NODE - данные по узлам;

Input.BRAN - данные по ветвям;

Пример вызова функции приведен ниже.

[NODE, RIB, DiagnRegim]=fDriveRegim\_PI(gSub(J), InputCalcReg);

Подробное описание данной структуры приведено в описании выходных данных функции fInputCalcReg\_PI, приведенного ниже.

Стоит отметить, что функция правильно работает лишь при наличии соответствующего полного окружения из управляемых функций, описание взаимосвязей приведено в документе «Описание функций».

Пример программного кода для расчета режима всех расчетных подсхем сети с последующей записью в соответствующие поля модели приведен ниже.

optSubGraph = struct('Size', 'Частн','Commt','С КА'…

,'Origin', 'Модель');

gSub= fGraphSub(g, hFN, hFV, hType, hComt, optSubGraph);

for J=1:length(gSub)-1

% создание ссылок на необходимые поля объекта модели

InputCalcReg=fInputCalcReg\_PI(gSub(J),hM);

[NODE, RIB, DiagnRegim]=fDriveRegim\_PI(gSub(J),InputCalcReg);

% запись в модель результатов расчета режима по узлам

hM.NODE(gSub(J).nod).QgR=NODE.QgR;

hM.NODE(gSub(J).nod).CurrType=NODE.CurrType;

hM.NODE(gSub(J).nod).U=NODE.U;

hM.NODE(gSub(J).nod).dU=NODE.dU;

% запись в модель результатов расчета режима по ветвям

clear NODE;

hM.BRAN(gSub(J).rib).Pn=RIB.Pn;

hM.BRAN(gSub(J).rib).Pk=RIB.Pk;

hM.BRAN(gSub(J).rib).Qn=RIB.Qn;

hM.BRAN(gSub(J).rib).Qk=RIB.Qk;

clear RIB;

end

% расчет суммарных потерь в сети

hM = CalcSum(hM);

% расчет напряжений по низкой стороне трансформаторов

hM = CalcUT(hM);

Функция fDriveRegim может быть применена в составе программного комплекса по оптимизации точек разреза, пример простейшего скрипта, использующего описываемую функцию для расчета экономичного потокораспределения, а также для последовательного переноса точки разреза на линии с двухсторонним питанием с расчетом режима в каждом положении, приведен ниже.

% создание ссылок на необходимые поля объекта модели для экономичного

% режима (X=0, все точки разреза замкнуты)

InputCalcReg\_Test=fInputCalcReg\_Ek(gSub(1),hM);

% запуск функции расчета режима

[Scalc, Ucalc]=fDriveRegim\_PI(gSub(1),InputCalcReg\_Test); %#ok<ASGLU>

% расчет величины модуля полной мощности

modS=abs((Scalc.n+Scalc.k)./2);

% определение потерь в экономичном режиме

dSsumEk=sum(abs(real(Scalc.n)-real(Scalc.k)));

dSsum(1:length(Rib))=0;

% перебор положений точек разреза по линии

for I=1:length(Rib)

% снятие разреза с предыдущего участка

if I~=1

CmStS(I-1)=2;

end

% установка разреза на текущий участок

CmStS(I)=1;

hComt = @(I)[CmStS(I) CmStF(I)];

% разделение на расчетные подсхемы при текущем положении разреза

optSubGraph = struct('Size', 'Частн','Commt','С КА','Origin', 'Граф');

gSub= fGraphSub(g, hFN, hFV, hType, hComt, optSubGraph);

for J=1:length(gSub)-1

% создание ссылок на необходимые поля объекта модели для реального

% режима (X=0)

InputCalcReg=fInputCalcReg(gSub(J),hM);

% расчет реального режима при текущем положении разреза

[Scalc, Ucalc]=fDriveRegim\_PI(gSub(J),InputCalcReg);

% определение суммарных потерь по линии

% при текущем положении точек разреза

dSsum(I)=dSsum(I)+sum(abs(real(Scalc.n)-real(Scalc.k)));

end

end

**6.5 Функция формирования исходных данных для расчета режима fInputCalcReg\_PI**

Функция осуществляет создание handle функций на параметры расчетной схемы с подменой индексации с используемой внутри графа на индексацию, используемую внутри модели. Вызов функции:

InputReg = fInputCalcReg\_PI(g, hM)

g - граф расчетной подсхемы в виде объекта класса CGraph;

hM - модель рассчитываемой распределительной сети класса CModelRS;

InputReg - структура ссылок на поля модели:

InputReg.COMM - структура общих данных;

InputReg.NODE - структура ссылок на данные о узлах;

InputReg.BRAN - структура ссылок на данные о ветвях;

InputReg.COMM.Un - номинальное напряжение

InputReg.COMM.TolFun - точность расчета

InputReg.NODE.Uu - данные о модуле напряжений

InputReg.NODE.dUu - данные о угле напряжения

InputReg.NODE.Pn - данные о активной нагрузке

InputReg.NODE.Pg - данные о активной генерации

InputReg.NODE.Qn - данные о реактивной нагрузке

InputReg.NODE.Qg - данные о реактивной генерации

InputReg.NODE.Qmin - данные о нижнем пределе реактивной мощности для PU-узла

InputReg.NODE.Qmax - данные о верхнем пределе реактивной мощности для PU-узла

InputReg.NODE.Type - данные о типе узла данные о ветвях

InputReg.BRAN.R - величина активного сопротивления ветви

InputReg.BRAN.X - величина реактивного сопротивления ветви

InputReg.BRAN.Pxx - величина активной мощности холостого хода ветви

InputReg.BRAN.Qxx - величина реактивной мощности холостого хода ветви

InputReg.BRAN.kt - величина коэффициента трансформации

InputReg.BRAN.Type - данные о типе ветви

**6.6 Функция формирования расчетной подсхемы для расчета режима fGraphSub**

Функция fGraphSub выполняет функцию декомпозии схемы сети на расчетные подсхемы, а также первоначальной подготовкой графа схемы для правильной работы всех функций, использующих данный граф (удаление не связанных со схемой узлов и ветвей и т.д.). Функция работает с графами, представленными в виде объектов класса CGraph. Работу функции можно настраивать при помощи ввода опций. Так, настраивается учет связанности схемы через центр питания и положения коммутационных аппаратов. При неучете связанности схемы через центр питания и неучете положения коммутационных аппаратов – создается подсхема для расчета экономичного потокораспределения, при неучете связанности через центра питания и учете положений коммутационных аппаратов – производится обычный расчет режима. Учет связанности через центр питания может быть необходим для решения ряда особых задач таких как, распределение нагрузки головного трансфоматора между узлами и др.

Также настраивается опорная нумерация подсхемы (может опираться на номера элементов в графе схемы, а может на номера элементов в модели).

Форма вызова функции и описание входных/выходных данных приведено ниже.

gSub = fGraphSub(g, hFN, hFV, hTypeN, hCommt, options)

gSub - выходной массив объектов класса CGraph подсхем

g - граф разделяемой схемы( объект класса CGraph)

hFN - handle ссылки на начало ветвей в модели сети

hFV - handle ссылки на концы ветвей в модели сети

hTypeN - Типы узлов в формате nTip

hCommt - состояние коммутационных аппаратов в формате cmStat

options - опции регулирующие способ разделения графа

options.Size - опция определяет учет или неучет связанности ветвей через центры питания

«Частн» - выделение подграфов без учета связанности через центр питания;

«Полн» - выделение подграфов с учетом связанности через центр питания;

options.Commt - опция определяет учет или неучет положения коммутационных аппаратов при построении подграфов.

«Без КА» - положение всех коммутационных аппаратов за исключением отключенных с запретом принимаем «включен»;

«С КА» - учитывается положение комутационных аппаратов;

options.Origin - опция определяет на какие номера участков и узлов опирается функция при построении подграфов.

«Модель» - номера элементов принимаются из модели;

«Граф» - номера элементов принимаются из опорного графа;

Результатом работы функции является массив объектов типа CGraph, притом последний элемент массива всегда или пустой или включает, при наличии, те узлы и ветви, которые не соединены с основной частью схемы (путем провеки на пустоту последнего элемента можно контролировать наличие отключенных элементов и, следовательно, правильность ввода данных о схеме).

Пример вызова функции и форма задания опции для случая без учета связности через центр питания с учета положения коммутационных аппаратов приведен ниже.

optSubGraph = struct('Size', 'Частн','Commt','С КА','Origin', 'Модель');

gSub= fGraphSub(g, hFN, hFV, hType, hComt, optSubGraph);

**6.7 Функция формирования контуров fGraphKont**

Функция fGraphKont выполняет функцию выделения контуров 3-х типов из расчетной подсхемы полученной в функции fGraphSub. Типы контуров это: «ЦП-ЦП» - контур между 2-мя центрами питания, «ЦП-ИГ» - контур между центром питания и источником генерации, а также контуры типа «цикл» - замкнутые кольца внутри графа подсхемы. Функция используется внутри fDriveRegim и у пользователя нет прямой необходимости работы с ней, однако она может полезна при проектировании иных систем в Matlab, которые будут работать с графами. Функция работает с графами, представленными в виде объектов класса CGraph. Форма вызова функции и описание входных/выходных данных приведено ниже.

[gKontBU, gKontPU, gCycle] = fGraphKont(g, hFN, hFV, TypeN)

gKontBU - выходной массив типа CGraph, содеражащий контура типа ЦП-ЦП, а

также контуры, созданные линиями, отходящими от одного ЦП;

gKontPU - выходной массив типа CGraph, содержащий контуры типа ЦП-ИГ

%(источник генерации);

gCycle - выходной массив типа CGraph, содержащий циклы;

g - объект CGraph, содержащий связанную подсхему, из которой будет происходить выделение контуров;

hFN - handle ссылки на начало ветвей в модели сети;

hFV - handle ссылки на концы ветвей в модели сети;

TypeN - Типы узлов в формате nTip.

Nop - (резервный) номер опорного центра питания;

Результатом работы функции является три массива объектов типа CGraph. Пример вызова функции приведен ниже.

[gKont.BU, gKont.PU, gKont.Cycle] = fGraphKont(g, [], [], Input.NODE.Type, Nop);

При оставлении пустым полей hFN и hFV для нумерации элементов в полученых графах контуров будет использоваться нумерация подсхемы (под нумерацией понимаются индексы, получаемые из графа по запросу вида g.nod).

**6.8 Функция вывода данных в файл Excel fExcelOut**

Функция осуществляет вывод исходных данных в файл Excel с заданным названием из заданного handle на объект, представляющего модель электрической сети. Запись осуществляется на 3 листа на первом листе хранятся данные об узлах, на втором листе – данные о ветвях. На 3-ем интегральные параметры

Форма:

StatePlot = fExcelOut(hM, filename)

hM – handle на объект класса CModelRS;

filename – строка с названием файла для вывода результатов;

StatePlot – флаг успешности записи в файл (1 – успешная запись на все листы, 0 – запись неуспешна хотя бы на один из листов).

Пример вызова функции приведен ниже.

xlsOut = '\Excel Data\Результаты';

StatePlot = fExcelOut(hM, xlsOut);

if StatePlot==1

display(['Расчет завершен. Результаты расчетов приведены в xls файле ', xlsOut]);

else

warning(['неудачная запись в файл результатов расчета режима ',xlsOut]);

end

**6.9 Класс для работы с исходными данными CData**

Класс CData осуществляет ввод исходных данных из Excel-файла заданного формата. Публичные методы включают чтение и запись содержимого, которое производится аналогично с CModelRS.

При необходимости использования другого формата исходных данных данный класс следует заменить пользовательским при сохранении названий полей.

hDt.NODE.Nn1 - Пользовательские номера узлов

hDt.NODE.Type - Типы узлов (BU, PU или PQ)

hDt.NODE.Pn - Активная мощность нагрузки

hDt.NODE.Qn - Реактивная мощность нагрузки

hDt.NODE.Pg - Активная мощность генерации

hDt.NODE.Qg - Реактивная мощность генерации

hDt.NODE.Qmin - Минимальный предел реактивной мощности

hDt.NODE.Qmax - Максимальный предел реактивной мощности

hDt.NODE.Uu - Уставка напряжения генератора (ЦП)

hDt.NODE.dUu - Угол напряжения (ЦП)

hDt.BRAN.Nb1 - Пользовательские номера ветвей

hDt.BRAN.NbSt - Номер узла начала

hDt.BRAN.NbF - Номер узла конца

hDt.BRAN.CmTpS - Тип КА в начале

hDt.BRAN.CmStS - Состояние КА в начале

hDt.BRAN.CmTpF - Тип КА в конце

hDt.BRAN.CmStF - Состояние КА в конце

hDt.BRAN.Type - Тип ветви

hDt.BRAN.R - Активное сопротивление

hDt.BRAN.X - Реактивное сопротивление

hDt.BRAN.Pxx - Активная мощность холостого хода

hDt.BRAN.Qxx - Реактивная мощность холостого хода

hDt.BRAN.kt - Коэффициент трансформации

hDt.COMM.TolFun - Точность вычислений

hDt.COMM.Un - Номинальное напряжение, кВ

Форма вызова конструктора приведен ниже

hDt = Cdata(xlsFile, options)

xlsFile - строковое имя файла MsExcel с данными по схеме сети;

options - структура параметров, необходимых для развётрывания модели;

Поля структуры options:

options.NodeSheet - номер листа в файле MsExcel с данными по узлам;

options.BranchSheet - номер листа в файле MsExcel с данными по ветвям;

options.KatLinesSheet - номер листа в файле MsExcel с каложными данными по линиям;

options.KatTransSheet - номер листа в файле MsExcel с каложными данными по трансформаторам

options.Method - метод ввода исходных данных (резервное поле – так как на момент реализации возможен ввод только из xls (тип 0)).

Пример вызова конструктора

inopts = struct('Method', methTip (0),'NodeSheet', 1, 'BranchSheet',2, 'TolFun', 1\*10^-2);

% ввод имени Excel файла

xlsFile = '\Excel Data\Исходные\_данные\_5';

oDt=CData(xlsFile, inopts);