

5. Составление скриптов на языке *Lua*

В разделе *Help* программы *FEMM* содержатся инструкции по написанию скриптов на языке *Lua*.

5.1. Из Энциклопедии языков программирования:

Lua (от португальского “луна”) — облегченный скриптовый язык с расширяемой семантикой.

Lua был создан и поддерживается представителями *Pontifical Catholic University* Рио-де-Жанейро. У него нет официального стандарта, и стандартом считается описание в руководстве пользователя.

В настоящее время *Lua* является самым популярным скриптовым языком в индустрии игр и используется в ряде приложений в других предметных областях.

5.2. Когда выгодно применять *Lua*-программы?

Если надо один раз решить какую-либо конкретную полевую задачу при заданных и не изменяющихся исходных данных, определяющих источники поля, граничные условия и свойства сред, применение *Lua*-программ вряд ли оправдано. В этом случае проще и быстрее решить задачу с помощью тех средств пре- и постпроцессоров *FEMM*, которые были описаны в предыдущих разделах. Если же предстоит решить *серию* полевых задач с изменяемыми определенным образом исходными данными – *размерами, взаимными положениями сердечников, свойствами сред, токами в обмотках, граничными условиями*, – тогда применение *Lua*-программ может значительно облегчить выполнение численного эксперимента, по существу, автоматизировав построения и вычисления.

5.3. Несколько советов

5.3.1. Впервые приступая к составлению *Lua*-скрипта (*script* – сценарий), внимательно ознакомьтесь со списком *Lua*-команд, имеющимся в приложении *Help*. В помощь тем, кто недостаточно знаком с английскими тер-

минами, сделан перевод основных слов, использованных в именах команд (см. раздел **5.4.**).

5.3.2. Удобнее писать сценарий не в *Lua*-консоли непосредственно, а в специальном *Word*-овском файле, копируя из него команды в *Input*-окно консоли.

5.3.3. При копировании и вставке команд будьте предельно внимательны – следите за тем, чтобы при этом во вставляемых в консоль командах не появлялось лишних пробелов или каких-либо других искажений.

5.3.4. Составляя большой сценарий, вводите в *Lua*-консоль по малому его фрагменту – всего по нескольким простым командам, сразу же проверяя их исполнение.

5.3.5. Блоки сценария можно отлаживать по частям, поочередно, возвращаясь при обнаружении ошибки не к началу сценария, а к началу отлаживаемого блока.

5.3.6. Обращайте внимание на размерность углов. В версиях *FEMM* до 4.2 значения углов и в алгебраических выражениях, и в выражениях аргументов команд подставляются в градусах, в то время как в поздних версиях *FEMM* значения углов в алгебраических выражениях подставляются в радианах.

5.4. Перевод английских слов, использованных в *Lua*-командах

Список слов составлен не в алфавитном порядке, а в той очередности, в которой эти слова встречаются в списке команд в разделе *Help* программы *FEMM*. Даются те переведенные значения, которые соответствуют смыслу образованных с их помощью команд.

clear - очистить

create -создать

open -открыть

pause – пауза

print – печатать

prompt – напоминание, подсказка

quit – закрыть

show – показать

add – добавить

node - узел

arc – дуга

select – выбрать

label – метка (обозначение блока)

prop – сокращение от *property* – свойство

mesh – ячейка (конечно-элементной сетки)

size – размер

circuit – цепь (в которую входят элементы модели)

magdirection – (от *magnetization direction*) – направление намагничен-

НОСТИ

turns – витки (число витков)

set – установить, назначить

hide – скрыть

deg – градусы

probdef - (от *problem definition*) – определение (данных) задачи

frequency – частота

units – единицы измерения

precision – точность

depth – глубина (имеется в виду размер в направлении, перпендикулярном плоскости модели, измеренный в заданных единицах длины)

minangle – (от *minimum angle*) – минимальный (допустимый) угол (конечного элемента)

analyze – запустить процедуру решения задачи

load – загрузить

solution – решение

purge – очистить

copy – копировать

editaction – параметр, определяющий тип объекта, к которому применяется команда (0 – узел, 1 – отрезок прямой, 2 – метка блока, 3 – дуга 4 – группа объектов)

rotate – вращать

translate – линейно переместить

move – передвинуть

scale – масштабировать (изобразить в другом масштабе)

factor – коэффициент

mirror – зеркально отразить

editmode – программа устанавливает тип объектов для последующих команд редактирования, в которых тип объекта не указан явно с помощью параметра *editaction*

snap – снимок, копия изображения, соответствующая определенному моменту времени

grid – решетка (множество узловых точек, нанесенных на область задачи, отстоящих по горизонтали и по вертикали друг от друга на равные расстояния)

density – плотность (может пониматься как *flux density* – плотность потока, т.е. индукция, и как *grid density* – плотность решетки)

refresh – обновить

width – ширина

height – высота

5.5. Примеры

5.5.1. Пример 1. Задача: построить равномерное магнитное поле с заданным значением индукции в гладком воздушном зазоре между двумя плоскими сердечниками при наличии в зазоре стального бруса

Составим Lua-сценарий для задачи, рассмотренной в разделе «Простейшие полевые задачи». На рис. 5.1., повторяющем рис. 1.5. этого раздела, приведен результат решения.

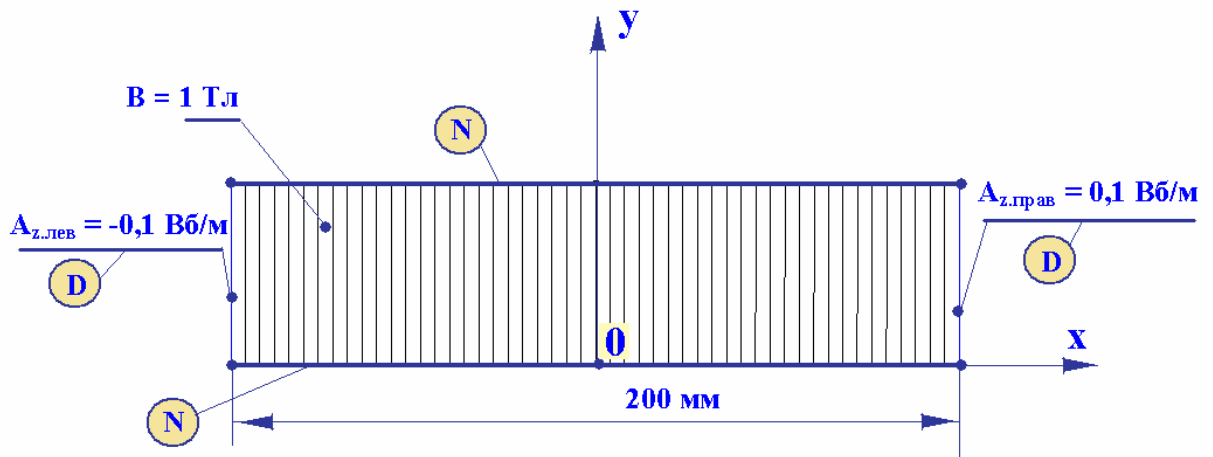


Рис. 5.1. Граничные условия и результат решения задачи о равномерном поле в плоском зазоре

Несколько усложним задачу, поместив в зазор стальной брус с прямоугольным поперечным сечением и относительной магнитной проницаемостью, равной 1000.

Начать работу можно следующим образом.

Открыть окно программы *FEMM*, открыть консоль (окно ввода) *Lua*, ввести команды: создания *fem*-файла, сохранения *fem*-файла, модификации (изменения параметров) *fem*-файла (рис. 5.2.).

Соответствующие команды (предварительно, как уже было сказано, их удобнее написать в *Word*-файле):

```
create("c:\\MyFEMMfiles\\Uniform_gap_field.fem",0)
mi_saveas("c:\\MyFEMMfiles\\Uniform_gap_field.fem")
mi_probdef(0,"millimeters","planar",1e-8,1,20)
```

Обратите внимание на то, что в адресах файлов следует писать двойные слэши.

Нулевое значение второго параметра первой команды означает, что создается файл магнитоэлектростатической задачи (электростатической задаче соответствует единица)

После выполнения этих команд появляется рабочее окно программы *FEMM* и в правильности задания характеристики созданного файла *Uniform_gap_field.fem* можно убедиться, нажав клавишу *Problem*.

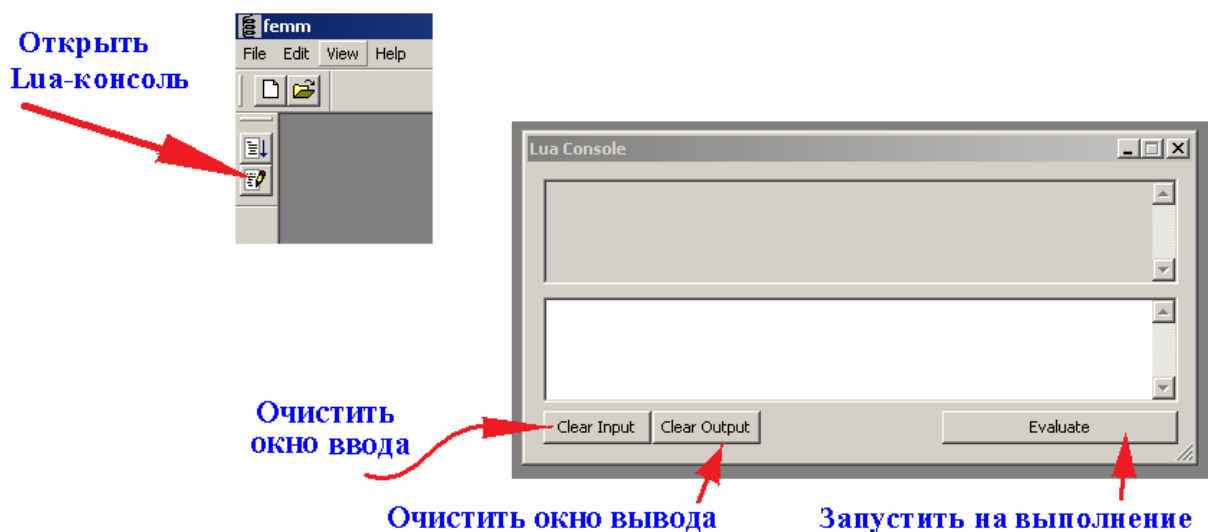


Рис. 5.2. Начало работы с *Lua*-сценарием

Далее следует создать и открыть текстовый файл для записи результатов:

```
outfile = "c:\\MyFEMMfiles\\Uniform_gap_field_res_1.txt "  
handle=openfile(outfile,"w")
```

Определим граничные условия и свойства сред (воздух и сталь):

```
mi_addboundprop("left",-0.1,0,0,0,0,0,0,0)  
mi_addboundprop("right",0.1,0,0,0,0,0,0,0)  
mi_addmaterial("air")  
mi_addmaterial("iron_1000",1000,1000)
```

Результаты исполнения этих команд можно посмотреть, нажав клавишу *Properties*.

Далее следуют графические построения – задаются координаты угловых точек и проводятся соединяющие их отрезки прямых:

```
mi_addnode(0,0)
mi_addnode(-100,0)
mi_addnode(100,0)
mi_addnode(-100,50)
mi_addnode(100,50)
mi_addsegment(-100,0,100,0)
mi_addsegment(-100,50,100,50)
mi_addsegment(-100,0,-100,50)
mi_addsegment(100,0,100,50)
mi_addnode(-20,10)
mi_addnode(20,10)
mi_addnode(-20,30)
mi_addnode(20,30)
mi_addsegment(-20,10, 20,10)
mi_addsegment(-20,30, 20,30)
mi_addsegment(-20,10, -20,30)
mi_addsegment(20,10, 20,30)
mi_zoomnatural()
```

Последняя команда позволяет увидеть результаты выполненных построений в «натуральном» масштабе на экране компьютера.

Следующие действия – присвоение граничных условий:

```
mi_selectsegment(-100,1)
mi_setsegmentprop("left")
mi_clearselected()
mi_selectsegment(100,1)
mi_setsegmentprop("right")
mi_clearselected()
```

и присвоение имен и свойств блоков:

```
mi_addblocklabel(-80,40)
mi_selectlabel(-80,40)
mi_setblockprop("air",0,1,"0",0,0,0)
mi_clearselected()
mi_addblocklabel(-10,20)
```

```
mi_selectlabel(-10,20)  
mi_setblockprop("iron_1000",0,1,"0",0,0,0)  
mi_clearselected()  
mi_refreshview()
```

Левой и правой границам области задачи присваиваются условия Дирихле – задаются заранее вычисленные значения векторного магнитного потенциала, обеспечивающие в зазоре при отсутствии стального стержня равномерное магнитное поле с индукцией 1 Тл. Задание условий Дирихле в рассматриваемом примере означает, что левую и правую границы мы полагаем совпадающими с силовыми линиями поля. Это положение, верное при отсутствии в зазоре ферромагнитных тел или проводов с токами, в данном случае является допущением.

На верхней и нижней границах области задачи граничные условия явно можно не задавать, т.к. однородные условия Неймана в программе *FEMM* задаются по умолчанию.

После того, как полевая задача полностью сформулирована, можно приступать к ее решению:

```
mi_analyze()
```

и загрузке результатов:

```
mi_loadsolution()
```

Для обработки результатов решения предназначены команды пост-процессора программы *FEMM*:

Для более наглядного представления картины поля можно, например, изменить число силовых линий в том же диапазоне значений векторного магнитного потенциала:

```
mo_showcontourplot(100,-0.1,0.1)
```


После этого картина поля, представляющая решение задачи, выглядит следующим образом:

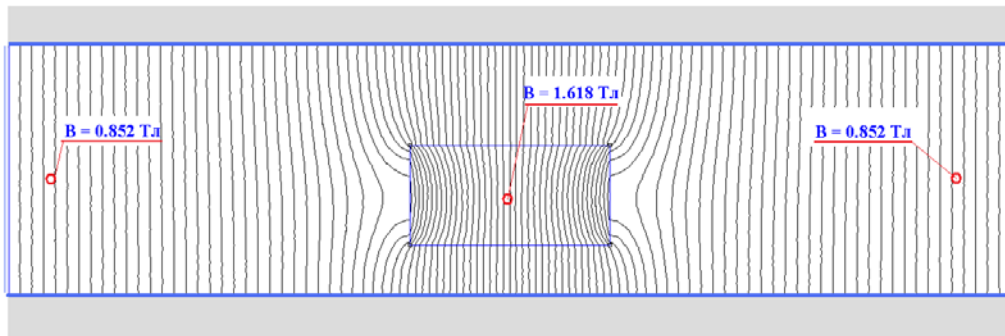


Рис. 5.3. Картина поля, представляющая решение задачи

Графики распределения индукции вдоль определенным образом выбранных контуров (в том числе и для последующего гармонического анализа) могут быть получены с помощью следующих команд:

Выделение контура на поверхности нижнего сердечника (с направлением слева направо) для считывания результатов:

```
mo_addcontour(-100,1)
mo_addcontour(100,1)
```

Запись результатов в текстовый файл (число считываемых значений в данном случае равно 2048):

```
mo_makeplot(2,2048,"c:\\MyFEMMfiles\\Uniform_gap_field_res_1.txt",1)
```

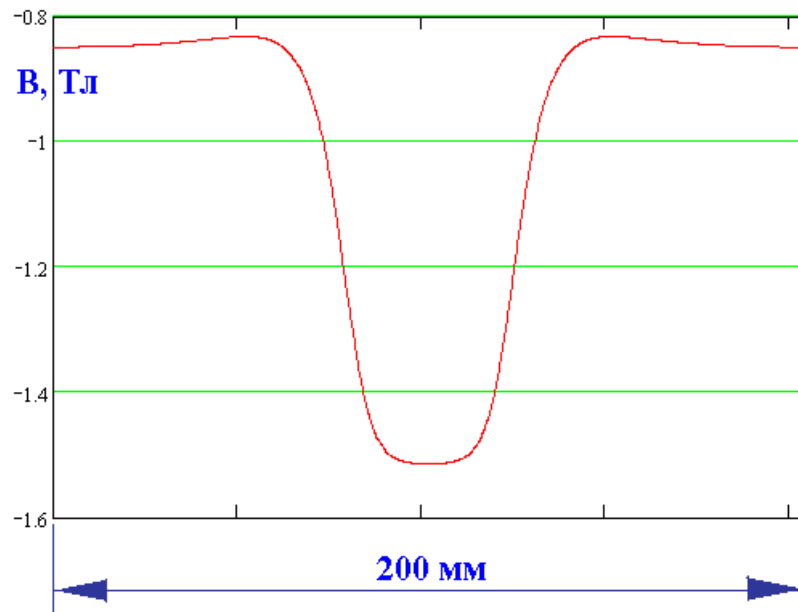


Рис. 5.4. Распределение индукции вдоль поверхности нижнего сердечника. Отрицательные значения означают, что векторы индукции направлены сверху вниз

5.5.1. Краткий анализ

Как уже было замечено, использование *Lua*-сценария для одноразового решения магнитостатической полевой задачи не дает выигрыша ни при формировании модели, ни в скорости решения, ни при обработке результатов. Однако, если в только что рассмотренной задаче мы попытаемся выяснить, как влияют размеры вносимого в зазор стального стержня на характер изменения магнитного поля, или как влияют размеры области задачи на погрешность, вызванную допущением о совпадении левой и правой границ с силовыми линиями результирующего поля, то мы очень быстро убедимся в том, что наличие легко редактируемого сценария существенно облегчает поиск ответов на поставленные вопросы.

Очевидно, что изменяемые размеры или свойства материалов удобнее задавать в виде параметров, присваивая им значения в специально выделен-

ном блоке *Lua*-программы, помещаемом в ее начале. В следующем примере программа организована именно таким образом.

5.5.3. Пример 2. Оптимизация размеров поперечного сечения магнитоэлектрической синхронной машины

Составим Lua-сценарий для формирования конечно-элементной модели поперечного сечения трехфазной четырехполюсной синхронной машины небольшой мощности с поверхностными постоянными магнитами на роторе.

Начало программы – такое же, как в предыдущем примере:

```
create("c:\\MyFEMMfiles\\PMSM_cross_section.fem",0)
mi_saveas("c:\\MyFEMMfiles\\PMSM_cross_section.fem")
mi_probdef(0,"millimeters","planar",1e-8,1,20)
```

Для записи результатов откроем два текстовых файла – один для данных о распределении индукции в воздушном зазоре и второй – для значений электромагнитного момента

```
outfile_1 = "c:\\MyFEMMfiles\\PMSM_res_Bdlt.txt"
handle=openfile(outfile_1,"w")
outfile_2 = "c:\\MyFEMMfiles\\PMSM_res_Trq.txt"
handle=openfile(outfile_2,"w")
```

При моделировании полного поперечного сечения машины в качестве граничных условий нам понадобится только однородное условие Дирихле, которое в этом случае задается на внешней границе области задачи (или на всех внешних границах, если область задачи не является односвязной)

```
mi_addboundprop("outer",0,0,0,0,0,0,0,0)
```

Кроме воздуха зададим материал сердечников статора и ротора, в качестве которого для начала выберем сталь с относительной магнитной проницаемостью, равной 1000.

```
mi_addmaterial("air")
mi_addmaterial("iron_1000",1000,1000)
```

В качестве материала постоянных магнитов выберем сплав *NdFeB* с максимальной энергией 40 МГсЭ (свойства этого материала приведены в библиотеке материалов программы *FEMM*).

```
mi_addmaterial("PM",1.049,1.049,979000,0,0,0.667,0)
```

Плотности тока на участках модели, соответствующих начальным и конечным сторонам катушечных сторон фаз обмотки статора, зададим вначале для момента времени, в который ток фазы *A-X* положителен и равен амплитудному значению:

```
JA=2
mi_addmaterial("A",1,1,0,JA)
mi_addmaterial("X",1,1,0,-JA)
mi_addmaterial("B",1,1,0,-JA*0.5)
mi_addmaterial("Y",1,1,0,JA*0.5)
mi_addmaterial("C",1,1,0,-JA*0.5)
mi_addmaterial("Z",1,1,0,JA*0.5)
```

Блок исходных данных (размеры даны в мм)

```
pi=3.14159
-- число пазов статора
Z1 = 24
-- радиус расточки статора
rstat=100
--ширина зубца статора
bz=7
-- глубина паза статора
```

```

hz=15
-- угол, соответствующий одному зубцовому делению статора
alphaz=360/Z1
--угол, соответствующий ширине одного зубца статора
alphabz=(bz/(pi*rstat))*180
-- угол, соответствующий одному пазу статора
alphasl=alphaz-alphabz
-- раскрытие (шлиц) паза статора
bs=5
-- глубина шлица паза статора
hs=1
--высота клиновой зоны паза статора
hw=4.5
--толщина ярма статора
ha=hz*0.8
--воздушный зазор
airgap=1
-- толщина магнитов ротора
hPM=10
-- толщина втулки ротора
hhub=15
-- угол, соответствующий ширине магнита
alphaPM=80

```

При построении окружности расточки статора следует задать достаточно малый угловой размер сегментов, из которых составляется дуга (последний параметр в соманде *mi_addarc*)

```

mi_addnode(0,0)
mi_addnode(-rstat,0)
mi_addnode(rstat,0)
mi_addarc(rstat,0,-rstat,0,180,1)
mi_addarc(-rstat,0,rstat,0,180,1)

```

В рассматриваемом здесь примере при графических построениях целесообразно предварительно вычислить некоторые характерные углы и координаты некоторых характерных точек поперечного сечения машины (см. рис. 5.3.)

```

dlt0=(bz*0.5)/sin(alphaz*0.5)

```

```

x2=-rstat-hs
y2=-bs*0.5
x3=x2-hw
y3=-((-x3-dlt0)*tan(alphaz*0.5))
x4=-((rstat+hz)*cos(alphaz*0.5))
y4=-((-x4-dlt0)*tan(alphaz*0.5))
x5=x4
y5=-y4
x6=x3
y6=-y3
x7=x2
y7=-y2

```

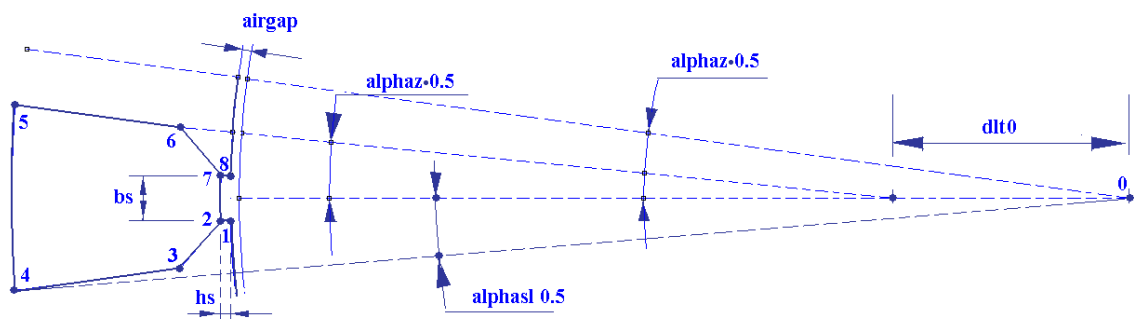


Рис. 5.5. К расчету координат характерных узлов поперечного сечения

```

mi_addnode(x2,y2)
mi_addnode(x3,y3)
mi_addnode(x4,y4)
mi_addnode(x5,y5)
mi_addnode(x6,y6)
mi_addnode(x7,y7)

```

```

mi_addsegment(x2,y2,x3,y3)
mi_addsegment(x3,y3,x4,y4)
mi_addarc(x5,y5,x4,y4,alphasl,1)
mi_addsegment(x5,y5,x6,y6)
mi_addsegment(x6,y6,x7,y7)

```

```

mi_selectnode(x2,y2)
mi_selectnode(x7,y7)
mi_copytranslate(hs*2,0,1,0)

```

```
mi_addsegment(x2,y2,x2+hs*2,y2)
mi_addsegment(x7,y7,x7+hs*2,y7)
mi_selectnode(x2+hs*2,y2)
mi_selectnode(x7+hs*2,y7)
mi_deleteselected()
```

```
mi_addarc(x7,y7,x2,y2,alpha1,1)
```

```
mi_selectnode(-rstat,0)
mi_deleteselected()
```

```
mi_zoomnatural()
```

Последняя команда позволяет посмотреть результаты выполненных построений. Для временной остановки процесса формирования модели здесь можно воспользоваться командой *pause*.

Убедившись в том, что паз построен правильно, его контуры следует скопировать ($Z1 - 1$) раз:

```
mi_selectnode(-rstat,-bs*0.5)
mi_copyrotate(0,0,-alphaz,1,0)
```

```
mi_selectarcsegment(rstat,0)
mi_deleteselected()
mi_selectarcsegment(rstat,0)
mi_deleteselected()
```

```
mi_selectsegment(-rstat,-bs*0.5)
mi_selectsegment(x2,y2)
mi_selectsegment(x4,y4)
mi_selectsegment(x5,y5)
mi_selectsegment(x7,y7)
mi_selectsegment(-rstat,bs*0.5)
mi_copyrotate(0,0,-alphaz,Z1-1,1)
```

```
mi_selectarcsegment(x4,y4)
mi_selectarcsegment(x2,y2)
mi_selectarcsegment(-rstat,bs*0.5)
mi_copyrotate(0,0,-alphaz,Z1-1,3)
```

При построении внешней окружности ярма размер образующих сегментов следует увеличить с тем, чтобы конечно-элементная сетка не была чрезмерно подробной там, где в этом нет необходимости.

```
mi_addnode(-rstat-hz-ha,0)
mi_addnode(rstat+hz+ha,0)
mi_addarc(rstat+hz+ha,0,-rstat-hz-ha,0,180,5)
mi_selectarcsegment(rstat+hz+ha,0)
mi_setarcsegmentprop(5,"outer")
mi_copyrotate(0,0,180,1,3)
mi_zoomnatural()
```

Закончим построение контуров блоков ротора и назначим граничное условие Дирихле на внутренней границе области задачи:

```
mi_addnode(-rstat+airgap,0)
mi_addnode(rstat-airgap,0)
mi_addnode(-rstat+airgap+hPM,0)
mi_addnode(rstat-airgap-hPM,0)
mi_addnode(-rstat+airgap+hPM+hhub,0)
mi_addnode(rstat-airgap-hPM-hhub,0)

mi_addarc(-rstat+airgap,0,rstat-airgap,0,180,1)
mi_addarc(rstat-airgap,0,-rstat+airgap,0,180,1)
mi_addarc(-rstat+airgap+hPM,0,rstat-airgap-hPM,0,180,5)
mi_addarc(rstat-airgap-hPM,0,-rstat+airgap+hPM,0,180,5)
mi_addarc(-rstat+airgap+hPM+hhub,0,rstat-airgap-hPM-hhub,0,180,5)

mi_selectarcsegment(-rstat+airgap+hPM+hhub,0)
mi_setarcsegmentprop(5,"outer",0,1)
mi_copyrotate(0,0,180,1,3)
mi_zoomnatural()
```

Назначение имен блоков.

Для наглядности в рассматриваемом примере эта часть сценария написана для конкретных значений чисел q и $Z1$, равных 2 и 24. Формализовать эту часть программы можно, организовав соответствующие циклы по пронумерованным именам блоков и углу поворота $alpha_z * n$.


```
mi_addblocklabel(-rstat-hz*0.5,0)
mi_selectlabel(-rstat-hz*0.5,0)
mi_setblockprop("A",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*2),(rstat+hz*0.5)*sin(alphaz*2))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*2),(rstat+hz*0.5)*sin(alphaz*2))
mi_setblockprop("Z",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*4),(rstat+hz*0.5)*sin(alphaz*4))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*4),(rstat+hz*0.5)*sin(alphaz*4))
mi_setblockprop("B",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*6),(rstat+hz*0.5)*sin(alphaz*6))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*6),(rstat+hz*0.5)*sin(alphaz*6))
mi_setblockprop("X",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*8),(rstat+hz*0.5)*sin(alphaz*8))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*8),(rstat+hz*0.5)*sin(alphaz*8))
mi_setblockprop("C",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*10),(rstat+hz*0.5)*sin(alphaz*10))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*10),(rstat+hz*0.5)*sin(alphaz*10))
mi_setblockprop("Y",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*12),(rstat+hz*0.5)*sin(alphaz*12))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*12),(rstat+hz*0.5)*sin(alphaz*12))
mi_setblockprop("A",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*14),(rstat+hz*0.5)*sin(alphaz*14))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*14),(rstat+hz*0.5)*sin(alphaz*14))
mi_setblockprop("Z",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*16),(rstat+hz*0.5)*sin(alphaz*16))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*16),(rstat+hz*0.5)*sin(alphaz*16))
mi_setblockprop("B",0,1)
```

```
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*18),(rstat+hz*0.5)*sin(alphaz*18))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*18),(rstat+hz*0.5)*sin(alphaz*18))
mi_setblockprop("X",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*20),(rstat+hz*0.5)*sin(alphaz*20))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*20),(rstat+hz*0.5)*sin(alphaz*20))
mi_setblockprop("C",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
```

```
mi_addblocklabel((-rstat-hz*0.5)*cos(alphaz*22),(rstat+hz*0.5)*sin(alphaz*22))
mi_selectlabel((-rstat-hz*0.5)*cos(alphaz*22),(rstat+hz*0.5)*sin(alphaz*22))
mi_setblockprop("Y",0,1)
mi_copyrotate(0,0,-alphaz,1,2)
mi_zoomnatural()
```

Назначение имен блоков ротора:

```
mi_addblocklabel((-rstat-hz*0.5-ha),0)
mi_selectlabel((-rstat-hz*0.5-ha),0)
mi_setblockprop("iron_1000",0,1)
mi_addsegment(-rstat+airgap,0,-rstat+airgap+hPM,0)
mi_addsegment(rstat-airgap,0,rstat-airgap-hPM,0)
mi_selectsegment(-rstat+airgap,0)
mi_selectsegment(rstat-airgap,0)
mi_copyrotate(0,0,-alphaPM,1,1)
mi_selectsegment(-rstat+airgap,0)
mi_selectsegment(rstat-airgap,0)
mi_selectsegment((-rstat+airgap)*cos(alphaPM),(rstat-airgap)*sin(alphaPM))
mi_selectsegment((rstat-airgap)*cos(alphaPM),(rstat-airgap)*(-sin(alphaPM)))
mi_copyrotate(0,0,-90,1,1)
```

```
V1=(-rstat+airgap+hPM*0.5)
mi_addblocklabel(V1*cos(alphaPM*0.5),(-V1)*sin(alphaPM*0.5))
mi_selectlabel(V1*cos(alphaPM*0.5),(-V1)*sin(alphaPM*0.5))
mi_setblockprop("PM",0,1,0,140,1,0)
mi_copyrotate(0,0,180,1,2)
V2=alphaPM*0.5+90
mi_addblocklabel(V1*cos(V2),(-V1)*sin(V2))
mi_selectlabel(V1*cos(V2),(-V1)*sin(V2))
mi_setblockprop("PM",0,1,0,140-90+180,1,0)
```

```

mi_copyrotate(0,0,180,1,2)
V3=(90-alphaPM)*0.5
mi_addblocklabel(V1*cos(V3),(-V1)*sin(-V3))
mi_selectlabel(V1*cos(V3),(-V1)*sin(-V3))
mi_setblockprop("air",0,1,0,0,1,0)
mi_copyrotate(0,0,90,3,2)
V4=V1+hhub
mi_addblocklabel(V4,0)
mi_selectlabel(V4,0)
mi_setblockprop("iron_1000",0,1,0,0,1,0)
mi_clearselected()
mi_addblocklabel(-rstat-airgap*0.5,0)
mi_selectlabel(-rstat-airgap*0.5,0)
mi_setblockprop("air",0,1)
mi_clearselected()
mi_addblocklabel(-rstat*0.3,0)
mi_selectlabel(-rstat*0.3,0)
mi_setblockprop("air")
mi_clearselected()
mi_zoomnatural()

```

Перед решением полезно убедиться в правильности формирования модели визуально и, проверив свойства блоков и границ.

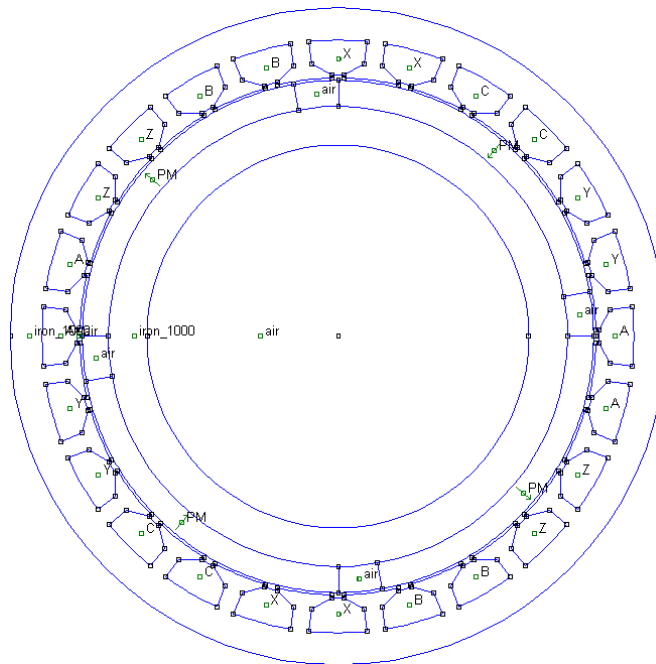


Рис. 5.6. Сформированная модель поперечного сечения машины

5.5.4. Пример 3. Расчет зависимости момента, обусловленного зубчатостью сердечника статора от углового положения ротора

В магнитоэлектрических машинах моменты, обусловленные зубчатостью сердечников, обычно вызывают целый ряд отрицательных явлений – вибрации и шум, пульсации частоты вращения. Эти моменты называют также зубцовыми (в англоязычной литературе установившееся название для них – *cogging torques*).

Для того чтобы получить зависимость зубцовых моментов от углового положения ротора, необходимо смоделировать поперечное сечение магнитоэлектрической машины достаточное число раз, поворачивая при этом все элементы модели, относящиеся к ротору, каждый раз на небольшой угол. При регулярной зубчатости на статоре, т.е. если все зубцы статора одинаковы, периодом изменения зубцовых моментов является угол, соответствующий одному зубцовому делению. В рассматриваемом примере этот период равен 15 пространственным градусам (24 зубца на статоре) и можно предположить, что при последовательном повороте элементов ротора на полградуса точность интересующей нас зависимости окажется достаточной. Поскольку, однако, вычислительная погрешность моделирования электромагнитного момента нами пока еще должным образом не оценена, имеет смысл увеличить диапазон изменения углового положения ротора до двух периодов. В этом случае о точности моделирования момента можно будет судить по повторяемости результатов. При такой постановке задачи серия численных экспериментов будет включать в себя 60 опытов моделирования.

В программе *FEMM* имеется возможность применять такие операции редактирования, как поворот и перемещение, к совокупности элементов, объединенных в *группу* (*group*). Объединение элементов в группу удобно производить на стадии формирования модели, особенно в том случае, если какая-то часть модели строится с помощью копирования повторяющейся

совокупности элементов (отрезков прямых, дуг, блоков). При формировании этой базовой совокупности следует каждому из ее элементов приписать принадлежность некоторой *группе*. При копировании принадлежность группе сохраняется и, таким образом, при необходимости изменить положение *группы*, мы можем выделить все ее элементы с помощью *одной* команды.

Из сказанного следует, что порядок выполнения серии численных экспериментов надо иметь в виду уже на стадии формирования модели.

Если же, как в нашем примере, объединение элементов в группы не было выполнено на стадии формирования модели, можно сделать это и позже, не вмешиваясь в текст готового и отлаженного сценария. Легко, однако, убедиться в том, что в этом случае сценарий выглядит менее привлекательно.

Итак, составим программу для определения зависимости зубцового момента от углового положения ротора, для рассмотренной в предыдущих двух примерах 4-полюсной магнитоэлектрической машины с поверхностными магнитами.

Объединение в группу с номером 1 *отрезков прямых*, являющихся боковыми (радиальными) границами постоянных магнитов:

```
mi_clearselected()
mi_selectsegment(-rstat+airgap+hPM,0)
mi_selectsegment((-rstat+airgap+hPM)*cos(80),(rstat-airgap-hPM)*sin(80))
mi_selectsegment((-rstat+airgap+hPM)*cos(90),(rstat-airgap-hPM)*sin(90))
mi_selectsegment((-rstat+airgap+hPM)*cos(170),(rstat-airgap-hPM)*sin(170))
mi_selectsegment((-rstat+airgap+hPM)*cos(180),(rstat-airgap-hPM)*sin(180))
mi_selectsegment((-rstat+airgap+hPM)*cos(260),(rstat-airgap-hPM)*sin(260))
mi_selectsegment((-rstat+airgap+hPM)*cos(270),(rstat-airgap-hPM)*sin(270))
mi_selectsegment((-rstat+airgap+hPM)*cos(350),(rstat-airgap-hPM)*sin(350))
mi_setsegmentprop("None",0,1,0,1)
```

То же для *внешних и внутренних дуг*, являющихся границами магнитов:

```
mi_clearselected()
r1=-rstat+airgap
```

```

r2=-rstat+airgap+hPM
r3=-r1
r4=-r2
mi_selectarcsegment(r1,0+0.1)
mi_selectarcsegment(r1*cos(90),r3*sin(90))
mi_selectarcsegment(r1*cos(180),r3*sin(180))
mi_selectarcsegment(r1*cos(270),r3*sin(270))
mi_setarcsegmentprop(1,"None",0,1)
mi_clearselected()
mi_selectarcsegment(r2,0+0.1)
mi_selectarcsegment(r2*cos(90),r4*sin(90))
mi_selectarcsegment(r2*cos(180),r4*sin(180))
mi_selectarcsegment(r2*cos(270),r4*sin(270))
mi_setarcsegmentprop(5,"None",0,1)
mi_clearselected()

mi_zoomnatural()

```

То же для *внешних и внутренних дуг*, ограничивающих промежутки между магнитами:

```

mi_selectarcsegment(r1,0-0.1)
mi_selectarcsegment(r1*cos(89),r3*sin(89))
mi_selectarcsegment(r1*cos(179),r3*sin(179))
mi_selectarcsegment(r1*cos(269),r3*sin(269))
mi_setarcsegmentprop(1,"None",0,1)
mi_clearselected()

mi_selectarcsegment(r2,0-0.1)
mi_selectarcsegment(r2*cos(89),r4*sin(89))
mi_selectarcsegment(r2*cos(179),r4*sin(179))
mi_selectarcsegment(r2*cos(269),r4*sin(269))
mi_setarcsegmentprop(5,"None",0,1)
mi_clearselected()

mi_zoomnatural()

```

То же для обозначений *блоков постоянных магнитов* (обратите внимание на то, что здесь приходится снимать выделение после модификации свойств каждого из магнитов. В противном случае все магниты окажутся намагничены одинаково – так же, как последний в этом цикле):

```

r5=-rstat+hPM*0.5
r6=-r5
mi_selectlabel(r5*cos(alphaPM*0.5),r6*sin(alphaPM*0.5))
mi_setblockprop("PM",0,1,"0",180-alphaPM*0.5,1,0)
mi_clearselected()
mi_selectlabel(r5*cos(alphaPM*0.5+90),r6*sin(alphaPM*0.5+90))
mi_setblockprop("PM",0,1,"0",180-alphaPM*0.5-90+180,1,0)
mi_clearselected()
mi_selectlabel(r5*cos(alphaPM*0.5+180),r6*sin(alphaPM*0.5+180))
mi_setblockprop("PM",0,1,"0",180-alphaPM*0.5-180,1,0)
mi_clearselected()
mi_selectlabel(r5*cos(alphaPM*0.5+270),r6*sin(alphaPM*0.5+270))
mi_setblockprop("PM",0,1,"0",180-alphaPM*0.5-270+180,1,0)
mi_clearselected()

mi_zoomnatural()

```

То же для обозначений *блоков промежутков* между магнитами:

```

mi_clearselected()
mi_selectlabel(r5*cos((90-alphaPM)*0.5),r6*(-sin((90-alphaPM)*0.5)))
mi_selectlabel(r5*cos((90-alphaPM)*0.5+90),r6*(-sin((90-alphaPM)*0.5+90)))
mi_selectlabel(r5*cos((90-alphaPM)*0.5+180),r6*(-sin((90-
alphaPM)*0.5+180)))
mi_selectlabel(r5*cos((90-alphaPM)*0.5+270),r6*(-sin((90-alphaPM)*0.5+270)))
mi_setblockprop("air",0,1,"0",180-alphaPM*0.5-270+180,1,0)

mi_zoomnatural()

```

Задание *начального* значения угла, определяющего положение ротора, *диапазона изменения* углового положения ротора и угла поворота ротора (*углового шага*) при переходе к следующему положению.

```

teta=0
teta_end=-15
teta_step=-1

```

Открытие файла для записи результатов измерения момента:

```

handle=openfile(outfile_2,"w")

```

Условие цикла:

```
while teta >= teta_end do
```

Содержание цикла:

```
mi_analyze()
mi_loadsolution()
mo_clearcontour()
mo_selectpoint(-rstat+airgap*0.5,0+0.01)
mo_selectpoint(rstat-airgap*0.5,0+0.01)
mo_selectpoint(rstat-airgap*0.5,0-0.01)
mo_selectpoint(-rstat+airgap*0.5,0-0.01)
Trq = mo_lineintegral(4)
print(teta.." "..Trq)
write(handle,Trq.."\n")
mo_close()
mi_selectgroup(1)
mi_moverotate(0,0,teta_step,4)
teta = teta + teta_step
```

Выход из цикла и окончание программы:

```
end
closefile(handle)
print("END");
```

---Результаты, выведенные на *Lua*-консоль:

```
--> 0      -0.06638789068414876
--> -1     -0.03839906469526368
--> -2     -0.02518947233224375
--> -3     -0.003760341567551533
--> -4     0.002797653064875037
--> -5     0.004103524251668446
--> -6     0.006939939583226135
--> -7     0.01378929753058581
--> -8     0.03410153435019589
--> -9     0.04817884324581996
--> -10    0.07575546411738791
```



```

--> -11    0.1002621018250477
--> -12    0.0433360466100355
--> -13    -0.03335392071313189
--> -14    -0.09138300988823311
--> -15    -0.06574964696989583
--> END

```

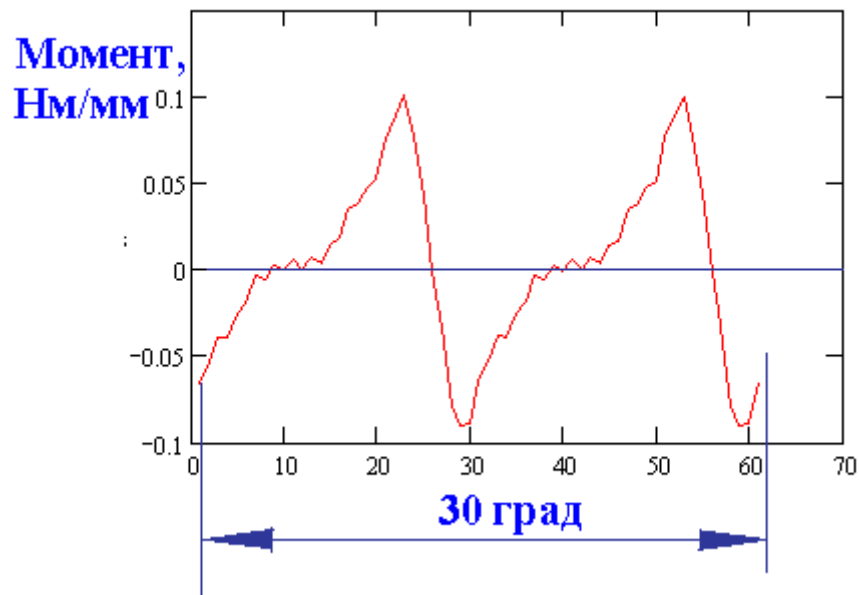


Рис. 5.7. Зависимость момента, обусловленного зубчатостью статора, от углового положения ротора. Ток в обмотке статора отсутствует

Угловой шаг поворота ротора 0,5 град., время счета – ок. полутора часов на ПК средней производительности. Симметричность полученной зависимости момента и хорошая повторяемость значений момента через период (зубцовое деление статора) свидетельствуют о достаточной точности моделирования.

5.5.5. Пример 4. Поле гармонических токовых слоев в гладком воздушном зазоре

Гармонический анализ является одним из основных инструментов исследования поля в воздушном зазоре электрических машин. Представление МДС обмоток в виде суммы гармонических составляющих позволяет рассчитать основной и дополнительные электромагнитные моменты, дополнительные потери, индуктивное сопротивление дифференциального рассеяния. Классической работой в этой области считается книга чехословацких авторов Б. Геллера и В. Гаматы «Высшие гармоники в асинхронных машинах» (Москва, «Энергия», 1981) («Дополнительные поля и моменты в асинхронных машинах» (Москва, «Энергия», 1968)).

В качестве примера рассмотрим следующую задачу. Смоделируем поле катушки с диаметральной шагом. Предположим, что стороны катушки уложены в пазы с малыми раскрытиями. В этом случае при достаточно малом зазоре (вернее, при достаточно малом отношении зазора к полюсному делению) распределение индукции имеет форму прямоугольной трапеции. Разложение в ряд Фурье функции в виде прямоугольной трапеции переменного знака содержит только нечетные гармоники. Если обозначить высоту трапеции $B_{\delta_{max}}$, то амплитуда основной гармоники оказывается равной $(4/\pi) B_{\delta_{max}}$, амплитуда третьей гармоники $(4/3\pi) B_{\delta_{max}}$, пятой $(4/5\pi) B_{\delta_{max}}$ и т.д.

Гармоническое распределение поле в гладком зазоре может быть создано тонким токовым слоем с гармонически изменяющейся линейной плотностью тока. Представим такой слой дискретным множеством тонких проводов, расположенных вблизи поверхности возбужденного сердечника. Значения токов в проводах должны изменяться в зависимости от координаты, отсчитанной вдоль окружности зазора по гармоническому закону.

При анализе результатов моделирования следует учитывать, что гармоники более высокого порядка быстрее затухают в зазоре машины.

Итак, составим *Lua*-сценарий для моделирования поля катушки с диаметральной шагом в гладком кольцевом зазоре с помощью гармонических токовых слоев.

```

--Создать файлы:
create("c:\\MyFEMMfiles\\CrntLrs_CylAG.fem",0)
mi_saveas("c:\\MyFEMMfiles\\CrntLrs_CylAG.fem")
mi_probdef(0,"millimeters","planar",1e-8,1,20)
outfile1 = "c:\\MyFEMMfiles\\CrntLrs_CylAG_Bdlt.txt"
handle=openfile(outfile1,"w")
outfile2 = "c:\\ MyFEMMfiles\\CrntLrs_CylAG_Trq.txt"
handle=openfile(outfile2,"w")
--Среда - воздух
mi_addmaterial("air")
--Размеры:
rinner=100
airgap=10
router=rinner+airgap
--Построение кольцевого зазора и контура интегрирования:
mi_addnode(0,0)
mi_addnode(-rinner,0)
mi_addnode(rinner,0)
mi_addnode(rinner+airgap*0.15,0)
mi_addnode(-rinner-airgap*0.15,0)
mi_addnode(rinner+airgap,0)
mi_addnode(-rinner-airgap,0)

mi_addarc(rinner,0,-rinner,0,180,1)
mi_addarc(-rinner,0,rinner,0,180,1)
mi_addarc(rinner+airgap*0.1,0,-rinner-airgap*0.15,0,180,1)
mi_addarc(-rinner-airgap*0.1,0,rinner+airgap*0.15,0,180,1)
mi_addarc(rinner+airgap,0,-rinner-airgap,0,180,1)
mi_addarc(-rinner-airgap,0,rinner+airgap,0,180,1)
--Свойства блоков:
mi_addblocklabel(-rinner-airgap*0.5,0)
mi_selectlabel(-rinner-airgap*0.5,0)
mi_setblockprop("air",0,0.5)
mi_addblocklabel(-rinner-airgap*0.03,0+0.5)
mi_selectlabel(-rinner-airgap*0.03,0+0.5)
mi_setblockprop("air",0,0.5)

```

Задавшись значением индукции в зазоре и зная размеры области задачи, можно определить амплитуды линейной плотности тока гармонических токовых слоев. Далее, задавшись числом точек (узлов), соответствующих тонким линейным проводам, моделирующим токовые слои (т.е. задавшись, в сущности числом участков, на которое разбит каждый токовый слой),

можно вычислить значение тока в каждом узле. Если принять, что каждый гармонический слой представлен одним и тем же числом узлов, то при моделировании нескольких слоев ток в каждом узле может быть найден как сумма токов в этом узле от каждого токового слоя. При учете достаточно большого числа гармоник распределение суммарного тока по поверхности сердечника должно приближаться к реальному, т.е. к двум линейным токам, соответствующим начальной и конечной катушечным сторонам. Ниже приведен фрагмент *MathCAD*'овского файла, в котором выполнены соответствующие расчеты для зазора 10 мм между двумя гладкими цилиндрическими сердечниками. Радиус внутреннего сердечника равен 100 мм. Амплитуда основной гармоники индукции принята равной 1 Тл (что соответствует $(\pi/4) = 0,785$ Тл трапецеидального распределения), число узлов слоя – 360 (по одному градусу на участок).

$$\begin{array}{lll}
 B_{1m} := 1 \cdot \text{Тл} & B_{1m} = 1 \text{Тл} & \text{airgap} := 10 \cdot \text{мм} \\
 \Psi_{1m} := \frac{1}{\mu_0} \cdot B_{1m} \cdot \text{airgap} & \Psi_{1m} = 7958 \text{ А} & k_{\text{harm}} := 1 \\
 r_{\text{inner}} := 100 \cdot \text{мм} & \tau_k := \frac{2 \cdot \pi \cdot r_{\text{inner}}}{2 \cdot k_{\text{harm}}} & \tau_k = 314.2 \text{ мм}
 \end{array}$$

Амплитуда линейной плотности тока гармоники:

$$J_{z1m} := \frac{\pi}{\tau_k} \cdot \Psi_{1m} \qquad J_{z1m} = 79577.5 \text{ А} \cdot \text{м}^{-1}$$

Максимальный ток на участке слоя шириной в 1 градус, соответствующий основной гармонике:

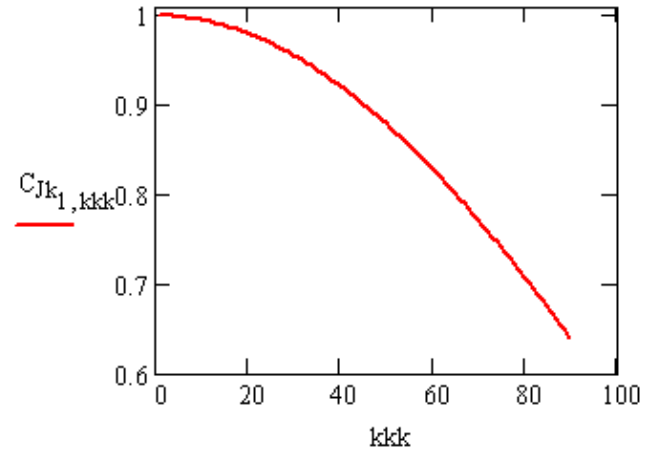
$$I_{z1m} := B_{1m} \cdot \frac{2 \cdot \text{airgap}}{\mu_0} \cdot \sin\left(0.5 \cdot \frac{\pi}{180}\right) \qquad I_{z1m} = 138.887 \text{ А}$$

Коэффициент, учитывающий уменьшение амплитуды линейной плотности тока (или максимального узлового тока) гармоники

$$k_{kk_harm_max} := 90$$

$$k_{kk} := 1..k_{kk_harm_max}$$

$$C_{Jk_1, k_{kk}} := \frac{\sin\left[\frac{1}{2} \cdot \frac{\pi}{180} \cdot (k_{kk} \cdot 2 - 1)\right]}{\frac{1}{2} \cdot \frac{\pi}{180} \cdot (k_{kk} \cdot 2 - 1)}$$



Токи в узлах токового слоя (сумма гармоник)

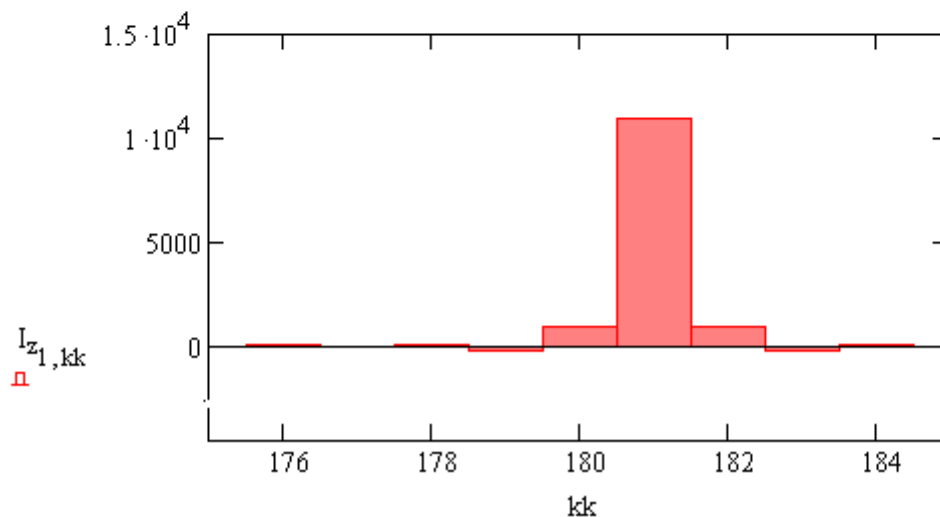
$$k_{k_max} := 360 \quad k_k := 1..k_{k_max} \quad k_{kk_harm_max} = 90$$

$$I_{z_1, k_k} := \sum_{k_{kk}=1}^{k_{kk_harm_max}} \left[-1 \cdot C_{Jk_1, k_{kk}} \cdot \left[I_{z1m} \cdot \cos\left[(k_k - 1) \cdot \left(\pi \cdot 180^{-1} \right) \cdot (k_{kk} \cdot 2 - 1) \right] \right] \right]$$

$$I_z =$$

	177	178	179	180	181	182	183	184	A
1	-51	91	-209	945	10908	945	-209	91	

$$10980 + 2 \cdot 945 - 2 \cdot 209 = 12452$$



Продолжение программы.

--Объявление массива узловых токов и исходные данные:

```
a = { }
```

```
Imax=138.9
```

```
nu_max=90
```

--Цикл по узловым точкам:

```
for i=1,361 do
```

--Цикл по гармоникам:

```
sum_nu=0
```

```
for nu=1,nu_max do
```

--Учитываем только нечетные гармоники:

```
nu_odd=nu*2-1
```

--Вычисляем коэффициенты уменьшения амплитуд и узловые токи:

```
C_j_nuodd=(sin(0.5*nu_odd))/((3.14159/360)*nu_odd)
```

```
sum_nu=sum_nu+1*( Imax* C_j_nuodd *cos((i-1)* nu_odd))
```

```
a[i] = sum_nu
```

--Окончание циклов:

```
end
```

```
end
```

--Массив имен узлов:

```
propname = { }
```

```
for i=1, 361 do
```

```
propname [i] = tostring(i)
```

```
end
```

--Построение узловых точек:

```
mi_addnode (-rinner-airgap*0.1,0)
```

```
mi_selectnode(-rinner-airgap*0.1,0)
```

```
mi_copyrotate (0,0,1,359, 0)
```