

В.И. Милых, Н.В. Полякова

**АВТОМАТИЗИРОВАННОЕ ФОРМИРОВАНИЕ РАСЧЕТНЫХ МОДЕЛЕЙ
ТУРБОГЕНЕРАТОРОВ ДЛЯ ПРОГРАММНОЙ СРЕДЫ FEMM**

Представлені принципи автоматизованого формування графічної і фізичної моделей турбогенераторів на алгоритмічній мові Lua для розрахунку їх магнітних полів і параметрів у програмному середовищі пакету FEMM. Можливості складеного скрипту Lua демонструються на прикладі реального турбогенератора. Бібл. 4, рис. 15.

Ключові слова: програма FEMM, метод скінчених елементів, турбогенератор, графічні та фізичні моделі, автоматизоване формування, Lua скрипт.

Представлены принципы автоматизированного формирования графической и физической моделей турбогенераторов на алгоритмическом языке Lua для расчета их магнитных полей и параметров в программной среде пакета FEMM. Возможности составленного скрипта Lua демонстрируются на примере реального турбогенератора. Библ. 4, рис. 15.

Ключевые слова: программа FEMM, метод конечных элементов, турбогенератор, графические и физические модели, автоматизированное формирование, Lua скрипт.

Введение. Существуют программные продукты для численных расчетов магнитных полей (МП) различных электротехнических устройств. Программы позволяют учитывать достаточно точно реальную геометрию устройств, а также нелинейные магнитные свойства ферромагнитных сердечников. Среди таких продуктов широкое распространение получила программа FEMM [1], основанная на методе конечных элементов и ориентированная на расчет двухмерных МП. Этого достаточно для электрических машин (ЭМ) классической конструкции, имеющих плоскопараллельное МП в пределах активной части.

Имеющиеся программные продукты уже не требуют от пользователя разработки каких-либо математических моделей. Достаточно следовать инструкциям при построении *графических моделей*, отображающих конструкцию ЭМ, а также *физических моделей*, отображающих свойства материалов, токи обмоток и параметры постоянных магнитов. Аналогично, надо следовать инструкциям при извлечении результатов расчета МП и их последующем использовании.

При эксплуатации программы FEMM обусловленные действия выполнялись и еще выполняются «вручную». Наибольшие затраты времени требуют формирование графических и физических моделей и преобразование результатов расчетов в форму, удобную для использования в процессе проектирования и исследования ЭМ. Это обуславливает низкую производительность расчетчиков и часто становится препятствием ввиду необходимости освоения большого объема инструкций и ошибок при следовании им.

Эти проблемы становятся еще серьезней при многовариантных расчетах МП и электромагнитных параметров ЭМ, что необходимо для варьирования токами их обмоток, геометрией и размерами, взаимным положением неподвижных и подвижных частей.

Проблемы можно преодолеть посредством автоматизации построения графических и физических моделей ЭМ, управления процессом расчета и перебора параметров, а также извлечения, обработки и приведения в необходимую для дальнейшего использования форму результатов. Такие возможности пользователям программы FEMM предоставляет интегри-

рованный в нее алгоритмический язык Lua. И задача теперь заключается в написании соответствующего комплекса программ (скриптов). Эта задача весьма сложна, так как требует комплексного владения знаниями в областях ЭМ, программирования и теории МП. Но эта задача вполне окупаема, так как программа пишется один раз, а используется бесчисленное число раз и любым количеством пользователей.

Для обеспечения высокой эффективности расчетов МП и электромагнитных параметров ЭМ в среде FEMM, нами разработан комплекс скриптов Lua. Эти программы позволяют многократно повысить производительность расчетчиков, а также обеспечивают возможность получения таких параметров и характеристик ЭМ, которые практически невозможно получить в «ручном» режиме работы с программой FEMM.

Цель работы. Все программы показать в одной статье не представляется возможным. Поэтому целью данной работы является представление принципов составления скрипта Lua для автоматизированного формирования графической и физической моделей турбогенератора (ТГ) – одного из наиболее крупных и ответственных представителей семейства ЭМ [2].

Требованием к такой программе является ее универсальность, с точки зрения варьирования геометрией и размерами проектируемых ТГ, при минимуме вводимой информации – и только в числовой форме.

Объект исследования. Для иллюстраций взяли ТГ мощностью 340 МВт. Его электромагнитная система представлена на рис. 1 поперечным сечением.

Здесь разной штриховкой стержней в пазах статора показаны фазные зоны его обмотки: $A-A'$, $B-B'$ и $C-C'$. При построениях моделей ТГ используется прямоугольная система координат x , y и принято: МДС фазной обмотки $A-A'$ и продольная ось ротора d ориентированы по оси y (при дальнейших расчетах ротор может вращаться); с осью x совпадают оси симметрии двух пазов статора. Но, в зависимости от относительного укорочения обмотки статора β_s , возможно расположение по оси x и зубцов статора [3]. Это регулируется в программе посредством ширины фазной зоны обмотки статора (в зубцовых делениях)

© В.И. Милых, Н.В. Полякова

$$z_s = q_s + \tau_p \cdot (1 - \beta_s). \quad (1)$$

Здесь используются число пазов на полюс и фазу и полюсное деление статора (в зубцовых делениях)

$$q_s = \frac{Q_s}{2 \cdot p \cdot m_s}; \quad \tau_p = \frac{Q_s}{2 \cdot p}, \quad (2)$$

где m_s, p, Q_s – числа фаз, пар полюсов и пазов статора.

Если z_s – нечетное число, то по оси x на статоре будет располагаться пазы, если четное – то зубцы.

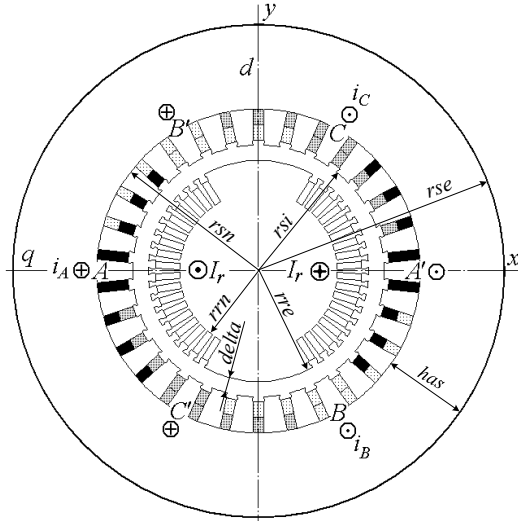


Рис. 1. Модель электромагнитной системы ТГ

Общая характеристика скрипта Lua. Тексты файлов скриптов и файлов данных можно писать в редакторе Блокнот. В строках этих файлов все после двойного дефиса (--) программа воспринимает как комментарии и для работы не использует.

В скриптах базовой является команда задания параметров новой *Задачи*: ее тип (МП или другое), единицы измерения размеров, глубина (активная длина) по аксиальной оси, тип координат, точность и т.д.

Построение графических моделей посредством Lua основано на командах расстановки *узлов* – координатных точек (x, y) . На их основе своими командами строятся *прямые отрезки* и *дуги*. В образующихся замкнутых подобластях – *блоках* ставятся их метки, через которые им приписываются физические свойства. Они задаются отдельными командами, в которых указываются все необходимые параметры материалов, токов, треугольников конечно-элементной структуры и др. Есть также свои команды для выбора и расстановки *граничных условий*. Перечисленные выше объекты можно объединять в нумерованные *группы* и оперировать со всеми объектами группы сразу.

Lua, как и другие языки программирования, позволяет создавать функции и процедуры, оперировать с циклами, условными операторами, делать вычисления по формулам, стандартным функциям: *ceil* – целая часть результата; *sqrt* – корень квадратный; *sin, cos, tan, atan* – синус, косинус, тангенс, арктангенс и др.

Для тригонометрических функций углы надо задавать в радианах, а при построении геометрических объектов Lua оперирует углами в градусах.

Скрипты вызываются (запускаются) из рабочего окна в FEMM «кнопкой» *Open Lua Script* в меню *File*.

Конструкторская модель турбогенератора.

Представим исходные параметры ТГ, которые задаются в файле данных для полного формирования графической и физической моделей ТГ. Далее используемые величины обозначены так, чтобы это соответствовало правилам идентификации в скрипте Lua.

К исходными схемно-конструктивным параметрам ТГ (рис. 1) относятся: *rre, rsi* – радиусы окружностей по поверхностям сердечников: внешней ротора и расточки статора; *has* – высота спинки сердечника статора; *p, ms* – числа пар полюсов и фаз; *Qs, Ncs, as, bss* – числа пазов, эффективных проводников в стержнях и параллельных ветвей, относительное укорочение шага обмотки статора (β_s); *Qr, Qru, Ncr* – числа пазов ротора обмотанных и условно-полное, а также эффективных проводников в пазу; *la* – активная длина статора; *dsl* – толщина листов его сердечника.

Кроме названного, детализации требуют зубцово-пазовые структуры статора и ротора, общепринятые формы пазов которых с обозначением необходимых размеров и опорных точек даны на рис. 2 и рис. 3.

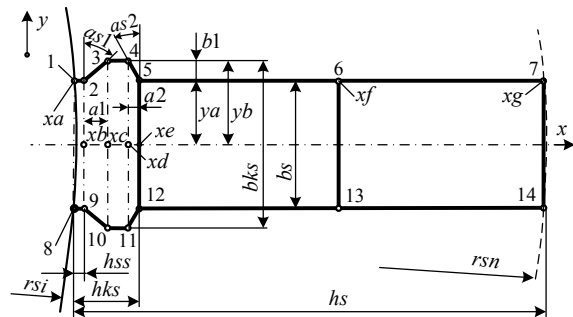


Рис. 2. Конструкция паза статора

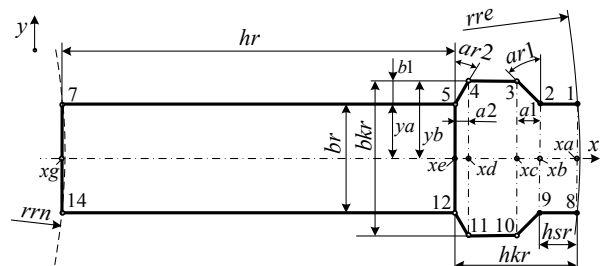


Рис. 3. Конструкция паза ротора

Паз статора имеет обозначения (выделены на рис. 2): *hs, bs* – высоты и ширина; *hss* – высота шлица; *hks, bks* – высота и ширина выемок под клин; *as1, as2* – 1-ый и 2-ой углы их скоса. Для паза ротора приняты обозначения (рис. 3): *hr, br* – высота и ширина; *hsr* – высота шлица; *hkr, bkr* – высота и ширина выемок под клин; *ar1, ar2* – 1-ый и 2-ой углы их скоса.

На основе перечисленных исходных данных вычисляются еще параметры графической модели: $rse = rsi + hs + has$ – внешний радиус сердечника статора; $delta = rsi - rre$ – воздушный зазор; $ts = 2 \cdot \pi \cdot rsi / Q_s$ – пазовый шаг по расточке статора; $tr = 2 \cdot \pi \cdot rre / Q_{ru}$ – пазовый шаг по поверхности ротора; $t_{sa} = 360 / Q_s$ – пазовый шаг статора (в градусах); $t_{ra} = 360 / Q_{ru}$ – пазовый шаг ротора (в градусах); $bts = ts - bs$ – ширина зубца статора по его поверхности; $btr = tr - br$ – ширина зубца ротора по его поверхности;

$rsn = \sqrt{x^2 + y^2}$ – радиус статора по углам дна паза,
 где $x = bs/2$; $y = hs + \sqrt{rsi^2 - x^2}$;
 $rtn = \sqrt{x^2 + y^2}$ – радиус ротора по углам дна паза,
 где $x = br/2$; $y = \sqrt{rre^2 - x^2} - hr$;
 $bzs = 2 * \pi * rsn / Qs - bs$ – ширина основания зубца статора;
 $bzr = 2 * \pi * rtn / Qru - br$ – ширина основания зубца ротора;
 $qsp = Qs / (2 * ms * p)$ – число пазов ст-ра на полюс и фазу;
 $tp = Qs / (2 * p)$ – полюсный шаг в зубцовых делениях;
 $zs = \text{ceil}(qsp + tp * (1 - bss))$ – ширина фазной зоны статора по его крайним стержням в зубцовых делениях.

Координатные размеры элементов паза статора:
 $ya = bs/2$; $yb = bks/2$; $xa = \sqrt{rsi^2 - ya^2}$; $xb = xa + hss$;
 $xe = xa + hks$; $xf = xa + (hks + hs)/2$; $xg = xa + hs$; $b1 = yb - ya$;
 $a1 = b1 * \tan(as1)$; $xc = xb + a1$; $a2 = b1 * \tan(as2)$; $xd = xe - a2$.

Координатные размеры элементов паза ротора:
 $ya = br/2$; $yb = bkr/2$; $xa = \sqrt{rre^2 - ya^2}$; $xb = xa - hsr$;
 $xe = xa - hkr$; $xg = xa - hr$; $b1 = yb - ya$; $a1 = b1 * \tan(ar1)$;
 $xc = xb - a1$; $a2 = b1 * \tan(ar2)$; $xd = xe + a2$.

Электромагнитные свойства и параметры турбогенератора в его физической модели.

Магнитные свойства сердечников ТГ представляются их кривыми намагничивания – зависимостями магнитной индукции от напряженности магнитного поля, т.е. $B(H)$. Эти кривые вводятся в программу FEMM парами значений В и Н (по сути – таблицами), а затем используются программой автоматически.

В ТГ используются кривые намагничивания для ярма сердечника статора, для его зубцов и для сердечника ротора [2]. Условные имена кривых предварительно вводятся из общего файла исходных данных, также как и имена соответствующих им файлов, откуда берутся массивы кривых намагничивания. Приняты (произвольно) такие обозначения: St_j_st, St_z_st, St_rot – имена сталей для спинки и зубцов сердечника статора и для сердечника ротора; bhj, bhz, bhr – имена файлов с кривыми намагничивания для этих сталей.

Для учета насыщения магнитопровода вводятся: kFes – коэффициент заполнения сердечника статора с учетом пакетирования; kFer – коэффициент приведения длины ротора к полной длине сердечника статора.

После формирования расчетной модели ТГ возможен тестовый расчет его МП. Для этого вводятся: Ir – ток обмотки ротора; Is – действующее значение фазного тока статора; beta – угол сдвига между направлениями действия МДС обмоток ротора и статора в соответствии с принятой системой фазовых соотношений ТГ [4]; fs – частота напряжения и тока статора.

Расчет МП проводится на конечно-элементной структуре, которая генерируется программой FEMM автоматически. Для варьирования ею вводятся максимально допустимые размеры сторон треугольников: fe1, fe2, fe3, fe4, fe5, fe6, fe7 – в зазоре; в ярме, зубцах и пазах статора; в пазах, зубцах и бочке ротора.

Структура программы построения графической и физической моделей турбогенератора.

Основные логические части программы таковы:

1. Задание имени файла исходных данных.
2. Открытие этого файла и чтение из него данных.
3. Задание имени расчетной модели ТГ.
4. Расчет дополнительных параметров на основе исходных данных.

5. Создание магнитной Задачи и ее параметров.
6. Создание внешней поверхности сердечника статора.
7. Создание зубцово-пазовой структуры статора.
8. Создание зубцово-пазовой структуры ротора.
9. Дорисовка поверхностей больших зубов ротора.
10. Создание свойства материалов с магнитной проницаемостью воздуха.
11. Создание параметров сердечников – имен магнитных свойств магнитопровода.
12. Ввод кривых намагничивания сталей.
13. Подготовка фазных токов обмотки статора.
14. Подготовка параметров геометрической структуры обмотки статора.
15. Задание материала и токов в стержнях шести фазных зон обмотки статора.
16. Задание материала и токов в пазах ротора.
17. Задание магнитных свойств воздушного зазора.
18. Создание граничных условий Дирихле и назначение их внешней поверхности статора.
19. Задание материала для ярма сердечника статора.
20. Задание материала для зубцов сердечника статора.
21. В зависимости от укорочения обмотки статора и ширины фазной зоны статора (1) поворачивается или нет его зубцовая структура на ползубцового шага.
22. Задание материала для зубцов сердечника ротора.
23. Задание материала для бочки ротора.
24. Запоминание созданной модели турбогенератора.
25. Выполнение тестового расчета магнитного поля.
26. Переход к показу результатов расчета.

Файлы исходных данных для скрипта Lua.

Эти файлы могут начинаться со строк комментариев, которые не участвуют в работе программы, но при вводе информации главной программой должны быть прочитаны – «сняты» как текстовые строки.

Программой вводятся числовые исходные данные, либо текстовые строки в двойных кавычках (" ").

Единица измерения геометрических размеров указывается в данных для главного процессора программы FEMM – обычно принимаются миллиметры.

Для примера подачи информации в файле данных дадим только отдельные фрагменты исходных параметров из перечня, представленного выше.

Примеры исходных данных в файле D_TG340.txt:
 "TG340" -- name_TG - имя расчетной модели ТГ
 "18.04.2015" -- data_dan - дата составления данных

```

.....
1 -- p - число пар полюсов
3 -- ms - число фаз обмотки статора
30 -- Qs - число пазов статора
36 -- Qr - число пазов ротора обмотанных
52 -- Qru - число пазов ротора общее (условное)
637.5 -- rsi - радиус расточки статора
.....
0.8 -- bss – относит. укорочение обмотки статора
2206 -- Ir - значение тока возбуждения, А
11547 -- Is - действующее значение тока статора, А
-160.47 -- beta - значение угла бета, градус
.....
"St3" -- St_rot - имя стали для сердечника ротора
"bhr" -- bhr - имя файла с кривой намагнич. для St_rot
.....
8 -- fe1 - макс. размер треугольника в зазоре
25 -- fe2 - макс. размер треугольника в ярме статора
.....
  
```

Содержимое файла "bhr" с кривой намагничивания.
 --Сталь Ст3 - для ротора: пары значений В-Н (Тл,А/м)
 0.1 80 0.2 160 0.3 240 0.4 320 0.5 400 0.6 488 0.7
 584 0.8 682 0.9 792 1.0 924 1.1 1090 1.2 1290 1.3
 1590 1.4 2090 1.5 2890 1.6 4100 1.7 5700 1.8 8000
 1.9 13000 2.0 23500 2.1 42500 2.15 62000 2.2 95000
 2.25 132000 2.3 170000 2.35 209000 2.4 249000 0 0

Нулевую точку кривых намагничивания программа FEMM добавляет сама, а нули в конце файлов сигнализируют о конце вводимого массива кривой.

Скрипт Lua для формирования модели ТГ.
 Далее приводятся фрагменты программы, которая после ее вызова в среде FEMM формирует графическую и физическую модели ТГ и запускает программу FEMM на расчет МП.

В тексте скрипта Lua команды, начинающиеся с *mi_*, являются стандартными процедурами. Их описание можно найти через кнопку *Help* в рабочем окне FEMM, войдя затем в раздел *Lua Scripting*.

Приведем часто встречающиеся команды:

1) добавление: *mi_addnode(x,y)* – узел; *mi_addsegment(x1,y1,x2,y2)* – отрезок между узлами; *mi_addarc(x1,y1,x2,y2,angle,maxseg)* – дуга между узлами с опорным углом 'angle' из угловых сегментов размером 'maxseg'; *mi_addblocklabel(x,y)* – ярлык блока, где (x,y), (x1,y1) и (x2,y2) – координаты узлов;

2) выбор ближайших к точке (x,y) объектов: *mi_selectnode(x,y)* – узел; *mi_selectsegment(x,y)* – отрезок; *mi_selectarcsegment(x,y)* – дуга; *mi_selectlabel(x,y)* – ярлык блока; *mi_selectgroup(n)* – n-ая группа;

3) копирование: *mi_mirror(x1,y1,x2,y2,edit)* – выбранных объектов относительно линии, проходящей через точки (x1,y1) и (x2,y2), *edit* – действие редактирования: 0 – для узлов, 1 – отрезков, 2 – дуг, 3 – меток, 4 – групп; *mi_copyrotate(bx, by, angle, copies, edit)* – *copies* – столько раз с поворотом на угол *angle* относительно базовой точки *bx, by*; *edit* – см. выше;

4) очистка выделения всех ранее выбранных объектов: *mi_clearselected()*;

5) установка свойства материала конкретной подобласти – блока: *mi_setblockprop("blockname", automesh, meshsize, "incircuit", magdir, group, turns)*, где: "blockname" – имя ярлыка для магнитных свойств блока, *automesh* – выбор способа триангуляции конечных элементов, *meshsize* – ограничивающий размер элемента, "incircuit" – символ тока, *magdir* – направление намагничивания, *group* – номер группы, *turns* – число эффективных проводников с током.

В программе используется нумерация точек пазов статора и ротора, приведенная на рис. 2 и 3.

По ходу программы вставлены рисунки, показывающие результат ее выполнения на данный момент.

Текст программы.

```
name_prog="Lua_M_TG" --имя программы
F_dan="D_TG340.txt" --имя файла данных
f_d=openfile(F_dan,"r") --его открытие
--считывание двух строк заголовка
for i=1,2,1 do str=read(f_d,"*1") end
--Ввод данных из File_dan по имени f_d
name_TG=read(f_d, "*n", "*1")
data_dan=read(f_d, "*n", "*1")
```

```
p=read(f_d, "*n", "*1") ms=read(f_d, "*n", "*1")
Qs=read(f_d, "*n", "*1") Qr=read(f_d, "*n", "*1")
Qru=read(f_d, "*n", "*1") rsi=read(f_d, "*n", "*1")
.....
bss=read(f_d, "*n", "*1") Ir=read(f_d, "*n", "*1")
Is=read(f_d, "*n", "*1") beta=read(f_d, "*n", "*1")
.....
-- имя стали для сердечника ротора
St_rot=read(f_d, "*n", "*1")
-- файл с кривой намагнич.для St_rot
bhr=read(f_d, "*n", "*1")
.....
fel=read(f_d, "*n", "*1")
fe2=read(f_d, "*n", "*1")
.....
-- Досчет геометрических параметров
gr=Pi/180 --коэф.из градусов в радианы
rse=rsi+hs+has delta = rsi-rre
rd=rre+delta/2 ts =2*Pi*rsi/Qs
tr=2*Pi*rre/Qru tsa =360/Qs
tra=360/Qru bts = ts-bs btr = tr-br
x=bs/2 y=hs+sqrt(rsi*rsi-x*x)
rsn=sqrt(x*x+y*y)
x=br/2 y=-hr+sqrt(rre*rre-x*x)
rrn=sqrt(x*x+y*y) bzs=2*Pi*rsn/Qs-bs
bzs=2*Pi*rrn/Qru-br qsp=Qs/(2*ms*p)
tp=Qs/(2*p) zs=ceil(qsp+tp*(1-bss))
closefile(f_d)--закрытие файла данных
--Создаем функцию для ввода кривых
--намагничивания В(Н) для сердечников
function get_B_H(bh_name, Steel_name, dsl, kFe)
--материал участка магнитопровода
mi_addmaterial(Steel_name, 0, 0, 0, 0, dsl, 0, kFe, 0, 0)
--открытие файла с кривой В(Н)
B_H = openfile(bh_name .. ".txt", "r")
str=read(B_H, "*1") --снятие заголовка
repeat --повтор ввода по условию
B,H=read(B_H, "*n", "*n") --чтение В и Н
--добавление точки кривой В(Н) в FEMM
mi_addbhpnt(Steel_name, B, H)
until (B==0) --условие конца кривой
closefile(B_H) end --function get_B_H
--Ввод кривых намагничивания сталей
--ротора, ярма и зубцов статора
mvi=get_B_H(bhr, St_rot, la, kFer)
mvi=get_B_H(bhj, St_j_st, dsl, kFes)
mvi=get_B_H(bhz, St_z_st, dsl, kFes)
--Составное имя модели ТГ - тип fem
geom_mod="Fe_Md" .. name_TG .. ".fem"
-- НАЧИНАЕТСЯ РАБОТА ПРОГРАММЫ FEMM
Start_date= date()
create(0) --Создаем магнитную Задачу
-- Параметры магнитной Задачи
mi_probdef(0, "millimeters", "planar", 1E-8, la)
--СОЗДАНИЕ ГЕОМЕТРИИ ТУРБОГЕНЕРАТОРА
mi_addnode(0, 0) --ставим точку центра
--точки верха и низа пов-ости статора
mi_addnode(0, rse) mi_addnode(0, -rse)
--размер сегмента окружности (град.)
max_seg = 5 --далее может меняться
--левая полуокружность пов-сти статора
mi_addarc(0, rse, 0, -rse, 180, max_seg)
--правая полуокружность
mi_addarc(0, -rse, 0, rse, 180, max_seg)
--Создание зубцового слоя статора
```

```

--координатные размеры паза статора
ya=bs/2 yb=bks/2 b1=yb-ya
xa=sqrt(rsi*rsi-ya*ya) xb=xa+hss
xe=xa+hks xf=xa+(hks+hs)/2 xg=xa+hs
a1=b1*tan(as1) xc=xb+a1 --1 скос паза
a2=b1*tan(as2) xd=xe-a2 -- 2 скос паза
--ставятся точки от 1 до 7-ой (рис.2)
mi_addnode(xa, ya) mi_addnode(xb, ya)
mi_addnode(xc, yb)
mi_addnode(xd, yb) mi_addnode(xe, ya)
mi_addnode(xf, ya) mi_addnode(xg, ya)
--линии между точками 1, 2, 3, 4, 5, 6, 7
mi_addsegment(xa, ya, xb, ya)
mi_addsegment(xb, ya, xc, yb)
mi_addsegment(xc, yb, xd, yb)
mi_addsegment(xd, yb, xe, ya)
mi_addsegment(xe, ya, xf, ya)
mi_addsegment(xf, ya, xg, ya)
--размечаются точки от 1 до 7-ой
mi_selectnode(xa, ya) mi_selectnode(xb, ya)
mi_selectnode(xc, yb) mi_selectnode(xd, yb)
mi_selectnode(xe, ya) mi_selectnode(xf, ya)
mi_selectnode(xg, ya)
--размечаем линии между тчк. от 1 до 7
mi_selectsegment(xa, ya) y=(ya+yb)/2
x=(xb+xc)/2 mi_selectsegment(x, y)
x=(xc+xd)/2 mi_selectsegment(x, yb)
x=(xd+xe)/2 mi_selectsegment(x, y)
x=(xe+xf)/2 mi_selectsegment(x, ya)
x=(xf+xg)/2 mi_selectsegment(x, ya)
--назначаем выделенным точкам и линиям
--группу 1 и снимаем выделение
mi_setnodeprop(" ", 1)
mi_setsegmentprop(" ", " ", 1, 0, 1)
mi_clearselected() --см. рис.4

```

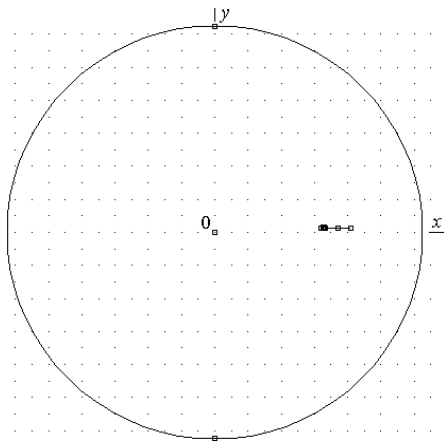


Рис. 4. Центр и поверхность сердечника статора с опорными точками дуг и стенка паза статора

```

mi_selectgroup(1) --выделяем стенку паза
--и отображаем симметрично оси x вниз
mi_mirror(0,0,rse,0,4) mi_clearselected()
--проводятся и размечаются линии 5-12,
-- 6-13 и 7-14, им назначается группа 1
mi_addsegment(xe, ya, xe, -ya)
mi_addsegment(xf, ya, xf, -ya)
mi_addsegment(xg, ya, xg, -ya)
mi_selectsegment(xe, 0) mi_selectsegment(xf, 0)
mi_selectsegment(xg, 0)
mi_setsegmentprop(" ", " ", 1, 0, 1)
mi_clearselected() --процесс на рис.5

```

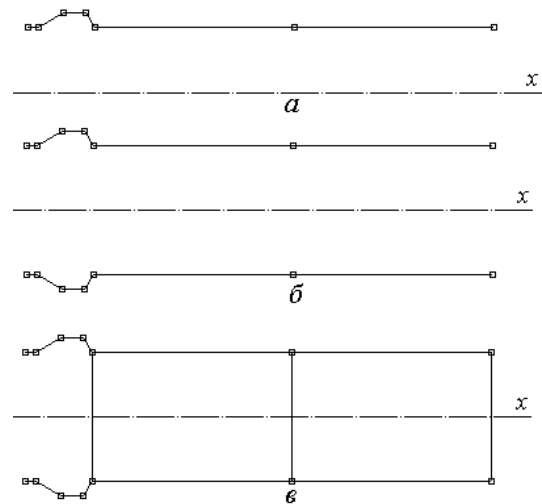


Рис. 5. Процесс построения паза статора: а – верхняя стенка паза; б – ее отражение относительно оси x; в – добавили линии дна паза и слоев обмотки

```

--Выделяем паз и поворачиваем его на
--ползубцового шага вверх над осью x
mi_selectgroup(1) mi_moverotate(0,0,tsa/2,4)
--готовим координаты точек 8 и 14 и
--копируем их симметрично под ось x
u1=atan(ya/xa) u2=0.5*tsa*gr-u1
x8=rsi*cos(u2) y8=rsi*sin(u2)
u3=atan(ya/xg) u4=0.5*tsa*gr-u3
x14=rsn*cos(u4) y14=rsn*sin(u4)
mi_selectnode(x8,y8) mi_selectnode(x14,y14)
mi_mirror(0,0,rse,0,0)
--дуги зубца от точек 8 и 14 вниз
max_seg=1 u2=2*u2/gr
mi_addarc(x8,-y8,x8,y8,u2,max_seg)
max_seg=2 u4=2*u4/gr
mi_addarc(x14,-y14,x14,y14,u4,max_seg)
--выделяем дуги и назначаем группу 1
mi_selectarcsegment(rsi,0) mi_selectarcsegment(rsn,0)
mi_setarcsegmentprop(1," ", 0, 1)
mi_clearselected() --итог на рис.6

```

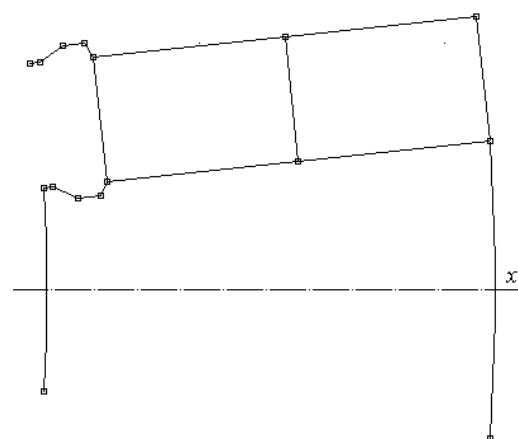


Рис. 6. Процесс построения зубца статора

```

--выделяем и копируем рис.6 (Qs-1)-раз
--и получаем зубцово-пазовую структуру
mi_selectgroup(1) mi_copyrotate(0,0,tsa,(Qs-1),4)
Аналогично создается геометрия ротора и объе-
диняется все в группу 9. Результат формирования
графической модели ТГ представлен на рис. 7.

```

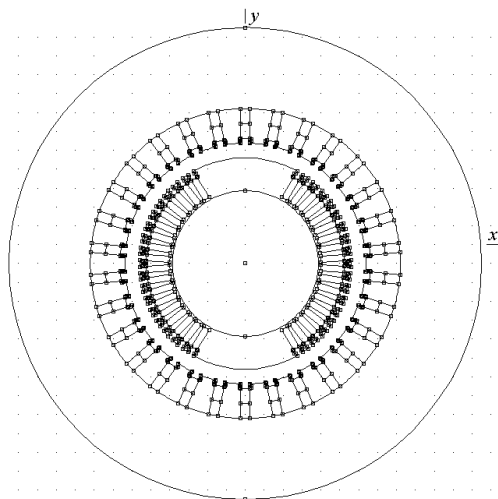


Рис. 7. Сформированная графическая модель ТГ

```

--Создаем свойства материалов ТГ
--магнитная проницаемость воздуха
mi_addmaterial("mu0",1,1)
--ЗАДАНИЕ ТОКОВ В ОБМОТКЕ СТАТОРА
Im=Is*2^0.5/as --амплитуда токов
--Фазные токи в обмотке статора
IA=Im*cos(beta*gr)
IB=Im*cos(beta*gr -2*Pi/3)
IC=Im*cos(beta*gr+2*Pi/3)
--Параметры обмотки статора
--радиусы ее нижнего и верхнего слоев
rn=rsi+0.75*hs rv=rsi+hs/4+hks
--числа пазов нижнего и верхнего слоев под осью q
qsn=0.5*(qsp-tp*(1-bss)) qsv=qsp-qsn
--углы расположения первых стержней
--нижнего и верхнего слоев фазы А
anl=180-(qsn-0.5)*tsa avl=180-(qsv-0.5)*tsa
--Функция задания тока в фазной зоне
--обмотки статора и материала стержней
--cir_name-имя зоны, i_s -ее ток,
--j_beg, j_end --номера начального и
--конечного проводников
--N_group -задаваемый номер группы
function tok_faz(cir_name, i_s, j_beg, j_end, N_group)
--связка имени тока с его значением
mi_addcircprop(cir_name, i_s, 1)
--цикл перебора проводн.фазной зоны
for j=j_beg,j_end,1 do ai=(j-1)*tsa
--координаты нижних стержней
ani=(anl+ai)*gr
x=rn*cos(ani) y=rn*sin(ani)
--добавление и выделение метки свойств
mi_addblocklabel(x, y) mi_selectlabel(x, y)
--задание свойств стержней
mi_setblockprop("mu0",0,fe4,cir_name,0,N_group,Ncs)
mi_clearselected()
--координаты верхних стержней
avi=(avl+ai)*gr x=rv*cos(avi) y=rv*sin(avi)
--добавление и выделение метки свойств
mi_addblocklabel(x, y) mi_selectlabel(x, y)
--задание свойств стержней обмотки
mi_setblockprop("mu0",0,fe4,cir_name,0,N_group,Ncs)
mi_clearselected()
end --конец перебора стержней по j
end --конец function tok_faz
--Задание токов в шести фазных зонах обмотки

```

```

--статора и отведение им групп 2 - 7
zap=tok_faz("IA+", IA, 1, qsp, 2)
zam=tok_faz("IA-", -IA, 3*qsp+1, 4*qsp, 5)

```

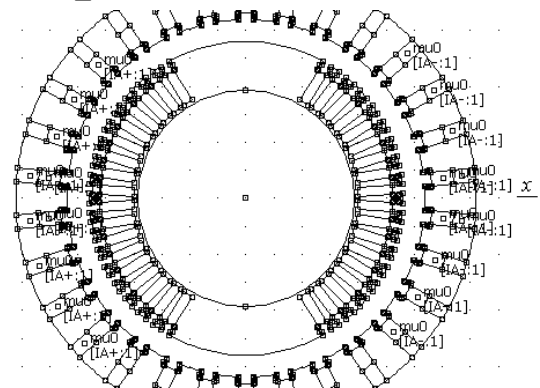


Рис. 8. Метки свойств в стержнях фазных зон фазы А

```

zap=tok_faz("IB+", IB, 2*qsp+1, 3*qsp, 3)
zam=tok_faz("IB-", -IB, 5*qsp+1, 6*qsp, 6)
zap=tok_faz("IC+", IC, 4*qsp+1, 5*qsp, 4)
zam=tok_faz("IC-", -IC, qsp+1, 2*qsp, 7)

```

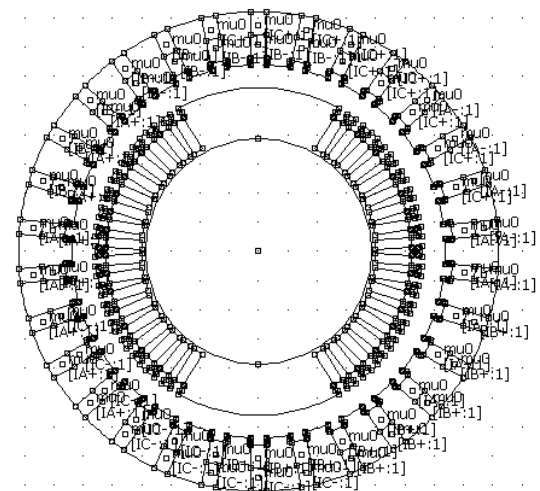


Рис. 9. Добавлены метки свойств в стержнях фазных зон фаз В и С обмотки статора

```

--Материал и токи в пазах ротора
--Связка имен тока с его значениями Ir
mi_addcircprop("IF-", -Ir, 1) mi_addcircprop("IF+", Ir, 1)
--добавление и выделение метки в пазу
y=rre-0.75*hr x=0.4*tr
mi_addblocklabel(y,x) mi_selectlabel(y,x)
--задание свойств стержней и группы 10
mi_setblockprop("mu0", 0, fe5, "IF-", 0, 10, Ncr)

```

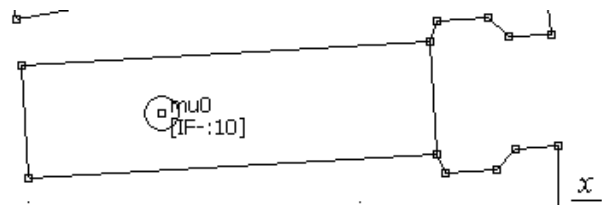


Рис. 10. Метка свойств стержня в пазу ротора

```

--задание токов "IF-" по пазам справа
--вверху, отражение вниз
mi_copyrotate(0, 0, tra, (Qr/4-1), 2)
mi_selectgroup(10) mi_mirror(0, rre, 0, 2)
--перевод пазов из 10-ой в 9-ую группу

```

```

mi_selectgroup(10)
mi_setblockprop("mu0", 0, fe5, "IF-", 0, 9, Ncr)
mi_clearselected()
--добавл. и выделение метки тока "IF+"
mi_addblocklabel(-y,x) mi_selectlabel(-y,x)
--задание свойств стержней, группа 10
mi_setblockprop("mu0", 0, fe5, "IF+", 0, 10, Ncr)
--задание токов "IF+" по пазам слева
--вверху, и отражение вниз
mi_copyrotate(0,0,-tra, (Qr/4-1), 2)
mi_selectgroup(10) mi_mirror(-rre, 0, 0, 0, 2)
--перевод пазов из 10-ой в 9-ую группу
mi_selectgroup(10)
mi_setblockprop("mu0",0,fe5,"IF+",0,9,Ncr)
mi_clearselected()

```

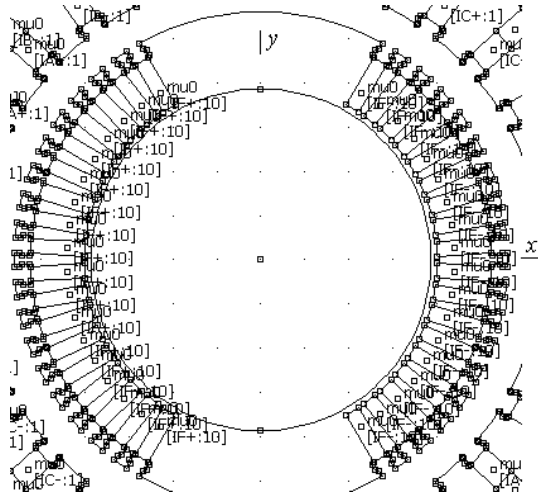


Рис. 11. Формирование свойств стержней в пазах ротора

```

--Ставится и выделяется метка, задание
--магнитных свойств зазора, группа 0
mi_addblocklabel(0, (rre+rsi)/2)
mi_selectlabel(0, (rre+rsi)/2)
mi_setblockprop("mu0",0,fe1, " ", 0, 0, 0)
mi_clearselected()
--Создаем граничные условия Дирихле
mi_addboundprop("Grn", 0, 0, 0, 0, 0, 0, 0, 0)
--выделяем дуги поверхности статора
mi_selectarcsegment(rse,0) mi_selectarcsegment(-rse,0)
max_sg=5
--Назначаем поверхности ст-ра условия
mi_setarcsegmentprop(max_sg, "Grn", 0, 0)
mi_clearselected()
--СОЗДАЕМ ПАРАМЕТРЫ СЕРДЕЧНИКОВ
--Ставится и выделяется метка, задание
--магнитных свойств ярма статора, гр.0
y=rsi+hs+has/2 mi_addblocklabel(0,y)
mi_selectlabel(0,y)
mi_setblockprop(St_j_st, 0, fe2, " ", 0, 0, 0)
mi_clearselected()

```

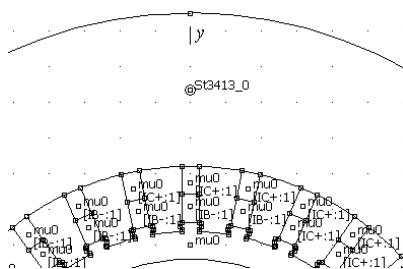


Рис. 12. Метки для зазора и ярма сердечника статора

```

--Задание метки и материала для зубца
--статора на оси x, группа 1
x=rsi+hs/2 mi_addblocklabel(x,0)
mi_selectlabel(x,0)
mi_setblockprop(St_z_st, 0, fe3, " ", 0, 1, 0)
--распространение по всем зубцам
mi_copyrotate(0,0,tsa, (Qs-1), 2)
--деление ширины фазной зоны zs на 2
z1=zs/2 z2=ceil(z1) --дробное и целое
--зависимо от этого поворачиваем или
--нет зубцовую структура на ползубца
if z1==z2 then str="zs-четное и нет"
else str="zs-нечетное и есть поворот")
mi_selectgroup(1) mi_moverotate(0,0,tsa/2,4) end

```

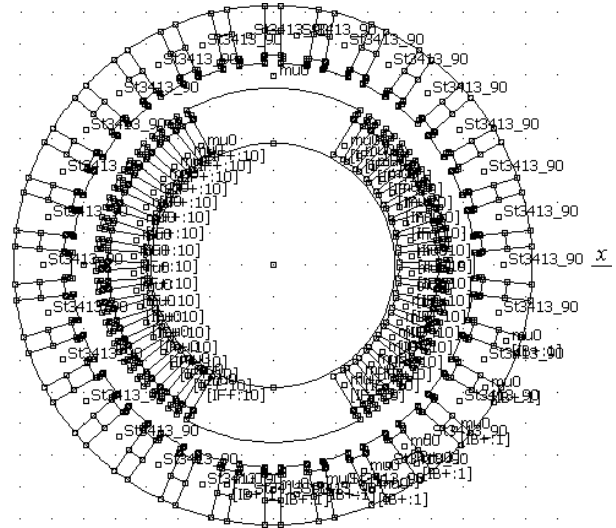


Рис. 13. Метки и свойств зубцов сердечника статора

```

--Метка св-тв, ее выделение и задание
--материала для зубцов ротора, гр. 9
x=rre-hr/2 mi_addblocklabel(x,0) mi_selectlabel(x,0)
mi_setblockprop(St_rot, 0, fe6, " ", 0, 9, 0)

```

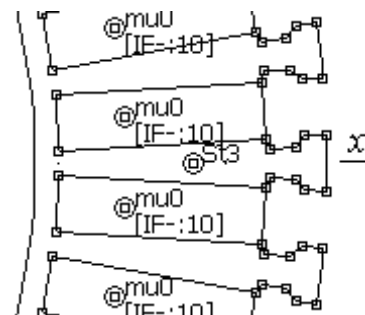


Рис. 14. Метка свойств для зубца сердечника ротора, она распространяется и на прилегающую кольцевую зону и через нее на все остальные зубцы ротора

```

--Задание материала для ярма сердечника ротора
y=rre/2 mi_addblocklabel(0,y) mi_selectlabel(0,y)
mi_setblockprop(St_rot, 0, fe7, " ", 0, 9, 0)
mi_clearselected()

```

```

mi_saveas(geom_mod) --Запомнить модель ТГ
mi_analyze(1) --Выполнить расчет МП
--Переход к показу результатов расчета
mi_loadsolution()
writeto() --Заккрытие файла результатов
mi_close() --Закрывается документ
--препроцессора магнетизма

```

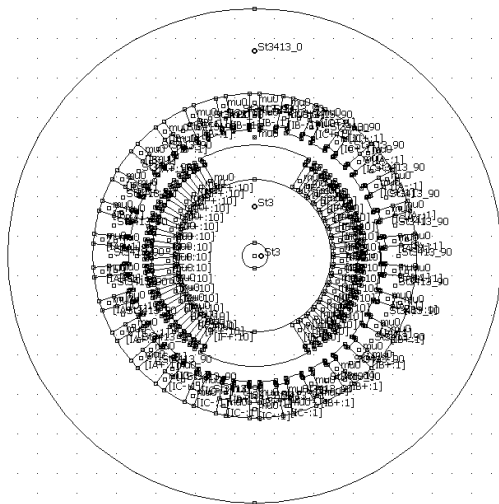


Рис. 15. Полная готовность модели турбогенератора

Инструкция по использованию скрипта.

1. Файл скрипта с расширением *.lua и все файлы с исходными данными должны находиться в одной папке на диске компьютера.

2. Запускаем программу FEMM и в меню File выбираем Open Lua script и в диалоговом окне запускаем файл с именем программы в работу.

3. Дальнейшие построения программа выполняет автоматически. В программу можно вставить команды Pause для ее останова в конкретном месте и просмотра текущего изображения на экране компьютера. Клавишей Enter продолжить выполнение программы.

4. Команды mi_saveas(name) в программе позволяют сохранить очередной файл с текущим результатом построенной модели ТГ под именем name.

Вывод. Представленный скрипт Lua дает широкие возможности пользователям программы FEMM в создании расчетных моделей для обеспечения расчетов магнитных полей и электромагнитных параметров ТГ. Этот скрипт универсален с точки зрения геометрических размеров и структуры электромагнитной системы ТГ в рамках их распространенной типовой конструкции. На базе рассмотренных принципов формирования расчетных моделей могут быть разработаны аналогичные скрипты для других типов ЭМ (например, уже создан и используется скрипт для формирования моделей асинхронных двигателей).

СПИСОК ЛИТЕРАТУРЫ

1. Meeker D. Finite Element Method Magnetics. FEMM 4.2 11 Oct 2010 Self-Installing Executable. Режим доступа: www.femm.info/wiki/OldVersions.
2. Титов В.В., Хуторецкий Г.М., Загородная Г.А. и др. Турбогенераторы. – Л.: Энергия. – 1967. – 895 с.
3. Милых В.И., Полякова Н.В. Численно-полевая оценка эффективности укорочения обмотки статора турбогенератора // *Электротехника і електромеханіка*. – 2014. – №4. – С. 12-16.

4. Милых В.И., Полякова Н.В. Система направлений и фазовых соотношений электромагнитных величин при численных расчетах магнитных полей в турбогенераторе // *Электротехника і електромеханіка*. – 2011. – №5. – С. 33-38.

REFERENCES

1. Meeker D. *Finite Element Method Magnetics. FEMM 4.2 32 bit 11 Oct 2010 Self-Installing Executable*. Available at: www.femm.info/wiki/OldVersions (accessed 10 March 2014).
2. Titov V.V., Hutoreckij G.M., Zagorodnaja G.A., Vartan'jan G.P., Zaslavskij D.I., Smotrov I.A. *Turbogenerator* [Turbogenerators]. Leningrad, Energiia Publ., 1967. 895 p. (Rus).
3. Milykh V.I., Polyakova N.V. Numerical field estimation of turbogenerator stator winding shortening efficiency. *Elektrotehnika i elektromekhanika – Electrical engineering & electromechanics*, 2014, no.4, pp. 12-16. (Rus).
4. Milykh V.I., Polyakova N.V. A system of directions and phase relationships for electromagnetic parameters at numerical calculations of magnetic fields in a turbogenerator. *Elektrotehnika i elektromekhanika – Electrical engineering & electromechanics*, 2011, no.5, pp. 33-38. (Rus).

Поступила (received) 20.04.2015

Милых Владимир Иванович¹, д.т.н., проф.,
Полякова Наталья Владимировна¹, инженер,
¹Национальный технический университет
«Харьковский политехнический институт»,
61002, Харьков, ул. Фрунзе, 21,
тел/phone +38 057 7076514, e-mail: mvikemkpi@gmail.com

V.I. Milykh¹, N.V. Polyakova¹

¹National Technical University «Kharkiv Polytechnic Institute»,
21, Frunze Str., Kharkiv, 61002, Ukraine.

Automated formation of calculation models of turbogenerators for software environment FEMM.

Attention is paid to the popular FEMM (Finite Element Method Magnetics) program which is effective in the numerical calculations of the magnetic fields of electrical machines. The main problem of its using - high costs in time on the formation of a graphical model representing the design and on the formation of the physical model representing the materials properties and the winding currents of machines – is solved. For this purpose, principles of the automated formation of such models are developed and presented on the turbogenerator example. The task is performed by a program written in an algorithmic language Lua integrated into the package FEMM. The program is universal in terms of varying the geometry and dimensions of the designed turbogenerators. It uses a minimum of input information in a digital form representing the design of the whole turbogenerator and its fragments. A general structure of the Lua script is provided, significant parts of its text, the graphic results of work's phases, as well as explanations of the program and instructions for its use are given. Performance capabilities of the compiled Lua script are shown on the example of the real 340 MW turbogenerator. References 4, figures 15.

Key words: program FEMM, Finite Element Method, turbogenerator, graphical and physical models, automated formation, Lua script.