

# Обзор компьютерной программы HELIOS

Веренич Кирилл Андреевич

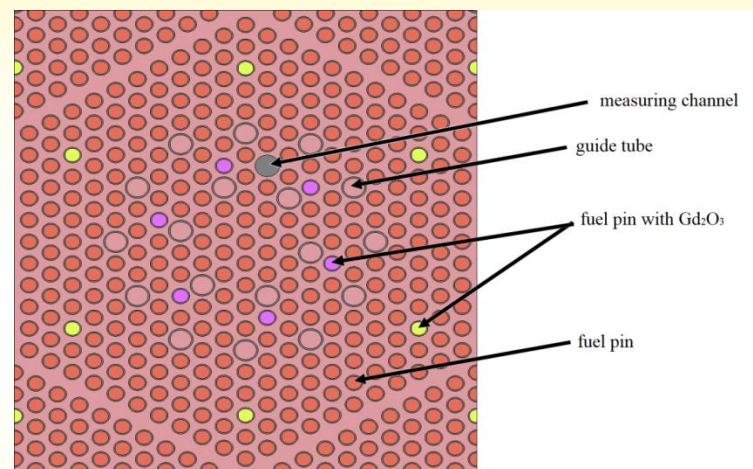
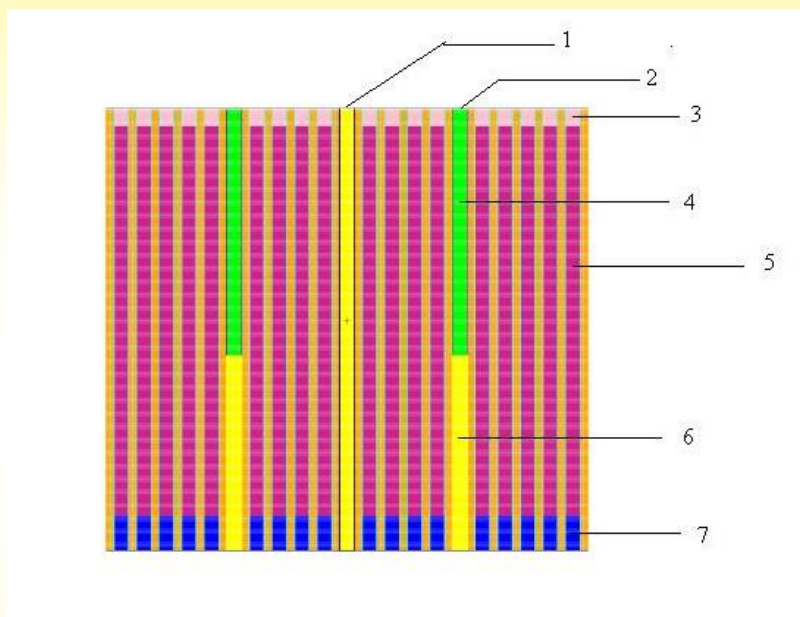
# Введение

- Программный комплекс HELIOS создан компанией Studsvik Scandpower (Швеция).
- Предназначен для нейтронно-физических расчетов двугрупповых сечений в процессе выгорания ядерного топлива.

Структура программного комплекса

1. AURORA – обработка входного файла.
2. ORION – графическое отображение геометрии.
3. HELIOS – 2-мерный расчет физических процессов.
4. ZENITH – вывод результатов.

# Двумерная геометрия



Вертикальный разрез по центру тепловыделяющей сборки ВВЭР [№ ГР 20122034].

- 1 – центральное отверстие,
- 2- канал для установки регулирующих стержней,
- 3 – верхняя часть твэла,
- 4 – регулирующий стержень,
- 5 – средняя часть твэла, содержащая уран,
- 6 – пустая часть регулирующего канала,
- 7 – нижняя часть твэла.

Модель геометрии тепловыделяющей сборки [doi:10.1088/1742-6596/1689/1/012043]

# Входной файл AURORA

```
+THEL
!-----1-----2-----3-----4-----5-----6-----7-----!
case   = CASE('E:\SCM\hellib-2.04.00\winT64\hy058n48g240_x64.dat'/'h.hrf'/
         'Pincell example')

$hp    = PAR("1.60/2") ! Half pitch !
$rfu   = PAR("0.5")    ! Fuel outer radius      [cm] !
$rc1   = PAR("0.6")    ! Clad radius            [cm] !

H2O    = MAT( 0.7/ 1001, 11.18;  8016,88.82)
UO2    = MAT(10   /92235,  3.0 ;  92238,97.00; 8001,0)
ZIR    = MAT( 6.5/40002,100.0)

!-----!
pin    = CCS(0.5,0.6//fuel, clad)
```

# Входной файл AURORA

```
cell = STR(("- $hp", "-$hp") ("-$hp", $hp) ($hp, $hp) ($hp, "- $hp") ! 1-4 !
          ("-$hp", 0 ) ( 0 , $hp) ($hp, 0 ) ( 0 , "- $hp") ! 5-8 !
          ("-$rc1", 0 ) ( 0 , $rc1) ($rc1, 0 ) ( 0 , "-$rc1") ! 9-12!
/ 4,cool / pin(0,0) /5,2,6,10,9,cool; 6,3,7,11,10,cool; 7,4,8,12,11,cool)
System = CNX(cell)
System = BDRY((1,1,1)1(0))
ovm      = OVLM(H2O/*-0-cool/
               UO2/*--fuel/
               ZIR/*--clad)
ovd      = OVLD(1/*-**)
ovt      = OVLT(555/*-**)
osm      = OVSM(ovm)
osd      = OVSD(ovd)
ost      = OVST(ovt)
st       = STAT(osm,osd,ost,38)
pa       = PATH(/(st),10000/10)
case     = RUN()
```

# Задание материального состава в HELIOS

## 1. Команда MAT.

Служит для задания изотопного состава материалов.

Использование:

```
name = MAT ( [NB/] [d] / i1, c1 [; i2, c2 ] ... )
```

- *name* – имя, представляющее определяемый здесь материал;
- NB – слово NB, указывающее, что материал не выгорающий;
- d – плотность в г/см<sup>3</sup>, [0]: ввод атомной плотности;
  - >0: плотность материала (ввод весовых долей);
  - <0: ввод атомной плотности в тяжелом металле при нулевом выгорании.

Пример:

```
H2O = MAT( 0.7/ 1001, 11.18; 8016, 88.82)
```

## 2. Команда OVLM (сокращение от OVerLay Material).

Использование:

```
name = OVLM ( mid/pid [, pid] ... [ /mid/pid [, pid]... ] ... )
```

- *name* – представляет присвоение материалов регионам системы.
- mid – идентификатор материала, относящийся к одному или нескольким материалам, или неявно через их регионы.
- pid – идентификатор места расположения, относящийся напрямую или косвенно к одному или нескольким регионам.

```
ovm = OVLM(H2O/*-0-cool/  
          UO2/*-*--fuel/  
          ZIR/*-*--clad)
```

# Задание температуры/плотности в HELIOS

1. Команда OVLT (сокращение от OVerLay Temperature).
2. Команда OVLD (сокращение от OVerLay Density).

Использование:

Пример:

ovd = OVLD (1/\*-\*\*) )

ovt = OVLT (555/\*-\*\*) )

# Задание геометрии в HELIOS. Оператор CCS

- `CCS()` – оператор, позволяющий определять систему, состоящую из круговых цилиндров.

Использование:

```
name = CCS(r , [r].../[na [(nf,nl)] ]/[cid [, cid]...])
```

- где *name* определяет систему, состоящую из кругового цилиндра, которая может содержать радиальные зоны, возможно разделенные на азимутальные сектора, таким образом определяя ее регионы (с постоянным потоком) или сеткой;
- *r* определяет радиус в см;
- *na* определяет число азимутальных регионов на радиальную зону;
- *nf*, *nl* – последовательности чисел первой и последней радиальной зоны, в которой действует *na* – если не определены, но берутся все зоны;
- *cid* – имя региона CCS.

Пример:

```
pin = CCS(0.5, 0.6//fuel, clad)
```

# Оператор STR

- Оператор STR используется для определения геометрической структуры, например, ячейки с твэлом, водяного канала или управляющего стержня.

Использование:

```
name = STR ( n [[,] n]... / p / [c [, c]...] / [r [; r]...] )
```

*name* обозначает геометрическую структуру, определяемую замкнутой периферией, состоящей из прямолинейных сегментов и ее внутренности, состоящей из CCS и регионов (в которых поток постоянный) или сеток.

*n* определяет узел,  $n=(x,y)$

*p* определяет периферию,  $p=p_n[,bid]$

*c* определяет CCS  $c = cname (cx, cy[, rot]) [np ]$

*R* определяет регион, не содержащий CCS,  $r = j,...k[, rid]$

Пример:

```
cell = STR(("-$hp", "$hp") ("-$hp", $hp) ($hp, $hp) ($hp, "$-$hp") ("-$rcl", 0) (0, $rcl) ($rcl, 0) (0, "$-$rcl") /  
4, cool / pin(0, 0) / 5, 2, 6, 10, 9, cool; 6, 3, 7, 11, 10, cool;  
7, 4, 8, 12, 11, cool)
```

# Задание геометрии. Оператор CNX.

## CNX

Сшивка или соединение структур и/или подсистем; угловая дискретизация токов в интерфейсах поверхностей.

Использование:

$name = \mathbf{CNX} ( s [ , s ] \dots [ / ( c , n [ ] ( c ] , n ) k ( c , n [ ] ( c ] , n ) ] \dots )$

*name* представляет систему или подсистемы, состоящие из одной или нескольких подсистем или структур, которые комбинируются друг за другом.

*s* указывает подсистему или структуру,  $s = sname$

*c* определяет структуру в *s* по цепочке,  $c = [ / ] \dots /$

*n* определяет узел в структуре,  $n = n_i$ , или  $n = (x, y)$

*k* определяет угловое представление токов на границе.

Где *sname* – имя одной из подсистем или структур, используемых для конструирования данной системы (подсистемы);

*/* – ссылка вдоль цепочки, которая определяет структуру (последняя ссылка), к которой относится узел *n*, используется при сшивке одной из подсистем или структур *s* в списке (первая ссылка);

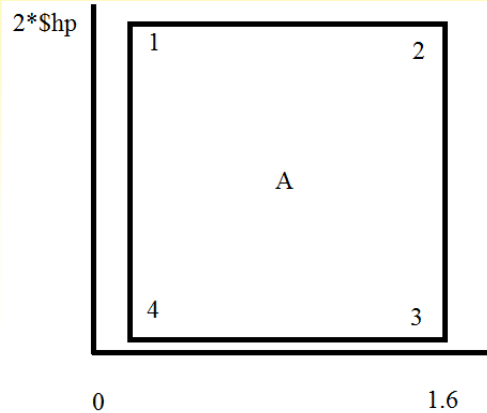
$n_i$  – номер последовательности узла в данной структуре;

$(x, y)$  – координаты (см), в круглых скобках, данной структуры узла, которого нет в списке узлов.

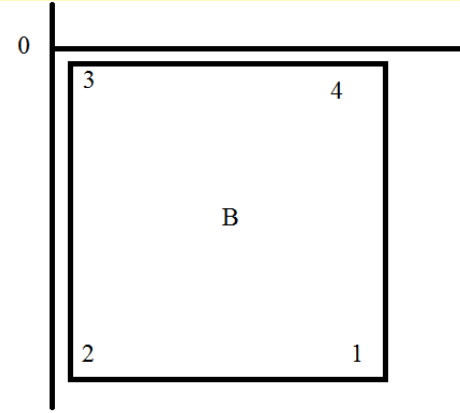
Пример 1:

```
System = CNX(cell)
```

## Оператор CNX. Пример 2

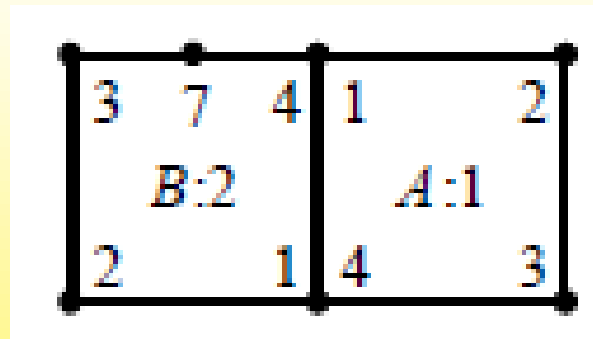


```
A = STR((0, "2*$hp") ("2*$hp", "2*$hp") ("2*$hp", 0)
(0, 0) / 4, cool / pin ($hp, $hp) //)
```

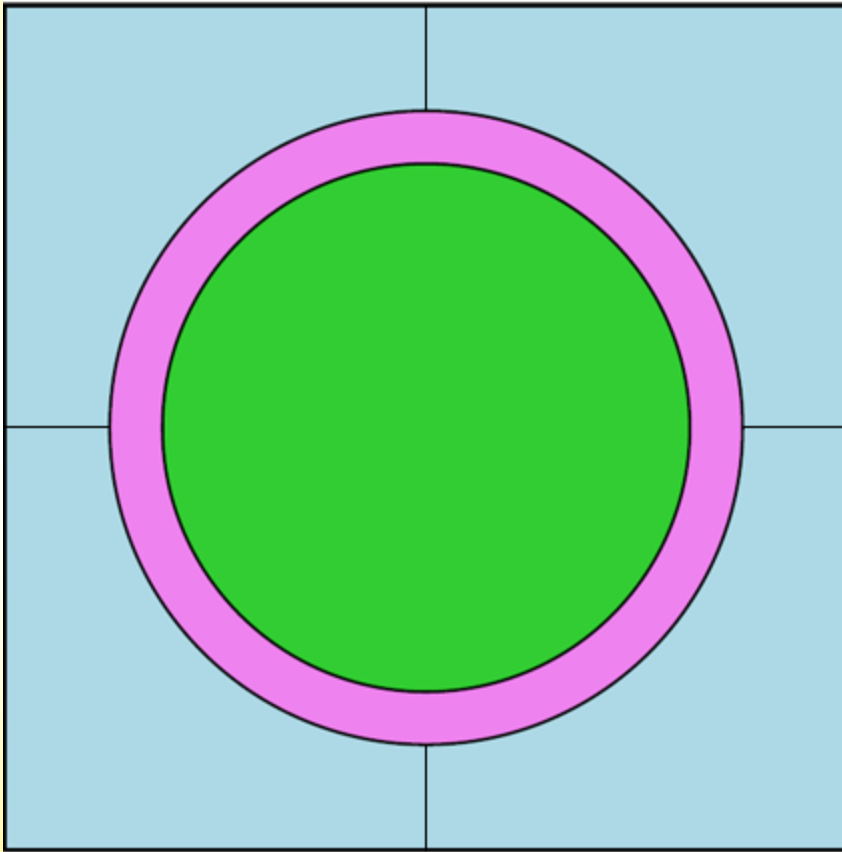


```
B=STR(("2*$hp", "-2*$hp") (0, "-2*$hp") (0, 0) ("2*$hp", 0) / 4, cool //)
```

$C = \text{CNX}(A, B / (B:2, 4, 1) 1 (A:1, 1, 4) )$



# Отображение геометрии в Orion



- ■ - UO<sub>2</sub>
- ■ - Zr
- ■ - H<sub>2</sub>O

```
pin      =  
CCS(0.5,0.6//fuel,clad)  
cell    = STR(("-$hp", "-  
$hp") ("-$hp", $hp) (  
$hp,$hp) ($hp,"-$hp") ("-$  
$rcl", 0 ) ( 0 , $rcl)  
($rcl, 0 ) ( 0 , "-$rcl") /  
4,cool / pin(0,0) /  
5,2,6,10,9,cool;  
6,3,7,11,10,cool;  
7,4,8,12,11,cool)
```

# Граничные условия для геометрии

- Оператор BDRY (от англ. **boundary** – граница)
- Использование:

*name* = **BDRY** ( (*c*<sub>1</sub>, *n*<sub>1</sub>[*l*] (*c*<sub>2</sub>], *n*<sub>2</sub>) *k* (*b*) [ / (*c*<sub>3</sub>, *n*<sub>3</sub>[*l*] (*c*<sub>4</sub>], *n*<sub>4</sub>) *k* (*b*) ]... )

где *name* представляет геометрическую систему, определенную оператором CNX;

*c*<sub>*i*</sub> задает структуру из *name* по соответствующей цепочке, *c*<sub>*i*</sub> = [*l*-]...*l*

*n* задает узел структуры, *n* = *n*<sub>*i*</sub>

*k* задает угловое распределение токов на границе.

*b* задает граничное условие, *b* = *aname*

или *b* = *c*, *n*[*l*] (*c*], *n*

где *l* – ссылка вдоль цепочки определяющая структуру — последняя ссылка — чьи узлы *n* используются для определения углового представления токов на границе;

*n*<sub>*i*</sub> – последовательность чисел узла в структуре, задаваемого пользователем;

*aname* – имя заданного пользователем альбеда (матрицы), или число 0 для зеркального отражения.

- Необходимо описать все ребра системы по следующему правилу:

A=BDRY ( (1, 1) (1, 2) 1 (White) / (1, 2) (1, 3) 1 (White) / (1, 3) (1, 1) 1 (White) )

где первое число в скобке обозначает номер системы (подсистемы), второе число обозначает номер узла в системе. 1 после скобки обозначает тип сопряжения между внутренней частью системы и наружностью. «White» – имя переменной, определенной оператором ALB.

# Расчет выгорания в HELIOS

Для расчета выгорания в HELIOS задаются операторы STAT и PATH

Оператор STAT определяет состояние (это полный набор конфигураций свойств, которые заполняют систему). Он также может включать в себя изотопные множители — по умолчанию или заданные пользователем — и уровень мощности.

Использование оператора STAT:

```
name = STAT (mos, dos, tos[, ix][, pw])
```

где *name* – имя состояния, которое может быть вычислено с помощью HELIOS.

*mos* - название набора для наложения материала — см. оператор OVSM.

*dos* - название набора для наложения плотности — см. оператор OVSD.

*tos* - название набора параметров с наложением температуры — см. оператор OVST.

*ix* - название комбинации изотопных множителей, определяемых оператором ISOX.

*pw* - уровень мощности системы в Вт/г (исходные тяжелые изотопы).

Пример:

```
st = STAT (osm, osd, ost, 38)
```

# Оператор PATH

Оператор PATH используется для указания, какие состояния (см. оператор STAT) должны вычисляться при каких уровнях выгорания и/или после каких этапов затухания.

Использование оператора PATH:

```
name = PATH([iis] / b [, b]... )
```

*name* представляет собой вычислительный путь, который должен быть выполнен HELIOS.

*iis* определяет исходные изотопы, полученные в реальном случае, или указывает изотопы, записанные ранее *iis* = *bname:eb*, *iis* = *dbname:eb*.

*b* указывает одну или несколько расчетных точек, *b*=[opt,] *stead*.

```
pa = PATH(/(st), 10000/10)
```

# Задание расчетных величин в HELIOS

- Оператор GROUP используется для определения структур групп при расчете выходных величин.
- Использование:

*name*=**GROUP** (NG/ [*g* [, *g*]...])

- *name* - представляет структуру группы для нейтронов и гамма-квантов.
- NG – или «N», или «G» – для нейтронов или гамма-квантов соответственно.
- *g* – нижняя граница (нижние границы) в эВ, в порядке убывания; если *g* не определено, *name* представляет все группы.
- Пример:

'N-gr.1' = GROUP (N/12)

где 'N-gr.1' обозначает единую группу нейтронов, которая включает все энергии. Согласно правилу № 4, нижний предел по библиотеке, например 10-5 эВ – даже при том, что в скобках указано 12 эВ.

# Оператор AREA

- Оператор AREA используется для объединения одной или нескольких частей геометрии или создания наборов таких объединений. Для таких объединенных частей геометрии результаты расчета указываются операторами MACRO и MICRO, после чего необходимые величины будут рассчитаны и записаны в базу данных. AREA является необязательным оператором.

Использование:

```
name = AREA ([<]aid [, aid][>] ...)
```

name — комбинация одного или нескольких областей, или набор регионов и/или комбинаций регионов.

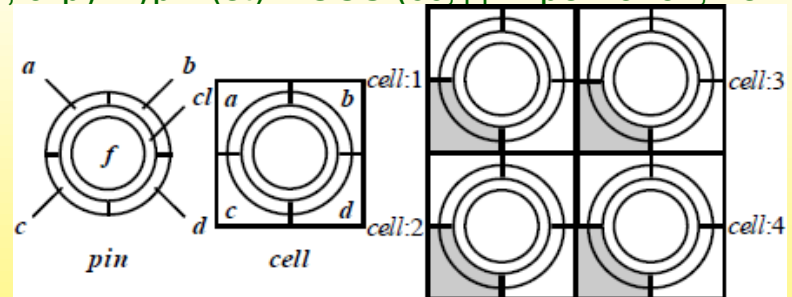
aid — идентификатор одного или нескольких регионов выбранной площадки:

```
aid = [ [<]ss - [ [<]ss - ] ... ] [<]st - [<]cc - [<]rg[>]
```

- обозначает один или несколько регионов (rg) как последняя ссылка на цепочку, которая проходит через подсистемы (ss), если они есть, структуры (st) и CCS (cc; для регионов, не содержащих CCS, cc=0).

Пример:

- SWcool = AREA( \*-<(0,pin)-c> )



# Оператор MACRO

## Назначение

Оператор MACRO используется для определения неизотропных свойств выбранных регионов, определенных оператором AREA , по структуре групп, определенной оператором GROUP. Эти данные сохраняются в базе данных для дальнейшей обработки выходной информации.

## Использование:

*name* = **MACRO** ([*T*] *gr*, *ar* / [*dt* [, *df*] ...])

где *name* представляет макро, т. е., один или несколько наборов данных о свойствах для выбранного региона (регионов) выбранной площади, данной площадки в выбранной структуре групп.

*T* - буква T для сохранения данных, усредненных по времени.

*gr* – имя структуры выбранных групп.

*ar* – имя площадки, т. е., один или несколько регионов.

*dt* – обозначает один или несколько типов величин; потоки сохраняются всегда, объемы сохраняются с другими данными оператора AREA.

## Пример:

'XS-ng2' = MACRO(T/ ng2,Assy/ab,fi,nf,kf,tr,p0)

- где *Assy* — набор регионов.

# Оператор MICRO

- Оператор MICRO используется для определения данных для частей геометрии, заданных оператором AREA, в структуре, определяемой оператором GROUP.

Спасибо за внимание