

Программное обеспечение для моделирования
физических процессов на установке SPD:

SpdRoot

Артур Ткаченко (ЛЯП)

12 октября 2017 г, ЛФВЭ ОИЯИ

SpdRoot: инсталляция.

Необходима предустановка **FairSoft, FairRoot**.

HOW TO INSTALL & RUN: spdroot/doc/Readme - локальная установка на ПК;
spdroot/doc/INSTALL - для работы на ферме ЛИТ ОИЯИ.

1. **FairSoft** - стандартный комплект программного обеспечения, включающий BOOST, GSL, ROOT, VMC ROOT, Geant, PYTHIA, HepMC, ...
 - Перед установкой FairSoft см. файл FairSoft/DEPENDENCIES.
 - В процессе установки должна быть выбрана версия ROOT (5 или 6).
2. **FairRoot** - пакет для физического моделирования и анализа (фактически - framework), расширяющий возможности VMC ROOT - интерфейса для работы с Geant(3&4) в среде ROOT.

Основная идея: возможность работы с отдельными частями установки - модулями (изменение геометрии, запись и обработка результатов моделирования) **НЕЗАВИСИМО** друг от друга.

3. **SpdRoot** - пакет программ для моделирования и анализа установки SPD, который базируется на FairRoot и имеет ту же структуру кода.
Пример пакетов на основе FairRoot, устроенных аналогичным способом:
CbmRoot, PandaRoot, MpdRoot, ...





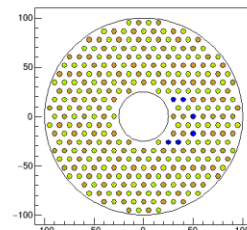
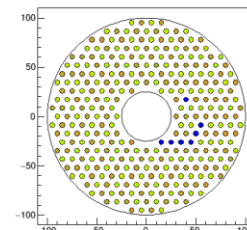
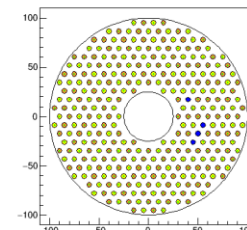
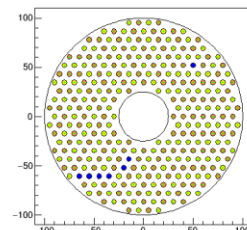
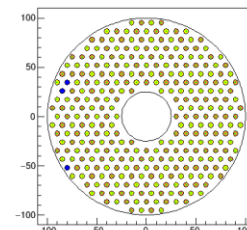
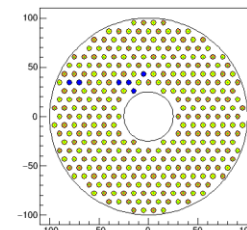
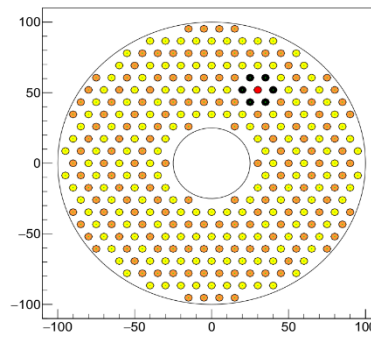
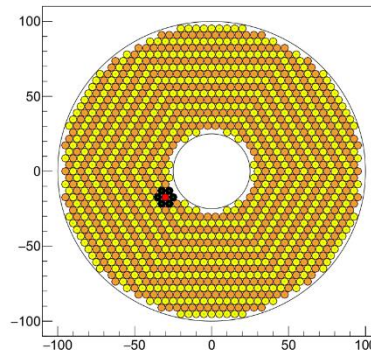
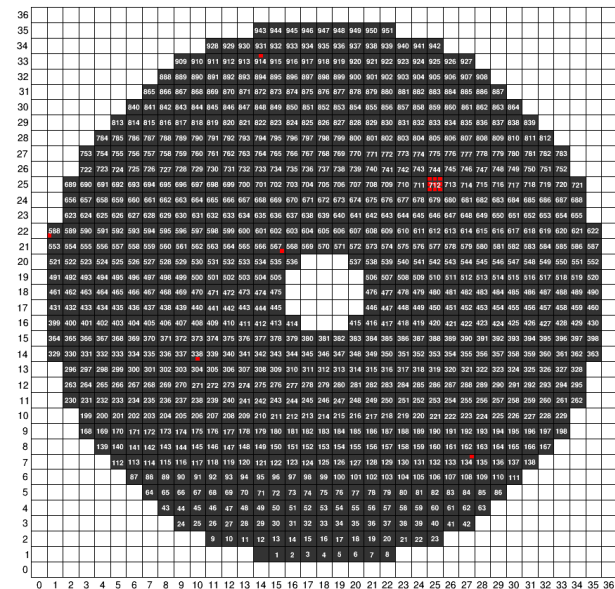
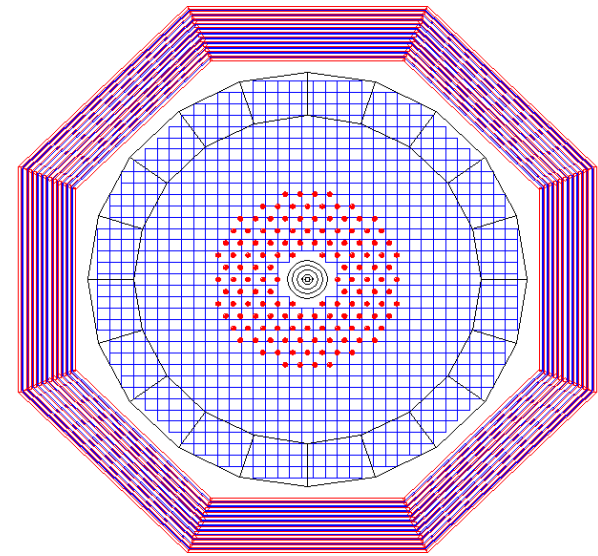
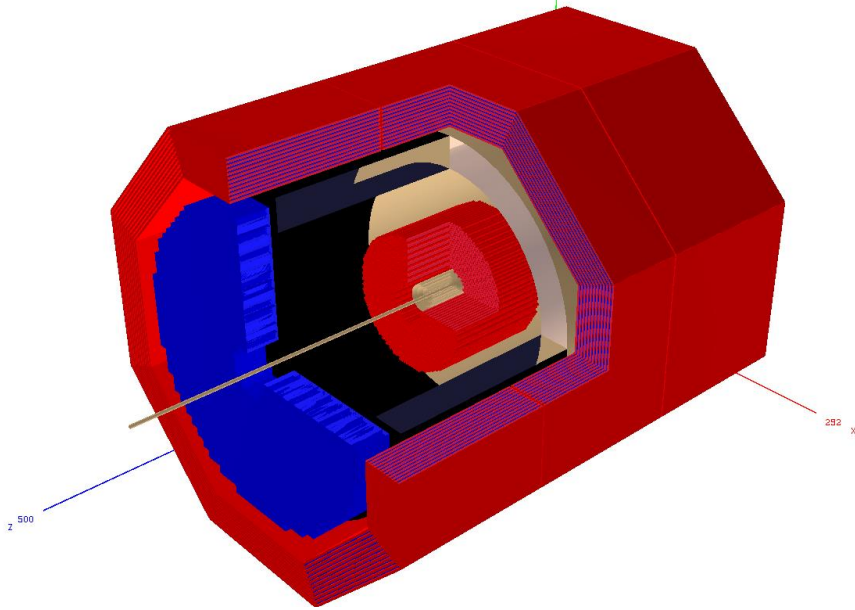
SpdRoot: список директорий.

| | |
|-----------------|--------------------------------------------------------------|
| doc/ | документация |
| tools/ | вспомогательные утилиты |
| scripts/ | скрипты для работы на ферме ЛИТ |
| data/ | файлы с входными данными |
| geometry/ | файлы с описанием материалов и геометрии детекторов |
| gconfig/ | конфигурационные файлы/скрипты для Geant |
| macro/ | ROOT - скрипты |
| field/ | код для работы с магнитным полем |
| spdgenerators/ | вершинные генераторы (Pythia6, Pythia8) |
| mpdgenerators/ | генераторы, портированные из mpdroot (MpdLAQGSM) |
| clustering/ | набор классов для задач кластеризации |
| test/ | набор классов для выполнения тестовых задач |
| passive/ | список “пассивных” модулей: Pipe, Magnet, ... |
| ect/ | end-cap tracking system |
| sts/ | вершинный детектор (silicon tracking stations, mpd-version) |
| emc/ | электромагнитный калориметр (центральная часть, mpd-version) |
| ecal/ | электромагнитный калориметр (end-cap части) |
| stt/ | главный трекер (straw tube tracker) |
| hmrs/ | мюонная система (hadron-muon range system) |
| spddata/ | контейнеры данных (треки, хиты, ...) |

А также:

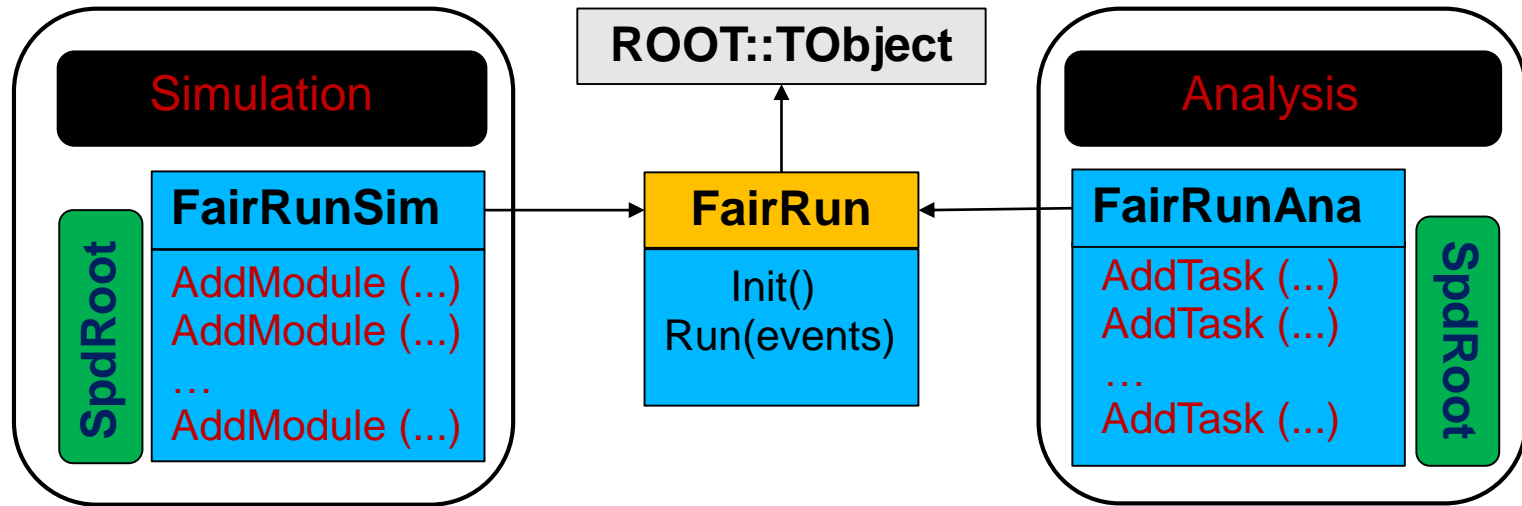
- примеры bash-скриптов, устанавливающих переменные окружения для начала работы в **SpdRoot**: SetEnv.sh, SetEnv_afs.sh, SetEnv_lxpub.sh;
- элементы системы сборки пакета (cmake).

Общий вид установки на данный момент.



Следы треков (энерговыведение) в главном трекаре (stt)

End-cap (ecal)



- 1) Моделирование выбранной конфигурации установки и сохранение результатов в формате *.root (TFile, TTree). Пример: macro/run_sim.C .

Результат выполнения скрипта: **run.root /TTree /TBranch's:**

| | |
|---------------------|----------------------------------------------------------|
| [FairMCEventHeader] | MCEventHeader (параметры события) |
| [TClonesArray] | SpdMCTrack (характеристики треков) |
| [TClonesArray] | SpdSttPoint, SpdStsPoint, MpdEmcPoint, SpdEcalPoint, ... |

params.root - параметры моделирования; в частности, параметры конфигураций (геометрии) отдельных модулей.

geofile_full.root - отдельный файл с геометрией установки, использованной при моделировании.

- 2) Анализ результатов моделирования: итерации по дереву TTree "event by event" (TFile, TTree, TChain) Примеры: macro/reco_ecal.C, macro/run_reco.C, ...

Параметры моделирования.

Задачи моделирования (via FairRunSim) и обработки/анализа (via FairRunAna) выполняются путем запуска соответствующего root-скрипта из директории `spdrroot/macro/` (например – `run_sim.C`).

Некоторые параметры моделирования, которые могут быть заданы в root-скрипте:

- имена output-файлов (<данные>.root, <параметры>.root);
- имя файла со списком материалов и их свойств ("<материалы>.geo");
- тип транспорт-генератора ("TGeant4", "TGeant3");
- список генераторов первичной вершины (один или более);
- конфигурации магнитного поля;
- **конфигурация установки** (список добавляемых модулей);
- **конфигурации отдельных частей установки** (ecal, stt);
- формат заголовка события (если необходимо;
 например информация о первичной вершине);
- формат output-файла (какую информацию сохранять?);
- количество событий для генерации;
- сохранение дополнительной информации о run'e
 (например, создание отдельного файла с геометрией установки);
- ...

Описание геометрии и сбор данных при моделировании для отдельных частей SPD.

Список и свойства материалов (базовых и композитных)

находится в файле `spdroot/geometry/media.geo`.

(<https://fairroot.gsi.de/?q=node/34>, HowTo / Detector Geometry and Media / Media file)

Пассивные части установки (пучковая трубка, магнит, элементы конструкции) находятся в директории `passive/` и описываются как объекты типа **FairModule**.

Активные (детектирующие) части установки (трекер, калориметр, ...) описываются как отдельные модули. Для каждого модуля отводится своя директория.

Основной объект, который одновременно является и «билдером» геометрии данной части SPD и интерфейсом для сбора данных при моделировании **FairDetector::FairModule**.

Два способа задать геометрию модуля:

1. ASCII files. Геометрия каждого модуля описана в отдельном файле.
(<https://fairroot.gsi.de/?q=node/32>, HowTo / Detector Geometry and Media / Geometry file format)
Чувствительные элементы детектора автоматически идентифицируются программой по материалу, для чего необходима соответствующая метка при описании материала в файле `*.geo`.

2. ROOT geometry. Описание производится непосредственно внутри модуля с использованием класса **TGeoManager**. Чувствительные элементы детектора должны быть заданы вручную в процессе конструирования геометрии модуля.

Пример: модуль esal.



| | |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| SpdEcal | FairDetector , основной класс, подключается в FairRunSim посредством метода FairRunSim::AddModule(FairDetector*) |
| SpdEcalGeoMapper | TObject , ключевой класс при описании геометрии, определяет список изменяемых параметров и обеспечивает навигацию по геометрии модуля |
| SpdEcalGeoPar | FairParGenericSet , содержит список параметров моделирования |
| SpdEcalPoint | FairMCPoint , информация, собираемая вдоль треков частиц в процессе моделирования |
| SpdEcalHitProducer | FairTask , класс для преобразования SpdEcalPoint в SpdEcalHit |
| SpdEcalHit | FairHit , информация, относящаяся к ячейкам калориметра |
| SpdEcalHitCont | TObject , дополнительная MC-информация, относящаяся к ячейкам калориметра |
| SpdEcalContFact | FairContFact , вспомогательный класс (обязателен) |
| SpdEcalGeo | FairGeoSet , вспомогательный класс, нужен при определении геометрии модуля в формате ascii |

clustering/

| | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| SpdEcalClManager | FairTask , основной класс, интерфейс задачи, подключается в FairRunAna посредством метода FairRunAna::AddTask(FairTask*) |
| SpdEcalPartClManager | ClManager , базовый класс для кластеризации |
| SpdEcalClusterElement | ClusterElement , элемент кластера в калориметре |
| SpdEcalCluster | Cluster , кластер - объединение ячеек калориметра |
| SpdEcalClusterCollection | ClusterCollection , объединения кластеров |
| SpdEcalClusterInfo | TObject , сохраняемая информация о кластерах (output) |
| SpdEcalCLinfo | TObject , параметры кластеризации и др. (output) |

Вопросы и комментарии.

1. Совместимость кода SpdRoot (компиляция) с ROOT5/ROOT6?
2. Использование стандарта C++11 и выше?
3. Стандарты описания геометрии модулей (ROOT/ASCII).
Отдельные модули для установки с различными типами магнита (соленоид/тороид)?
4. Администрирование:
 - поддержка репозитория для SpdRoot;
 - определение прав участников проекта для выполнения обновлений;
 - поддержка по внешним пакетам FairSoft и FairRoot, проверка обновлений и совместимости, компиляция;
 - вопросы компиляции SpdRoot;
 - свой репозиторий для FairSoft, FairRoot?
5. Координация:
 - вопросы по проекту в целом;
 - глобальные изменения архитектуры: перемещение кода, добавление новых директорий и т.п.;
 - технические вопросы по SpdRoot/FairRoot/ROOT/C++/...;
 - поддержка целостности кода и вопросы компиляции.
6. Конкретные исполнители по отдельным частям детектора.
7. Расположение самой новой версии FairSoft (feb 2017, ~6.5 Gb) на ферме ЛИТ: afs/jinr.ru/fairsoft ;
FairRoot на текущий момент устанавливается локально (рабочая версия FairRoot – spdroot/afs).