NEW MATERIAL MODEL INSTRUCTIONS

tkLayout developers meeting

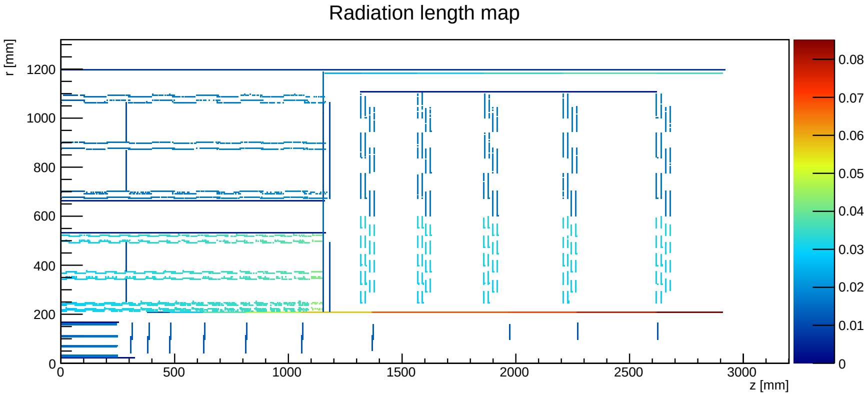
Stefano MARTINA

stefano.martina@cern.ch

European Organization for Nuclear Research

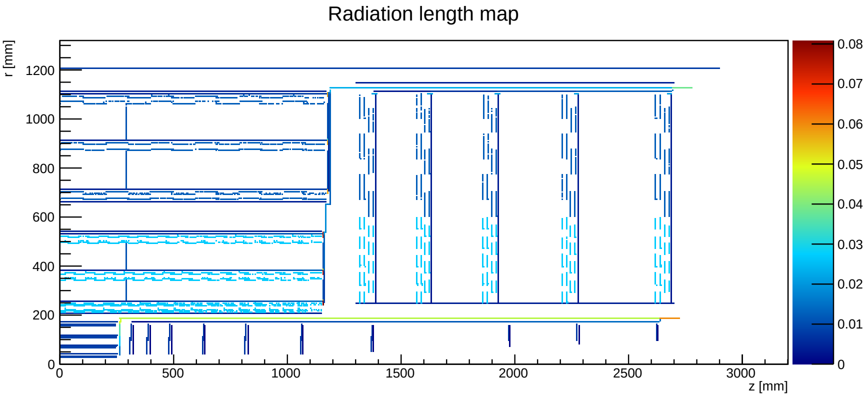
March 3, 2015

Old model



- ✓ Cables material distributed inside modules volumes
- ✓ Possible to model cooling pipe along rods, manifold in the flange and bigger cooling pipe out of the barrel

New model



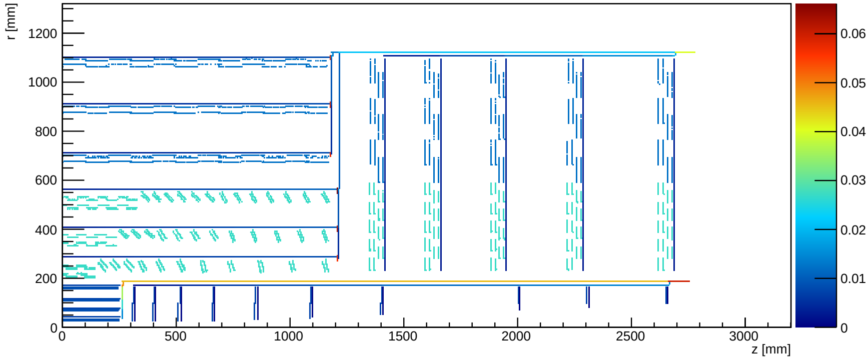
- ✓ Cables material in dedicated volumes
- ✓ More detailed
- ✓ Better routing algorithm
- ✓ More functionalities

Advantages

Correct description for tilted modules

- ✓ In old model the cables were distributed over the modules
  - Not feasible in case of tilted modules
- ✓ Now is possible to model this design

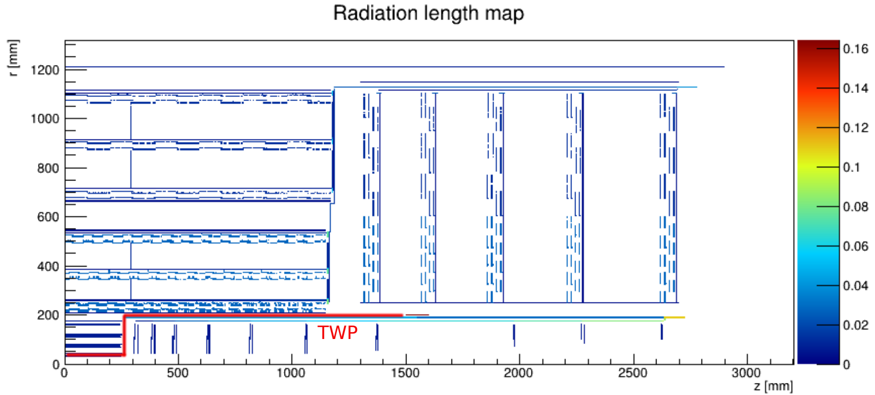
Radiation length map



New feature

Model for pixel-like materials

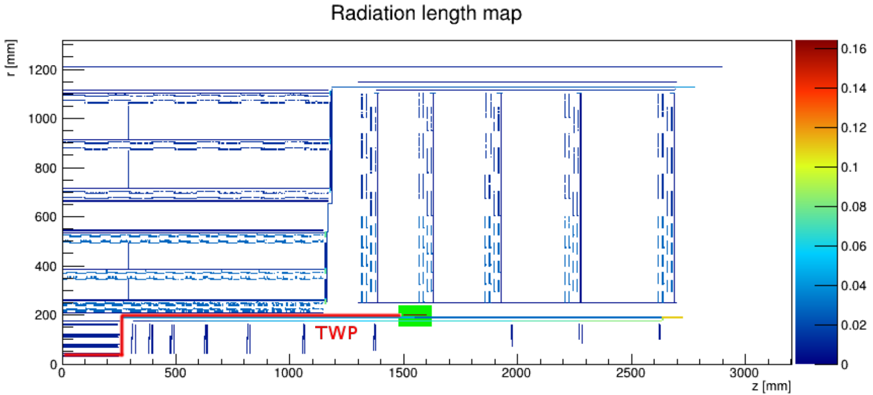
- ✓ For instance twisted pair from modules, electrical optical transducer, and optic fibers after it



New feature

Model for pixel-like materials

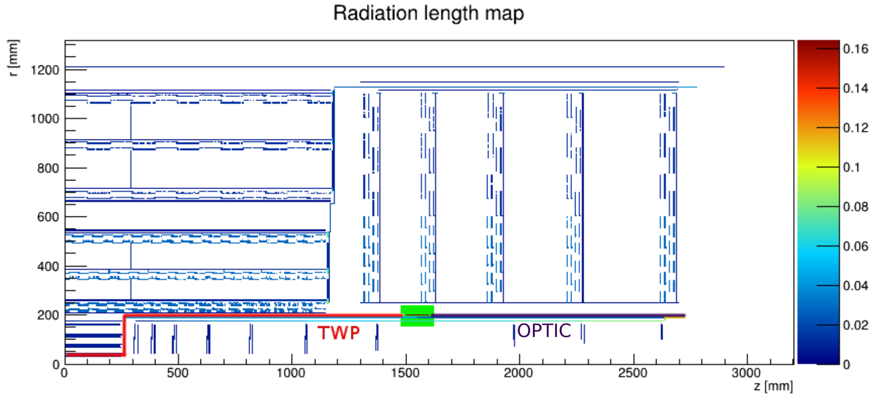
✓ For instance twisted pair from modules, electrical optical transducer, and optic fibers after it



New feature

Model for pixel-like materials

✓ For instance twisted pair from modules, electrical optical transducer, and optic fibers after it



Advantages

✓ The new algorithm use the same underlying C++ objects of the old

✓ This means that the XML export is working as usual

  - only more detailed than before

Configuration files

- ✓ `config/stdinclude/Materials/` material definition
- ✓ `config/stdinclude/Conversion/` conversions definition

- ✓ Include materials in geometries at layer/disk, rod or module levels, depending of the material type (or before)
- ✓ Include conversions in geometries at layer or disk level (or before)

Material

- `type` define the destination of material, can be:
  - `module`
  - `rod` only for barrels
  - `layer` also for the disks
- `ReferenceSensor` only for `module` material, define the reference sensor for scaling material (if specified), each one must have:
  - `numStripsAcross`
  - `numSegments`
- `Component` inside can have other `Component` or:
  - `Element` define a single material and have the properties:
    - `componentName` the name of inner component, not mandatory
    - `elementName` the name of the element, mandatory
    - `quantity` mandatory
    - `unit` the unit between `g/m`, `mm`, `g`, mandatory
    - `service` if the material is locally to the element or exiting from it, false by default
    - `scaleOnSensor` used only on `module` materials, if 0 (by default) the element don't scale on sensor, otherwise scale on the specified sensor
    - `referenceSensor` used only on `module` materials with `scaleOnSensor` active, specify the sensor of reference between the list provided before
    - `targetVolume` used only on `module` materials, specify the target volume index inside the module (sensor, hybrids, etc..), 0 by default
    - `destination` if defined specify the second level conversion on wich the element is converted
    - `debugInactivate` specify if deactivating the element or not for debugging, false by default

Conversion

- `stationName` is the identifier, used also for the `destination` property, is mandatory
- `type` specify the kind of station, is mandatory and can be `flange` or `second`
- `minZ` used only for `second` stations, specify the position
- `maxZ` as the previous
- `Conversion` can be more than one and is constituted by:
  - `Input` mandatory, only one for `Conversion` and have
    - `Element` mandatory, only one for `Input`, is the element to be converted, have:
    - `elementName`
    - `quantity`
    - `unit`
  - `Output` mandatory, only one for `Conversion` and have
    - `Element` not mandatory, can be more than one, is the result of the conversion, have:
    - `elementName`
    - `quantity`
    - `unit`
    - `service` if false (by default) the material go inside the conversion object (flange or custom for second), if true go out

	Unit=g/m	Unit=mm	Unit=g
Module Service=false	Module $\times moduleLength$ No accumulation No conversion Scaling possible	Module $\times moduleSurface \times \rho$ (sensor surface) No accumulation No conversion Scaling possible	Module $\times 1$ No accumulation No conversion Scaling possible
Module ring R of N Service=true	Following supports $S_{R+1} \dots S_i \dots S_N$ $\times numModules_R \times supportLength_i$ Accumulation Conversion(1:1 by default, with warning) Scaling possible	Following supports $S_{R+1} \dots S_i \dots S_N$ $\times numModules_R \times supportSurface_i \times \rho$ Accumulation Conversion(1:1 by default, with warning) Scaling possible Deprecated warning	Error

Rod(barrel)
Service=false
	All supports $S_1 \dots S_i \dots S_N$
	$\times numModules_1 \times supportLength_i$
	No accumulation
	No conversion
	Scaling not possible
		All supports $S_1 \dots S_i \dots S_N$
		$\times supportSurface_i \times \rho$
		No accumulation
		No conversion
		Scaling not possible
			All supports $S_1 \dots S_i \dots S_N$
			$\times numModules_1 \times \frac{supportLength_i}{\sum_{j=1}^{N} supportLength_j}$
			No accumulation
			No conversion
			Scaling not possible

Rod(barrel)
Service=true
	All supports $S_1 \dots S_i \dots S_N$
	$\times numModules_1 \times supportLength_i$
	No accumulation
	Conversion
	Scaling not possible
		All supports $S_1 \dots S_i \dots S_N$
		$\times supportSurface_i \times \rho$
		No accumulation
		Conversion
		Scaling not possible
		Deprecated warning
			Error

Layer/Disk Service=false	All supports $S_1 \dots S_i \dots S_N$ $\times supportLength_i$ No accumulation No conversion Scaling not possible	All supports $S_1 \dots S_i \dots S_N$ $\times supportSurface_i \times \rho$ No accumulation No conversion Scaling not possible	All supports $S_1 \dots S_i \dots S_N$ $\times \frac{supportLength_i}{\sum_{j=1}^{N} supportLength_j}$ No accumulation No conversion Scaling not possible
Layer/Disk Service=true	All supports $S_1 \dots S_i \dots S_N$ $\times supportLength_i$ No accumulation Conversion Scaling not possible	All supports $S_1 \dots S_i \dots S_N$ $\times supportSurface_i \times \rho$ No accumulation Conversion Scaling not possible Deprecated warning	Error

Example

geometries/.../TechnicalProposal2014_Types.cfg

```
Barrel TBPS {
  @includestd ModuleTypes/ptPS
  @includestd Conversions/flange
  Layer 1 {
    triggerWindow 5
    dsDistance 2.6
  }

...
```

config/stdinclude/ModuleTypes/ptPS

```
...

@includestd Materials/ptPS
@includestd Materials/rodPtPS
```

Example

.../Materials/ptPS

```
Materials module-ptPS {
  type module

  // Default sensor:
  ReferenceSensor 1 {
    numStripsAcross 960
    numSegments 32
  }
  ReferenceSensor 2 {
    numStripsAcross 960
    numSegments 2
  }

  // Sensor
  Component {
    componentName Sensor
    service false
    scaleOnSensor 0
    targetVolume 1
    Element {
      elementName SenSi
      quantity 0.2
      unit mm
    }
  }


...
```

.../Materials/rodPtPS

```
Materials rodPtPS {
  type rod

  Component {
    componentName Cooling
    service true
    scaleOnSensor 0
    Element {
      elementName Steel
      quantity 7.860696517
      unit g/m
    }
    Element {
      elementName CO2
      quantity 1.791044776
      unit g/m
    }
  }


...
```

Example

.../Conversions/flange

```
Station {
  stationName flange
  type flange
...
  Conversion {
    Input {
      Element {
        elementName Cu_MV
        quantity 10
        unit g/m
      }
    }
    Output {
      Element {
        elementName Cu
        quantity 10
        unit g/m
        service true
      }
      Element {
        elementName Cu
        quantity 0.423
        unit g
        service false
      }


...
```

.../Conversions/endcap1

```
Station {
  stationName endcap1
  type second

  minZ 1500
  maxZ 1600

  Conversion {


...
```

Supports

Custom supports

```
Support {
  type custom
  customZMin 2300
  customRMin 200
  customLength 600
  customDir vertical
  Component {
    componentName supp
    Element {
      elementName Steel
      quantity 1000000
      unit g
    }
  }
}
```

Supports

Top/bottom supports

```
Support {
  type top //or bottom

  Component {
...
```

Auto supports

```
Support {
  type auto
  autoPosition 500

  Component {
...
```