Geant 4

Detector Description - Materials

Gabriele Cosmo, CERN/IT

Geant4 Users' Workshop, CERN 11-15 November 2002

Materials

- *The System of units & constants*
- *Definition of elements*
- *Materials and mixtures*
- *Some examples ...*

Unit system

- Geant4 has no default unit. To give a number, unit must be “multiplied” to the number.
  - for example :

    ```
    G4double width = 12.5*m;
    G4double density = 2.7*g/cm3;
    ```

  - If no unit is specified, the *internal* G4 unit will be used, but this is discouraged !
  - Almost all commonly used units are available.
  - The user can define new units.
  - Refer to CLHEP: `SystemOfUnits.h`
- Divide a variable by a unit you want to get.

    ```
    G4cout << dE / MeV << “ (MeV)” << G4endl;
    ```

HEP system of Units

- System of units are defined in CLHEP, based on:
  - millimetre (**mm**), nanosecond (**ns**), Mega eV (**MeV**), positron charge (**eplus**) degree Kelvin (**kelvin**), the amount of substance (**mole**), luminous intensity (**candela**), radian (**radian**), steradian (**steradian**)
- All other units are computed from the basic ones.
- In output, Geant4 can choose the most appropriate unit to use. Just specify the *category* for the data (Length, Time, Energy, etc…):

```
G4cout << G4BestUnit(StepSize, “Length”);
```

StepSize will be printed in km, m, mm or … fermi, depending on its value

Defining new units

- New units can be defined directly as constants, or (suggested way) via `G4UnitDefinition`.
  - `G4UnitDefinition ( name, symbol, category, value )`
- Example (mass thickness):
  - ```
    G4UnitDefinition ("grammpercm2", "g/cm2",
                      "MassThickness", g/cm2);
    ```
  - The new category "MassThickness" will be registered in the kernel in **G4UnitsTable**
- To print the list of units:
  - From the code

    ```
    G4UnitDefinition::PrintUnitsTable();
    ```
  - At run-time, as UI command:

    ```
    Idle> /units/list
    ```
    ```

Definition of Materials

- Different kinds of materials can be defined:
  - isotopes <> G4Isotope
  - elements <> G4Element
  - molecules <> G4Material
  - compounds and mixtures <> G4Material
- Attributes associated:
  - temperature, pressure, state, density

Isotopes, Elements and Materials

- **G4Isotope** and **G4Element** describe the properties of the *atoms*:
  - Atomic number, number of nucleons, mass of a mole, shell energies
  - Cross-sections per atoms, etc...
- **G4Material** describes the *macroscopic* properties of the matter:
  - temperature, pressure, state, density
  - Radiation length, absorption length, etc...

Elements & Isotopes

- Isotopes can be assembled into elements

```
G4Isotope (const G4String& name,
                 G4int      z,     // number of atoms
                 G4int      n,     // number of nucleons
                 G4double   a );   // mass of mole
```

- ... building elements as follows:

```
G4Element (const G4String& name,
           const G4String& symbol, // element symbol
                 G4int      nIso ); // # of isotopes
G4Element::AddIsotope(G4Isotope* iso,  // isotope
                      G4double relAbund); // fraction of atoms
                                          // per volume
```

Material of one element

- Single element material

```
G4double density = 1.390*g/cm3;
G4double a = 39.95*g/mole;
G4Material* lAr =
 new G4Material("liquidArgon",z=18.,a,density);
```

- Prefer low-density material to vacuum

Material: molecule

- A Molecule is made of several elements (composition by number of atoms):

```
a = 1.01*g/mole;
G4Element* elH  =
   new G4Element("Hydrogen",symbol="H",z=1.,a);
a = 16.00*g/mole;
G4Element* elO  =
   new G4Element("Oxygen",symbol="O",z=8.,a);
density = 1.000*g/cm3;
G4Material* H2O =
   new G4Material("Water",density,ncomp=2);
H2O->AddElement(elH, natoms=2);
H2O->AddElement(elO, natoms=1);
```

Material: compound

- Compound: composition by fraction of mass

```
a = 14.01*g/mole;
G4Element* elN  =
   new G4Element(name="Nitrogen",symbol="N",z= 7.,a);
a = 16.00*g/mole;
G4Element* elO  =
   new G4Element(name="Oxygen",symbol="O",z= 8.,a);
density = 1.290*mg/cm3;
G4Material* Air =
   new G4Material(name="Air",density,ncomponents=2);
Air->AddElement(elN, 70.0*perCent);
Air->AddElement(elO, 30.0*perCent);
```

Material: mixture

- Composition of compound materials

```
G4Element* elC  = …;    // define "carbon" element
G4Material* SiO2 = …;   // define "quartz" material
G4Material* H2O = …;    // define "water" material

density = 0.200*g/cm3;
G4Material* Aerog =
   new G4Material("Aerogel",density,ncomponents=3);
Aerog->AddMaterial(SiO2,fractionmass=62.5*perCent);
Aerog->AddMaterial(H2O ,fractionmass=37.4*perCent);
Aerog->AddElement (elC ,fractionmass= 0.1*perCent);
```

Example: gas

- It may be necessary to specify temperature and pressure
  - (dE/dx computation affected)

```
G4double density = 27.*mg/cm3;
G4double temperature = 325.*kelvin;
G4double pressure = 50.*atmosphere;

G4Material* CO2 =
   new G4Material("CarbonicGas", density, ncomponents=2
                  kStateGas, temperature, pressure);
CO2->AddElement(C,natoms = 1);
CO2->AddElement(O,natoms = 2);
```

Example: vacuum

- Absolute vacuum does not exist. It is a gas at very low density !
  - Cannot define materials composed of multiple elements through Z or A, or with $\rho = 0$.

```
G4double atomicNumber = 1.;
G4double massOfMole = 1.008*g/mole;
G4double density = 1.e-25*g/cm3;
G4double temperature = 2.73*kelvin;
G4double pressure = 3.e-18*pascal;
G4Material* Vacuum =
   new G4Material(“interGalactic", atomicNumber,
                  massOfMole, density, kStateGas,
                  temperature, pressure);
```