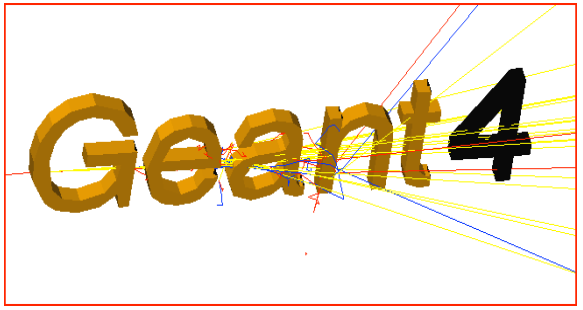


Material Definition

I. Hrivnacova, IPN Orsay

Geant4 tutorial for ED MIPEGE,
13 - 24 May 2013, Orsay

Acknowledgment

- Most of the slides shown were originally created for previous Geant4 tutorials and provided with courtesy for this course
- We wish to thank our Geant4 colleagues for allowing us to re-use their material
- The credits for slides re-used in this presentation
  - I. Hrivnacova, IPN Orsay; T. Nikitina, CERN; G4 tutorial at Annecy, 2008 with acknowledgment to J.Apostolakis, G.Cosmo, A. Lechner, CERN
  - S.Incerti, CENBG, G4 tutorial at KISTI, 2012

Outline

- *The System of units & constants*
- *Definition of elements*
- *Materials and mixtures*
- *NIST Database*

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;
```

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

- Element can be defined by name, symbol, atomic number and mass of mole:

```
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);
```

Elements & Isotopes

- When necessary, elements can be built from pre-defined isotopes:

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

- ... 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);
```

- Avoid “effective” average A, Z with hadronic physics !
  - Cross-section are not a function of material properties, but a function of nuclear properties.
    → At least correct set of elements is needed
- 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 and elements by mass fraction

```
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);
```

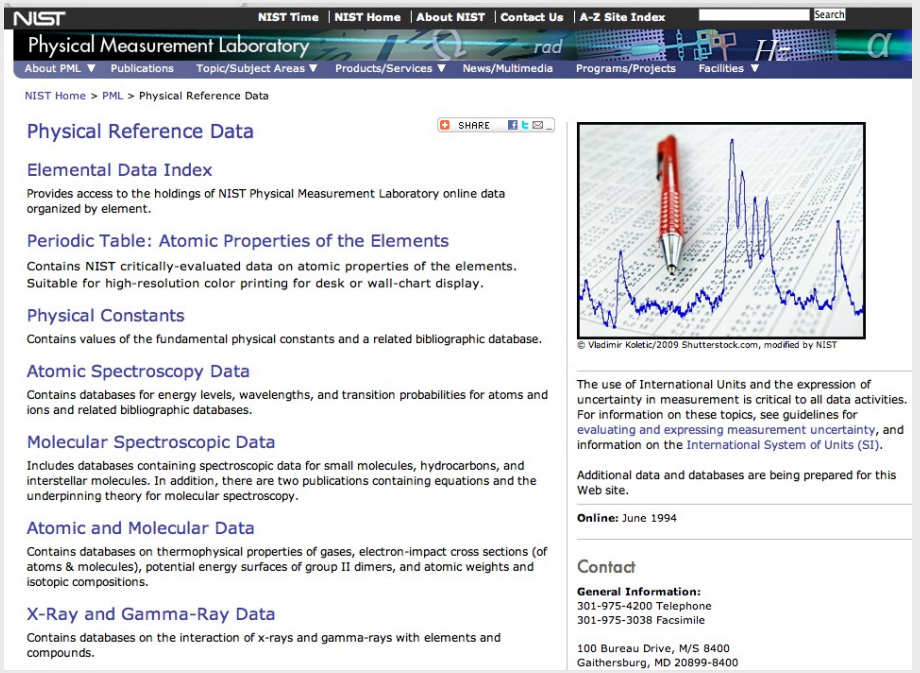
Example: vacuum

- 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);
```

NIST Database

- NIST database:
  http://physics.nist.gov/PhysRefData
- Guarantees accuracy in major parameters:
  - Density, Mean excitation potential, Elemental composition, Isotopic composition, ...
- Imported inside Geant4



NIST materials in Geant4

Elements

```
=================================================
###   Elementary Materials from the NIST Data Base
=================================================
 Z   Name  ChFormula   density(g/cm^3)  I(eV)
=================================================
1    G4_H    H_2        8.3748e-05       19.2
2    G4_He              0.000166322      41.8
3    G4_Li              0.534            40
4    G4_Be              1.848            63.7
5    G4_B               2.37             76
6    G4_C               2                81
7    G4_N    N_2        0.0011652        82
8    G4_O    O_2        0.00133151       95
9    G4_F               0.00158029       115
10  G4_Ne               0.000838505      137
11  G4_Na               0.971            149
12  G4_Mg               1.74             156
13  G4_Al               2.6989           166
14  G4_Si               2.33             173
```

Compounds

```
=================================================
###    Compound Materials from the NIST Data Base
=================================================
 N  Name      ChFormula      density(g/cm^3)  I(eV)
=================================================
13  G4_Adipose_Tissue       0.92             63.2
          1      0.119477
          6      0.63724
          7      0.00797
          8      0.232333
         11      0.0005
         12      2e-05
         15      0.00016
         16      0.00073
         17      0.00119
         19      0.00032
         20      2e-05
         26      2e-05
         30      2e-05
4   G4_Air                  0.00120479       85.7
          6      0.000124
          7      0.755268
          8      0.231781
         18      0.012827
2   G4_CsI                  4.51             553.1
         53      0.47692
         55      0.52308
```

Example: material from NIST

- Element/Material is retrieved from Geant4 material database by its name:

```
G4NistManager* manager = G4NistManager::Instance();
G4Element* elC
  = mananger->FindOrBuildElement("G4_C");
G4Material* matWater
  = mananager->FindOrBuildMaterial("G4_WATER");
```

- The list of currently available material names can be found in the Geant4 User's Guide for Application Developers
  - Appendix 10: Geant4 Material Database
  - The list is permanently being extended

Example: material from NIST (2)

- A compound can be built by specifying a vector of atomic numbers and weights using

```
G4NistManager::ConstructNewMaterial(
                    const G4String& name,
                    const std::vector<G4int>& Z,
                    const std::vector<G4double>& weight,
                    G4double density, G4bool iso);
```

- Isotope masses are accessible using

```
G4NistManager::GetMass(G4int Z, G4int N);
```