

# 1 Simulation configuration

**Pencil** beam

Proton : 200MeV

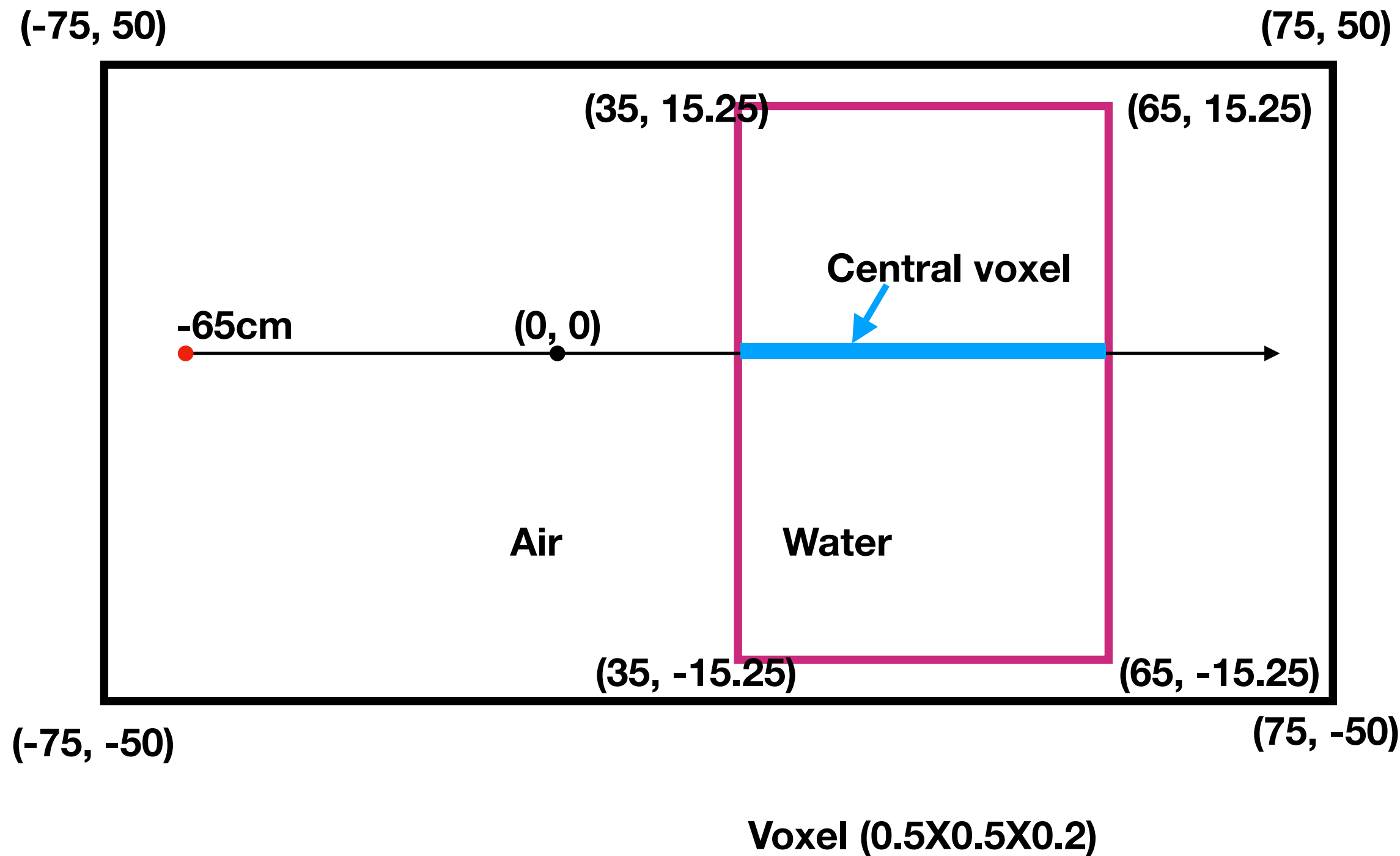
Mean excitation energy: 77eV

Geant4 version: 10.2.p03, 10.5.p01, 10.6

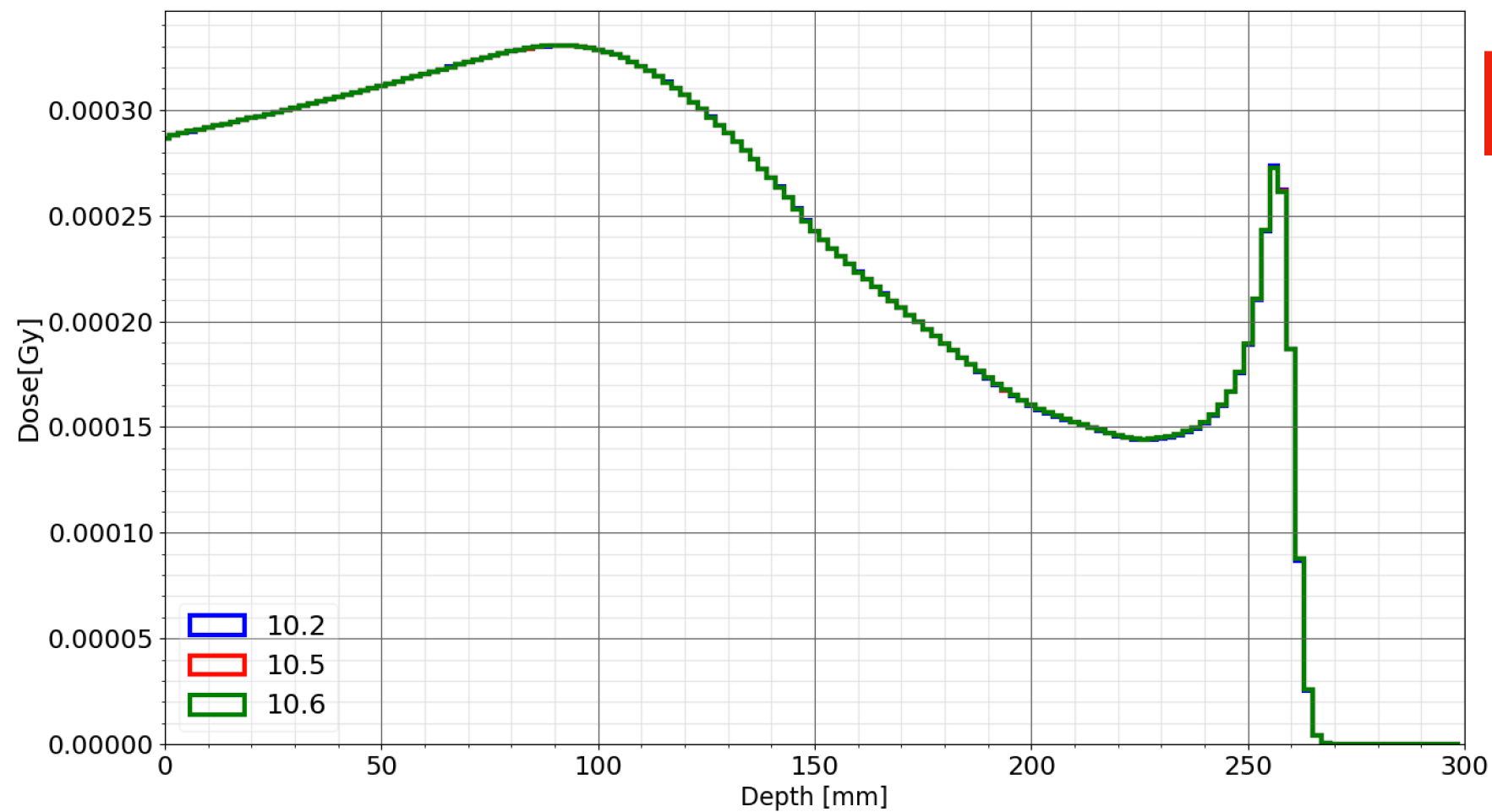
Events: 1M

# Geometry setup

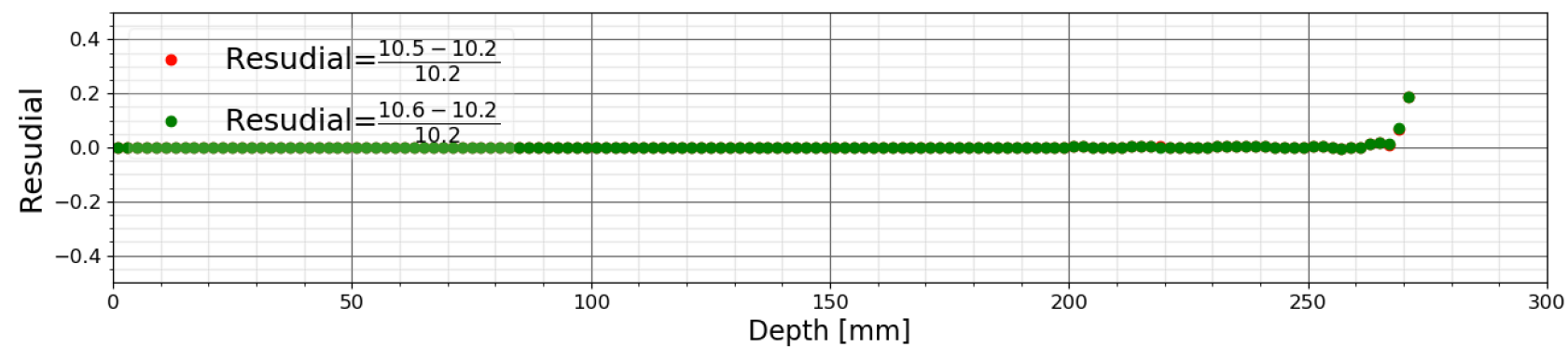
unit : cm



# Central voxel

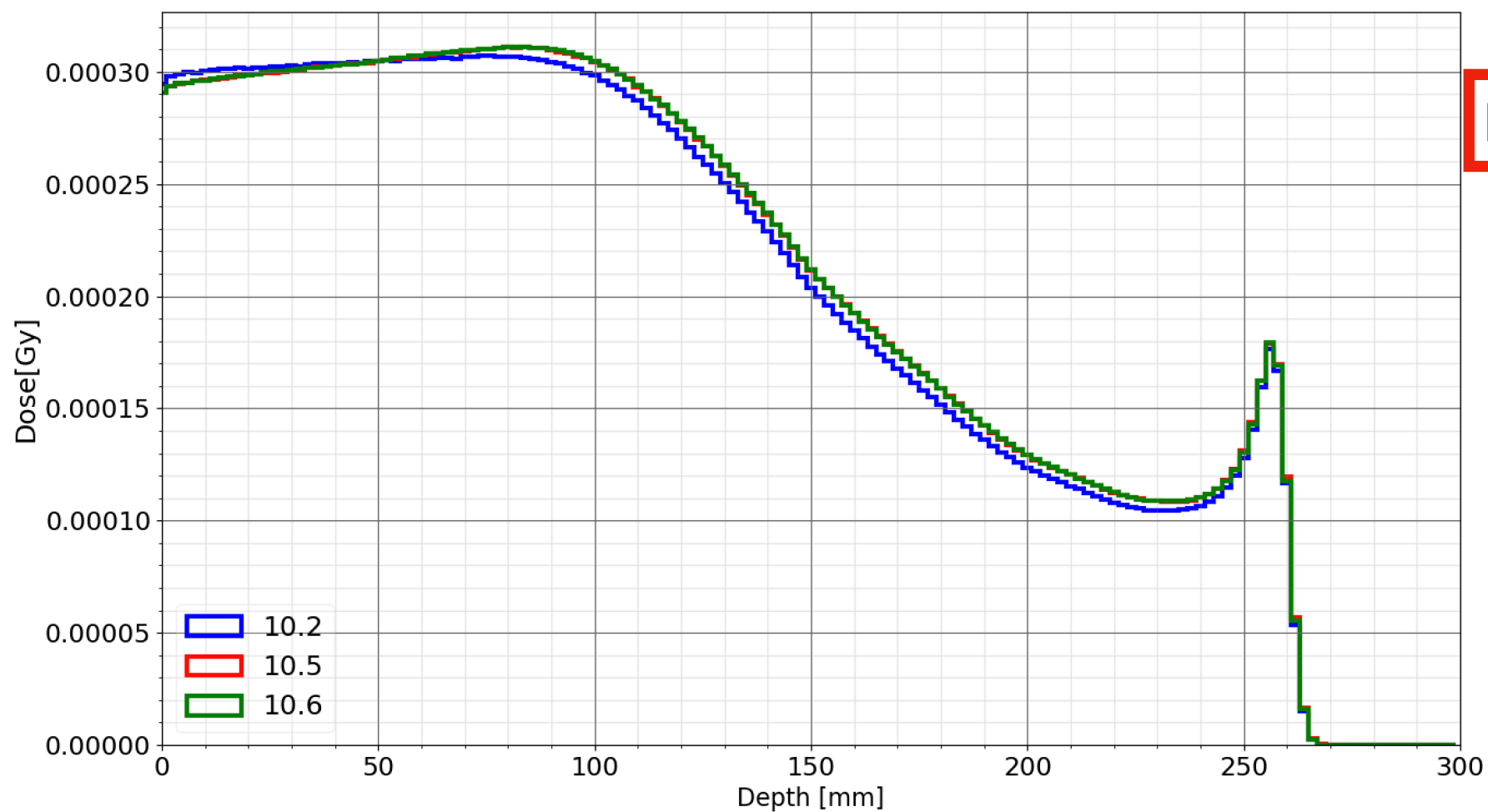


Only EM processes

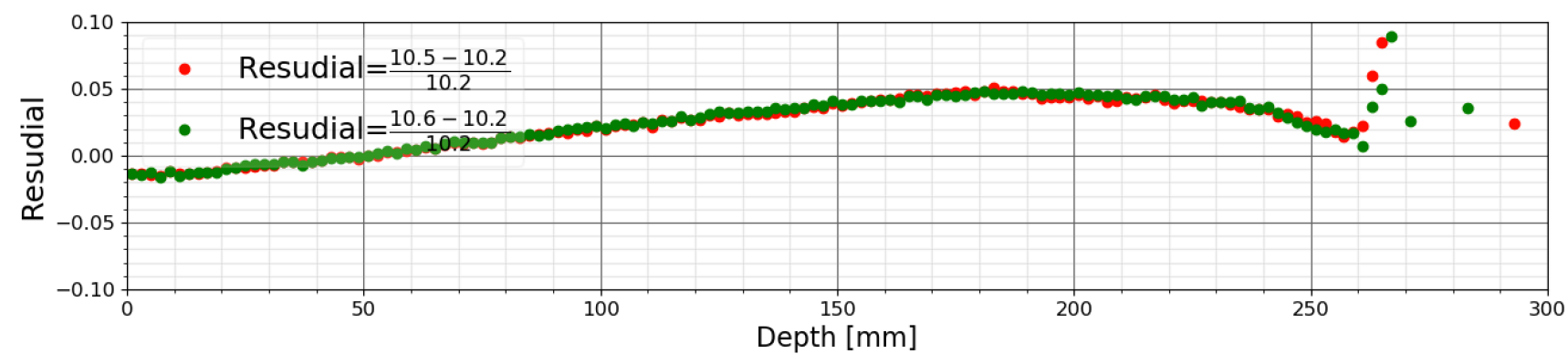


$$\text{Residual} = \frac{10.6 - 10.2}{10.2}$$

# Central voxel



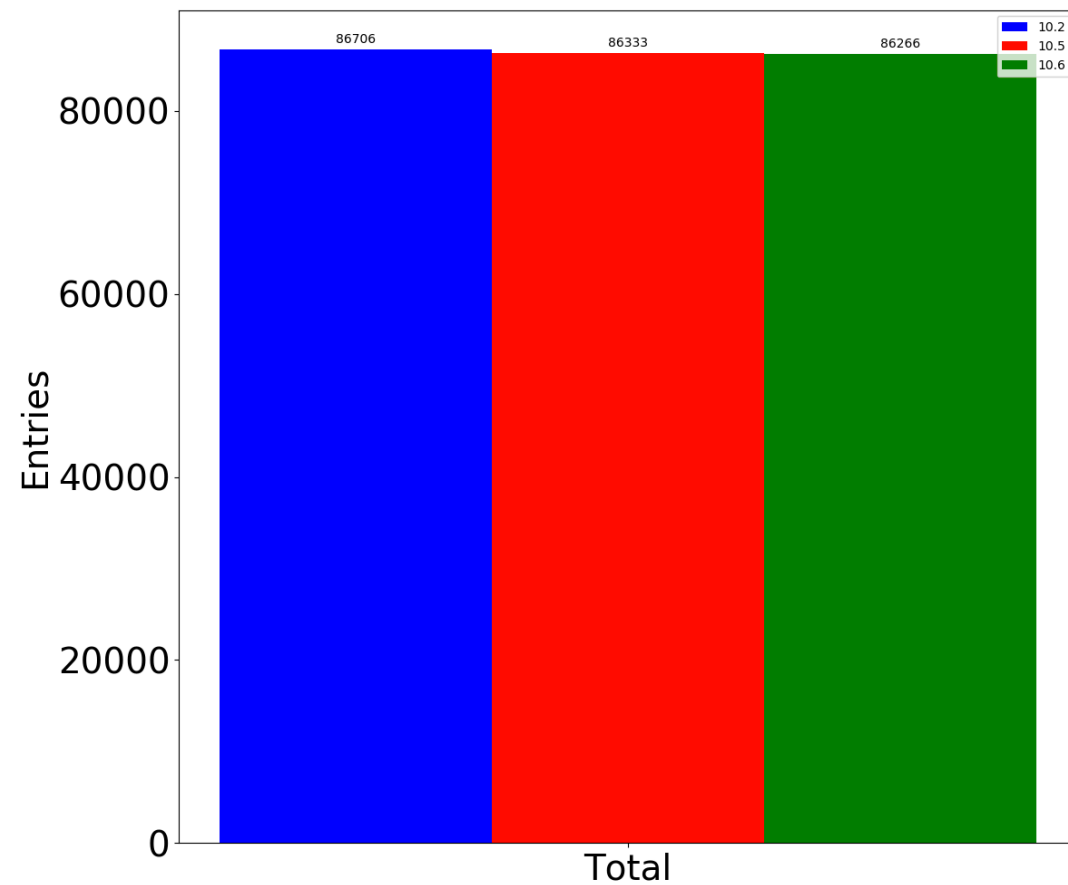
**EM + hadronic processes**



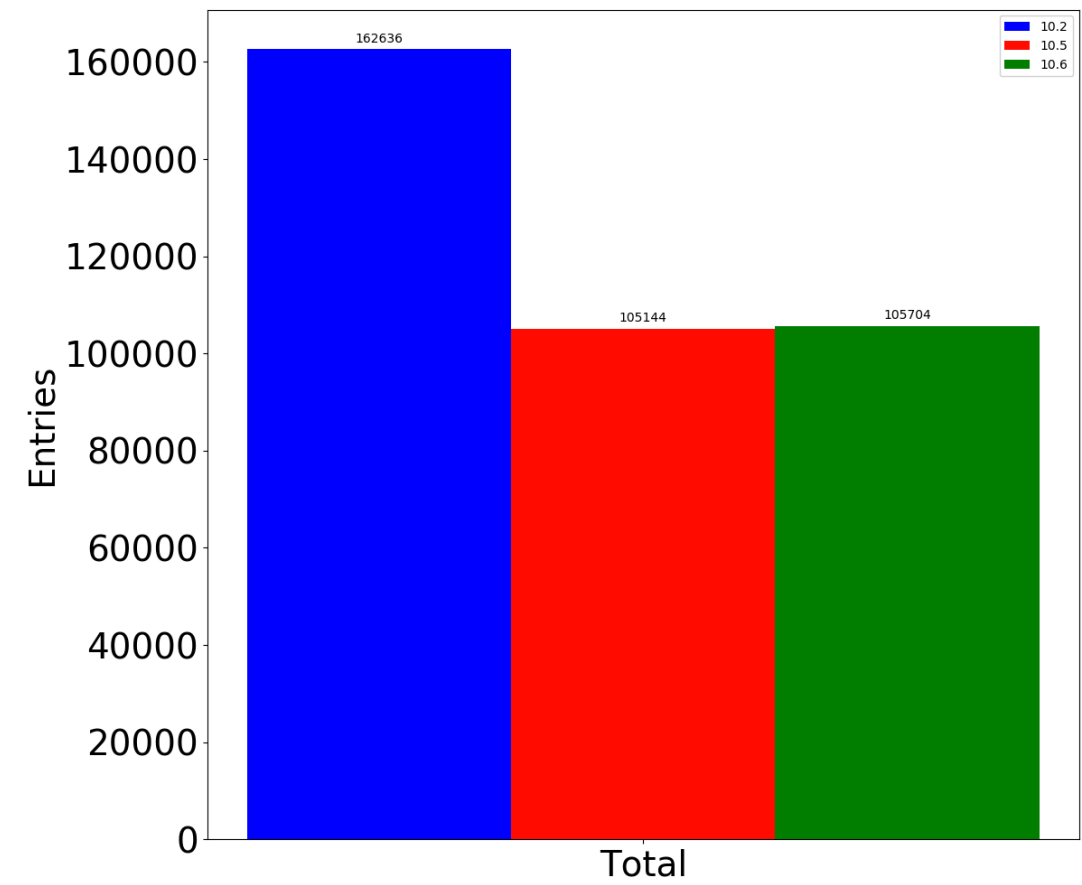
$$\text{Residual} = \frac{10.6 - 10.2}{10.2}$$

# Hadronic cross section

## Elastic



## Inelastic



Using Gheisha data set

```

38 G4HadronInelasticDataSet::G4HadronInelasticDataSet(const G4String& nam)
39 : G4VCrossSectionDataSet(nam)
40 {
41     theHadronCrossSections = G4HadronCrossSections::Instance();
42 }
43
44
45 G4HadronInelasticDataSet::~G4HadronInelasticDataSet() {}
46
47
48 void G4HadronInelasticDataSet::CrossSectionDescription(std::ostream& outFile) const
49 {
50     outFile << "G4HadronInelasticDataSet contains inelastic cross sections\n"
51             << "for all long-lived hadrons at all incident energies.  It was\n"
52             << "developed as part of the Gheisha hadronic package\n"
53             << "by H. Fesefeldt, and consists of a set of parameterizations\n"
54             << "of inelastic scattering data.\n";
55 }
56
57
58 G4bool
59 G4HadronInelasticDataSet::IsElementApplicable(const G4DynamicParticle* aParticle,
60         G4int /*Z*/,
61         const G4Material*)
62 {
63     return theHadronCrossSections->IsApplicable(aParticle);
64 }
65
66 G4double
67 G4HadronInelasticDataSet::GetElementCrossSection(const G4DynamicParticle* aParticle
68         G4int Z,
69         const G4Material*)
70 {
71     G4int A = G4rint(G4NistManager::Instance()->GetAtomicMassAmu(Z));
72     return theHadronCrossSections->GetInelasticCrossSection(aParticle, Z, A);
73 }

```

10.2.3

## Inelastic data set

```

39 G4HadronInelasticDataSet::G4HadronInelasticDataSet(const G4String& nam)
40 : G4VCrossSectionDataSet(nam), theZ(0), fInelasticXS(0.0),
41     fKinEnergy(0.0), fParticle(nullptr)
42 {
43     fGheishaXS = G4HadronCrossSections::Instance();
44     fNIST = G4NistManager::Instance();
45 }
46
47 G4HadronInelasticDataSet::~G4HadronInelasticDataSet() {}
48
49 void G4HadronInelasticDataSet::CrossSectionDescription(std::ostream& outFile) const
50 {
51     outFile << "G4HadronInelasticDataSet contains inelastic cross sections\n"
52             << "for all long-lived hadrons at all incident energies.  It was\n"
53             << "developed as part of the Gheisha hadronic package\n"
54             << "by H. Fesefeldt, and consists of a set of parameterizations\n"
55             << "of inelastic scattering data.\n";
56 }
57
58 G4bool
59 G4HadronInelasticDataSet::IsElementApplicable(const G4DynamicParticle*,
60         G4int, const G4Material*)
61 {
62     return true;
63 }
64
65 G4double
66 G4HadronInelasticDataSet::GetElementCrossSection(
67     const G4DynamicParticle* aParticle, G4int Z, const G4Material*)
68 {
69     G4double ekin = aParticle->GetKineticEnergy();
70     const G4ParticleDefinition* pd = aParticle->GetDefinition();
71     if(Z != theZ || ekin != fKinEnergy || pd != fParticle) {
72         theZ = Z;
73         fKinEnergy = ekin;
74         fParticle = pd;
75         G4int A = G4rint(fNIST->GetAtomicMassAmu(Z));
76         fInelasticXS = fGheishaXS->GetElasticCrossSection(aParticle, Z, A);
77     }
78     return fInelasticXS;
79 }

```

10.5/10.6

10.2.3

## Elastic data set

```

40 G4HadronElasticDataSet::G4HadronElasticDataSet(const G4String& nam)
41 : G4VCrossSectionDataSet(nam)
42 {
43     theHadronCrossSections = G4HadronCrossSections::Instance();
44 }
45
46
47 G4HadronElasticDataSet::~G4HadronElasticDataSet() {}
48
49
50 void G4HadronElasticDataSet::CrossSectionDescription(std::ostream& outFile) const
51 {
52     outFile << "G4HadronElasticDataSet contains elastic cross sections for\n"
53             << "all long-lived hadrons at all incident energies. It was\n"
54             << "developed as part of the Gheisha hadronic package\n"
55             << "by H. Fesefeldt, and consists of a set of parameterizations\n"
56             << "of elastic scattering data.\n";
57 }
58
59
60 G4bool
61 G4HadronElasticDataSet::IsElementApplicable(const G4DynamicParticle* aParticle,
62                                             G4int /*Z*/,
63                                             const G4Material*)
64 {
65     return theHadronCrossSections->IsApplicable(aParticle);
66 }
67
68 G4double
69 G4HadronElasticDataSet::GetElementCrossSection(const G4DynamicParticle* aParticle,
70                                             G4int Z,
71                                             const G4Material*)
72 {
73     G4int A = G4rint(G4NistManager::Instance()->GetAtomicMassAmu(Z));
74     return theHadronCrossSections->GetElasticCrossSection(aParticle, Z, A);
75 }

```

10.5/10.6

```

41 G4HadronElasticDataSet::G4HadronElasticDataSet(const G4String& nam)
42 : G4VCrossSectionDataSet(nam), theZ(0), fElasticXS(0.0),
43   fKinEnergy(0.0), fParticle(nullptr)
44 {
45     fGheishaXS = G4HadronCrossSections::Instance();
46     fNIST = G4NistManager::Instance();
47 }
48
49 G4HadronElasticDataSet::~G4HadronElasticDataSet() {}
50
51 void
52 G4HadronElasticDataSet::CrossSectionDescription(std::ostream& outFile) const
53 {
54     outFile << "G4HadronElasticDataSet contains elastic cross sections for\n"
55             << "all long-lived hadrons at all incident energies. It was\n"
56             << "developed as part of the Gheisha hadronic package\n"
57             << "by H. Fesefeldt, and consists of a set of parameterizations\n"
58             << "of elastic scattering data.\n";
59 }
60
61 G4bool
62 G4HadronElasticDataSet::IsElementApplicable(const G4DynamicParticle*,
63                                             G4int, const G4Material*)
64 {
65     return true;
66 }
67
68 G4double G4HadronElasticDataSet::GetElementCrossSection(
69     const G4DynamicParticle* aParticle, G4int Z, const G4Material*)
70 {
71     G4double ekin = aParticle->GetKineticEnergy();
72     const G4ParticleDefinition* pd = aParticle->GetDefinition();
73     if(Z != theZ || ekin != fKinEnergy || pd != fParticle) {
74         theZ = Z;
75         fKinEnergy = ekin;
76         fParticle = pd;
77         G4int A = G4rint(fNIST->GetAtomicMassAmu(Z));
78         fElasticXS = fGheishaXS->GetElasticCrossSection(aParticle, Z, A);
79     }
80     return fElasticXS;
81 }

```