

Refinement and extension of photon coherent interaction models with matter

G. Paternò

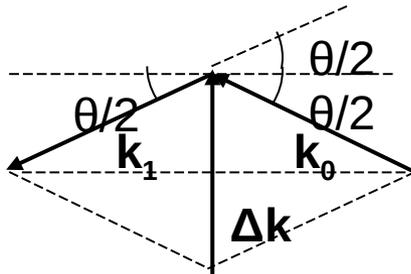
INFN Sezione di Ferrara, Dipartimento di Fisica
e Scienze della Terra, Università di Ferrara

Outline

- Development of interference effects in coherent X-ray scattering model.
- Development of diffraction process in polycrystalline materials.
- Development of refraction/reflection of X-rays.

Development of interference effects in coherent X-ray scattering model

In **Rayleigh (Coherent) Scattering**, photons are scattered by **bound atomic electrons** without excitation of the target atom, i. e., the energy of incident and scattered photons is the same.



$$\frac{d\sigma_{Ra}}{d\Omega} = r_e^2 \frac{1 + \cos^2 \theta}{2} |F(q, Z)|^2 \quad \text{Differential cross-section (DCS)}$$

Momentum transfer

$$q = \hbar |k_1 - k_0| = 2k \hbar \sin(\theta/2) = \frac{4\pi \hbar}{\lambda} \sin(\theta/2) = 2 \frac{E}{c} \sin(\theta/2)$$

$$x = \frac{q}{2h} = \frac{1}{\lambda} \sin(\theta/2) \quad [\text{nm}^{-1}]$$

Total cross-section

$$\sigma_{Ra} = \int \frac{d\sigma_{Ra}}{d\Omega} d\Omega = \pi r_e^2 \int (1 + \cos^2 \theta) |F(q, Z)|^2 \sin \theta d\theta$$

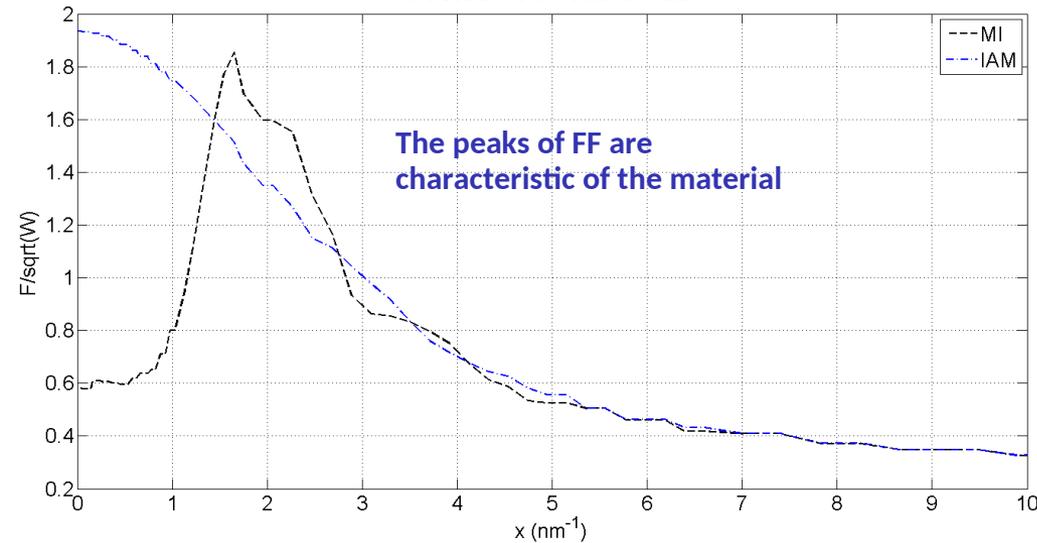
Common MC approach for molecules: IAM

$$F_{mol, IAM}^2(q) = W \sum_i \frac{w_i}{A_i} F^2(q, Z_i) \quad \rightarrow \text{No interference effects}$$

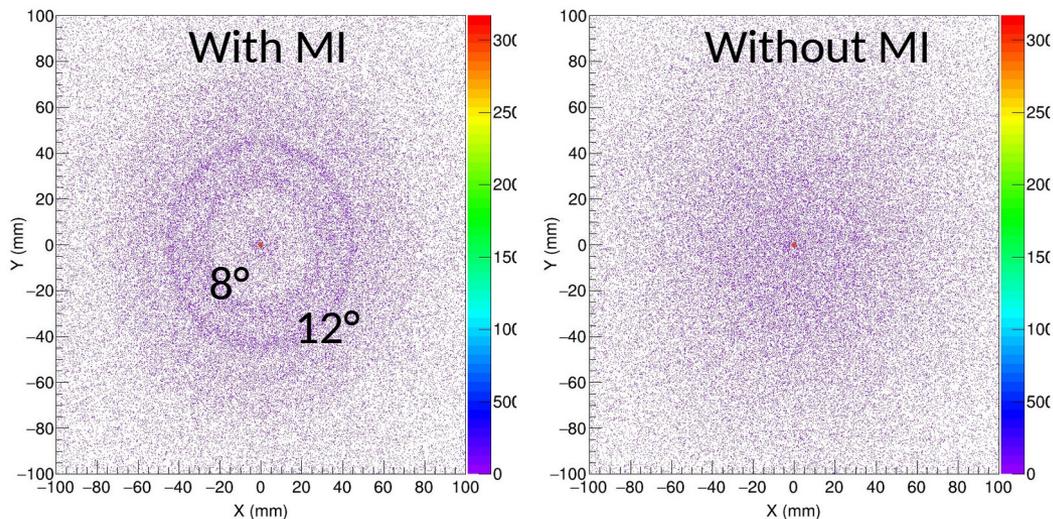
For each material a proper Form Factor including interference effects is required

Development of interference effects in coherent X-ray scattering model

Molecular Form Factor of water



- The **Penelope model was modified to read molecular form factors (FF) with interference effects** [G. Paterno et al, *Physica Medica* 51 (2018) 64–70].
- A **library of about 70 FFs** is made available.
- Every **biological tissue can be segmented in 4 basis components**.
- The user can introduce **custom form factors with interference** (using the support of **G4ExtendedMaterial**).
- The **cross-section is re-calculated integrating the DCS**.
- The **code allows** the user to:
 - remove scatter from images,
 - simulate WAXS/SAXS experiments,
 - classify unknown tissues.
- Coherent scattering **dominates** Compton scattering at low angles and it is **distinguishable** from primary beam.



Scattering of a 20 keV photon beam in a **human breast** sample

Development of interference effects in coherent X-ray scattering model

Read the Form Factors with Molecular Interference (MIFF)

A new method of G4PenelopeRayleighModel: **ReadMolInterferenceData(G4Material*)** is used to read the form factor (FF) with molecular interference (MI) of a **selected variety of materials**, according to a well-defined association “matname” → “MIFF”.

```
if (matname == "bone_MI") FFfilename = "FF_bone_King2011.dat";
if (matname == "FatLowX_MI") FFfilename = "FF_fat_Tartari2002_joint_lowXdata_ESRF2003.dat";
if (matname == "BoneMatrixLowX_MI") FFfilename = "FF_bonematrix_Tartari2002_joint_lowXdata.dat";
if (matname == "PMMALowX_MI") FFfilename = "FF_PMMA_Tartari2002_joint_lowXdata_ESRF2003.dat";
if (matname == "dryBoneLowX_MI") FFfilename = "FF_drybone_Tartari2002_joint_lowXdata_ESRF2003.dat";
if (matname == "CIRS30-70_MI") FFfilename = "FF_CIRS30-70_Poletti2002.dat";
if (matname == "CIRS50-50_MI") FFfilename = "FF_CIRS50-50_Poletti2002.dat";
if (matname == "CIRS70-30_MI") FFfilename = "FF_CIRS70-30_Poletti2002.dat";
if (matname == "RMI454_MI") FFfilename = "FF_RMI454_Poletti2002.dat";
if (matname == "PMMA_MI") FFfilename = "FF_PMMA_Tartari2002.dat";
if (matname == "Lexan_MI") FFfilename = "FF_lexan_Peplow1998.dat";
if (matname == "Kapton_MI") FFfilename = "FF_kapton_Peplow1998.dat";
if (matname == "Nylon_MI") FFfilename = "FF_nylon_Kosanetzky1987.dat";
if (matname == "Polyethylene_MI") FFfilename = "FF_polyethylene_Kosanetzky1987.dat";
if (matname == "Polystyrene_MI") FFfilename = "FF_polystyrene_Kosanetzky1987.dat";
if (matname == "Formaline_MI") FFfilename = "FF_formaline_Peplow1998.dat";
if (matname == "Acetone_MI") FFfilename = "FF_acetone_Cozzini2010.dat";
```

MIFF then are stored in a dedicated map (matname is the key):

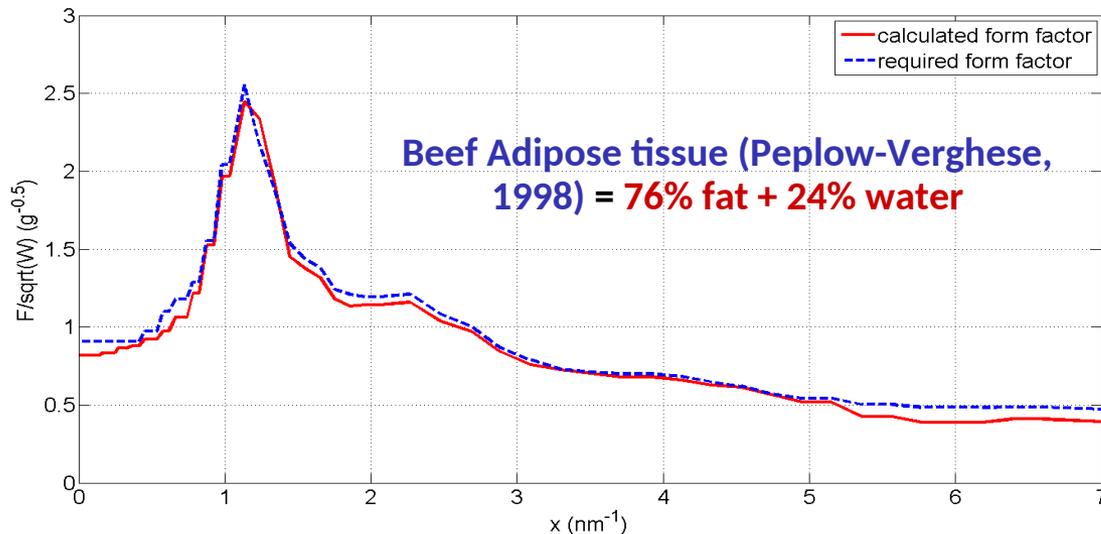
```
std::map<G4String,G4PhysicsFreeVector*> * MolInterferenceData;
```

This map is recalled in **BuildFormFactorTable(const G4Material*)** method.

For materials with MIFF, the cross-section is calculated by integrating the DCS (overriding of **ComputeCrossSectionPerVolume(const G4Material*)** of G4VEmModel). If a coherent scattering event occurs, the scattering angle is sampled according to RITA algorithm, which was not modified).

Development of interference effects in coherent X-ray scattering model

Decomposition of tissues in basis components



$$\frac{F^2}{W} = \sum_{i=1}^4 a_i \frac{F_i^2}{W_i}$$

i=fat, water, collagen, hydroxyapatite

[G. Paterno et al, PMB (2020), <https://doi.org/10.1088/1361-6560/aba7d2>]

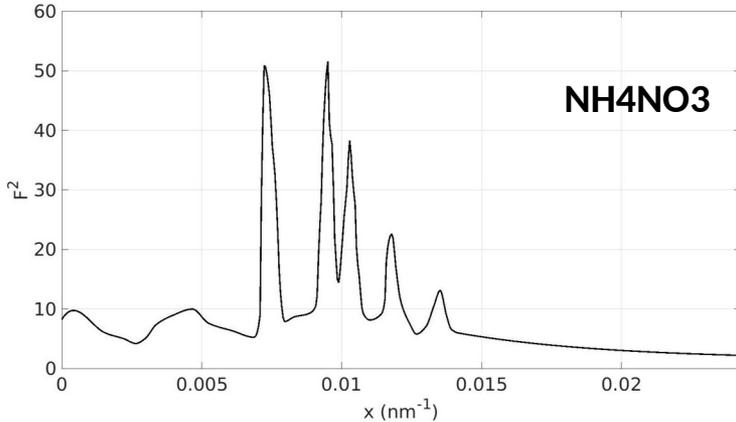
Avoid to use unclassified tissues

No extension needed, simply define a material and label it as “**MedMat_a1_a2_a3_a4**”, where a_i are the weight fractions of the 4 basis materials.

In the **BuildFormFactorTable(const G4Material*)** method, the basis FFs are mixed accordingly.

Development of interference effects in coherent X-ray scattering model

Form factor defined by the user



Useful for materials with partial crystalline behavior

In the DetectorConstrucion.cc, use the new **MIdata** extension to provide the path of the FF file of the material (through **setFilenameFF(G4String)** method).

```
d = CustomMatDensity*g/cm3;
matname = "CustomMat";
CustomMat = new G4ExtendedMaterial(matname, d, nel);
if (CustomMatHmassfract) CustomMat->AddElement(eLH, CustomMatHmassfract);
if (CustomMatCmassfract) CustomMat->AddElement(eLC, CustomMatCmassfract);
if (CustomMatNmassfract) CustomMat->AddElement(eLN, CustomMatNmassfract);
if (CustomMatOmassfract) CustomMat->AddElement(eLO, CustomMatOmassfract);
if (CustomMatNamassfract) CustomMat->AddElement(eLNa, CustomMatNamassfract);
if (CustomMatPmassfract) CustomMat->AddElement(eLP, CustomMatPmassfract);
if (CustomMatSmassfract) CustomMat->AddElement(eLS, CustomMatSmassfract);
if (CustomMatClmassfract) CustomMat->AddElement(eLCl, CustomMatClmassfract);
if (CustomMatKmassfract) CustomMat->AddElement(eLK, CustomMatKmassfract);
if (CustomMatCmassfract) CustomMat->AddElement(eLCa, CustomMatCmassfract);
//Extend the material
CustomMat->RegisterExtension(std::unique_ptr<MIdata>(new MIdata("MI")));
MIdata* dataMICustomMat = (MIdata*)CustomMat->RetrieveExtension("MI");
dataMICustomMat->SetFilenameFF(fCustomMatFF);
```

It will be used in the **BuildFormFactorTable(const G4Material*)** method.

Development of diffraction process in polycrystalline materials

The **cross-section for Bragg diffraction** from a **crystalline powder** sample is given by:

$$\sigma_B = \frac{r_e^2 \lambda^2}{2NV} \sum_{i=hkl} \left(\frac{1 + \cos^2 \theta}{2} \right) m_i d_i |F_i|^2 \quad (1)$$

Bragg's law states that scattering can only occur when $2d \sin(\theta/2) = \lambda$ and being $x = 1/\lambda \sin(\theta/2)$, it follows that **$d = 1/(2x)$** . Therefore:

$$\sigma_B = \frac{r_e^2}{4} \frac{(1 + \cos^2 \theta) \sin^2(\theta/2)}{2} |F_B(x)|^2 \quad (2)$$

$$|F_B(x)|^2 = \frac{1}{NV} \sum_{i=hkl} \frac{m_i |F_i|^2}{x^3} \delta(x - x_i) \quad (3)$$

The scattering angle $\theta = 2\theta_B$ is determined by sampling x from the **Bragg form factor F_B** , which contains all of the material-dependent terms (**$F_i = f_i * G_i * DW_i$**). The middle term in equation (2) depends only on angle and is used as the rejection function applied to the sampled x values.

Development of diffraction process in polycrystalline materials

The **atomic form factor f** can be calculated using the Cromer-Mann coefficients (a_i, b_i, c):

$$f(x) = c + \sum_{i=1}^4 a_i e^{-b_i x^2}$$

The **geometric factor G** depends on the crystal structure

$$G = \sum_j e^{-\vec{G}_m \cdot \vec{\rho}_j}$$

Where G_m is the **reciprocal lattice vector** and ρ_j is the position of the j -th atom in the **unit cell**.

The **Debye-Waller factor DW** , due to the atom vibration, depends also on crystal structure. There are semi-empirical formulas that allow one to estimate it.

Peak broadening (FWHM): $W_i = \frac{K \lambda}{L_G \cos \theta_B}$

Development of diffraction process in polycrystalline materials

- Development, from scratch, of a **specific physical process**, which implements the theory seen before exploiting the **G4CrystalExtension** (unit cell management, plane spacing and structure factor calculation).
- Development of **ExtraCrystalData** class, extension of G4Material, to add further properties to crystalline materials, such as, structure type, Debye temperature, crystallite size L_G , Scherrer GrainShape Factor K.
- **Automatic calculation of diffraction multiplicity, main diffraction lines** for a given crystal structure **has yet to be implemented** (by extending G4CrystalExtension).

Development of diffraction process in polycrystalline materials

In the DetectorConstruction:

```
584 ////////////////////////////////////////////////// Crystal Definition ////////////////////////////////////////////
585 GraphitePowder = new G4ExtendedMaterial("graphite_powder", 2.26*g/cm3, 1);
586 G4NistManager* NistMan = G4NistManager::Instance();
587 G4Element* elC = NistMan->FindOrBuildElement("C");
588 GraphitePowder->AddElement(elC, 1);
589
590 GraphitePowder->RegisterExtension(std::unique_ptr<G4CrystalExtension>(new G4CrystalExtension(GraphitePowder)));
591
592 G4CrystalExtension* crystalExtension = (G4CrystalExtension*)GraphitePowder->RetrieveExtension("crystal");
593
594 crystalExtension->SetUnitCell(new G4CrystalUnitCell(2.464 *CLHEP::angstrom,
595                                                    2.464 *CLHEP::angstrom,
596                                                    6.711 *CLHEP::angstrom,
597                                                    CLHEP::halfpi,
598                                                    CLHEP::halfpi,
599                                                    4/3*CLHEP::halfpi,
600                                                    194));
601
602 G4CrystalAtomBase* atomBase = new G4CrystalAtomBase();
603 atomBase->AddPos(G4ThreeVector(0. , 0. , 0. ));
604 atomBase->AddPos(G4ThreeVector(0. , 0. , 1./2. ));
605 atomBase->AddPos(G4ThreeVector(-1./3. , +2./3. , 0. ));
606 atomBase->AddPos(G4ThreeVector(+1./3. , -2./3. , 1./2. ));
607 crystalExtension->AddAtomBase(0, atomBase);
608
609 GraphitePowder->RegisterExtension(std::unique_ptr<MIdata>(new MIdata("MI")));
610
611 GraphitePowder->RegisterExtension(std::unique_ptr<ExtraCrystalData>(new ExtraCrystalData("extraDATAforCrystal")));
612 ExtraCrystalData* extraData = (ExtraCrystalData*)GraphitePowder->RetrieveExtension("extraDATAforCrystal");
613 extraData->SetStructureType("Hexagonal");
614 extraData->SetDebyeTemperature(1550.);
615 extraData->SetCrystalliteSize(80.); //nm
616 extraData->SetGrainShapeFactor(0.94);
617 ////////////////////////////////////////////
---
```

Development of diffraction process in polycrystalline materials

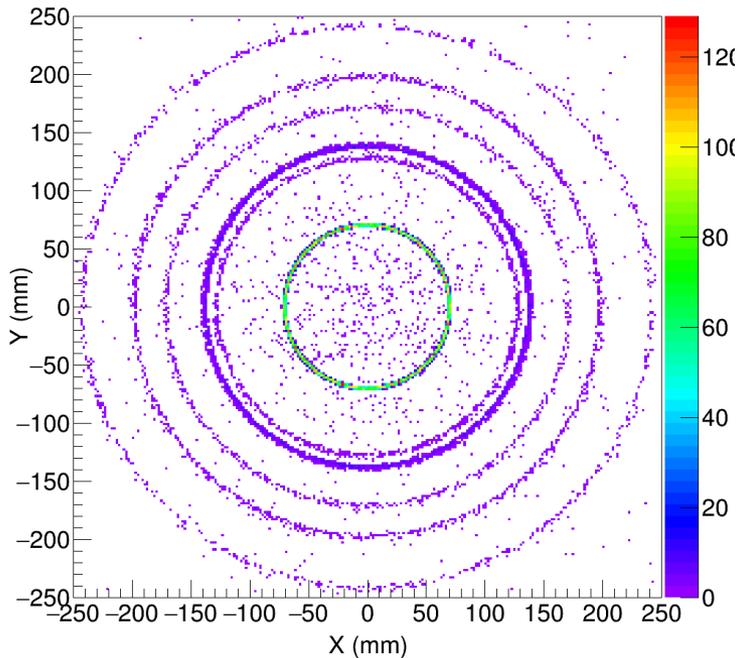
PowderDiffraction::ComputeSinglePlaneScatteringAmplitude (G4LogicalCrystalVolume* aLCV, G4int h, G4int k, G4int l)

```
164 //Retrieve ExtraCrystalData (new)
165 ExtraCrystalData* extraData = GetExtraCrystalData(aLCV->GetMaterial());
166 if (extraData != 0) {
167     fDebyeTemperature = extraData->GetDebyeTemperature();
168     fScherrerCrystalliteSize = extraData->GetCrystalliteSize() * CLHEP::nanometer;
169     fScherrerGrainShapeFactor = extraData->GetGrainShapeFactor();
170     fStructureType = extraData->GetStructureType();
171 }
172
173 G4double TonDT = aLCV->GetMaterial()->GetTemperature() / fDebyeTemperature;
174
175 //G4double A = aLCV->GetMaterial()->GetA() / (CLHEP::g/CLHEP::mole);
176 //G4Material->GetA() gives errors if the material is not a single element but a compound.
177 //This case is managed with a control.
178 G4double A;
179 if (aLCV->GetMaterial()->GetNumberOfElements() == 1)
180     A = aLCV->GetMaterial()->GetA() / (CLHEP::g/CLHEP::mole);
181 else
182     A = aLCV->GetMaterial()->GetMassOfMolecule() / (CLHEP::amu);
183 //G4cout << "A: " << A << G4endl;
184
185 G4double u_DWF = 1.149*10000./ (A*fDebyeTemperature)*(TonDT+0.25*std::exp(-4.*TonDT))*CLHEP::angstrom*CLHEP::angstrom;
186 G4double DWF = std::exp(-u_DWF / (4. * intsp * intsp));
187 //G4cout << "DWF: " << DWF << G4endl;
188
189 //Structure Factor (SF)
190 G4double SF = std::abs(aXtal->ComputeStructureFactorGeometrical(h,k,l));
191
192 //Atomic Form Factor (AFF)
193 G4double kScatteringVector = X * 3.1415926536 / 0.125 * CLHEP::angstrom * CLHEP::angstrom;
194
195 //Implemented a weighted mean of FF of the elements that form the compound.
196 G4Material* material = aLCV->GetMaterial();
197 G4int nElements = material->GetNumberOfElements();
198 const G4ElementVector* elementVector = material->GetElementVector();
199 const G4double* fractionVector = material->GetFractionVector();
200 G4double AFF = 0.;
201 for (G4int i=0; i<nElements; i++) {
202     G4int iZ = (G4int)(*elementVector)[i]->GetZ();
203     AFF += G4AtomicFormFactor::GetManager()->Get(kScatteringVector, iZ)*fractionVector[i];
204 }
205
206 //multiplicity (mult)
207 G4double mult = ComputePlaneMultiplicity(h,k,l);
208
209 G4double unitCellVolume = aXtal->GetUnitCell()->GetVolume();
210 G4int natoms = aXtal->GetAtomBase(0)->GetPos().size();
211 G4double Fi = AFF * SF * DWF;
212 G4double X3 = X * X * X;
213 return 1. / (unitCellVolume * natoms) * Fi * Fi * mult / X3;
214 }
```

Development of diffraction process in polycrystalline materials

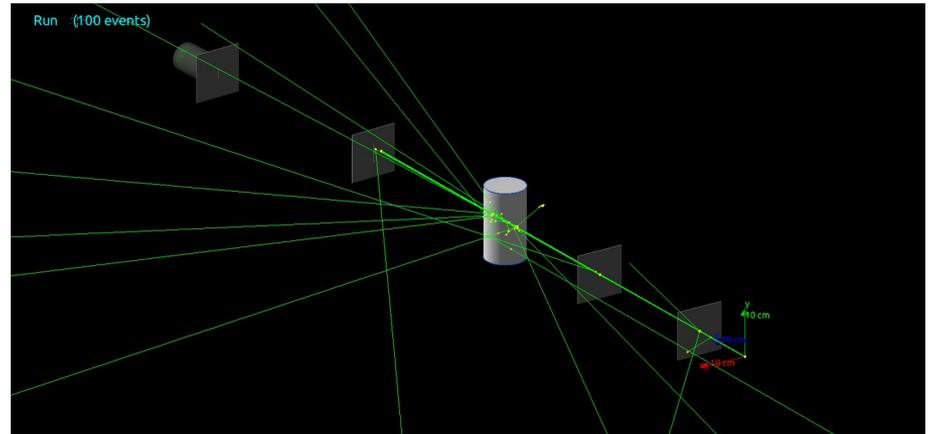
Simple test: diffraction of a **8 keV** X-ray beam from a thin “slab” of a **graphite powder**.

Spatial Distribution

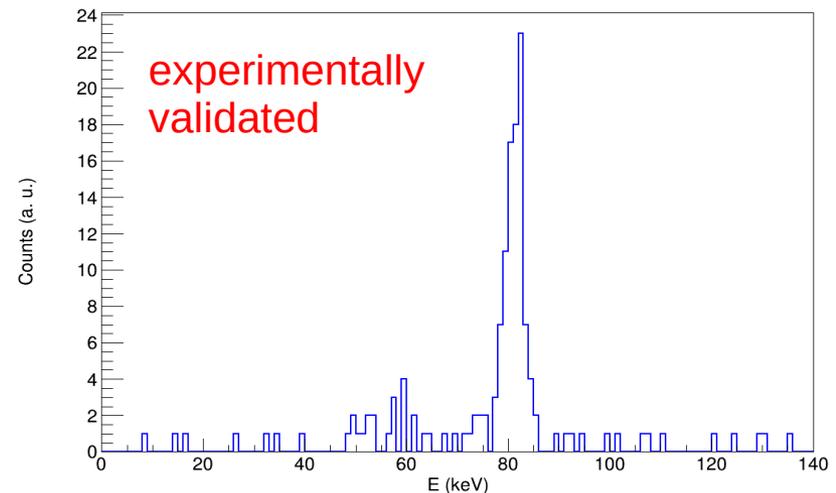


Line position, width and relative intensity **in agreement with analytical calculation.**

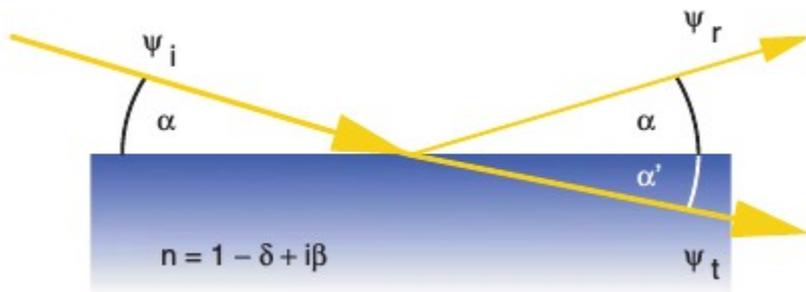
Advanced test: diffraction of a **polychromatic** X-ray beam from a cylinder of **graphite powder**. The set-up is conceived to select the most intense line.



Spectrum of energy deposited inside the Ge detector



Development of refraction/reflection of X-rays



$$n = 1 - \frac{r_0}{2\pi} \lambda^2 \sum_i N_i f_i(0),$$

complex
refraction index

$$\cos \alpha = n \cos \alpha'$$

Snell's law

$$\alpha_c = \sqrt{2\delta}$$

Total external reflection for $\alpha \leq \alpha_c$

$$r \equiv \frac{a_R}{a_I} = \frac{\alpha - \alpha'}{\alpha + \alpha'}; \quad t \equiv \frac{a_T}{a_I} = \frac{2\alpha}{\alpha + \alpha'}$$

$$r = \frac{\sin \alpha - (n^2 - \cos^2 \alpha)^{1/2}}{\sin \alpha + (n^2 - \cos^2 \alpha)^{1/2}}$$

Fresnel equations for
perfect surfaces (In
this form, they are
exact for s-pol and
approx for p-pol)

$$R_r = R_f \cdot \exp[-(2k\alpha\sigma)^2]$$

$R=|r|^2$ reflectivity. Fresnel equations for
rough surfaces (σ = roughness). In addition,
the normal to the surface varies randomly
according to a Gaussian distribution

Development of refraction/reflection of X-rays

- **STATUS:**

- Defined the main algorithms and new classes to be implemented.
- Identified a database for index of refraction (CXRO).
- Concrete implementation has yet to be done.

- A bibliographic research highlighted that these effects were already implemented in Geant4/Gate by different groups [1,2,3,4], with different levels of accuracy and subsequent application. These works can be used as **benchmarks**.

[1] Langer M. et al, *Towards Monte Carlo simulation of X-ray phase contrast using GATE*, Optics Express 28, 2020.

[2] Sanctorum J. et al, *X-ray phase-contrast simulations of fibrous phantoms using GATE*, 2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings.

[3] Buis E.J, G. Vacanti, *X-ray tracing using Geant4*, NIM-A 599 (2009) (code freely download-able).

[4] Wang Z. et al, *Implement X-ray refraction effect in Geant4 for phase contrast imaging*, 2009 IEEE Nuclear Science Symposium Conference Record.

Conclusions

- Development of interference effects in coherent X-ray scattering model → Ready (almost) for the official release (thanks to Luciano Pandola)
- Development of diffraction process in polycrystalline materials → Under development
- Development of refraction/reflection of X-rays → We are at the beginning

Back-up slides

Development of interference effects in coherent X-ray scattering model

First, the occurrence of a coh. scatt. event is determined from σ_{Ra} , then the angular deflection is sampled

$$P_{Ra}(\cos\theta) = \frac{1 + \cos^2\theta}{2} F^2(q) \quad 0 \leq q \leq q_{\max} = 2E/c = 2m_e c k$$

$$P_{Ra}(\cos\theta) = g(\cos\theta) \pi(q^2) \quad g(\cos\theta) = \frac{1 + \cos^2\theta}{2} \quad \pi(q^2) = F^2(q)$$

rejection method

1. Using the **RITA algorithm**, sample a random value of q^2 from the distribution $\pi(q^2)$, restricted to the interval $[0, q_{\max}^2]$.
2. Set $\cos\theta = 1 - 1/2 * q^2/k^2$ ($k = E/m_e c^2$). *(it comes from the definition of $q = 2E/c[\sin(\theta/2)] = (E/c[2(1-\cos\theta)])^{1/2}$)*
3. Generate a new random number ξ (uniformly distributed in the interval $[0,1]$).
4. If $\xi > g(\cos\theta)$, go to step 3. *(note that g is a valid rejection function since $0 < g \leq 1$)*
5. Deliver $\cos\theta$.

Sampling efficiency higher than 66%