

Verification of absolute intensity

BornAgain report
ba-intern/theory/IntensityScale.tex
Scientific Computing Group of JCNS-MLZ
Joachim Wuttke, November 4, 2020
Last change November 6, 2020.

It has been asked whether BornAgain yields intensities in correct absolute units so that scattering and reflection can be directly compared. This short report justifies the affirmative answer: Yes, simulated intensities have been in correct absolute units ever since the first release of BornAgain.

1 Scattering intensity

BornAgain has been validated extensively and on absolute scale against IsGisaxs. If scattering intensities were not correct in absolute units in BornAgain, then they would not be correct in IsGisaxs either. IsGisaxs takes all parameters that are necessary to compute scattered intensities in correct absolute units; nothing in the IsGisaxs documentation indicates that intensities are in arbitrary units only. It would be most surprising if such a grave misunderstanding had passed unnoticed through more than fifteen years, while IsGisaxs has been applied to many hundred experiments. So this is a first strong reason to expect BornAgain to yield correct absolute scattering units.

Nonetheless, we proceed with a direct verification of BornAgain against a paper-and-pencil computation. A 2D scattering simulation yields an array where each element represents the integral scattering intensity $I(q)$ in one pixel q . So

$$I = J \Delta\Omega, \quad (1)$$

where J is the scattered flux, and $\Delta\Omega$ is the solid angle taken by pixel q .

From transport theory, the scattering from thin samples is linear,

$$J = \rho L \frac{d\sigma}{d\Omega}, \quad (2)$$

where ρ is the number density of scatterers, and L is the beam path through the sample. We consider a layer of height h . With glancing angle α_i , we have

$$h = L \sin \alpha_i. \quad (3)$$

Our model is parameterized with the surface density

$$s = \rho h. \quad (4)$$

So

$$J = \frac{s}{\sin \alpha_i} \frac{d\sigma}{d\Omega}. \quad (5)$$

The scattering cross section of one single particle in vacuum is

$$\frac{d\sigma}{d\Omega} = \left| \int d^3r \psi_i(\mathbf{r}) u(\mathbf{r}) \psi_f(\mathbf{r}) \right|^2 \quad (6)$$

$$= (\rho b)^2 |F(\mathbf{q})|^2, \quad (7)$$

where ρb is the scattering length density. The simplest way to verify the absolute scale is to consider the channel $q \simeq 0$, for which the shape transform $F(\mathbf{q})$ is just the particle volume V .

The refractive index is [e. g. Squires eq. 6.1]

$$n = 1 - \frac{\lambda^2 \rho b}{2\pi}. \quad (8)$$

We resolve for

$$\rho b = \frac{2\pi}{\lambda}(1 - n). \quad (9)$$

In the absence of absorption, we have $n = 1 - \delta$ and thereby

$$\rho b = \frac{2\pi}{\lambda}\delta. \quad (10)$$

Wrapping up,

$$I(0) = \frac{s}{\sin \alpha_i} \frac{4\pi^2}{\lambda^4} \delta^2 V^2 \Delta\Omega. \quad (11)$$

For spherical particles,

$$I(0) = \frac{s}{\sin \alpha_i} \frac{4\pi^2}{\lambda^4} \delta^2 \left(\frac{4\pi}{3}\right)^2 R^6 \Delta\Omega. \quad (12)$$

To verify, we run the following script against BornAgain:

```

import bornagain as ba
from bornagain import deg, angstrom, nm

def get_sample():
    m_vacuum = ba.HomogeneousMaterial("Vacuum", 0.0, 0.0)
    m_particle = ba.HomogeneousMaterial("Particle", 6e-4, 0)

    ff = ba.FormFactorFullSphere(5*nm)
    cylinder = ba.Particle(m_particle, ff)
    particle_layout = ba.ParticleLayout()
    particle_layout.setTotalParticleSurfaceDensity(0.01)
    particle_layout.addParticle(cylinder, 1.0)

    vacuum_layer = ba.Layer(m_vacuum)
    vacuum_layer.addLayout(particle_layout)

    multi_layer = ba.MultiLayer()
    multi_layer.addLayer(vacuum_layer)
    return multi_layer

def get_simulation():
    simulation = ba.GISASSimulation()
    simulation.setDetectorParameters(200, -2.0*deg, 2.0*deg,
                                     200, -2.0*deg, 2.0*deg)
    simulation.setBeamParameters(1*angstrom, 0.2*deg, 0.0*deg)
    return simulation

```

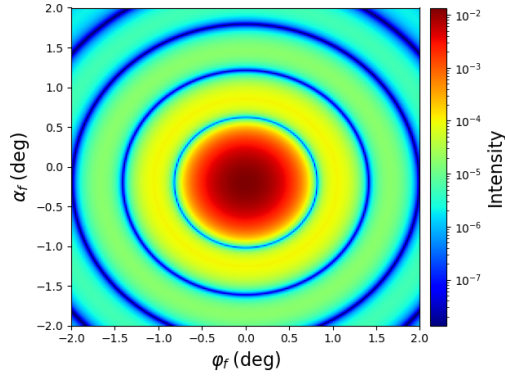


Figure 1: Result of BornAgain simulation: grazing-incidence scattering by a layer of uncorrelated spherical particles in vacuum.

```

def run_simulation():
    simulation = get_simulation()
    simulation.setSample(get_sample())
    simulation.runSimulation()
    return simulation.result()

if __name__ == '__main__':
    result = run_simulation()
    print("max: ", result.max()) # fct added after release 1.18
    ba.plot_simulation_result(result, cmap='jet', aspect='auto')

```

This script models a monolayer of uncorrelated spheres in vacuum. As there are no reflections from interfaces, BornAgain automatically discards the DWBA and computes scattering in Born Approximation. Parameters are chosen as in many of our examples,

$$\begin{aligned}
 \alpha_i &= 0.2^\circ, \\
 s &= 0.01 \text{ nm}^{-2}, \\
 \delta &= 6 \cdot 10^{-4}, \\
 R &= 5 \text{ nm}, \\
 \lambda &= 1 \text{ \AA} = 0.1 \text{ nm}.
 \end{aligned}$$

In α_f and φ_f , we choose 200 bins, extending from -2° to 2° . So one bin covers $\Delta\alpha = \Delta\varphi = 4^\circ/200$, which yields

$$\Delta\Omega = \Delta\alpha \Delta\varphi = 1.22 \cdot 10^{-7}. \quad (13)$$

With eq. (12) we expect

$$I(0) \simeq 0.0136. \quad (14)$$

And indeed, our simulation (Fig. 1) yields

$$I(0) \simeq 0.01358. \quad (15)$$

So yes, BornAgain computes scattering in correct absolute units.

2 Reflected intensity

A specular reflection has the flux

$$J = R\delta(\mathbf{q} + 2k_{iz}\hat{\mathbf{z}}), \quad (16)$$

where R is the reflectivity of the sample. Under total reflection conditions, it is just $R = 1$. The integral intensity in the pixel that contains the specular peak is

$$I(q \simeq -2k_{iz}\hat{\mathbf{z}}) = R, \quad (17)$$

regardless of the pixel size $\Delta\Omega$.

This is verified with the following script:

```
import bornagain as ba
from bornagain import deg, angstrom, mm

def get_sample():
    m_vacuum = ba.HomogeneousMaterial("Vacuum", 0.0, 0.0)
    m_substrate = ba.HomogeneousMaterial("Substrate", 6e-6, 0)
    vacuum_layer = ba.Layer(m_vacuum)
    substrate_layer = ba.Layer(m_substrate)
    multi_layer = ba.MultiLayer()
    multi_layer.addLayer(vacuum_layer)
    multi_layer.addLayer(substrate_layer)
    return multi_layer

def get_simulation():
    simulation = ba.GISASSimulation()
    simulation.setDetectorParameters(101, -2.0*deg, 2.0*deg,
                                     101, -2.0*deg, 2.0*deg)
    simulation.setBeamParameters(1.0*angstrom, 0.01*deg, 0.0*deg)
    simulation.getOptions().setIncludeSpecular(True)
    return simulation

def run_simulation():
    sample = get_sample()
    simulation = get_simulation()
    simulation.setSample(sample)
    simulation.runSimulation()
    return simulation.result()

if __name__ == '__main__':
    result = run_simulation()
    print("max:  $\square$ ", result.max())
    ba.plot_simulation_result(result, cmap='jet', aspect='auto')
```

The simulated detector image (Fig. 2) contains exactly one pixel with nonzero intensity. Python reports the intensity in this pixel as 0.9999999999999999.

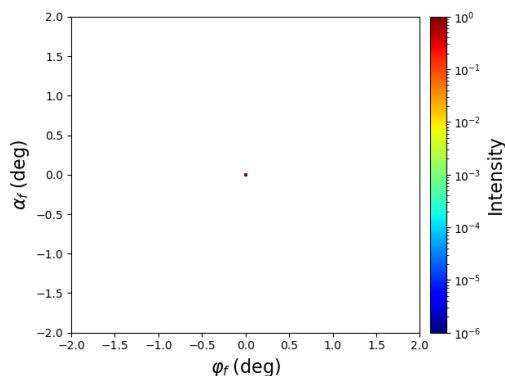


Figure 2: Result of BornAgain simulation: specular reflection from a flat interface under total reflection conditions. As expected, the intensity in the single illuminated pixel is 1 within floating-point accuracy.

3 Conclusion

By comparing simple BornAgain simulations with paper-and-pencil computations, we have demonstrated that scattering and reflection intensities are obtained in correct absolute units.

From its earliest versions, BornAgain has been quantitatively validated against Is-Gisaxs. Furthermore, the test suite of BornAgain ensures that simulation outcomes do not change in unexplained ways as the code evolves. This allows us to affirm that absolute units have been correct since the earliest versions of BornAgain.

So far, reflection and refraction of the direct beam have only been considered in unperturbed Fresnel theory. If there is scattering, then this theory must be modified to ensure particle, or energy, conservation. As some intensity goes into scattering, the total transmitted and reflected intensity must be smaller than 1. This can be enforced ad hoc by adding some damping to the imaginary part of the refractive index. It remains to be seen whether future versions of BornAgain can achieve the same automatically, through some self-consistent formalism based on the optical theory.