Feasibility study of new metrology for x-ray capillary development

Brenden Roberts

Advisors: Dr. Rong Huang & Dr. Peter Revesz

2012 CLASSE REU Final Presentations
August 10, 2012
Outline

1 Background
   - X-ray capillary theory
   - Capillaries at Cornell

2 Simulation methods & results
   - Ray-tracing code
   - Numerical results

3 Experiment
   - Capillary testing
   - Testing results
The role of x-ray capillaries at CHESS

- Synchrotron radiation emerges in a relatively collimated narrow cone tangent to the beam path.
- Upstream beamline optics can be used to select a bandwidth or perform initial focusing.
- Capillaries are designed to perform the final microfocusing before the x-ray beam reaches a sample.
Capillary design

What properties are we looking for?
- High intensity gain
- Low focal spot size
- High optical efficiency
- Effective over a wide energy range

Ideal gain from capillary focusing

Graph showing X-ray flux gain versus vertical position (microns).
Capillary design

What properties are we looking for?
- High intensity gain
- Low focal spot size
- High optical efficiency
- Effective over a wide energy range

What tradeoffs can we make? The answer is dependent on the type of experiment the x-rays are being used to perform.
Total external reflection of x-rays

Because $n_2$ for glass is less than 1 by $\delta$ ($\sim$1 part in $10^5$), any reflection below the critical angle (3 mrad at 10 keV) is referred to as *total external reflection* and preserves almost all ray intensity.
Capillary techniques

Cornell’s pulling process:

1. Program used describing a parabolic shape
2. Glass tube is loaded into the puller
3. Mobile furnace heats part of the glass
4. Tension is applied to the tube’s ends
5. Capillary inner radius is locally reduced via conservation of mass
6. Furnace moves to new location, based on the stretched length
Capillary techniques

Is there a way to know the capillary ID as it’s in the process of being stretched?

Use a UV light and a filtered photodiode to measure light transmission. Why UV?

\[
\text{Reflected intensity} = f(\text{Incident angle (rad)})
\]

- UV LED
- Tube
- Furnace
- Photodiode
Program structure flowchart

create model
Program structure flowchart

Input (experimental):
- Maximum divergence
- X-ray focal length
- X-ray source distance
- etc.

*Includes both random scattering (Gauss-Markov process) and preset centerline deflection error*
Simulation methods & results

Ray-tracing code

Program structure flowchart

create model

generate random initial ray conditions

Input (experimental):
- Maximum divergence
- X-ray focal length
- X-ray source distance
- etc.

Output:
- Includes both random scattering (Gauss-Markov process) and preset centerline deflection error

Monte Carlo: uniform distribution within extended source; specific angular distribution are there rays in the capillary?
- yes
- no

Assumption: photons do not refract through the capillary wall and reenter the inner space.

A separate, recursive 3D algorithm tracked all refracted rays and showed that they contributed <1% of total intensity, even in ideal cases like the one pictured.

Fresnel intensity

\[ I = R_\parallel I_0 \left( \frac{n_2}{n_2 \cos \theta_i - i} \right) \left( \frac{n_2}{n_2 \cos \theta_i + i} \right)^2 \]

\[ R_\perp = \cos \theta_i - i \sqrt{\sin^2 \theta_i - n_2^2} \]

\[ R = R_\parallel + R_\perp^2 \]

\[ \vec{u}_r = \vec{u}_i - 2 \left( \vec{u}_i \cdot \vec{n} \right) \vec{n} \]

\[ \theta = f(r, \phi, z) \]

\[ \vec{n} = \frac{\partial f}{\partial \phi} \times \frac{\partial f}{\partial z} \]

remove dead rays

STOP

collect rays
Program structure flowchart

create model → generate random initial ray conditions

Monte Carlo: uniform distribution within extended source; specific angular distribution

Monte Carlo

\[ \vec{u}_r = \vec{u}_i - 2(\vec{u}_i \cdot \vec{n}) \vec{n} \]

\[ R_{\parallel} = n_2 \cos \theta_i - i \sqrt{\sin^2 \theta_i - n_2^2 \cos^2 \theta_i} \]

\[ R_{\perp} = \cos \theta_i - i \sqrt{\sin^2 \theta_i - n_2^2 \cos^2 \theta_i} \]

\[ R = R_{\parallel}^2 + R_{\perp}^2 \]

\[ \vec{f}(r, \phi, z) \]

Fresnel intensity

\[ I = R I_0 R_{\parallel} \]

Power/Radiant Intensity
Program structure flowchart

create model → generate random initial ray conditions → are there rays in the capillary?

- yes → continue simulation
- no → remove dead rays

Input (experimental):
- Maximum divergence
- X-ray focal length
- X-ray source distance

Output:
- Includes both random scattering (Gauss-Markov process) and preset centerline deflection error

Monte Carlo: uniform distribution within extended source; specific angular distribution

Specular reflection

\[ \vec{n} = \frac{\partial f}{\partial \phi} \times \frac{\partial f}{\partial z} \parallel \frac{\partial f}{\partial \phi} \times \frac{\partial f}{\partial z} \parallel \]

\[ \vec{u}_r = \vec{u}_i - 2(\vec{u}_i \cdot \vec{n}) \vec{n} \]

\[ \theta_{\text{f}}(r, \phi, z) \]

Fresnel intensity

\[ I = R \left| \frac{\n_2 \cos \theta_i - \n_1 \sqrt{\sin^2 \theta_i - \n_2^2 \cos^2 \theta_i}}{\n_2 \cos \theta_i + \n_1 \sqrt{\sin^2 \theta_i - \n_2^2 \cos^2 \theta_i}} \right| \]

\[ R_\parallel = \cos \theta_i - \sqrt{\sin^2 \theta_i - \n_2^2 \cos^2 \theta_i} \]

\[ R_\perp = \cos \theta_i + \sqrt{\sin^2 \theta_i - \n_2^2 \cos^2 \theta_i} \]

\[ R = R_\parallel + R_\perp \]
Assumption: photons do not refract through the capillary wall and reenter the inner space.

A separate, recursive 3D algorithm tracked all refracted rays and showed that they contributed <1% of total intensity, even in ideal cases like the one pictured.
create model

create model

generate random initial ray conditions

generate random initial ray conditions

are there rays in the capillary?

are there rays in the capillary?

yes

specular reflection

specular reflection

no

STOP

collect rays

collect rays
**Program structure flowchart**

1. **create model**
2. **generate random initial ray conditions**

**Ray-tracing code**

- **Input (experimental):**
  - Maximum divergence
  - X-ray focal length
  - X-ray source distance
  - etc.

- **Output:**
  - Includes both random scattering (Gauss-Markov process) and preset centerline deflection error

**Monte Carlo:** uniform distribution within extended source; specific angular distribution

- Are there rays in the capillary? **no**
  - Stop
- Yes
  - Specular reflection

**Fresnel intensity**

\[ I = R I_0 \]

\[ R_\parallel = \frac{n_2 \cos \theta - i \sqrt{\sin^2 \theta - n_2^2 \cos^2 \theta}}{n_2 \cos \theta + i \sqrt{\sin^2 \theta - n_2^2 \cos^2 \theta}} \]

\[ R_\perp = \cos \theta - i \sqrt{\sin^2 \theta - n_2^2 \cos^2 \theta} \]

\[ R = R_\parallel + R_\perp^2 \]

**remove dead rays**

\[ \vec{u}_r = \vec{u}_i - 2(\vec{u}_i \cdot \vec{n})\vec{n} \]

\[ \vec{n} = \frac{\frac{\partial f}{\partial \phi} \times \frac{\partial f}{\partial z}}{\left\| \frac{\partial f}{\partial \phi} \times \frac{\partial f}{\partial z} \right\|} \]
Program structure flowchart

create model

\[ \text{generate random initial ray conditions} \]

\[ \text{Fresnel intensity} \]

\[ \text{specular reflection} \]

\[ \text{are there rays in the capillary?} \]

\[ \text{yes} \]

\[ \text{no} \]

\[ \text{collect rays} \]

\[ \text{remove dead rays} \]

\[ \vec{n} = \frac{\partial f}{\partial \phi} \times \frac{\partial f}{\partial z} \parallel \frac{\partial f}{\partial \phi} \times \frac{\partial f}{\partial z} \parallel \]

\[ \vec{u}_r = \vec{u}_i - 2 (\vec{u}_i \cdot \vec{n}) \vec{n} \parallel \]

\[ I = R I_0 \]

\[ R \parallel = n_2 \cos \theta_i - i \sqrt{\sin^2 \theta_i - n_2^2} \]

\[ R \perp = \cos \theta_i - i \sqrt{\sin^2 \theta_i - n_2^2} \]

\[ R = R \parallel + R \perp \]

\[ \vec{f}(r, \phi, z) \]

\[ \theta \]

\[ \nabla \]

\[ \Delta \]

\[ \mathcal{O} \]

\[ \mathcal{E} \]

\[ \mathcal{I} \]

\[ \mathcal{P} \]

\[ \mathcal{R} \]

\[ \mathcal{S} \]

\[ \mathcal{T} \]

\[ \mathcal{U} \]

\[ \mathcal{V} \]

\[ \mathcal{W} \]

\[ \mathcal{X} \]

\[ \mathcal{Y} \]

\[ \mathcal{Z} \]
Program structure flowchart

create model → generate random initial ray conditions →

are there rays in the capillary?

yes → Fresnel intensity

no → remove dead rays

\[
I = R I_0
\]

\[
R_{\parallel} = \frac{n^2 \cos \theta_i - i \sqrt{\sin^2 \theta_i - n^2}}{n^2 \cos \theta_i + i \sqrt{\sin^2 \theta_i - n^2}}
\]

\[
R_{\perp} = \frac{\cos \theta_i - i \sqrt{\sin^2 \theta_i - n^2}}{\cos \theta_i + i \sqrt{\sin^2 \theta_i - n^2}}
\]

\[
R = \frac{R_{\parallel} + R_{\perp}}{2}
\]
Program structure flowchart

1. Create model
2. Generate random initial ray conditions
3. Are there rays in the capillary?
   - No
     - Collect rays
   - Yes
     - Specular reflection
     - Fresnel intensity
     - Remove dead rays
Program structure flowchart

create model → generate random initial ray conditions → are there rays in the capillary? → yes

→ remove dead rays → Fresnel intensity → specular reflection

→ no

stop collect rays
Simulation methods & results
Ray-tracing code

Program structure flowchart

create model → generate random initial ray conditions

are there rays in the capillary?

no → STOP collect rays

yes → remove dead rays

Fresnel intensity → specular reflection

B. C. Roberts 
X-ray capillary development
REU 2012 8 / 16

Input (experimental):
- Maximum divergence
- X-ray focal length
- X-ray source distance
- etc.

Output:
- Includes both random scattering (Gauss-Markov process) and preset centerline deflection error

Monte Carlo:
- uniform distribution within extended source; specific angular distribution

are there rays in the capillary?

yes

no

remove dead rays

specular reflection

Fresnel intensity

STOP collect rays
Why is this important?

Similar programs have been written by Dr. Huang and previous REU students, but

- This is the first truly 3-dimensional capillary raytracer at Cornell, and accounts for non-ideal profiles (bending & scattering)
- This program employs parallel raytracing strategy rather than serial
- My method is more difficult to implement but much quicker

Both parallel and serial tracing methods run in $\Theta(n)$ time. However the coefficient on the parallel tracer is orders of magnitude smaller.
Primary simulation results

Straight capillary transmission during pull

Shape:
- Max. radius 0.4 mm
- Min. radius 0.12 mm
- Init. length 20 cm
Primary simulation results

Capillary transmission during pull with deflection

Shape:
- Max. radius: 0.4 mm
- Min. radius: 0.12 mm
- Init. length: 20 cm
Spatial intensity distributions

All rays

Only indirect

1 mm

Total intensity:
54.0% indirect

10 mm
For the experimental verification, instead of using pre-drawn capillaries, I used straight (unmodified) glass tubes.
Normalized transmitted light

Capillary transmission during pull with deflection

Transmission (arbitrary units) vs. Length (mm)

Simulation vs. Experimental

Justification:
- Ray reflection counts and error are likely to be correlated
- Longer tubes will result in more bounces
- The UV filter is not 100% opaque to visible light
Reflection count vs. raytracer accuracy

Relationship between bounces and simulation error

![Graph showing the relationship between average number of ray bounces and simulation percent error. The x-axis represents the average number of ray bounces ranging from 1 to 10, while the y-axis represents the simulation percent error ranging from 0% to 40%. The graph includes data points at average number of bounces: 15.2 mm, 50.5 mm, 89.5 mm, 149.0 mm, and 201.5 mm corresponding to simulation errors at 0%, 10%, 20%, 30%, and 40% respectively.]
Conclusions

- Cornell now has a working, versatile 3D capillary raytracer.
- The fundamental idea behind the methodology works. A linear trend exists in UV intensity as a capillary is being pulled, so we can monitor the ID in realtime.
- This trend is unaffected by up to moderate profile imperfections.
- Preliminary experiment indicates that the simulation matches real data well but will tend to deviate for long capillary lengths.

Acknowledgments

This REU program is sponsored by NSF grant PHY-1156553. Dr. Rong Huang and Dr. Peter Revesz served as my advisors. Tom Szebenyi provided invaluable technical explanations. Finally Monica Wesley and Ivan Bazarov did a great deal of organizing for this program.