Phonon Specular Reflection on Curved Surfaces

Michael Kelsey, Wade Lamberson, Nolan Tenpas
Texas A&M University

G4CMP Consortium Meeting, 15 May 2025

Outline

Phonon Propagation: Wavevector vs. Group Velocity

Problem: Specular Reflection Fails on Sidewalls

G4CMP-317: Model Surface Propagation

Outline of Basic Search Algorithm

Complications from Flats and Sidewall Edges

Technical Implementation Notes

Status and Plans

Phonon Propagation

Phonons propagate through crystal with different *phase* vs. *group* velocities.

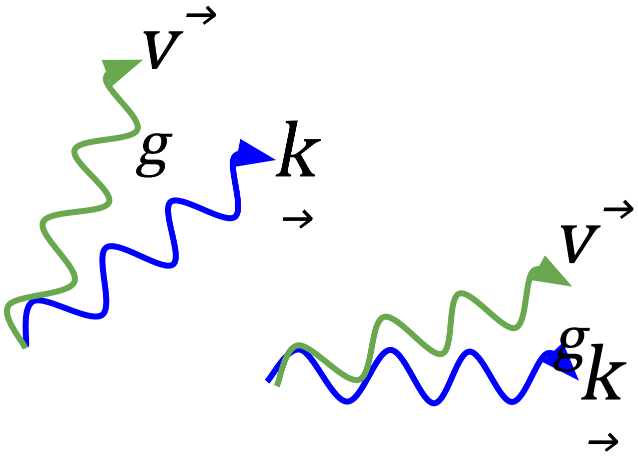
G4CMP computes dispersion relationship, and uses that for transport

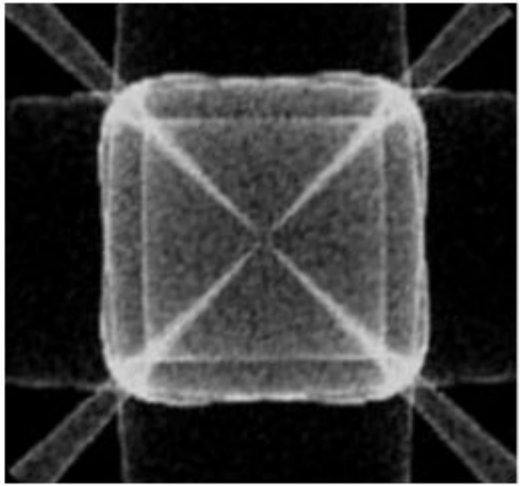
- Phonons travel along $\vec{v}_g$ (group velocity)
- Uses lookup tables for group vs. phase velocities at runtime

Can plot resulting "caustic" distributions (right)

Phonons reflect via $\vec{k}$, propagate via $\vec{v}_g$

- Reflected propagation depends on local normal



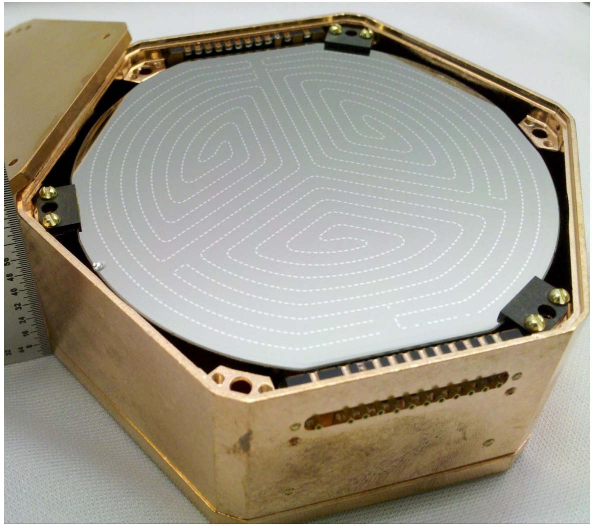


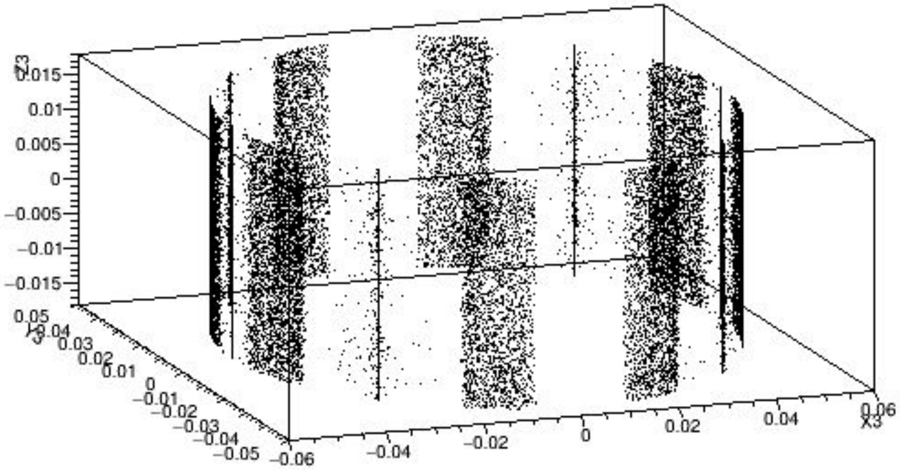
Phonons Can Fail Reflection

CDMS detectors are cylindrical. We see about 30% of phonons fail to reflect specularly on sidewalls and get killed, always in specific regions

To avoid losing 30% of event energy, CDMS simulation jobs only use "diffuse" Lambertian reflection

Phonons incident on surface at any angle are reflected with a cos(Θ) distribution around the surface normal





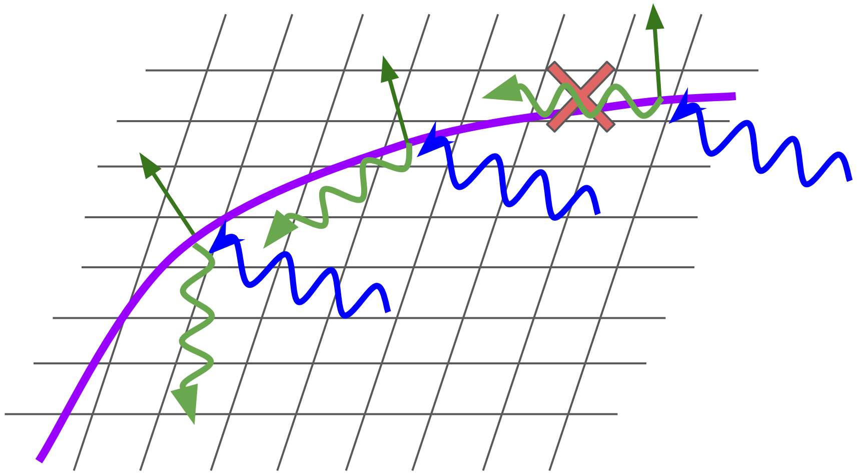
Why Do Reflections Fail Only in Some Regions?

Grey grid represents crystal lattice structure, which determines $\vec{k}-\vec{v}_g$ relationship.

Purple is curved sidewall of CDMS detector, which is sliced through lattice structure.

Green are normals to detector surface at different places. Notice that they are at different angles to lattice.

Result will be reflected phonons travelling in different directions, sometimes "outward"!



Phonons (blue) propagate along preferred directions, and avoid other directions, in the lattice ($\vec{v}_g$ is not isotropic)

G4CMP-317: Simplified Surface Propagation

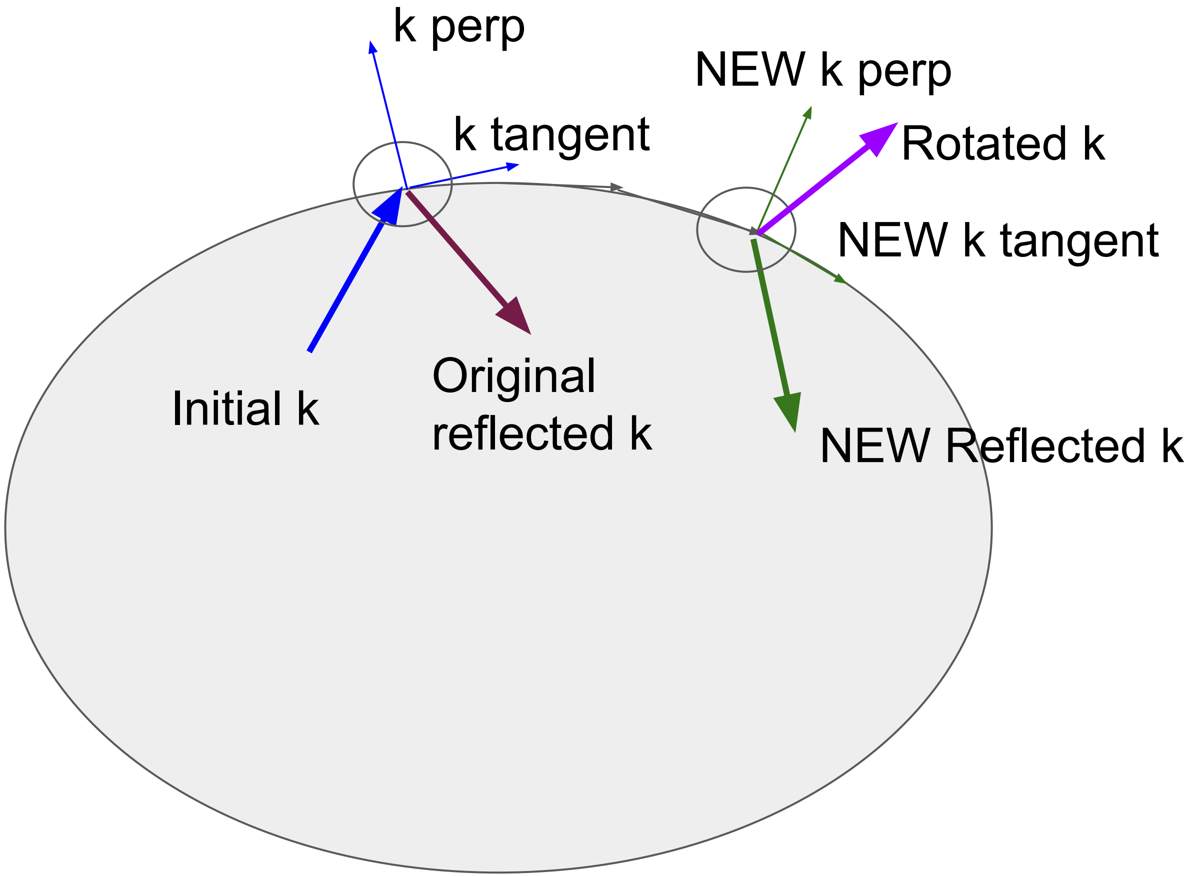
Phonons at a crystal surface may downconvert* to a reflected bulk mode and a surface propagation mode, if the putative reflected wave vector cannot propagate in the bulk (i.e., the reflected $\vec{k}$ has a $\vec{v}_g$ which is not inwardly directed).

The surface wave will propagate along $\vec{k}_{\mathrm{tangent}}$ following surface curvature until it reaches a point where the wave vector does have an inwardly directed $\vec{v}_g$.

Defining surface mode particles, with their own transport and interactions, seems too complex to address the simple problem we have. Instead, we can do a search along the surface direction to find a position where the lattice orientation allows inward bulk propagation.

* In this presentation, we are neglecting the "downconversion" process, and focusing only on how to implement the surface propagation.

Basic Displacement Search Algorithm



Scale is exaggerated!

In practice, 50-100 μm steps are appropriate

Usually need tens of mm to find successful endpoint

At each step, get $\vec{v}_g$ for new attempt at reflected $\vec{k}$

If still failing, take another step until maximum limit reached

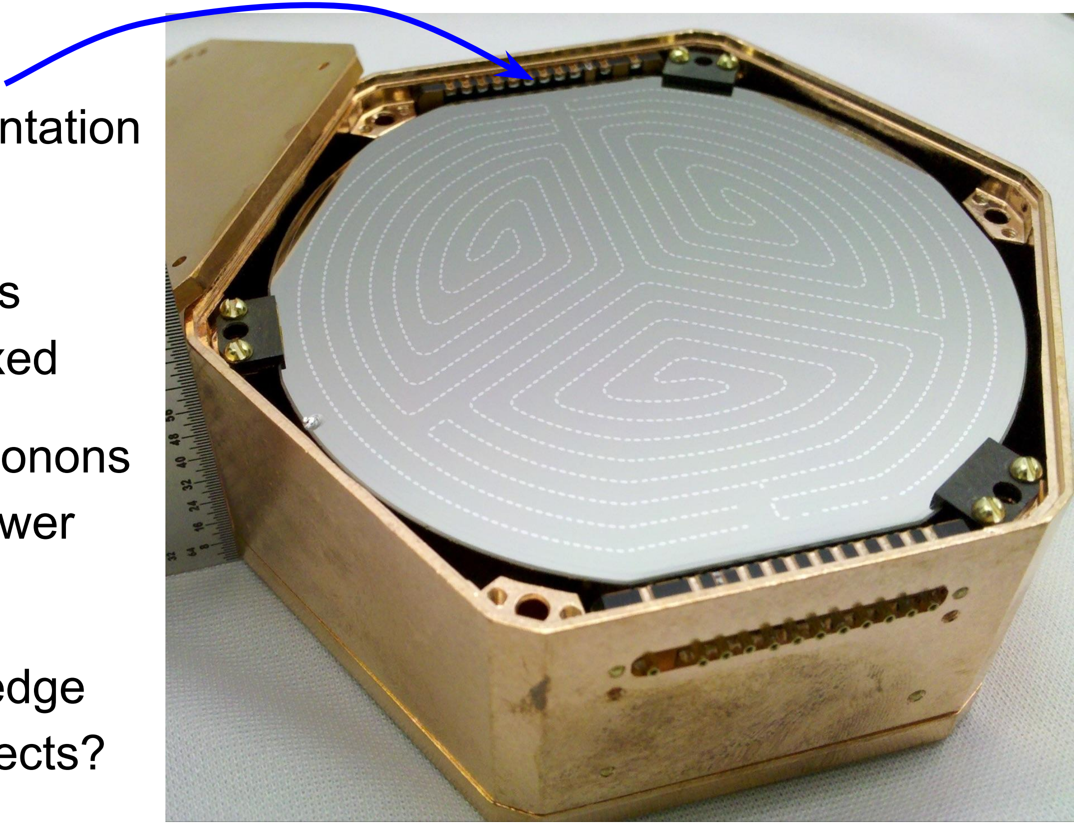
Complications In Real Detectors

CDMS "cylindrical" detectors have machined flats to identify lattice orientation and alignment in apparatus

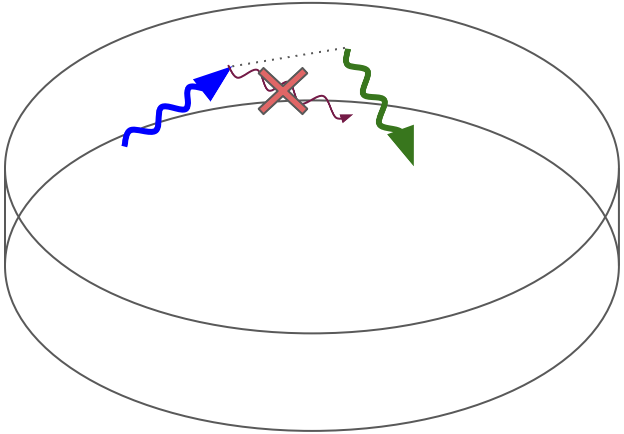
Searching "along" flat for reflection is waste of time, since orientation is fixed

Detectors much wider than thick; phonons could be "aimed" toward upper or lower faces when they hit sidewall

How do we treat surface search at edge discontinuities, including Geant4 effects?

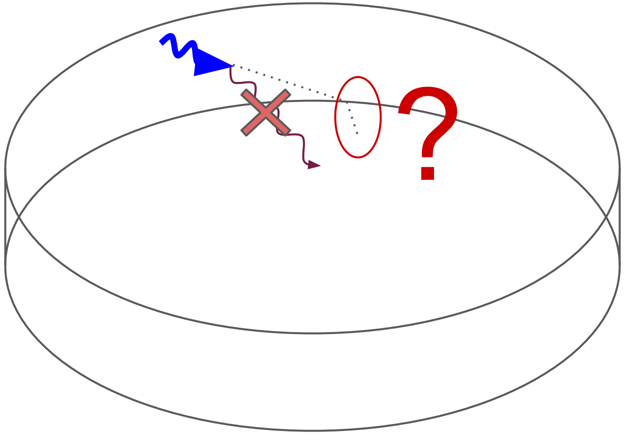


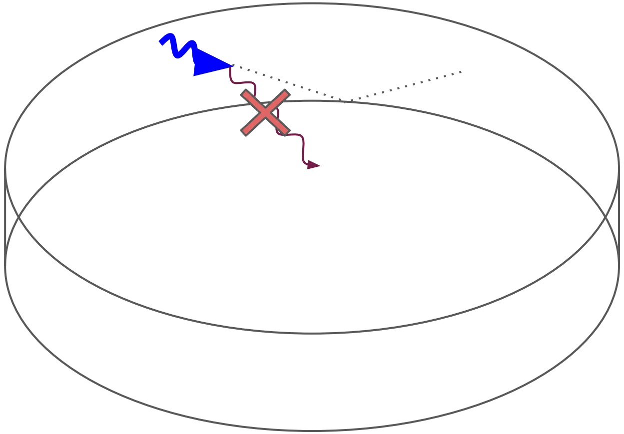
Phonon Reflection via Surface Displacement



Blue phonon can't reflect at incident point. Search along direction to find point where reflected phonon (green) can propagate

What if the search hits an edge? Would a surface mode really "turn the corner?"

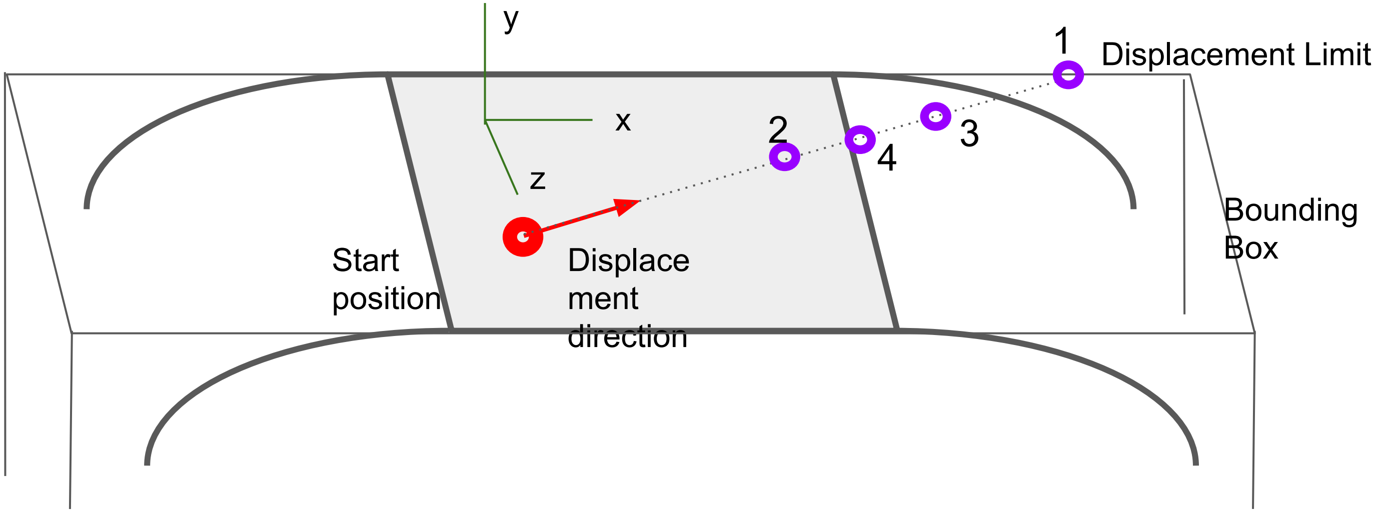




We guess not. We "reflect" the search at the hard edge, until we find a good position.

Finding an Edge When Stuck on a Flat

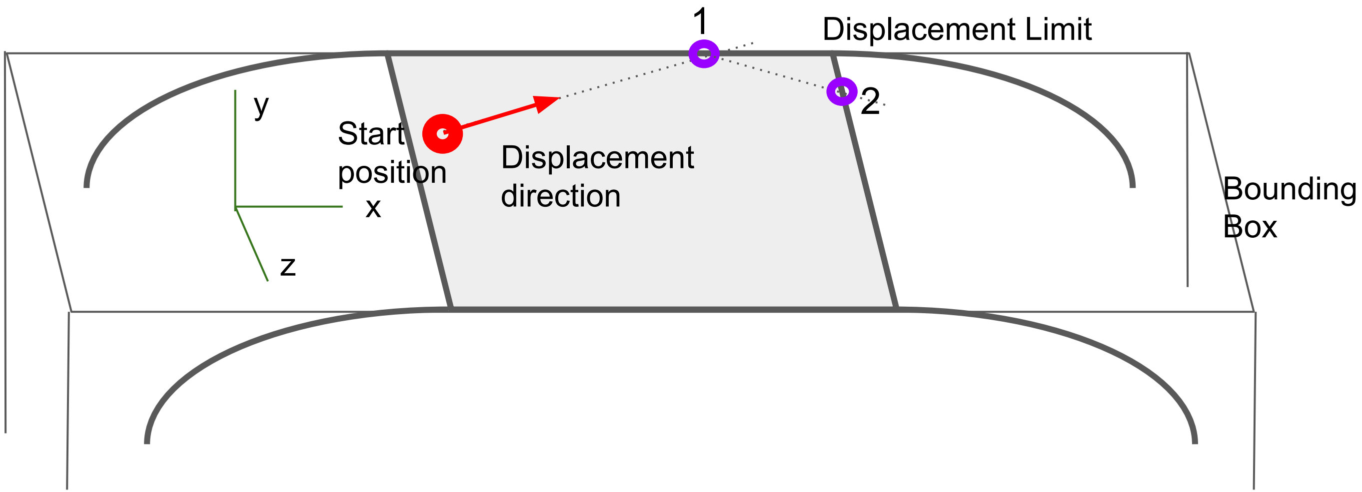
- Purple (1) is where the displacement direction intersects the bounding box.
- Points (2), (3) and (4) are successive binary search points (each is half the distance from the previous).
- At each point, test whether the search has left the surface.



- When the search distance matches (or less than) the "displacement step" length, and you've got one point on the surface and one point off, then you've found the edge of the flat.
- If (4) is still on the flat surface, then you jump that far plus one "displacement step."

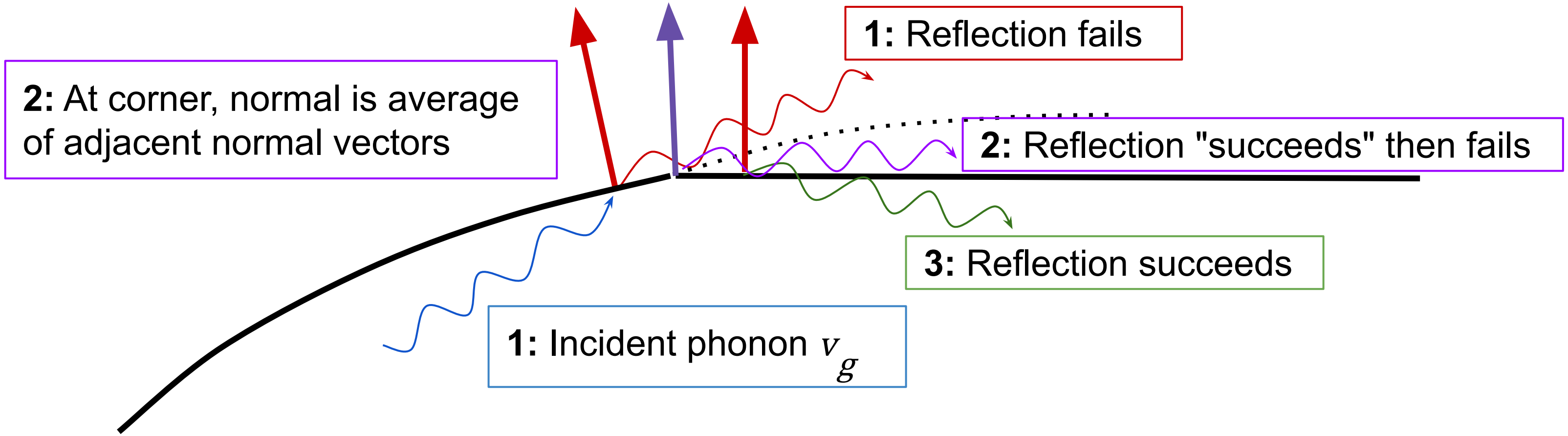
What To Do at Edge of Sidewalls?

- Purple (1) is where the displacement direction intersects the bounding box.
- Since (1) is still on the flat, what do we do?
- If we follow the displacement direction around the corner, we end up on the flat face (here -Z) of the detector.



- Another option would be to "reflect" the displacement direction at (1), and find the edge of the flat along the new direction (2), starting from (1).
- This same discussion applies during the search along the sidewalls, if a ±Z edge is encountered during the displacement search.

Reflection succeeds and fails at edge of flat

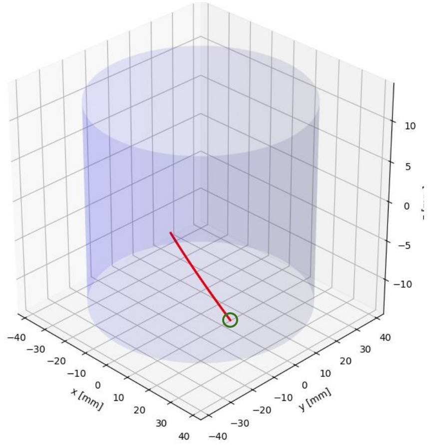


Reflection fails at curved surface (**1**): reflected k gives an outward v_g. We search along surface to find point where v_g is inward.

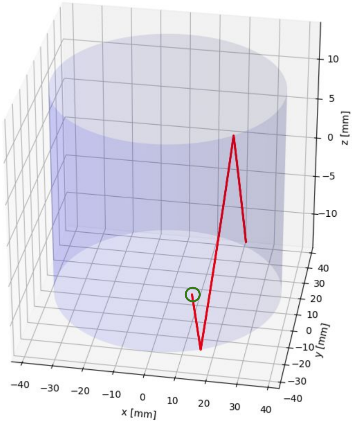
At edge of flat, reflected v_g appears to succeed, as it is inward of curving surface. But it is outward of flat (**2**), so on next step phonon is killed. Need to be fully on flat (**3**).

Performance of Specular Reflection Code

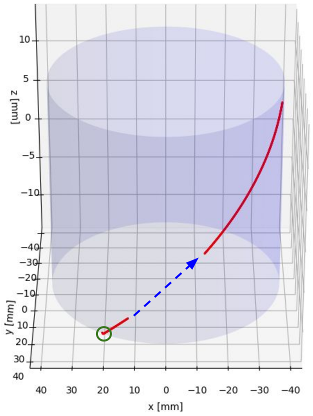
Simple Search



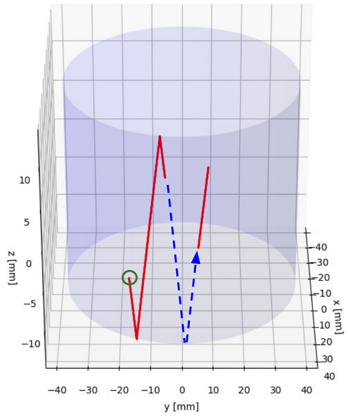
Hard Edges



Skipping Flat Face



Hard Edges & Flat



Green circle shows bulk phonon incident on surface

Dotted arrows show search skipping across flats

Generated 10 keV "phonon burst" at center of Ge detector: 1.21M phonons

Displacement search in 100 µm steps, up to 1000 steps before falling back

Detector Orientation	**Diffuse**
Origin & Unrotated	1033
Rotated (120 degree)	991
Displaced (.5 .5 .5 m)	1023
Rotated & Displaced	974

Technical Implementation Notes

Main displacement search is linear steps, does increase CPU time, but extremely small compared to overall charge and phonon tracking

Displaced reflection does not behave like normal "optical reflection" in Geant4

- Track cannot be allowed to leave volume or have "zero-length reflection step"
- Custom **`G4CMPParticleChangeForPhonon`** which handles this

Geant4 has "glitches" when trying to deal with points very close to surfaces

- "Inside" vs. "Outside" vs. "Surface" are problematic
- Especially near corners, tolerance can confuse which surface is relevant
- Several internal routines in reflection code to work around glitches

Status and Plans

Implemented on branch G4CMP-317 by Wade Lamberson and Nolan Tenpas

Validation of software performance completed, release on **g4cmp-V09-07-00**

Using feature branch with CDMS simulations to compare with real data

Some open issues still need to be developed and incorporated phonon transport

- Implement downconversion-like splitting of incident phonon into bulk and surface mode
- Should reflection include mode conversion? (G4CMP doesn't do polarization)
- Is re-entry of surface mode to bulk simple, or also downconversion-like?
- These are likely to be several months of work

Backup Slides

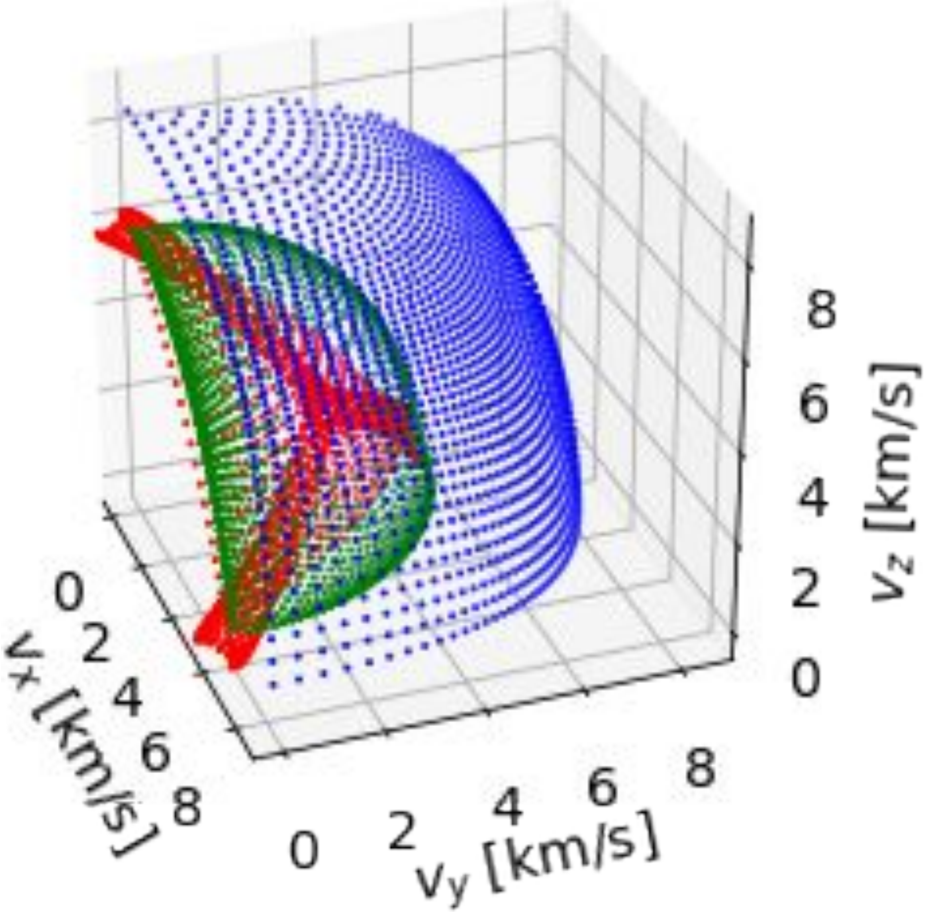
Group Velocities in 3D

phononTS (red)
phononTF (green)
phononL (blue)

Both transverse modes are anisotropic, but in complementary ways

- TF mode clusters along a linear combination of 2 axial vectors
- TS clusters along linear combination of axial vector and octant central vector

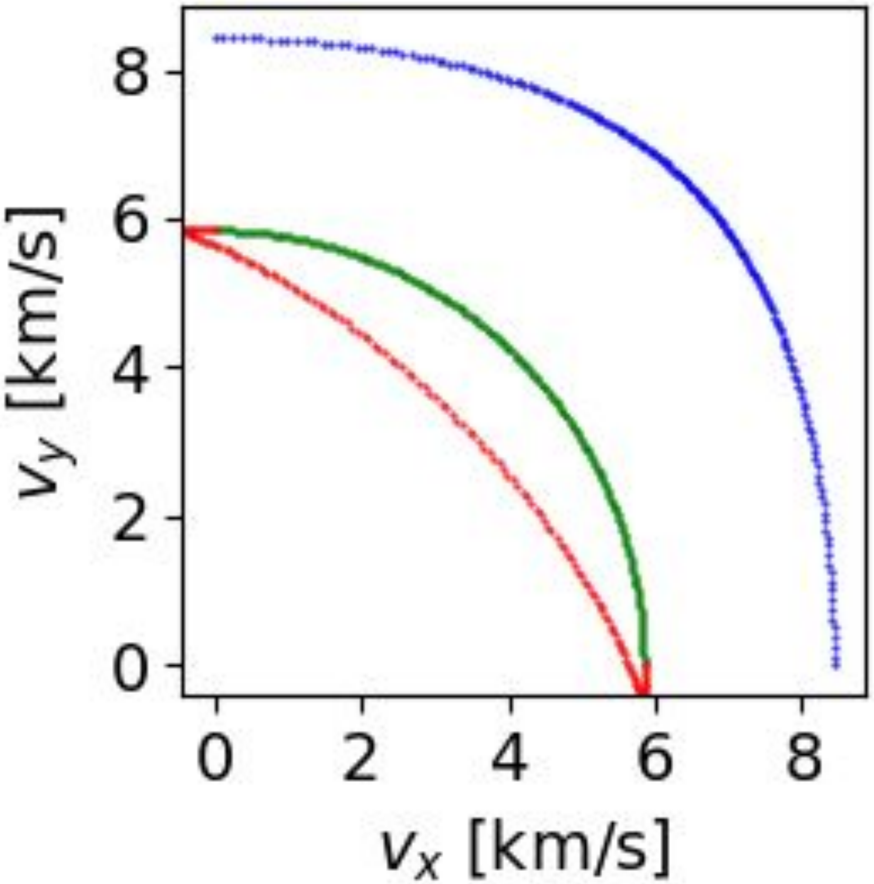
Plots are for silicon, but generally the same for germanium, though velocity magnitudes are different



Group Velocities in 2D

phononTS (red)
phononTF (green)
phononL (blue)

Plots are for silicon, but generally the same for germanium, though velocity magnitudes are different



Plotting package courtesy of Richard Lawrence; available at GitHub: kelseymh/TAMU_Jupyter/phonon_Vg-K_RL.ipynb

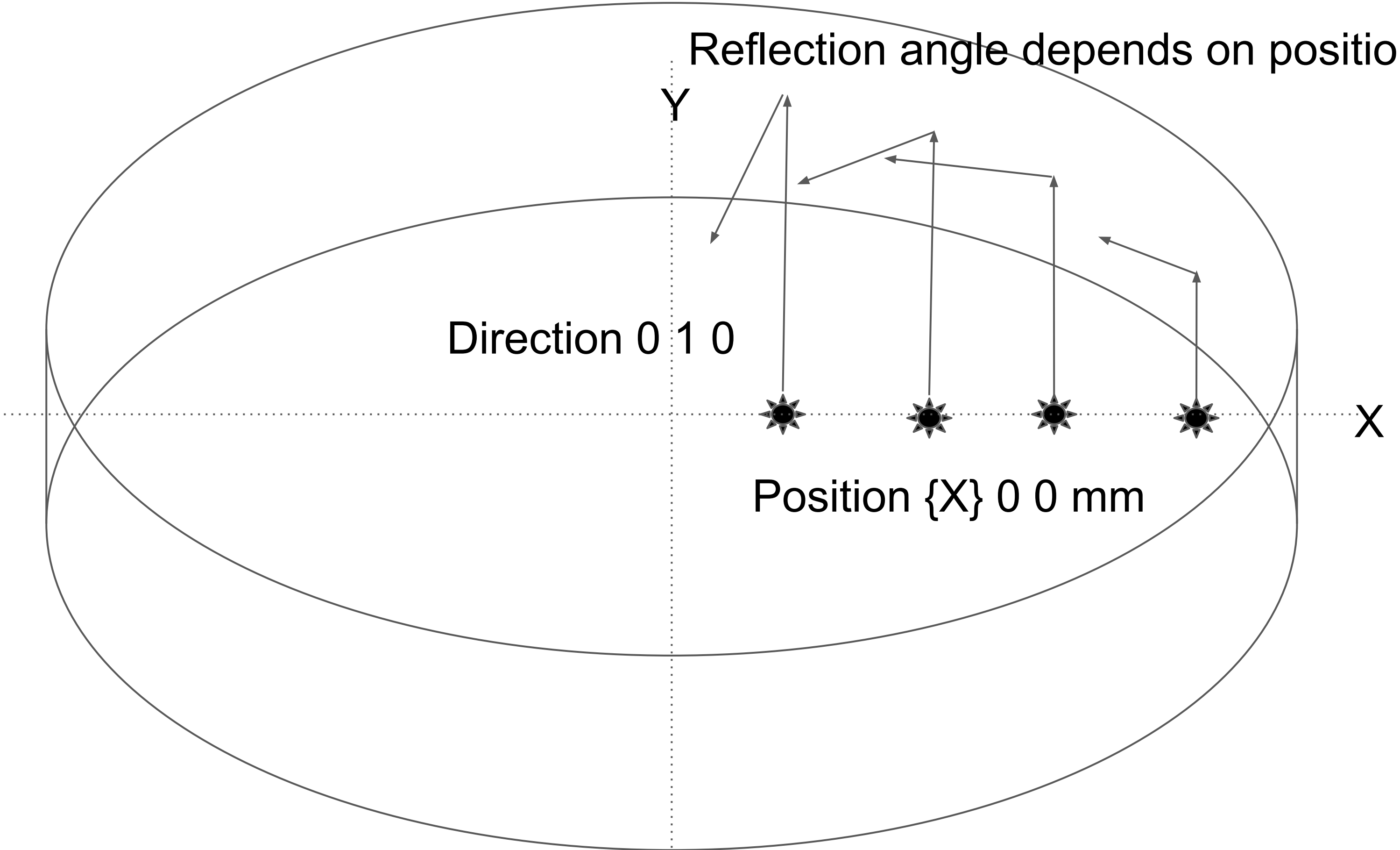
G4CMP-317: Offset Point for Phonon Reflection

We are proposing to implement this pragmatically, as part of **`G4CMPPhononBoundaryProcess::DoReflection()`**

1. Start at the reflection point
2. Step along the surface in the tangential direction of $\vec{k}$ (or $\vec{v}_g$)
3. Get the local normal at the new surface point
4. Calculate $\vec{k}_{\mathrm{refl}}$ and $\vec{v}_g$ using the new local normal, test if $\vec{v}_g$ is inward
5. Repeat until 4 succeeds.
6. Update ParticleChange to shift to the new position with the new $\vec{k}_{\mathrm{refl}}$ and $\vec{v}_g$.

Note that this simple description assumes a curved surface, and that both the initial reflection failure and the entire search stays on the curved sidewall.

Single Phonon Testing of Specular Reflection



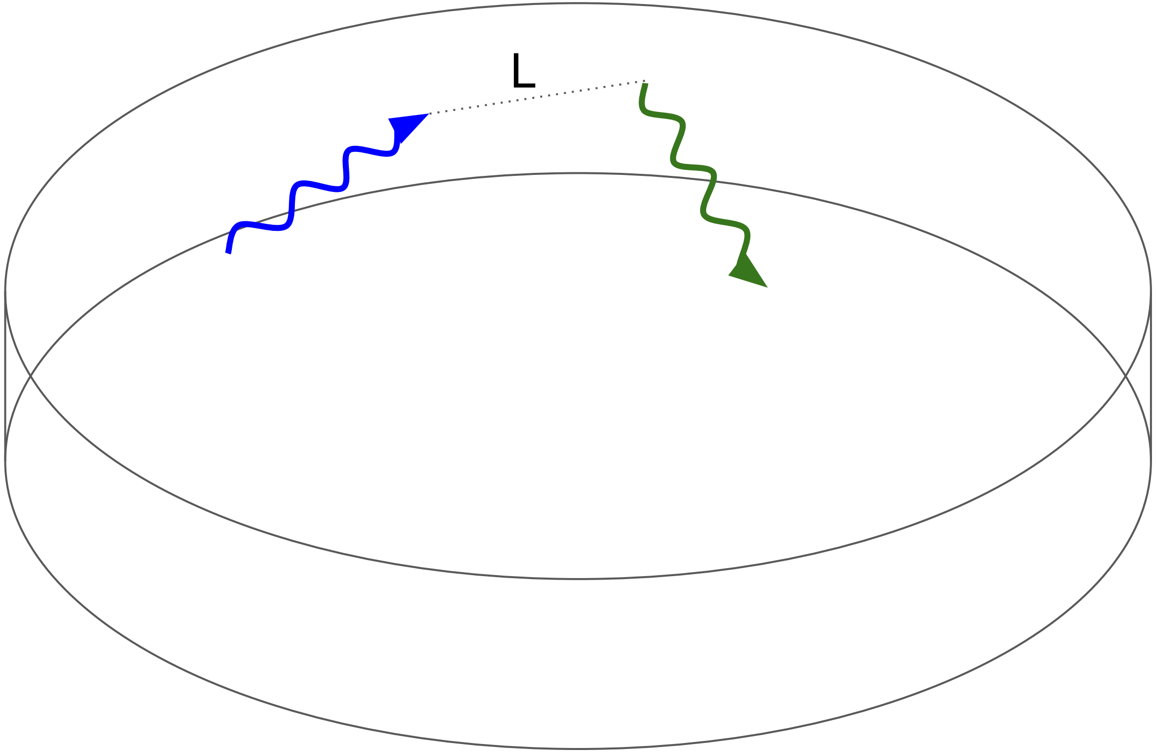
Arc Length of Displacement

Treat detector as simple cylinder; don't correct for flats

$L = [\ \Delta z^2 + (r\Delta\varphi)^2\ ]^{½}$

$2\Delta\varphi = \sin^{-1}(L_{xy}/2r)$

$L_{xy} = [\ \Delta x^2 + \Delta y^2\ ]^{½}$



Implementation Notes

`G4CMPPhononBoundaryProcess` inherits from **`G4CMPProcessUtils`**, so all of its features are automatically available

`GetCurrentVolume()` returns physical volume (PV) of track

- Use **`PV->GetLogicalVolume->GetSolid()`** to retrieve the **`G4VSolid*`** "volume shape"

Use **`GetLocalPosition(track)`** to get the position in local coordinates

Use **`GetSurfaceNormal(track)`** to get outward normal at track position

Use **`GetLocalWavevector()`** to get wavevector in local coordinates at the start of the track

While stepping in local coordinates along surface, use **`solid->SurfaceNormal(pos)`** to get new surface normal

With final result in local coordinates, transform back to global coordinates for ParticleChange using **`RotateToGlobalPosition(pos)`**, **`RotateToGlobalDirection(wv)`**