Geant4 Condensed Matter Package

From the SuperCDMS Detector Monte Carlo Group

Rob Agnese

September 12, 2016

What is G4CMP?

Library built on top of Geant4 to add support for low energy condensed matter physics.

Some Features:

- Awareness of various crystal symmetries (and amorphous solids)
- Transport and interactions of (acoustic) phonons
- Transport and interactions of charge carriers (electrons and holes)
- Support for local electric fields from 3D potential maps.
- Seamlessly transition from "normal" Geant4 energy deposits to G4CMP particle physics.

Using G4CMP

Install from GitHub.com,

```
git clone https://github.com/ragnese/g4cmp
```

Set up Geant4 environment as usual.

Add **G4CMPPhysicsList** and a **G4LatticePhysical** to your application.

Link application to libG4CMP.{so,a} and then G4. (Order matters!)

Lattice Configuration

G4LatticeLogical - Physics properties

- Crystal symmetry group, lattice constants
- Stiffness tensor
- Electron/hole effective masses, conduction band "valley" directions
- Scattering rates/parameters

G4LatticePhysical - Orientation of G4LatticeLogical

- Attaches to G4VPhysicalVolume
- Owns G4LatticeLogical (calls through for physics params)
- Crystal orientation from Miller indices and rotation angles

Using G4Lattice{Logical,Physical}

```
G4VPhysicalVolume* G4DMCDetectorConstruction::Construct() {
[...]

// Load the lattice for the material we're using
G4LatticeManager* latManager = G4LatticeManager::GetLatticeManager();
G4Material* germanium = nistManager->FindOrBuildMaterial("G4_Ge");
latManager->LoadLattice(germanium, "Ge"); // latManager has LogicalLattice info

// Define our detector volume as usual
G4VSolid* germaniumSolid = new G4Tubs("germaniumCyl", 0.*cm, 3.81*cm,
                                      zipThickness/2., 0.*deg, 360.*deg);
G4LogicalVolume* germaniumLogical = new G4LogicalVolume(zipSolid, germanium,
                                                        "germaniumLogical");
G4VPhysicalVolume* germaniumPhysical = new G4PVPlacement(0, DMCcoords,
                                                         germaniumLogical,
                                                         "germaniumPhysical",
                                                         worldLogical,
                                                         false, 0);

// Attach a physical lattice to our physical volume
G4LatticePhysical* detLattice = new G4LatticePhysical(latManager->GetLattice(germanium));
detLattice->SetMillerOrientation(1,0,0,0.);
// Register pVol <-> pLat relationship for look up by processes
latManager->RegisterLattice(pv, detLattice);

[...]
}
```

Configuration Files

Germanium

```
# Crystal parameters
cubic 5.658
stiffness 1 1 1.26e11
stiffness 1 2 0.44e11
stiffness 4 4 0.67e11
# Phonon parameters
beta -0.732 gamma -0.708
   lambda 0.376 mu 0.561
scat 3.67e-41
decay 1.6456e-54
LDOS 0.097834
STDOS 0.53539
FTDOS 0.36677
# Charge carrier parameters
vsound 5324.2077
l0_e 257e-6
l0_h 108e-6
# Hole and electron masses
hmass 0.350
emass 1.588 0.081 0.081
valley  -45 35.2644 90 deg
valley   45 35.2644 90 deg
valley  135 35.2644 90 deg
valley -135 35.2644 90 deg
# Intervalley scattering
ivField 217
ivRate 6.72e-2
ivPower 3.24
```

Silicon

```
# Crystal parameters
cubic 5.431
stiffness 1 1 1.656e11
stiffness 1 2 0.639e11
stiffness 4 4 0.795e11
# Phonon parameters
beta -0.429 gamma -0.945
   lambda 0.524 mu 0.680
scat 2.43e-42
decay 7.41e-56
LDOS  0.093
STDOS 0.531
FTDOS 0.376
# Charge carrier parameters
vsound 9000.0
l0_e 16.9e-6
l0_h 7.5e-6
# Hole and electron masses
hmass 0.50
emass 0.91 0.19 0.19
valley   0  0 90 deg
valley  90  0 90 deg
valley   0 90 90 deg

# Intervalley scattering
ivField 217
ivRate 6.72e-2
ivPower 3.24
```

```
# angstrom (Lattice constant)
# C11, C12, C44 in Pa


# S. Tamura et al., PRB31(4), 1985
# 10\^11 Pa

# s\^3
# s\^4
# S. Tamura et al., PRB44(7), 1991


# m/s (approximate)
# m
# m

# per m(electron)
# per m(electron)
# Euler angles phi,theta,psi


# V/m
# s (Using Ge rate for Si for now)
# exponent in sqrt(field^ivPower)
```

Physics Processes

Mode mixing (due to isotope scattering)

Anharmonic decay (L $\rightarrow$ L$'$ T, L $\rightarrow$ T T)

Intervalley scattering (no $\vec{k}$ change, just v $\rightarrow$ v$'$ index)

Luke-Neganov phonon emission (discrete, one step per emission)

Boundary processes á là **G4OpBoundaryProcess**

- Specify absorption, reflection, transmission probabilities
- Thresholds for absorption, total reflection
- Can fill HitCollection for digitization, sensor simulation
- If **G4CMPVElectrodePattern** is attached to surface, absorb under specified conditions

Particle Kinematics

Non-Newtonian (non-SR) kinematics, not G4 compatible

Geant4 Assumes:

- $E^2 = p^2 + m^2$
- $\hat{v} = \hat{p}$
- if (m == 0) v = c;

G4CMP Needs:

- Phonon $\vec{k}$ vs. $\vec{v}_g$; m=0, $v \neq c$
- Electrons have m_{ij} tensor
- Brillouin valleys, Herring-Vogt space

Particle Kinematics

G4CMP processes **CAN NOT** use the usual G4Track accessors: GetMomentum(), GetKineticEnergy(), etc.

G4CMPProcessUtils Implements the necessary helpers: GetMomentum(G4Track*), etc. Also applies to filling the ParticleChange.

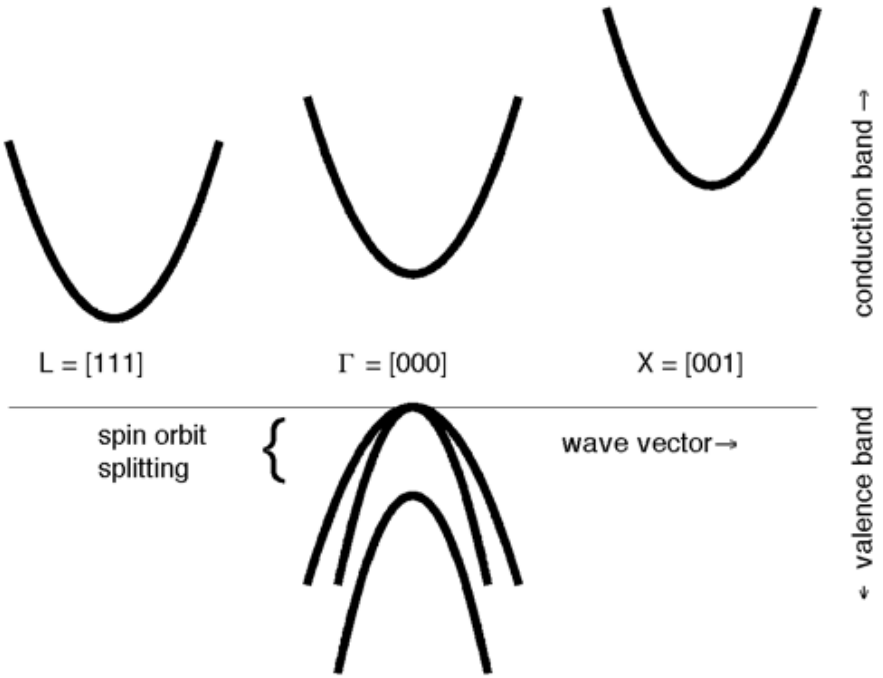
G4CMP uses **G4AuxiliaryTrackInformation** to store and retrieve extra kinematic quantities for each track (e.g. wave vector direction of phonons)

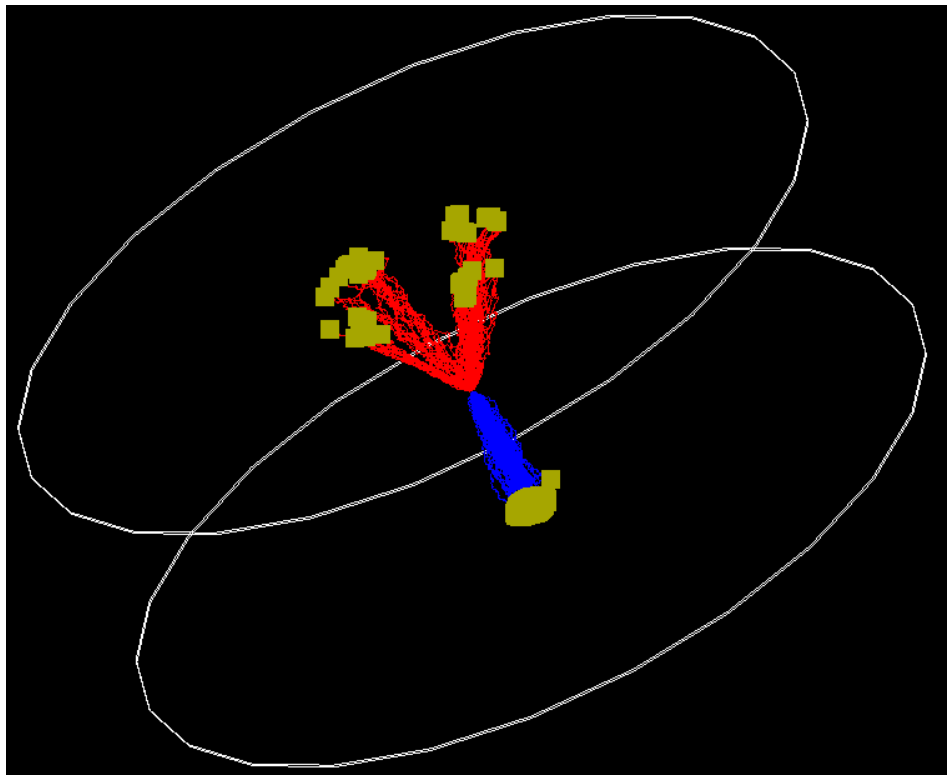
G4CMPStackingAction populates TrkInfo for generated primaries

Secondaries created with TrkInfo filled automatically by inheriting from **G4VPhononProcess** or **G4CMPVDriftProcess**

Charge Carrier Propagation

G4CMP uses a tensor effective mass for transportation of charges

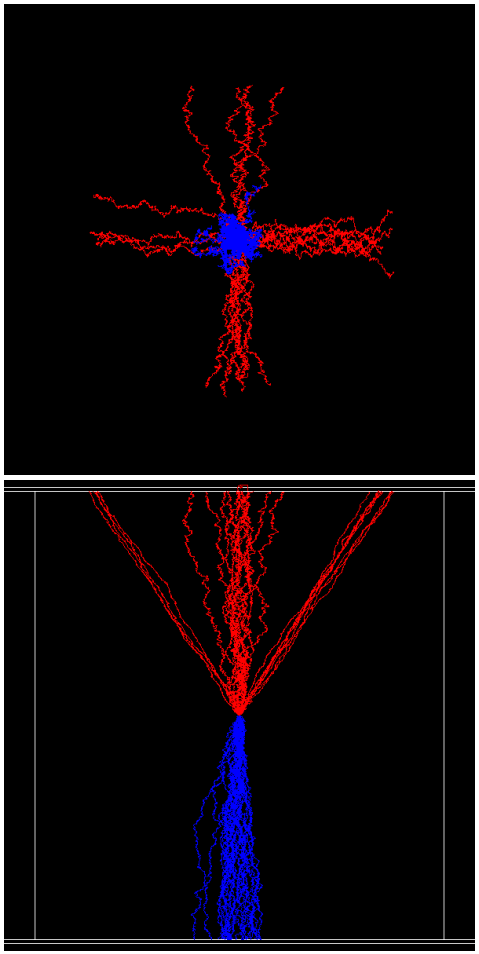




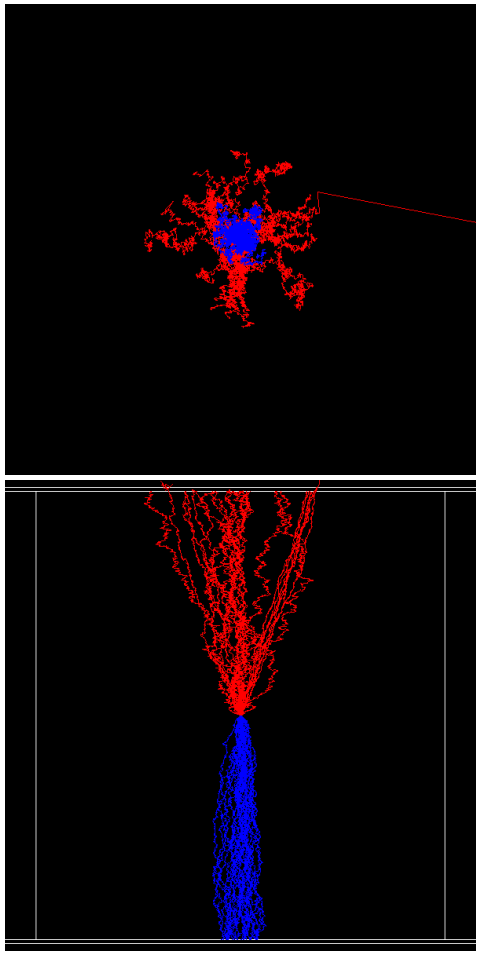
Electron Trajectories

76×25 mm crystals, uniform 4 V field, no IV scattering

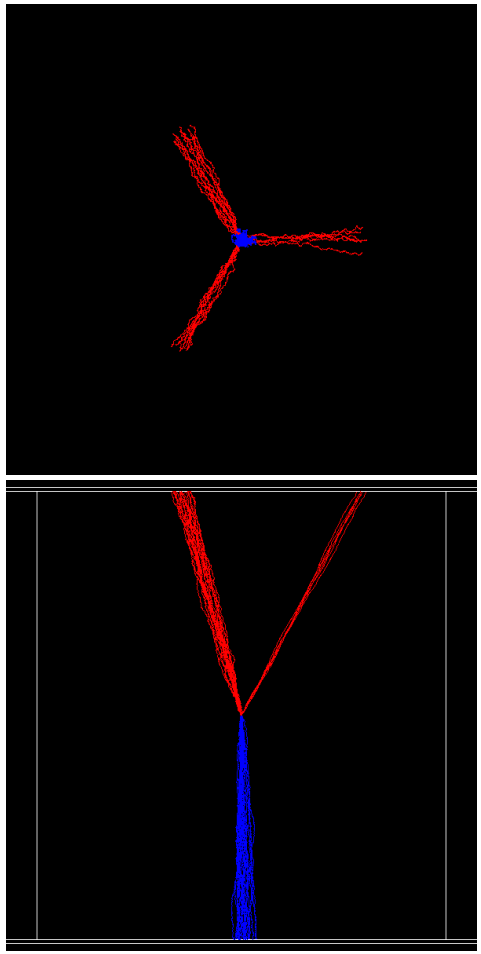
Ge (100)



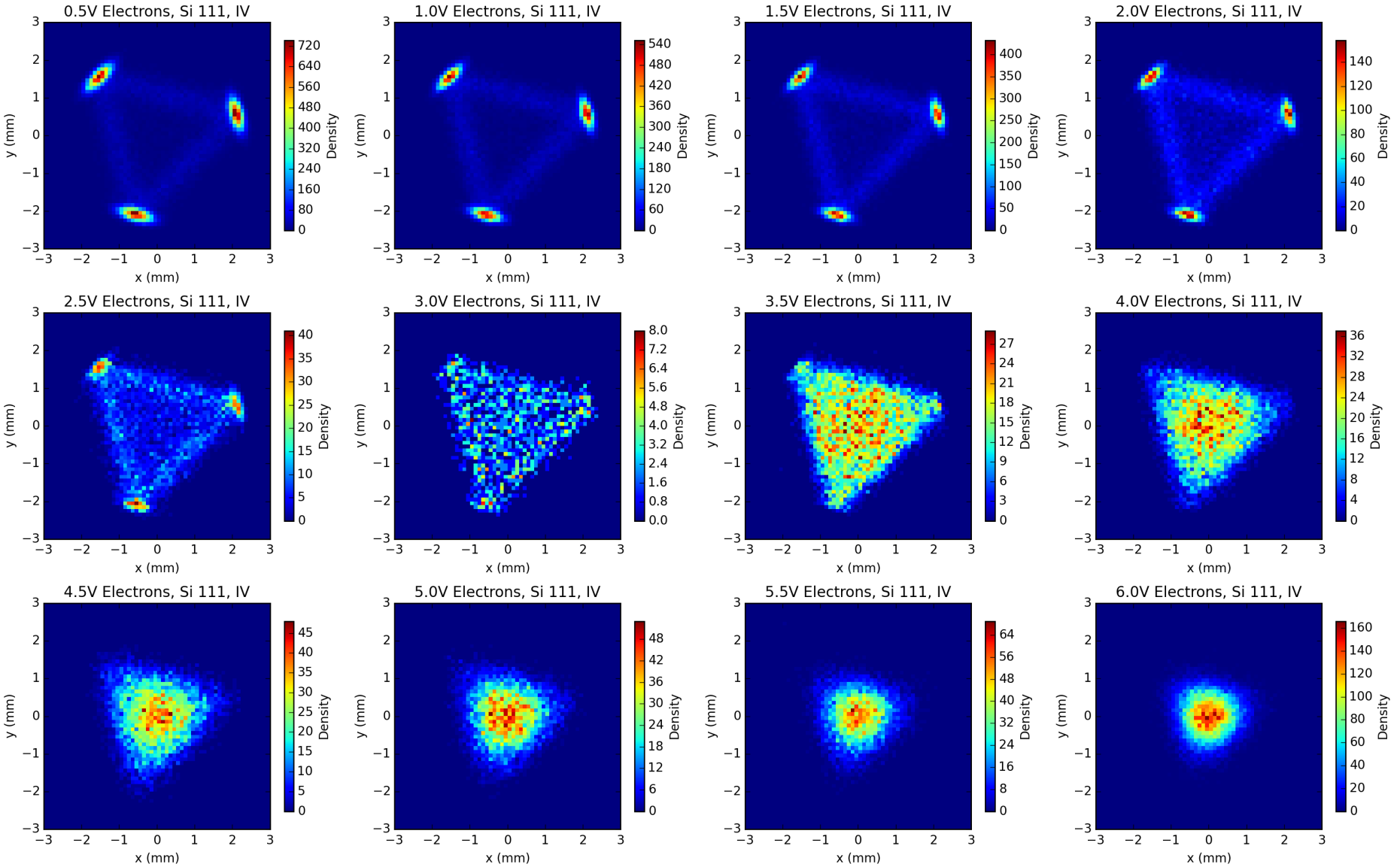
Ge (111)



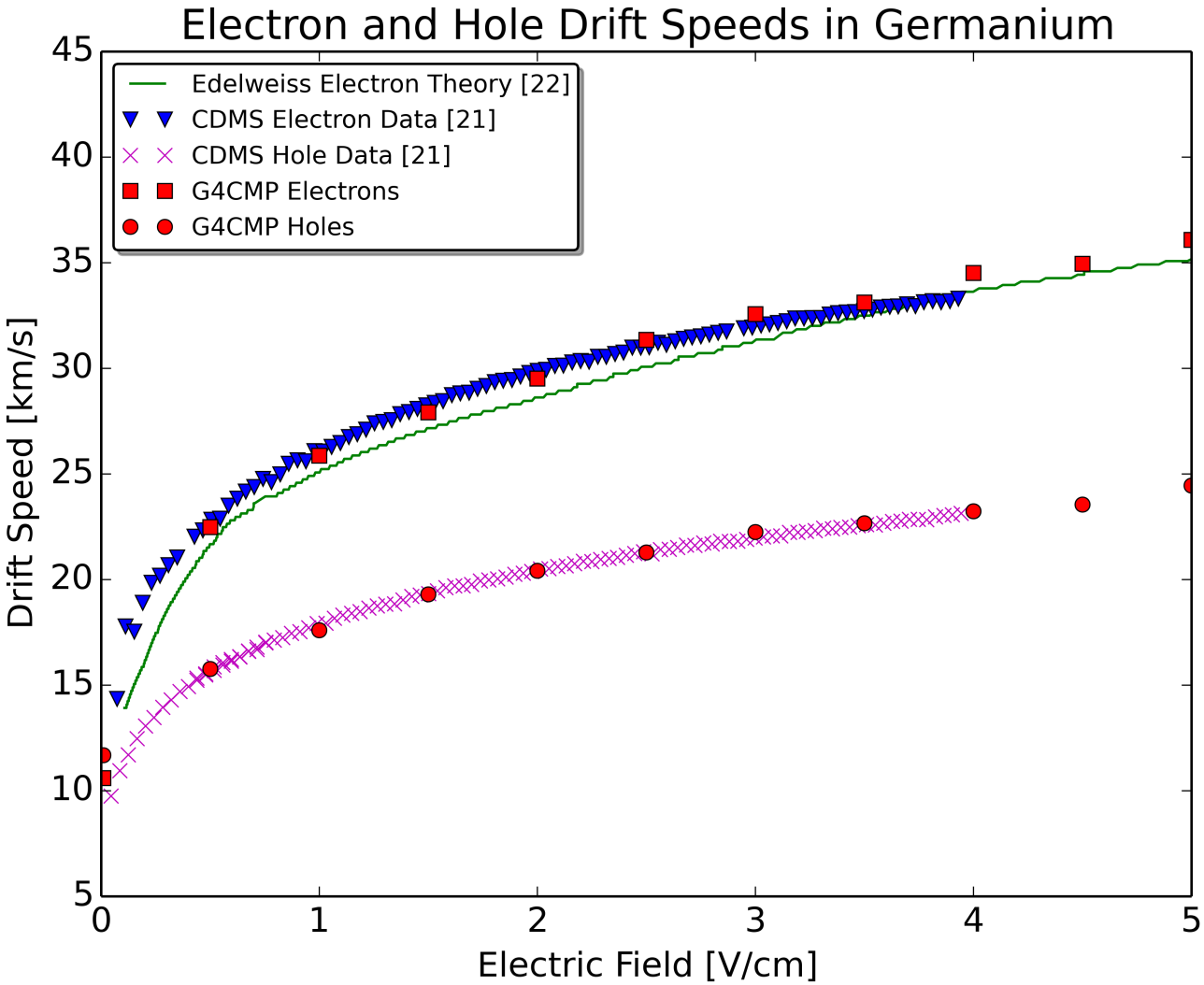
Si (111)



Electron Hits in Silicon (111)



Charge Carrier Drift Speed



Phonon Propagation

Phonon mode m (L, ST, FT), wavevector $\vec{k}$, $\hat{n} = \vec{k}/|k|$

Compute Christoffel matrix $D_{ik} = n^j n^l C_{ijkl}$

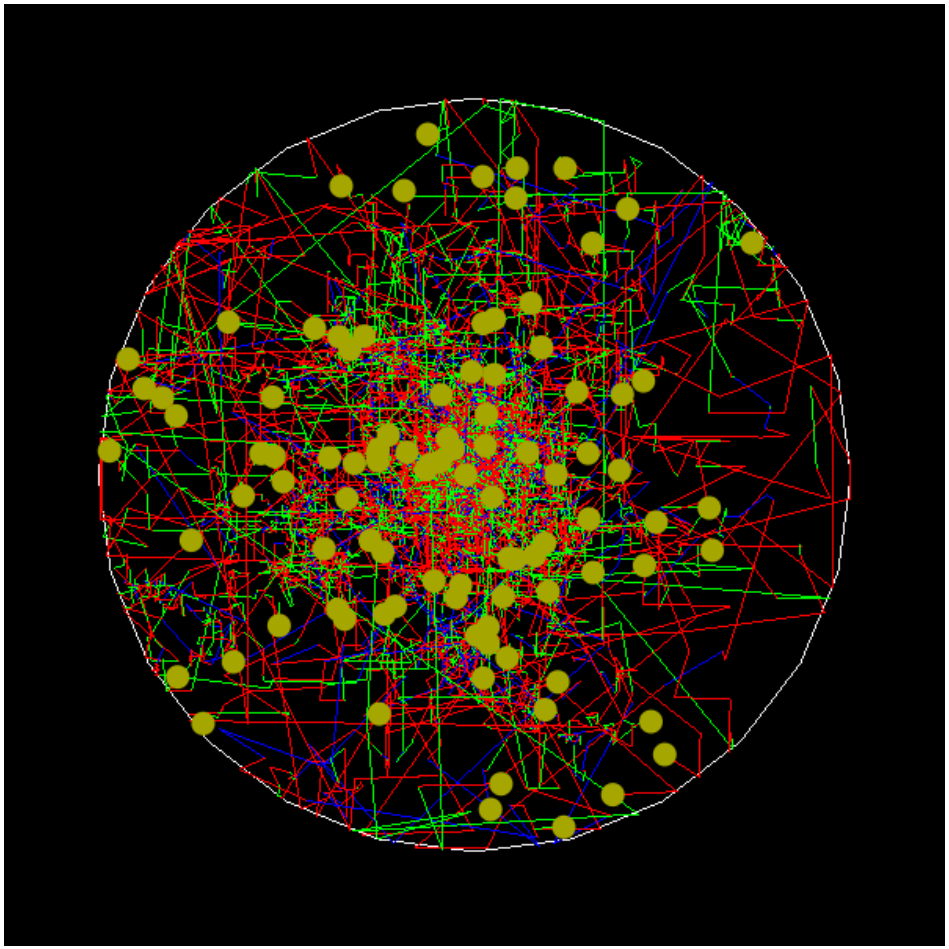
Use eignvector solver on D_{ik}

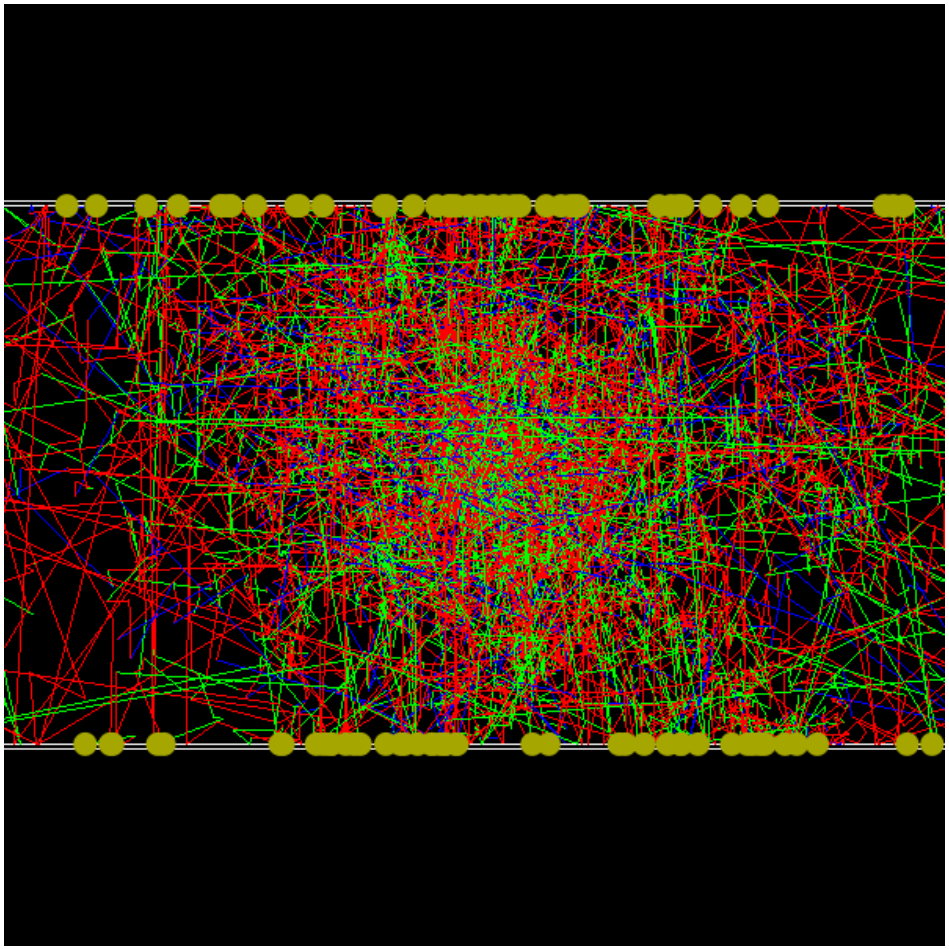
- Eigenvalues are phase speeds $v_s^{(m)}$
- Eigenvectors are true polarizations $\vec{e}^{(m)}$
- Group velocities extracted including material density ρ
  $\vec{v}_{g,l}^{(m)} = \frac{1}{\rho v_s} e^{(m),i} n^j e^{(m),k} C_{ijkl}$

Solver not fast enough to use during running ($\times 3$ lookup)

Populate lookup tables in $\hat{n}$ (ϕ, θ) by **G4LatticeLogical** ctor

Phonons in Germanium





What's New in 2016?

- Support for crystal symmetries other than simple cubic
- Lattice config file for Si (already had Ge)
- Support for sensitive electrode patterns at boundaries (G4CMPVElectrodePattern)
- Charges recombine and release bandgap energy at rest
- Phonons reflect (optionally) diffusely from surfaces
- Kaplan quasi-particle downconversion for phonons that absorb in a superconductor NOTE: The quasi-particles are not actually simulated
- Utilize Geant4 track biasing by weight
- User configurable minimum cutoff energies
- Lindhard partitioning of energy deposits to create primaries/secondaries

Symmetry Groups

Group	Arguments	Stiffness Components Cpq
cubic	*a*	C11 C12 C44
tetragonal	*a c*	C11 C12 C13 C16
		C33 C44 C66
orthorhombic	*a c*	C11 C12 C13 C22
		C23 C33 C44 C55 C66
hexagonal	*a b c*	C11 C12 C13 C33 C44 C66
rhombohedral	*a* α `deg|rad`	C11 C12 C13 C14 C15
		C33 C44 C66
monoclinic	*a b c* α `deg|rad`	orthorhombic +
		C45 C16 C26 C36
triclinic	*a b c* α β γ `deg|rad`	All 21 Cpq
amorphous	none	C11 C12

Energy Partitioning For New Tracks

G4CMPEnergyPartition uses Lindhard theory to partition deposited energy between phonons and charges. Adapted from Lewin And Smith, 1996:

$$\epsilon = 0.0115E \times Z^{-7/3}$$

$$k = 0.133\frac{Z^{2/3}}{\sqrt{A}}$$

$$h = \epsilon + 0.7\epsilon^{0.6} + 3\epsilon^{0.15}$$

$$LindhardFactor = \frac{kh}{1+kh}$$

Also applies the Fano factor from the crystal to vary the number of charge carrier pairs created.

G4CMPSecondaryProduction is a continuous process that picks up all energy depositions from **non-G4CMP** particle tracks and uses **G4CMPEnergyPartition** to spawn G4CMP secondaries.

TODO

Add more materials

Add support for optical phonon modes

Convert Luke phonon emission to a continuous process

Use more experimental data to validate/tune various parameters (e.g., recombination phonon distribution)

Create Bogoliubov quasiparticle type for tracking through superconductors