Stuck Charges in EPot High Voltage

Michael Kelsey
Texas A&M University
SuperCDMS DMC Meeting, 4 May 2025

The Problem (e-Log #1714, G4CMP-358)

In 2023, high voltage jobs using EPot files would occasionally get "stuck", using up full batch time without completing

Unable at the time to isolate a cause, but after some unrelated updates, problem "went away" in test jobs, so dropped it

DC3 and most CDMSlite simulations use uniform field, which doesn't have this behaviour

Recent pre-commissioning tests using the "large EPot" files for SNOLAB show same behaviour: 1 or 2 threads in a fraction of jobs would "hang" at 100V; no problems at 50V

TL;DR: Apply limit on number of steps per track; kill track when limit exceeded

Debugging: Finding the Symptom

Ran single track (technically, single e/h pair) events in HV100mm

Able to get "stuck" with 500 or more events per job, from 1 to 4 threads per job

Modified **`CDMSEventAction::BeginOfEventAction()`** to enable tracking verbosity for specific event number, re-ran identical job to identify problem

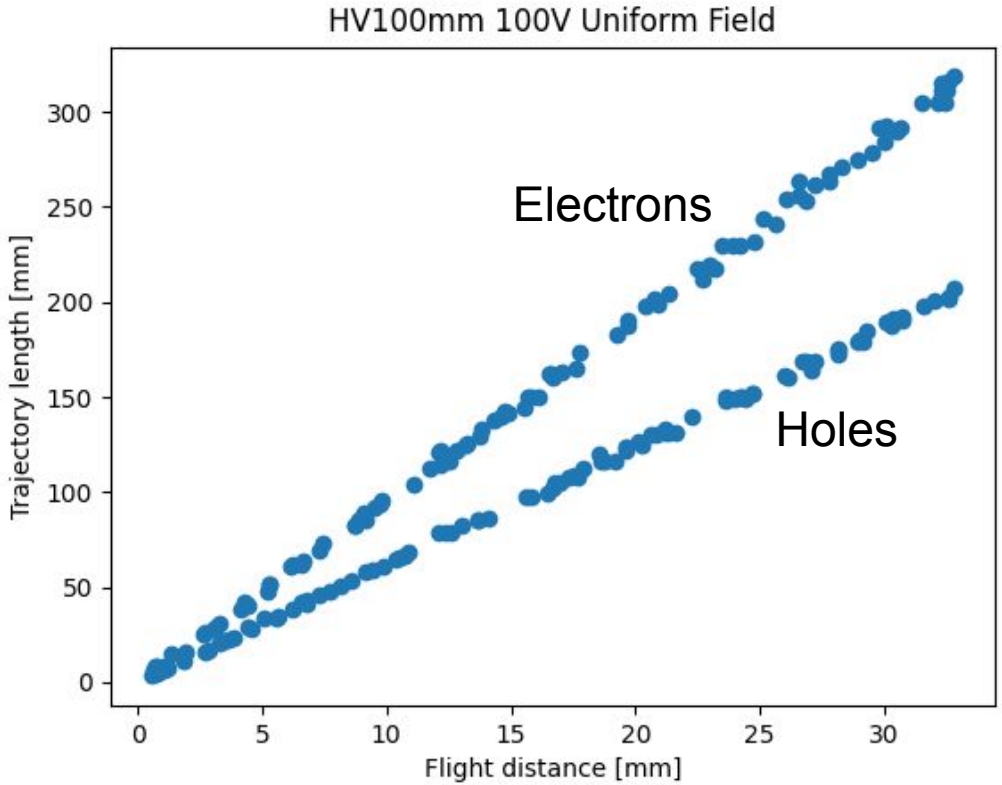
Stuck event was still running until I killed the job: Track had nearly ***100 million steps*** without changing "final" position by 100 microns! Kinetic energy below NTL threshold, just IV scattering to change direction

Hypothesis: Undesirable local minimum in EPot potential map. Is it worth trying to isolate? Update UofT COMSOL work from last year?

Trajectory vs. Flight Distance (Uniform Field)

Charged tracks have frequent large-angle scatters, so total length of trajectory can be significantly longer than "flight distance" from initial to final positions

Scattering processes depend on electric field, which is mostly uniform, except close to surfaces

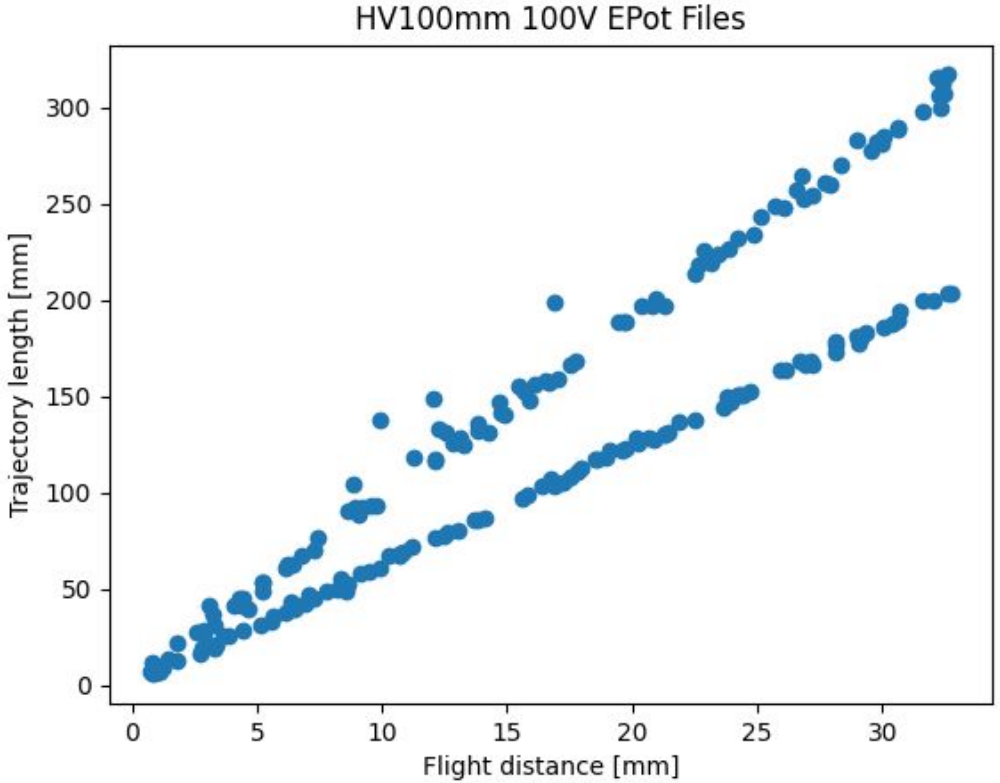


Trajectory length is stored in **mczip0.StepLen** for DMC hits starting with e-Log #2125

Trajectory vs. Flight Distance (EPot)

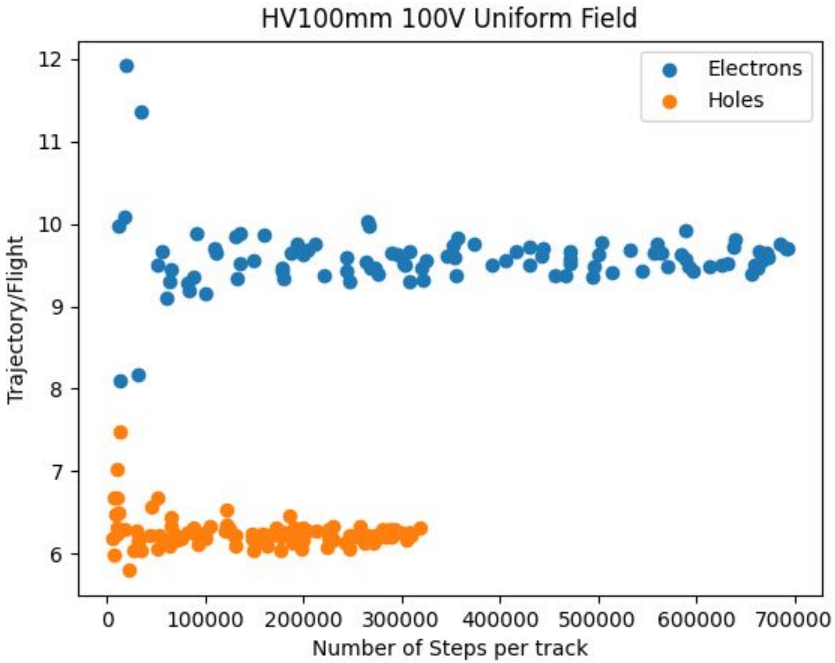
Charged tracks have frequent large-angle scatters, so total length of trajectory can be significantly longer than "flight distance" from initial to final positions

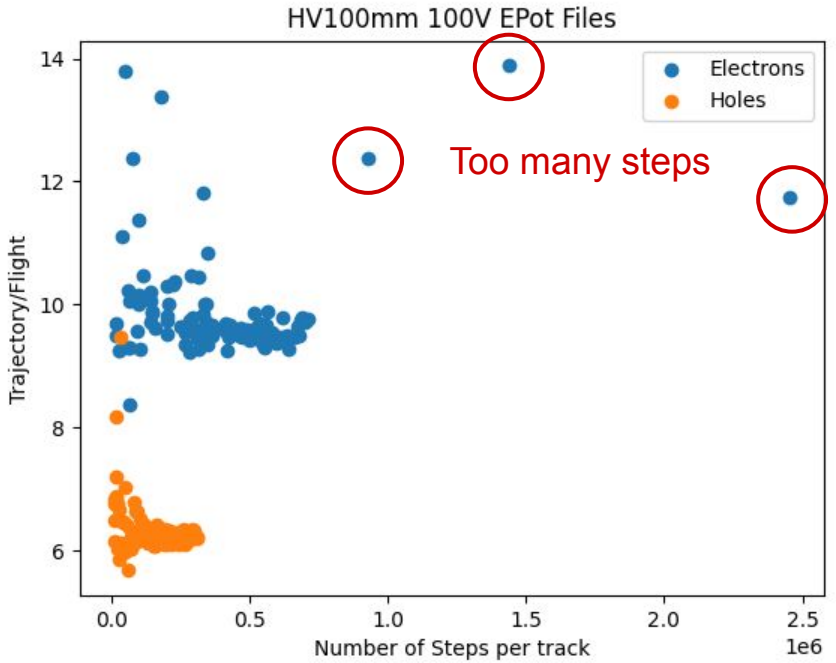
Scattering processes depend on electric field, which is mostly uniform, except close to surfaces



Trajectory length is stored in **mczip0.StepLen** for DMC hits starting with e-Log #2125

Path Ratio, Number of Steps

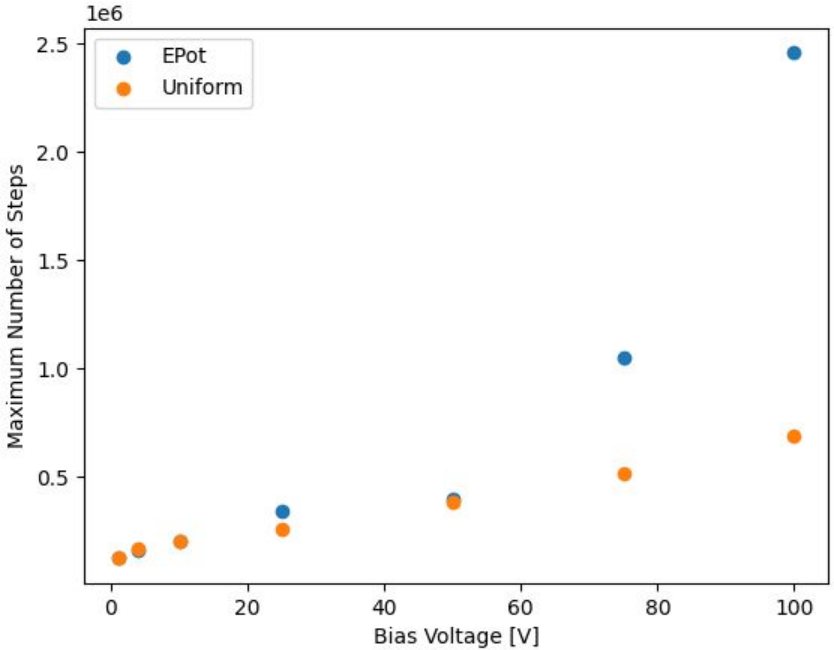


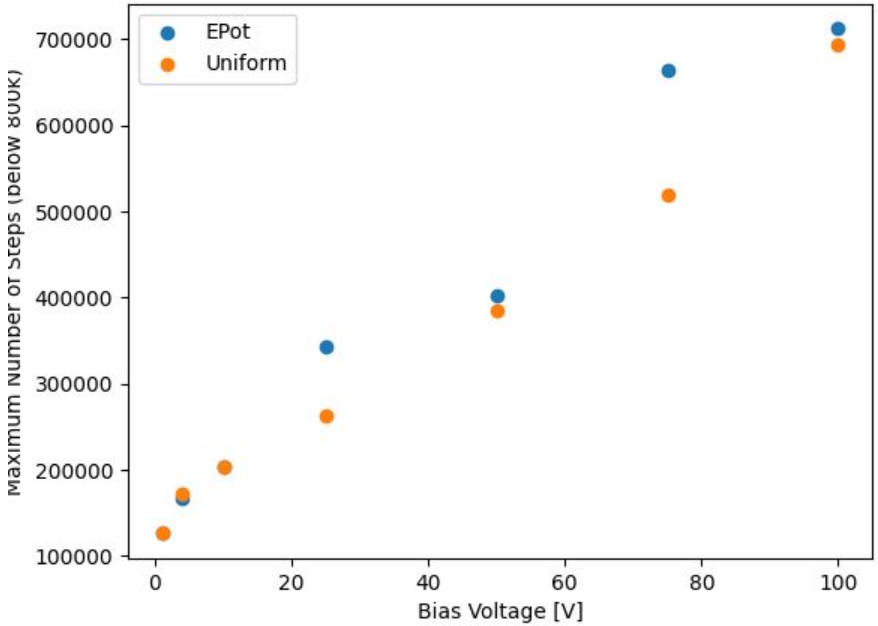


Electrons take more steps with IV

Maximum Steps vs. Voltage

Applying cut of 800k steps per track, EPot and Uniform field tracks look similar





Applying Limit on Number of Steps?

Previous page shows clear and desirable effect – no outlier stepping tracks!

10,000 track job with HV100mm, no stuck tracks, distributions look sensible

Maximum steps scale roughly linearly with voltage up to 100V

- $200{,}000 + 7{,}000*\mathrm{V}$ is a reasonable match to HV100mm (and HV100mmSi)
- Maximum flight distance, hence maximum steps, must also depend on geometry

Configurable limit will require new G4CMP release with **G4CMP-358** included

SuperSim: Can derive empirical limit for supported detectors (SNOLAB)

G4CMP: Is there a way to estimate limit from user's electric field (bias voltage) and crystal geometry, to set automatically?

Steps Per Track Not Linear Above 100V

For electrons, intervalley scattering is power-law vs. voltage (electric field)

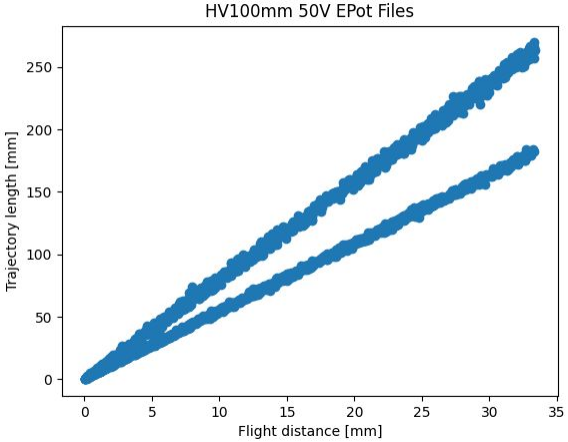
Ge: $\Gamma(IV) \sim \sqrt{(E^2 - E_0^2)}^{3.42}$
Si: $\Gamma(IV) \sim 1.5\ Hz + 1.5\ MHz\ E^4$

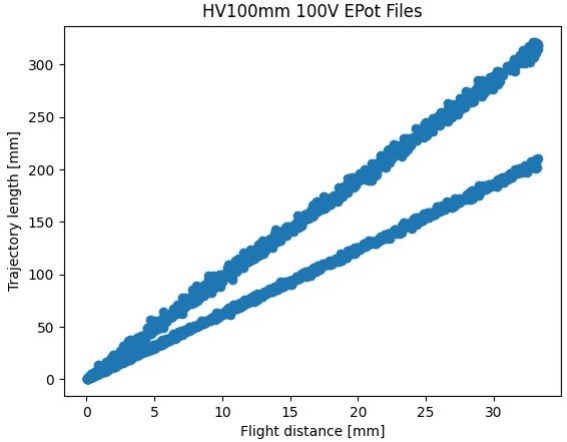
NTL scattering rate is linear in bias voltage. Above 100V (maybe sooner), intervalley rate dominates over NTL

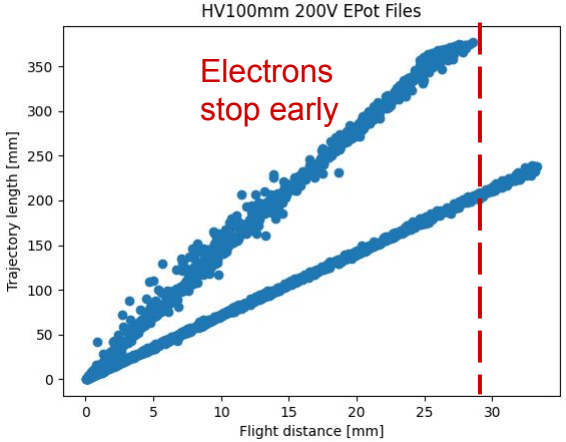
Applying linear step limit $(200{,}000 + 7{,}000*V)$ cuts off electron tracks early in their trajectory, and even earlier in flight distance, as shown on next page

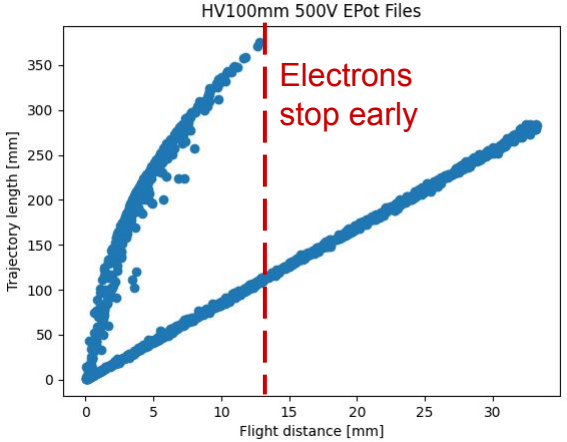
Need to find reasonable way to estimate number of steps vs. voltage for electrons, to prevent both early killing and stuck tracks

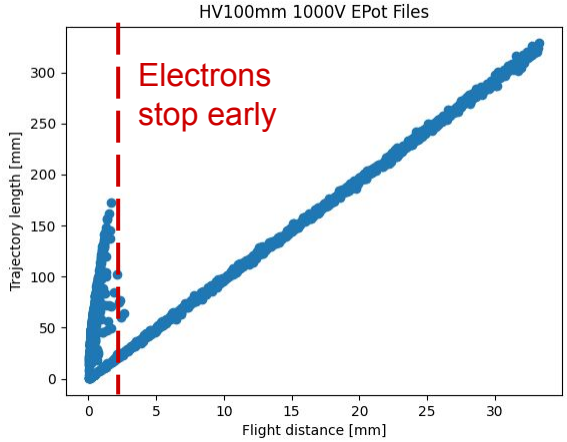
Steps per Track Not Linear Above 100V

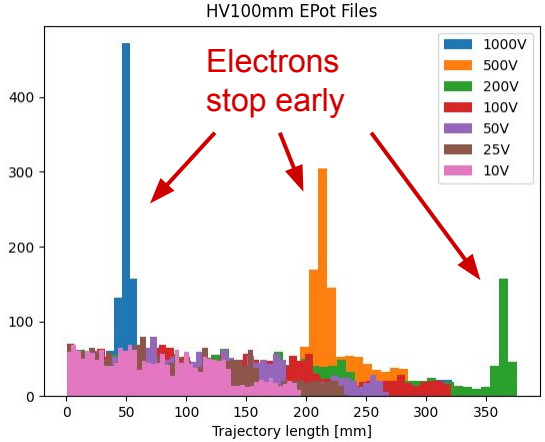












Use Field, or Bias & Length, To Compute Limit

Want to set default limit on step count in G4CMP, without every user having to do it themselves after having jobs "hang"

- No assumptions about geometry or voltage which are CDMS-centric

Option 1: Have access to track's current volume

- Get center of volume in global coordinates
- Retrieve electric field manager, get field vector at center of volume
- Get thickness of volume along field direction (e.g., Z axis)

Option 2: Ask `G4CMPTimeStepper` to do calculation of desired "step length", and apply that with volume dimensions, instead of voltage calculation

Other possibilities?

Summary and Next Steps

Electron tracks get "stuck" in odd corners of EPot files at high voltage (> 70V), spin in place for tens of millions of steps without reaching surface

Track should be stopped and killed (with recombination) if number of steps exceeds maximum possible for detector

- **`G4CMPTrackLimiter`** has been updated on branch G4CMP-358 to do this, with UI command to set step limit for job

Once released, **`G4DMCPhysics`** in SuperSim will need to set step limit to 900000

Preferably, have **`G4CMPTrackLimiter`** compute step limit when track starts, based on geometry and voltage

Backup Slides

Final Steps of Stuck Track (Before Killing Job)

	Step#	X(mm)	Y(mm)	Z(mm)	KinE(MeV)	dE(MeV)	StepLeng	TrackLeng	NextVolume	ProcName
G4WT7 >										
[...]										
G4WT7 >	99614999	-3.12	47.4	16.6	1.22e-09	0	2.18e-06	282	Zip	G4CMPInterValleyS
G4WT7 >	99615000	-3.12	47.4	16.6	1.16e-09	0	2.87e-06	282	Zip	G4CMPTimeStepper
G4WT7 >	99615001	-3.12	47.4	16.6	1.16e-09	0	7.61e-08	282	Zip	G4CMPInterValleyS
G4WT7 >	99615002	-3.12	47.4	16.6	1.17e-09	0	3.63e-07	282	Zip	G4CMPInterValleyS
G4WT7 >	99615003	-3.12	47.4	16.6	1.2e-09	0	1.86e-06	282	Zip	G4CMPTimeStepper
G4WT7 >	99615004	-3.12	47.4	16.6	1.2e-09	0	1.58e-07	282	Zip	G4CMPInterValleyS
G4WT7 >	99615005	-3.12	47.4	16.6	1.14e-09	0	1.76e-06	282	Zip	G4CMPTimeStepper

	Step#	X(mm)	Y(mm)	Z(mm)	KinE(MeV)	dE(MeV)	StepLeng	TrackLeng	NextVolume	ProcName
G4WT16 >										
[...]										
G4WT16 >	14881966	47.4	7.57	16.5	1.55e-10	0	4.54e-08	64.3	Zip	G4CMPInterValleyS
G4WT16 >	14881967	47.4	7.57	16.5	1.56e-10	0	6.64e-08	64.3	Zip	G4CMPInterValleyS
G4WT16 >	14881968	47.4	7.57	16.5	1.56e-10	0	2.55e-08	64.3	Zip	G4CMPInterValleyS
G4WT16 >	14881969	47.4	7.57	16.5	1.57e-10	0	1.39e-07	64.3	Zip	G4CMPInterValleyS
G4WT16 >	14881970	47.4	7.57	16.5	1.49e-10	0	8.26e-07	64.3	Zip	G4CMPTimeStepper
G4WT16 >	14881971	47.4	7.57	16.5	1.48e-10	0	2.43e-07	64.3	Zip	G4CMPInterValleyS