

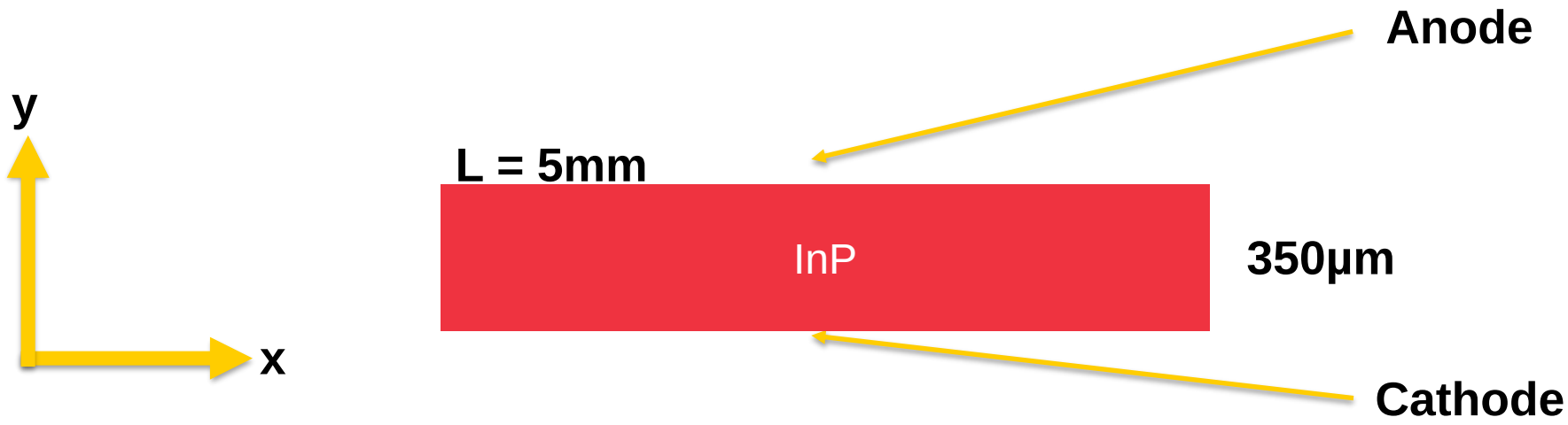


COLLEGE OF ENGINEERING
UNIVERSITY of HAWAI'I at MĀNOA

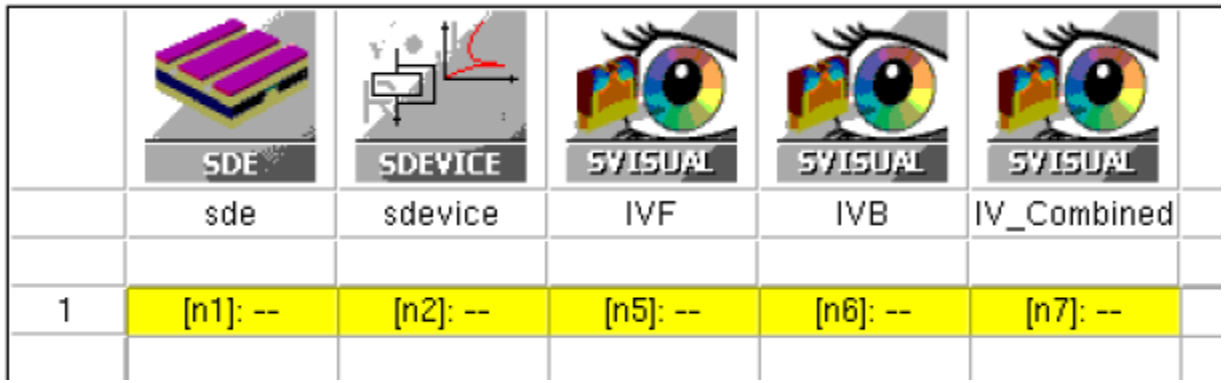
HEP InP Thin Films Update June 9, 2026

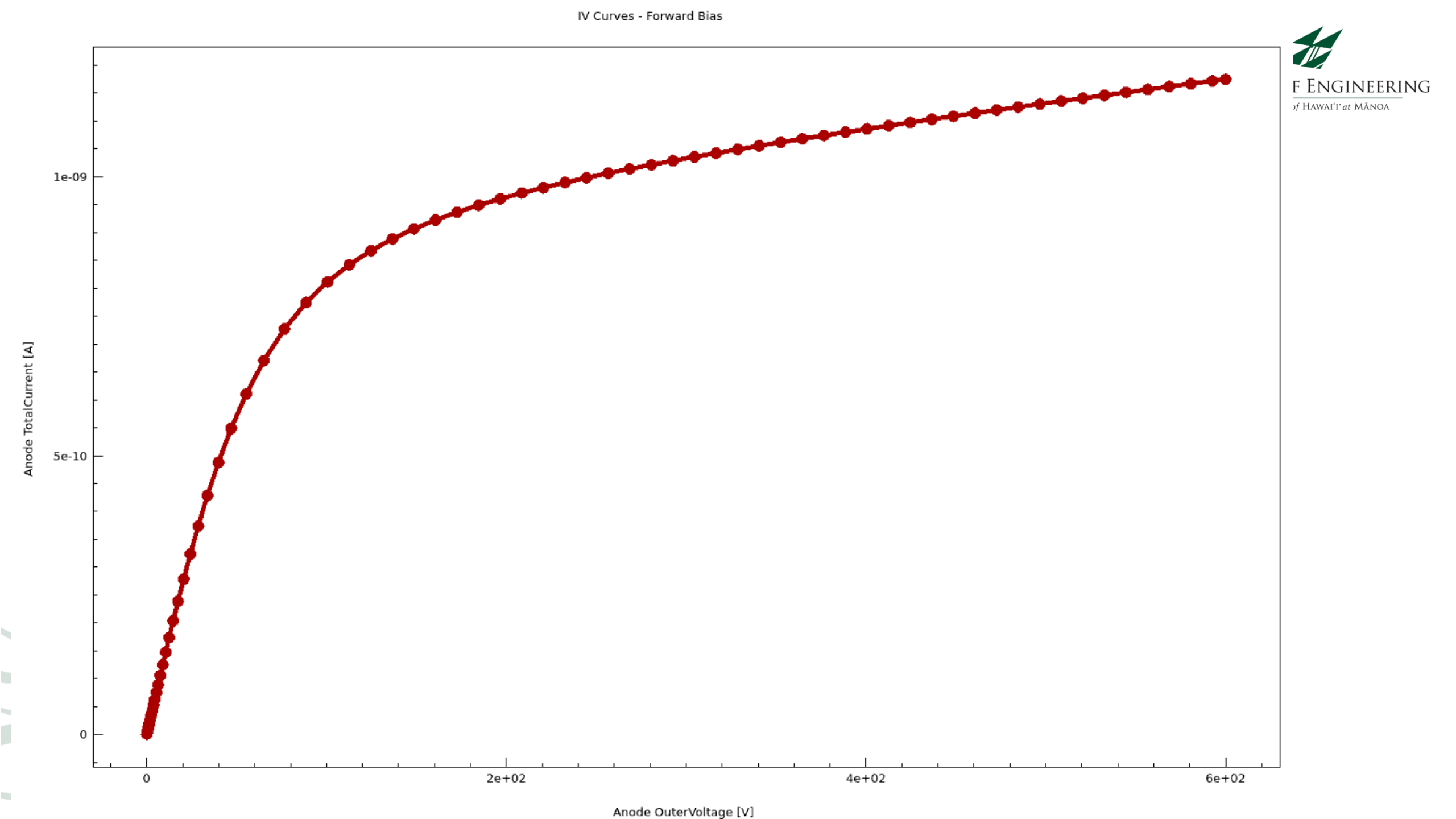
Richville Fagaragan

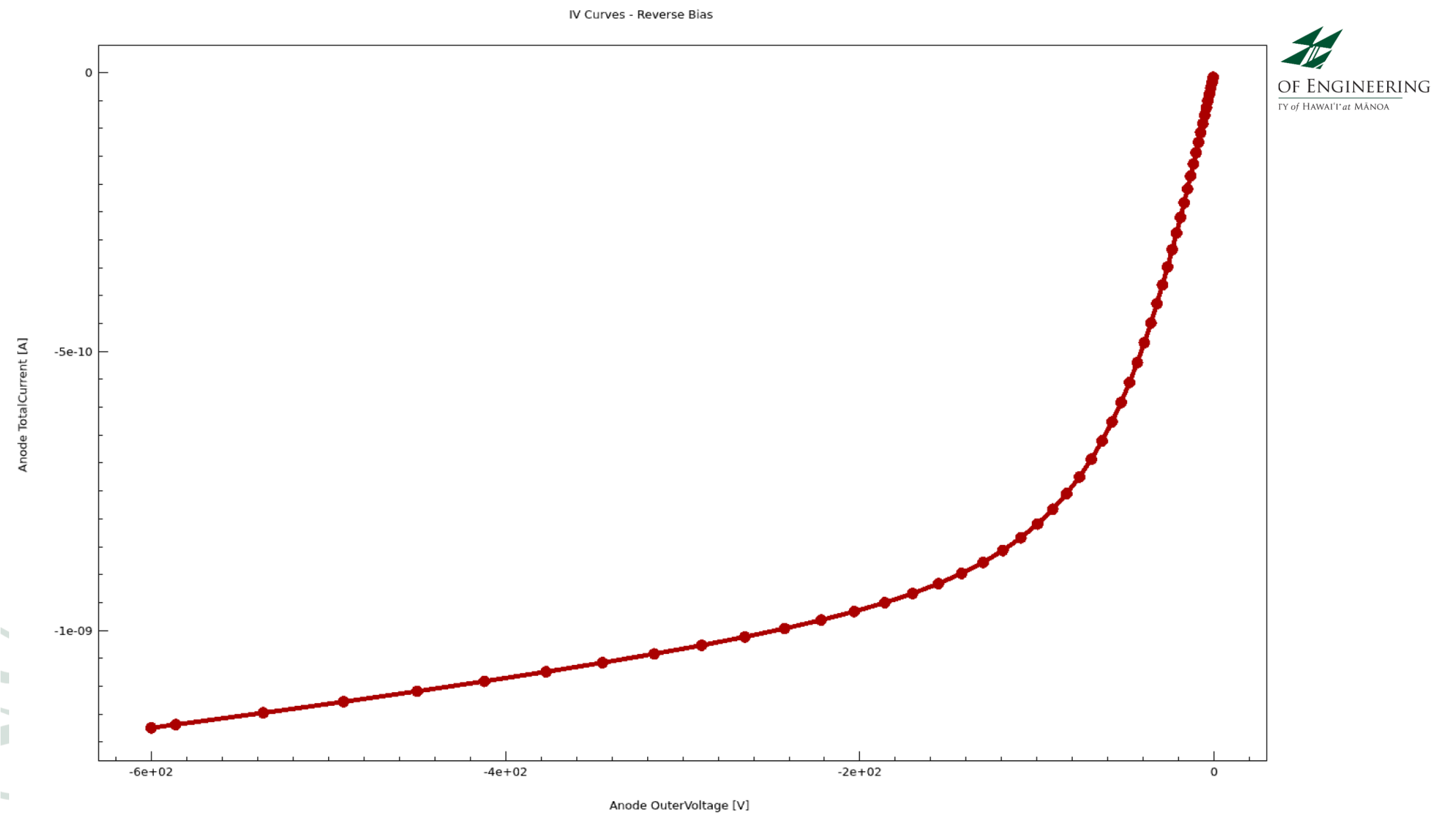
Undoped Structure for TCAD 2D Simulation



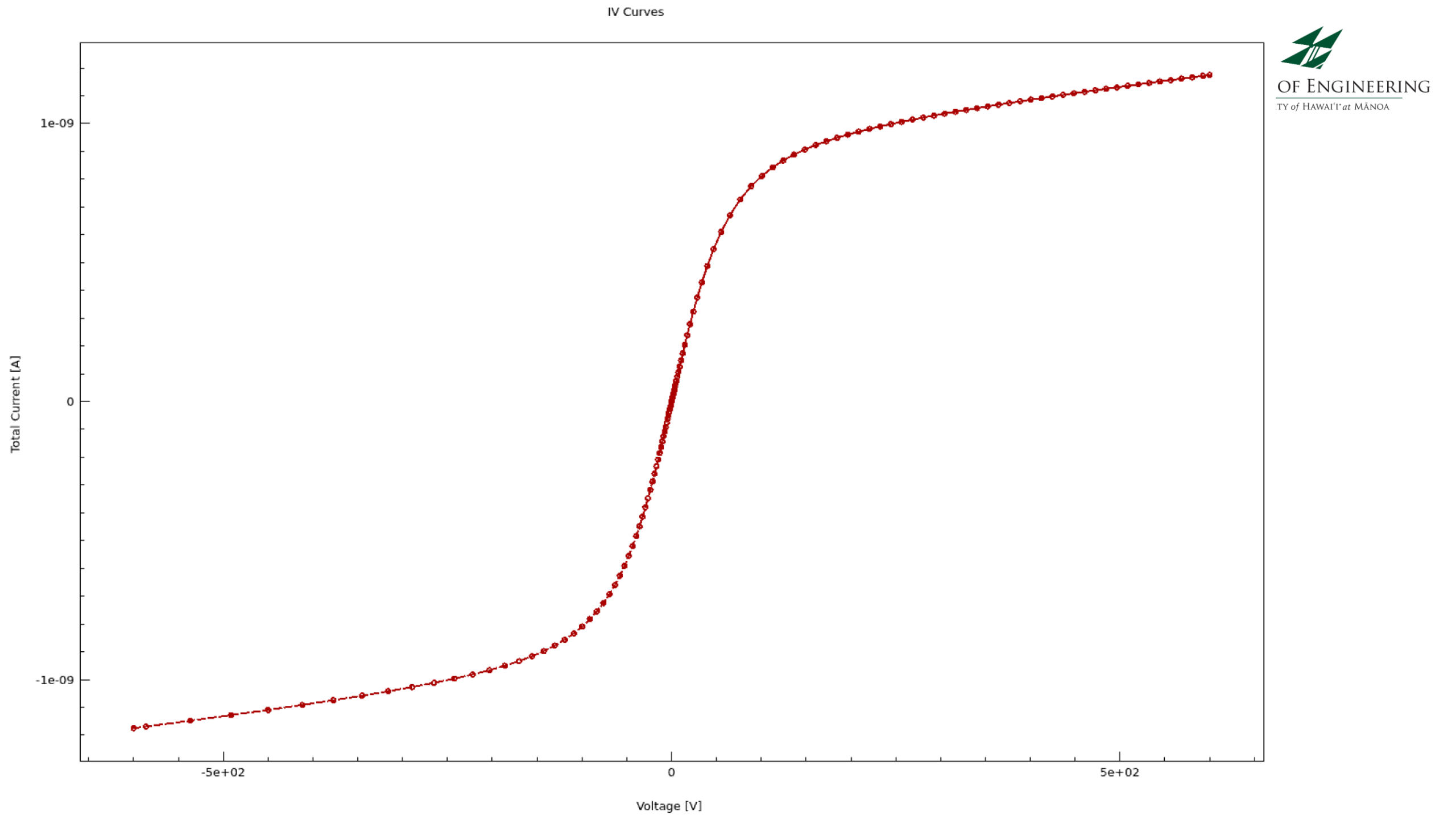
Workbench (for T=300K) + Trap Off



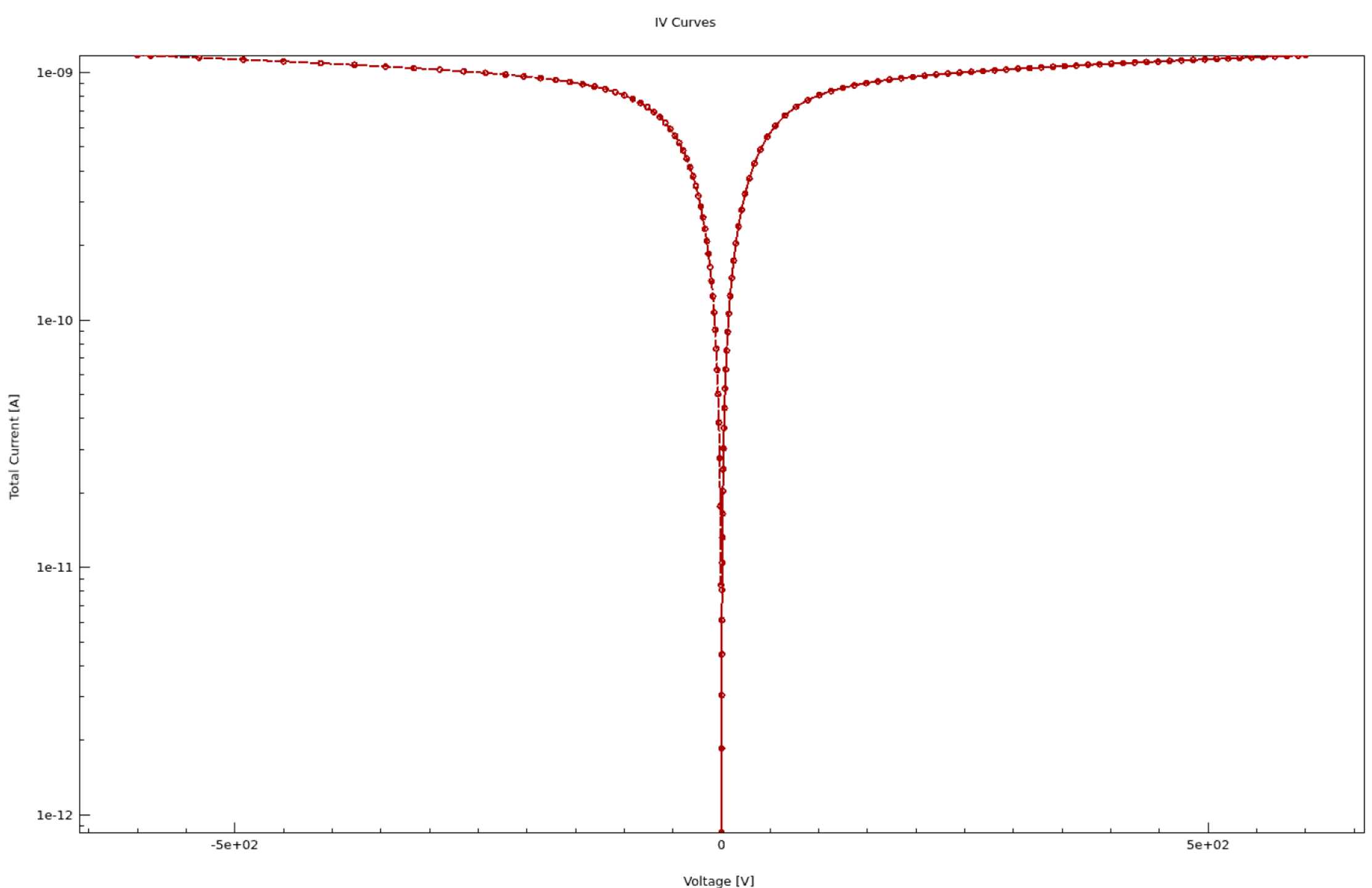

IV Curves - Forward Bias
F ENGINEERING
of HAWAI'I at MĀNOA
Anode TotalCurrent [A]
1e-09
5e-10
0
0
2e+02
4e+02
6e+02
Anode OuterVoltage [V]



OF ENGINEERING
TY of HAWAI'I at MĀNOA



OF ENGINEERING
TY of HAWAI'I at MĀNOA

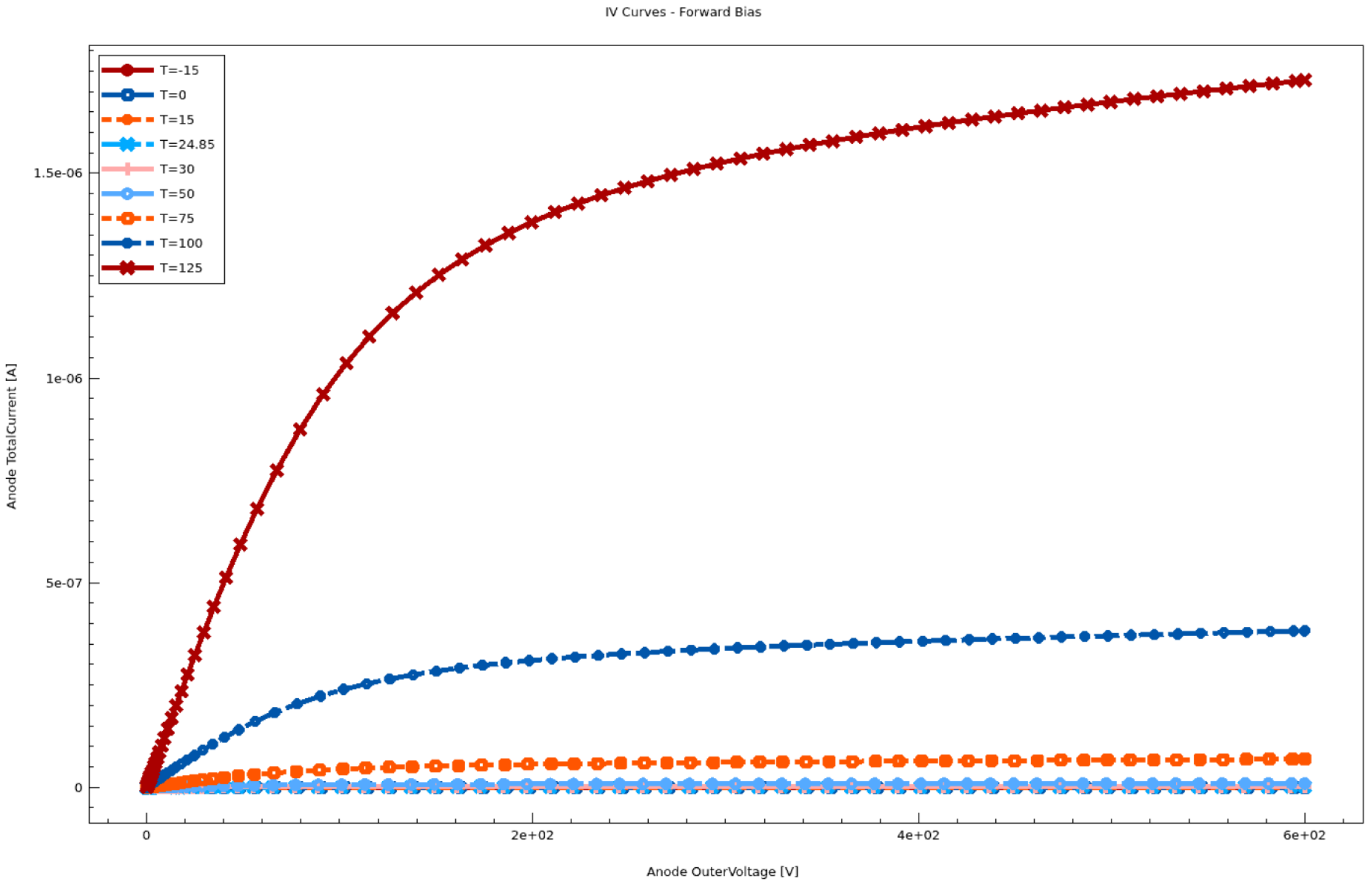


OF ENGINEERING
ITY of HAWAI'I at MĀNOA

Log

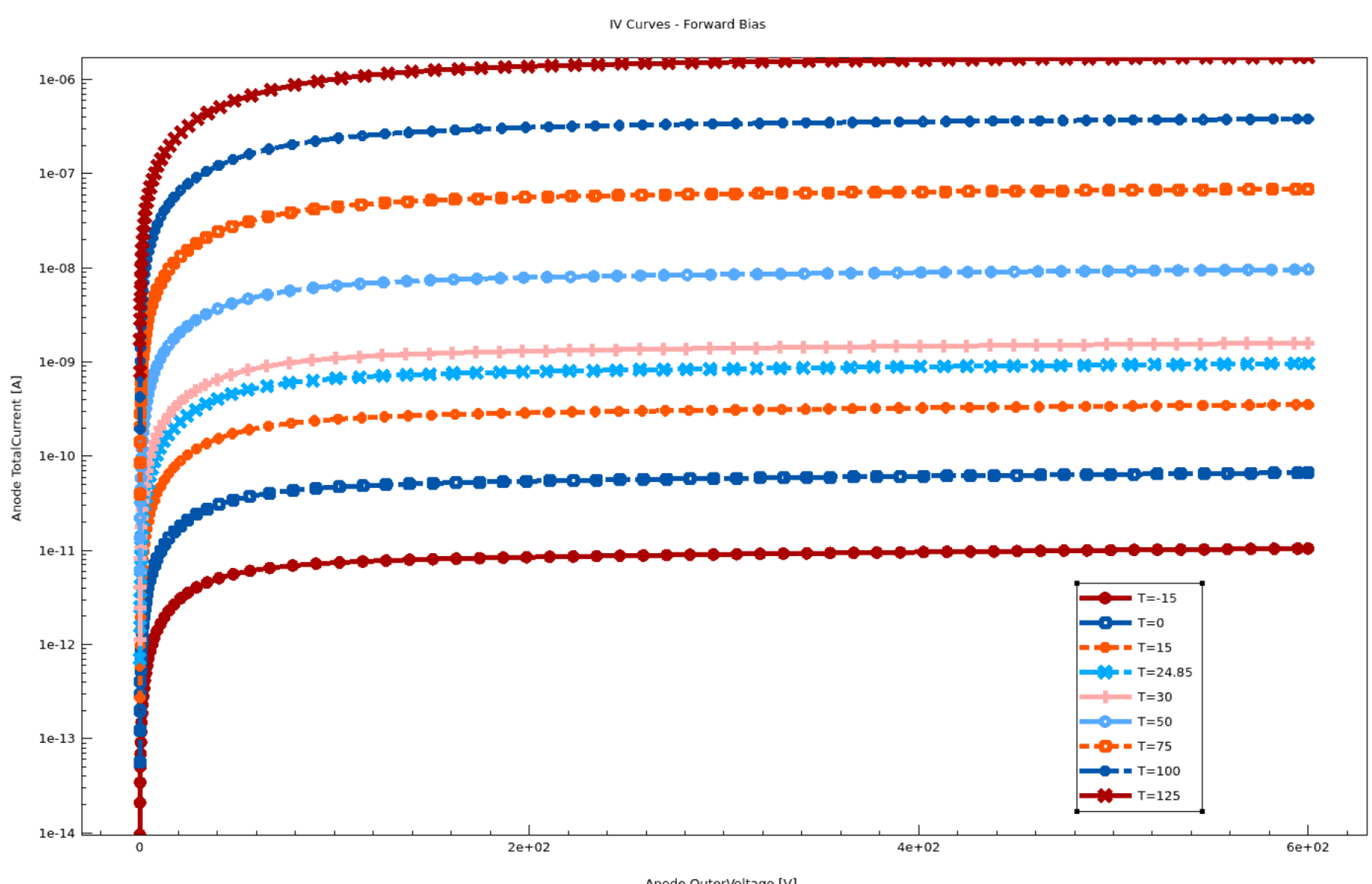
Workbench for all different temperatures

	SDE	SDEVICE		SVISUAL	SVISUAL	SVISUAL
	sde	sdevice		IVF	IVB	IV_Combined
			Temp			
1	[n1]: --	[n8]: --	[n2]: -15	[n5]: --	[n6]: --	[n7]: --
2			[n9]: 0	[n10]: --	[n11]: --	[n12]: --
3			[n13]: 15	[n14]: --	[n15]: --	[n16]: --
4			[n17]: 24.85	[n18]: --	[n19]: --	[n20]: --
5			[n21]: 30	[n22]: --	[n23]: --	[n24]: --
6			[n25]: 50	[n26]: --	[n27]: --	[n28]: --
7			[n29]: 75	[n30]: --	[n31]: --	[n32]: --
8			[n33]: 100	[n34]: --	[n35]: --	[n36]: --
9			[n37]: 125	[n38]: --	[n39]: --	[n40]: --



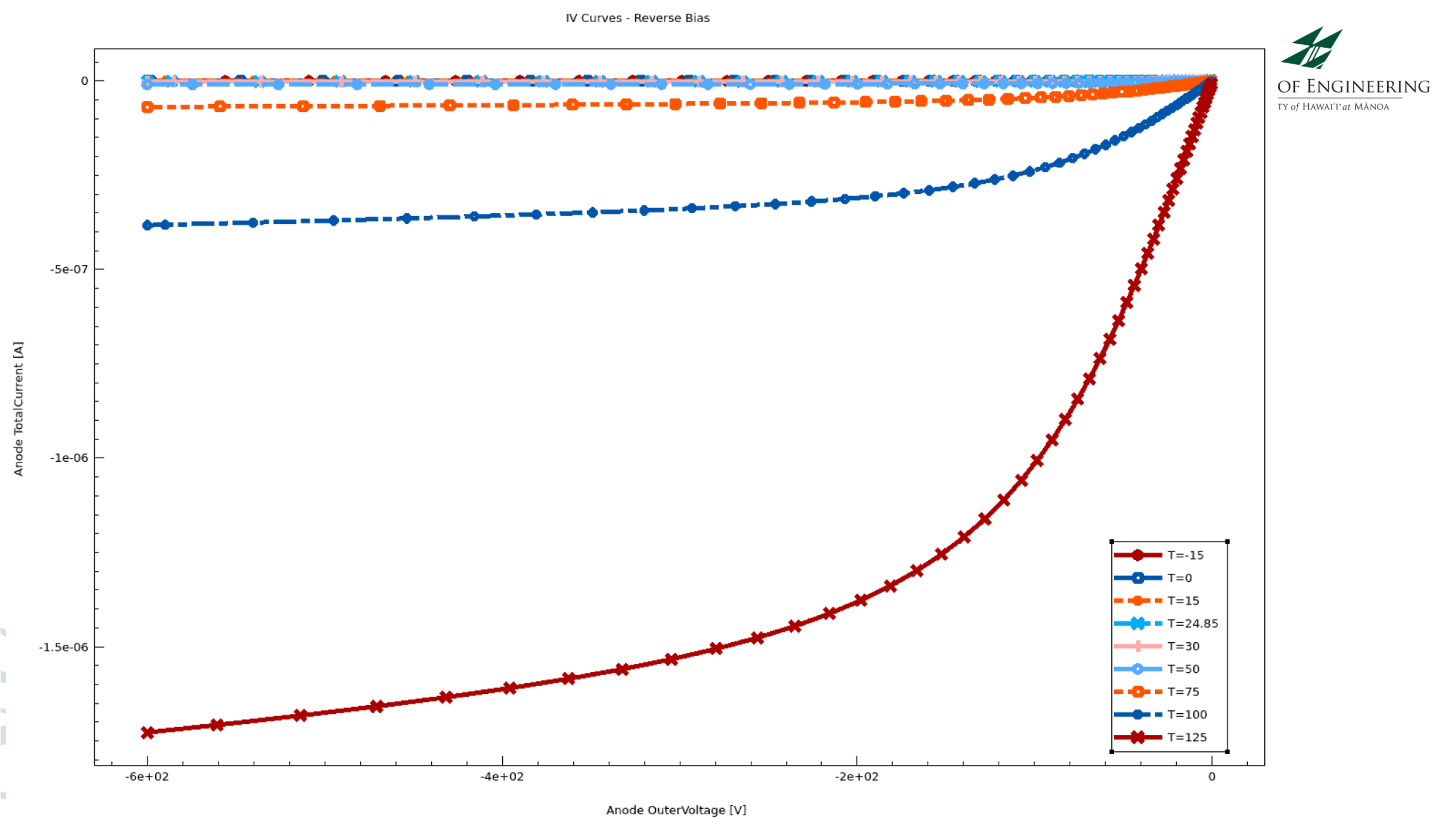
F ENGINEERING

of HAWAI'I at MĀNOA



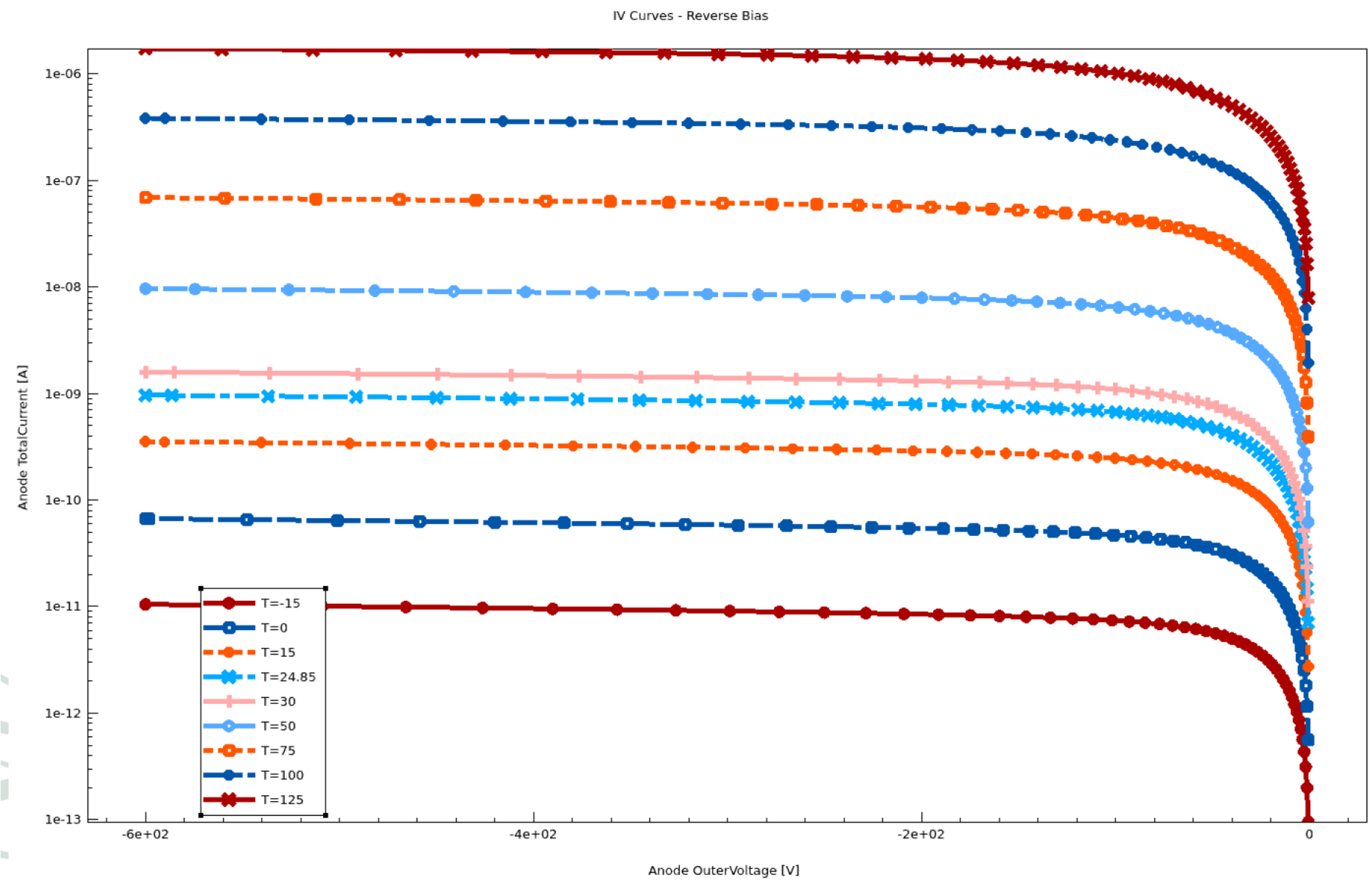
OF ENGINEERING
TY of HAWAI'I at MĀNOA

Log



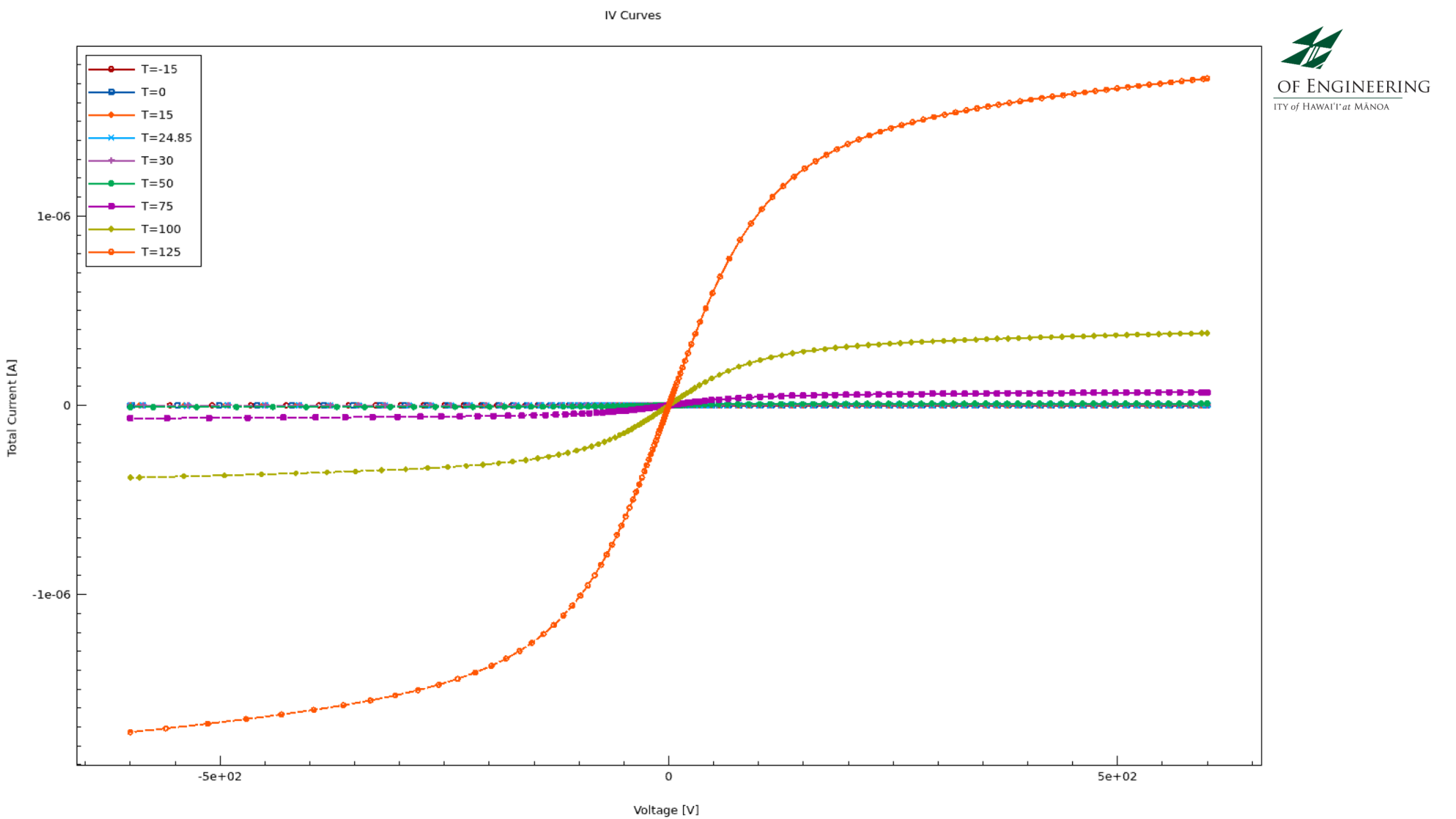
OF ENGINEERING

TY of HAWAI'I at MĀNOA

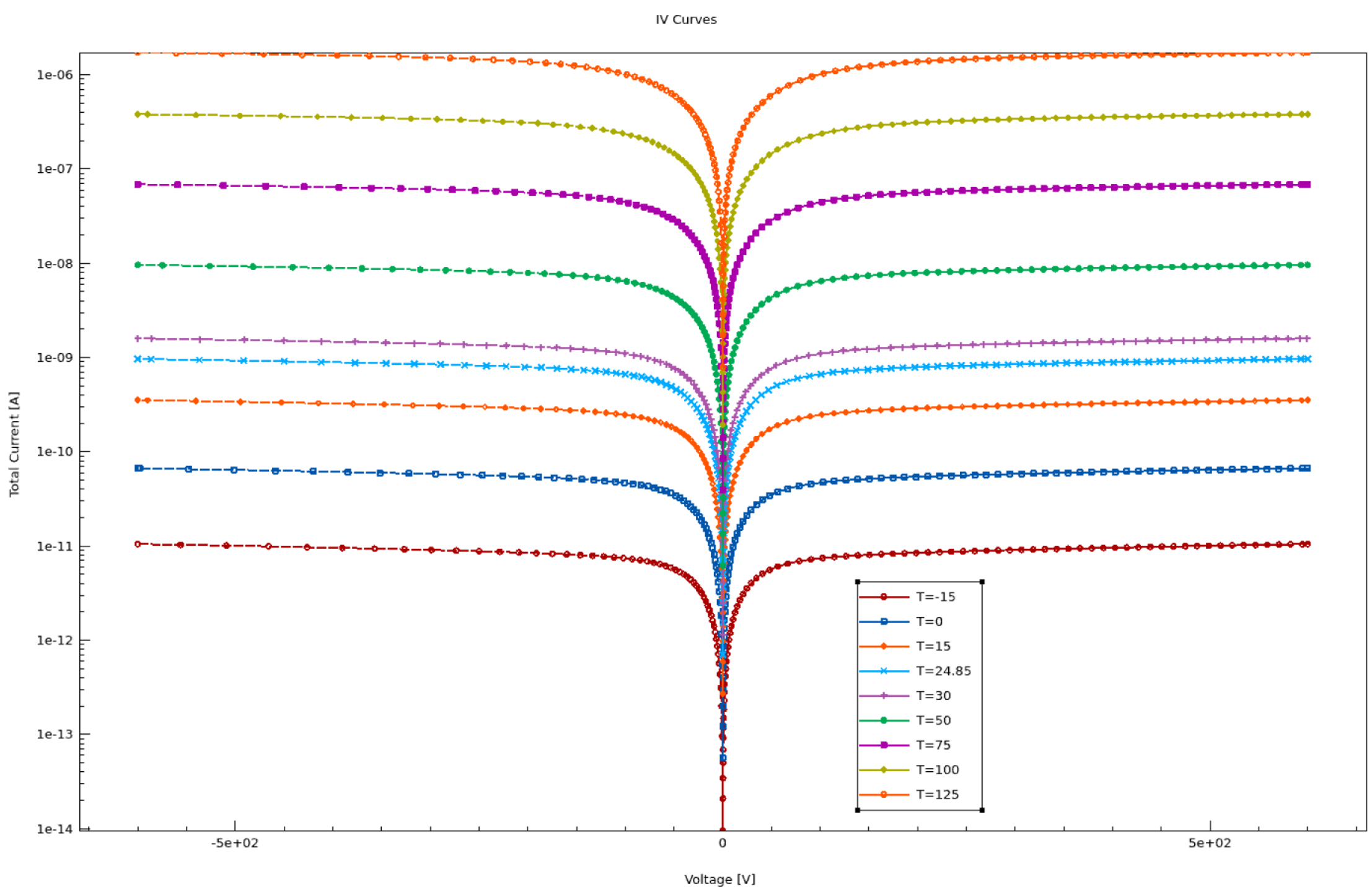


OF ENGINEERING
of HAWAI'I at MĀNOA

Log

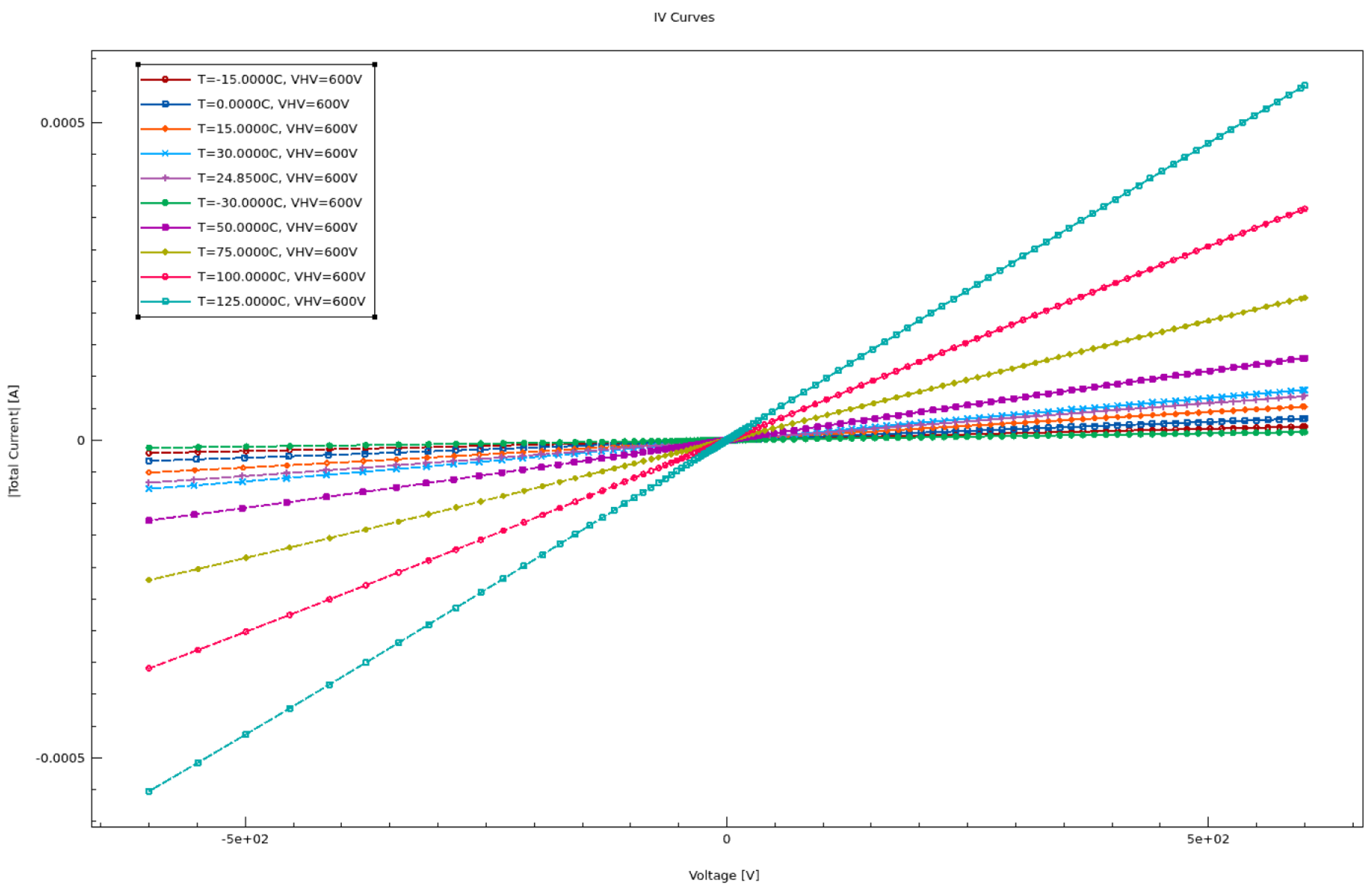


OF ENGINEERING
ITY of HAWAI'I at MĀNOA



E OF ENGINEERING
SITY of HAWAI'I at MĀNOA

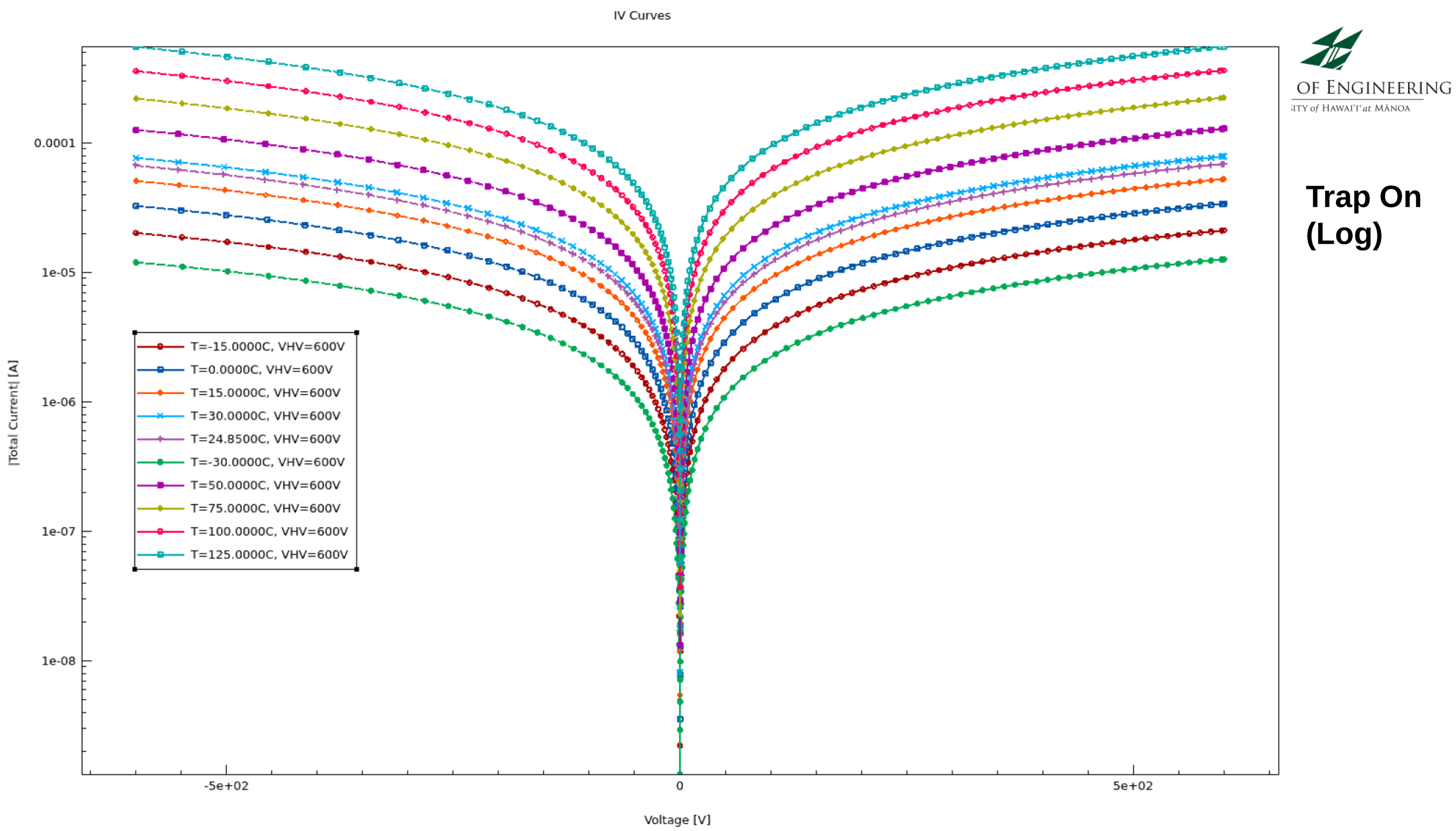
Log



OF ENGINEERING

TY of HAWAI'I at MĀNOA

Trap On



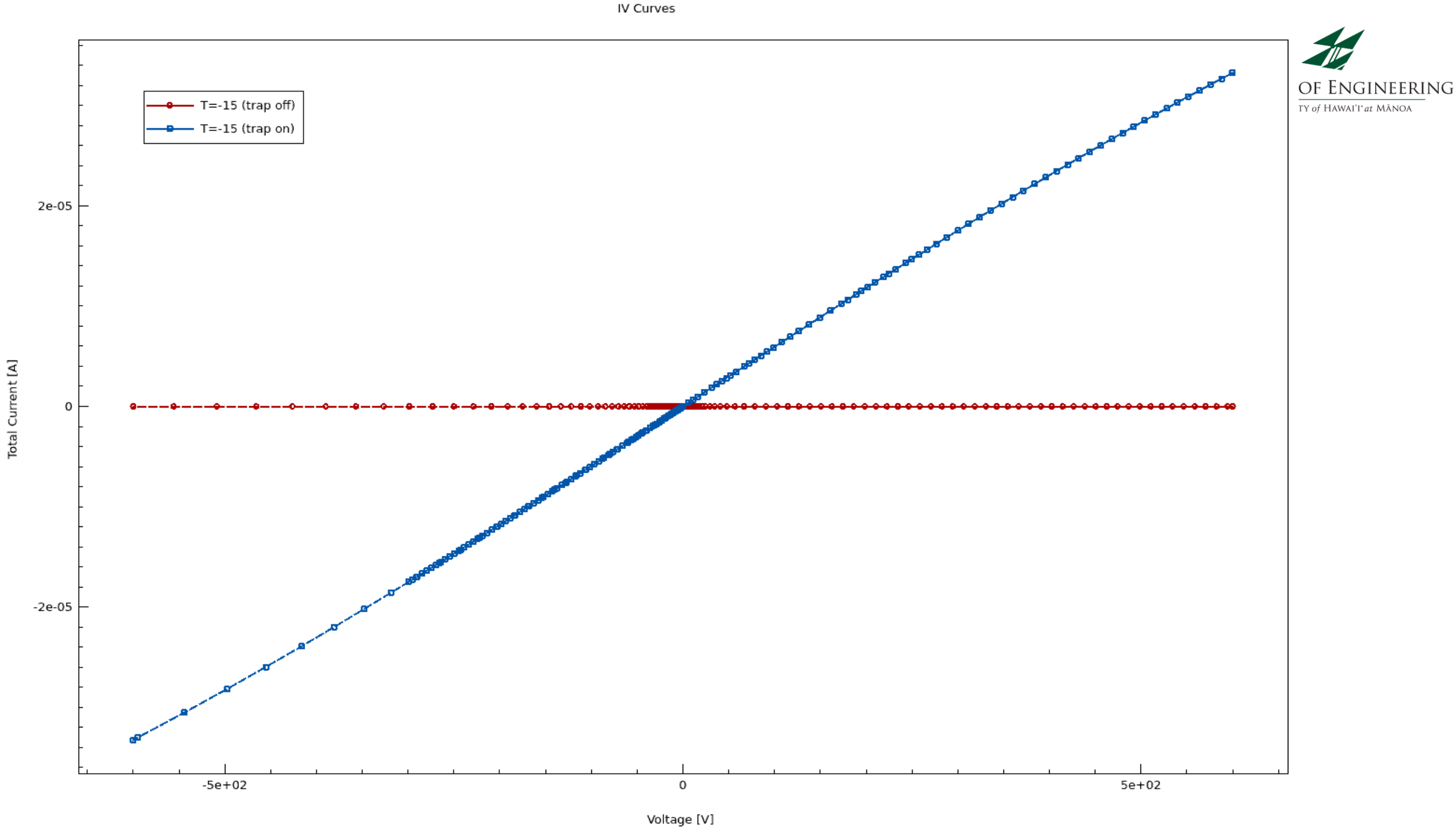
OF ENGINEERING

ITY of HAWAI'I at MĀNOA

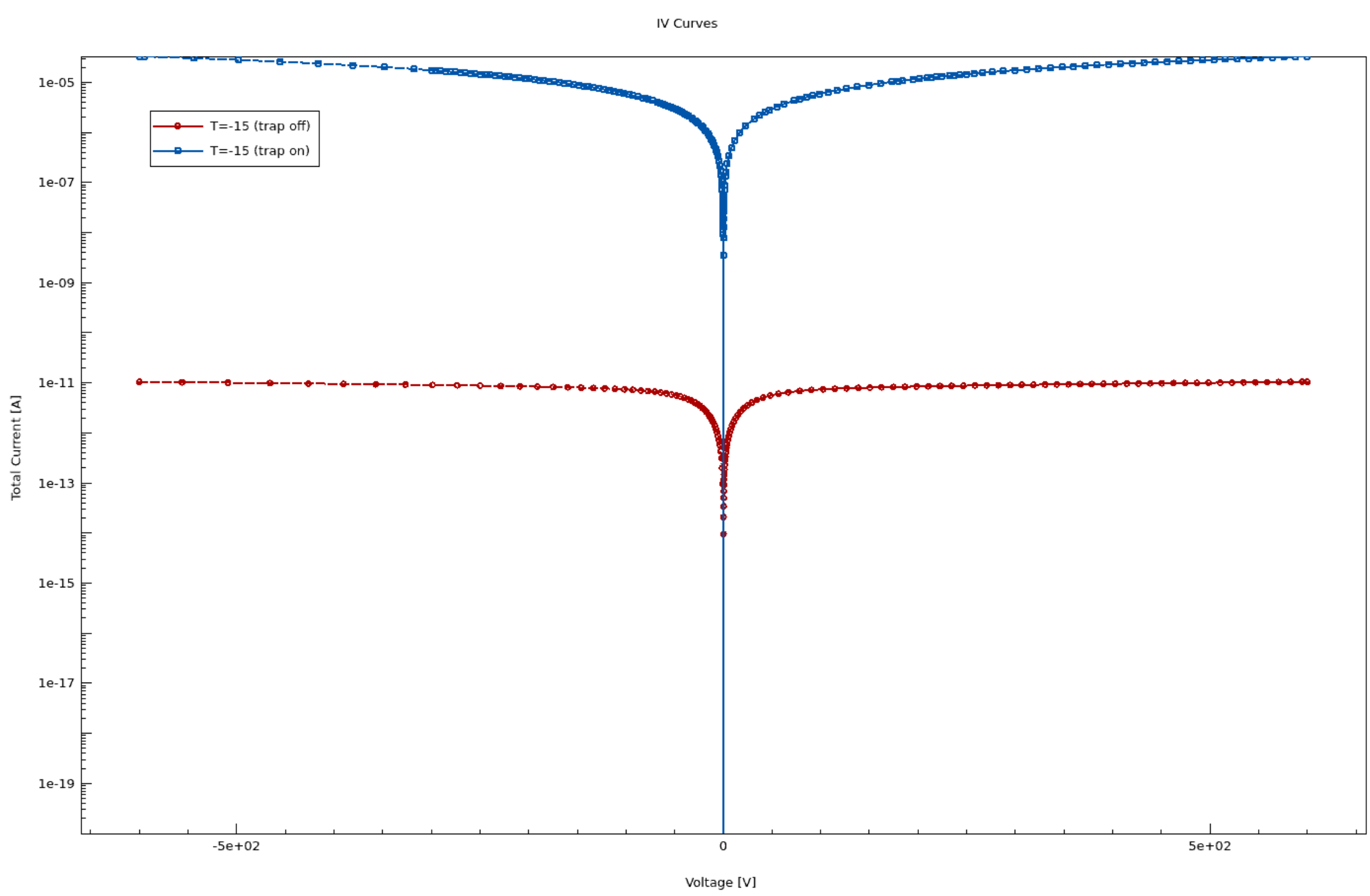
Trap On (Log)

COLLEGE OF ENGINEERING
UNIVERSITY of HAWAI'I at MĀNOA

In Progress Work



OF ENGINEERING
TY of HAWAI'I at MĀNOA



E OF ENGINEERING
ERSITY of HAWAI'I at MĀNOA

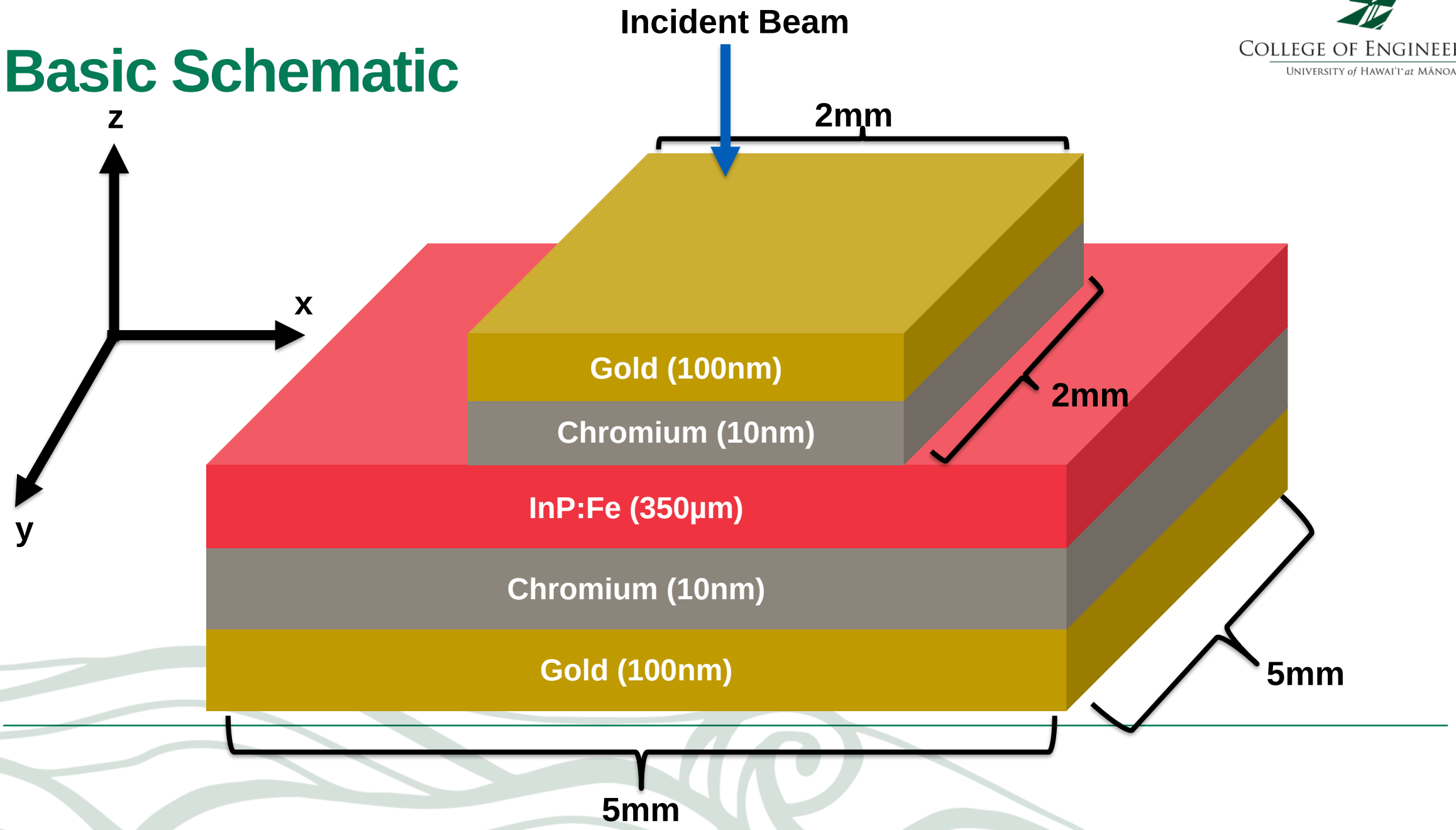
Log


UNIVERSITY OF HAWAI'I
MĀLAMALAMA
1907
UA MAU KE EA O KA 'ĀINA I KA PONO

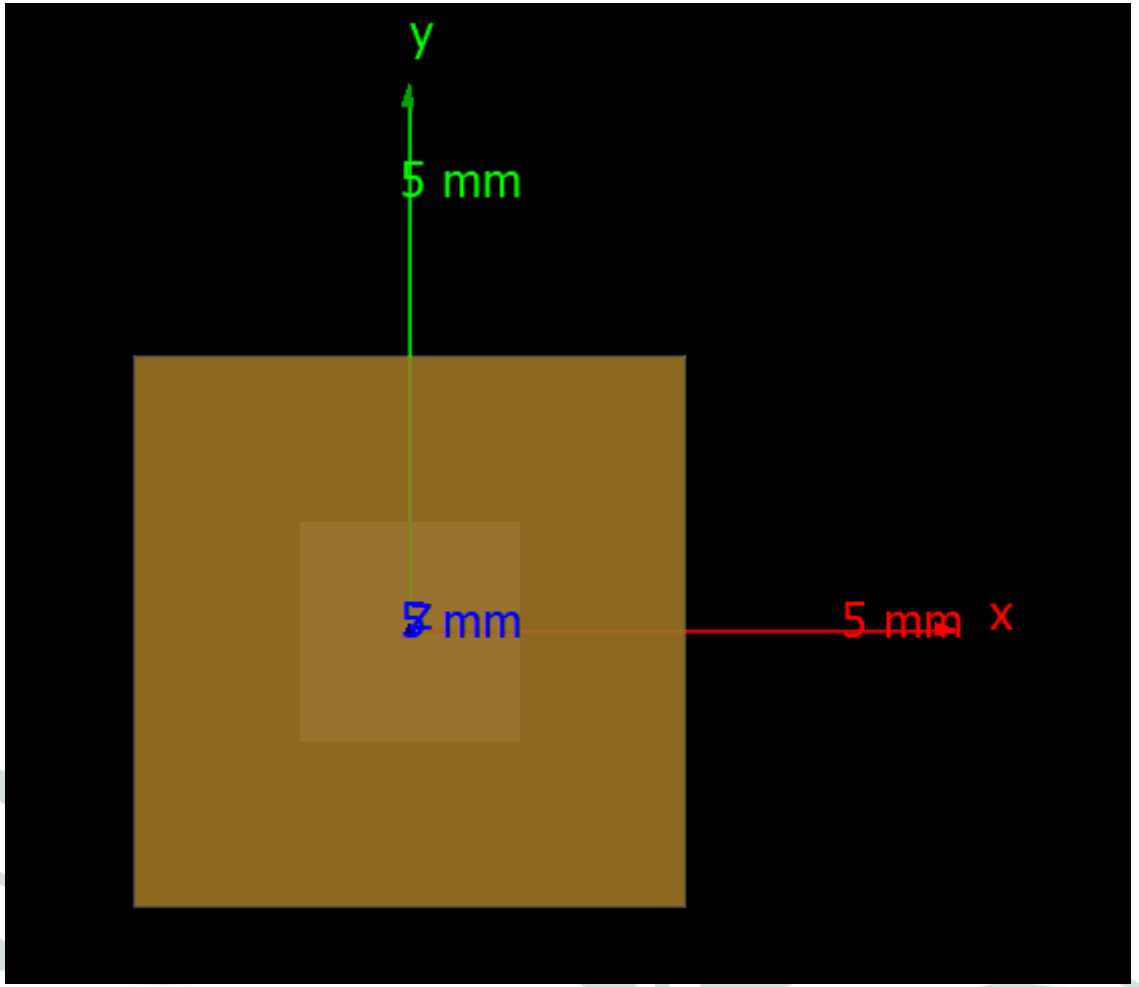

COLLEGE OF ENGINEERING
UNIVERSITY of HAWAI'I at MĀNOA

Geant4

Basic Schematic



Geant4



Beam Parameters

COLLEGE OF ENGINEERING
UNIVERSITY of HAWAI'I at MĀNOA

Table 5: Beam Line Properties

Facility	Beam Energy [keV]	Beam Diameter [μm]	Operational Temperature [C]	Beam Flux [ph/s]	Sample support
CLS	15	40	2.5	1.25×10^{11}	Thick PCB
DLS	15	2	-15	1.18×10^{8}	Thin PCB with enhanced thermal control

```cpp
#ifndef GENERATOR_HH
#define GENERATOR_HH

#include "G4VUserPrimaryGeneratorAction.hh"
#include "G4GeneralParticleSource.hh"
#include "G4SystemOfUnits.hh"
#include "G4ParticleTable.hh"
#include "G4Event.hh"
#include "G4ThreeVector.hh"

class MyPrimaryGenerator : public G4VUserPrimaryGeneratorAction
{
public:
    MyPrimaryGenerator();
    virtual ~MyPrimaryGenerator();
    virtual void GeneratePrimaries(G4Event* event);

    // Getter for incident energy
    // so it always reflects whatever the macro set
    G4double GetParticleEnergy() const {
        return fParticleGun->GetCurrentSource()->GetEneDist()->GetMonoEnergy();
    }

    // Getter for particle name
    G4String GetParticleType() const {
        G4ParticleDefinition* pd =
            fParticleGun->GetCurrentSource()->GetParticleDefinition();
        return pd ? pd->GetParticleName() : "unknown";
    }

    // Getter for position (centre of source)
    G4ThreeVector GetParticlePosition() const {
        return fParticleGun->GetCurrentSource()->GetPosDist()->GetCentreCoords();
    }

    // Getter for direction
    G4ThreeVector GetParticleDirection() const {
        return fParticleGun->GetCurrentSource()
                    ->GetAngDist()->GetDirection();
    }

private:
    G4GeneralParticleSource* fParticleGun;
};

#endif
```

Used Geant4 GeneralParticleSource

COLLEGE OF ENGINEERING
UNIVERSITY of HAWAI'I at MĀNOA

The `G4GeneralParticleSource` (GPS) is part of the GEANT4 toolkit for Monte-Carlo, high-energy particle transport. Specifically, it allows the specifications of the spectral, spatial and angular distribution of the primary source particles. An overview of the GPS class structure is presented here. Configuration covers the configuration of GPS for a user application, and Macro Commands describes the macro command interface. Example Macro File gives an example input file to guide the first time user.

GPS allows the user to control the following characteristics of primary particles:

- Spatial sampling: on simple 2D or 3D surfaces such as discs, spheres, and boxes.
- Angular distribution: unidirectional, isotropic, cosine-law, beam or arbitrary (user defined).
- Spectrum: linear, exponential, power-law, Gaussian, blackbody, or piece-wise fits to data.
- Multiple sources: multiple independent sources can be used in the same run.

CLS Beam Macro

```
/gps/particle gamma
/gps/energy 15 keV

/gps/pos/type Beam
/gps/pos/centre 0 0 -176.11 um

# CLS: 40 um diameter, Gaussian sigma = diameter/2 =
20 um
/gps/pos/sigma_x 20 um
/gps/pos/sigma_y 20 um

# Perfectly collimated
/gps/ang/type beam1d
/gps/ang/sigma_r 0 rad

/gps/direction 0 0 1

/run/initialize
/run/beamOn 10000
```

DLS Beam Macro

```
/gps/particle gamma
/gps/energy 15 keV

/gps/pos/type Beam
/gps/pos/centre 0 0 -176.11 um

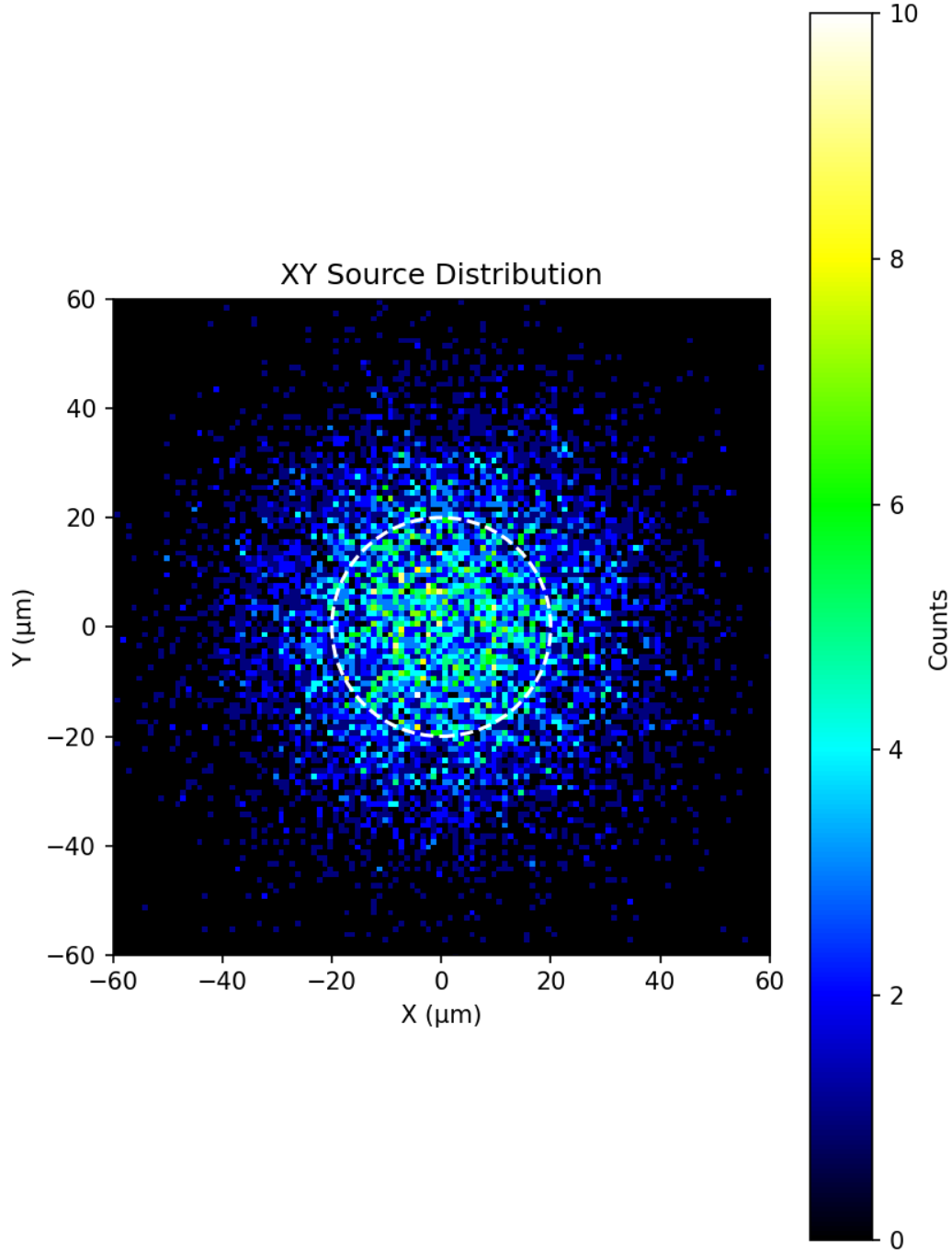
# DLS: 2 um diameter, Gaussian sigma = diameter/2 = 1
um
/gps/pos/sigma_x 1 um
/gps/pos/sigma_y 1 um

# Perfectly collimated
/gps/ang/type beam1d
/gps/ang/sigma_r 0 rad

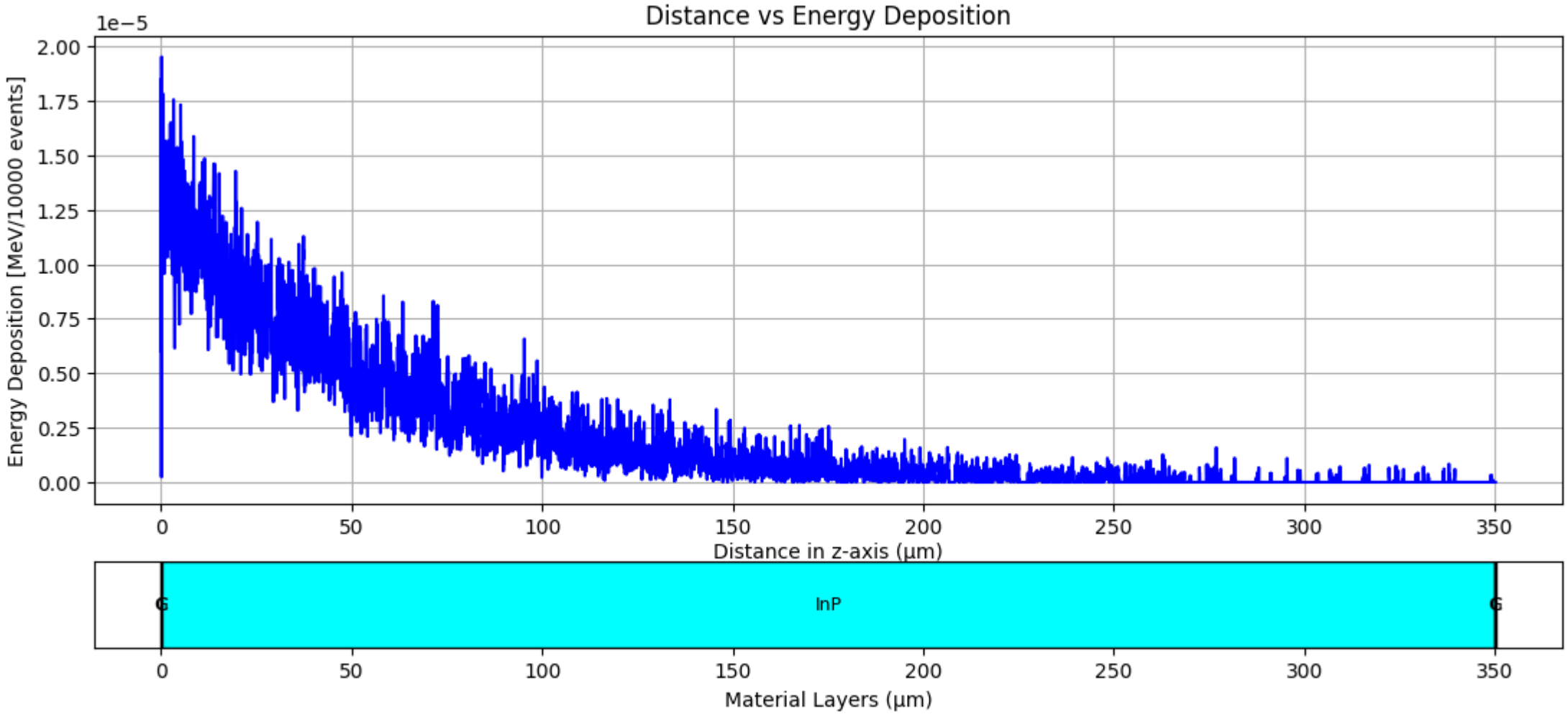
/gps/direction 0 0 1

/run/initialize
/run/beamOn 10000
```

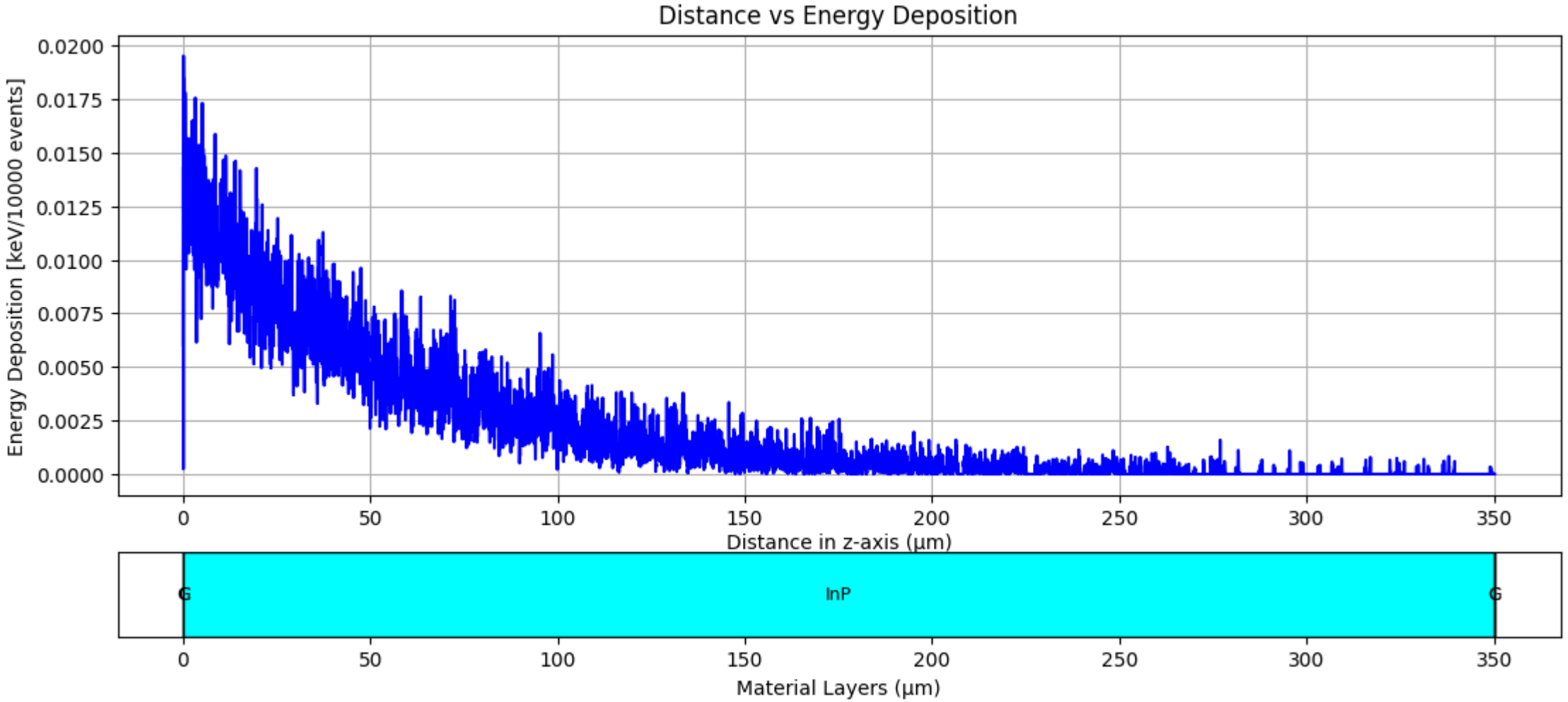
CLS 40µm Diameter Beam


XY Source Distribution
Y (µm)
X (µm)
60
40
20
0
−20
−40
−60
Counts
10
8
6
4
2
0

Energy Deposition (in MeV/10000 events)

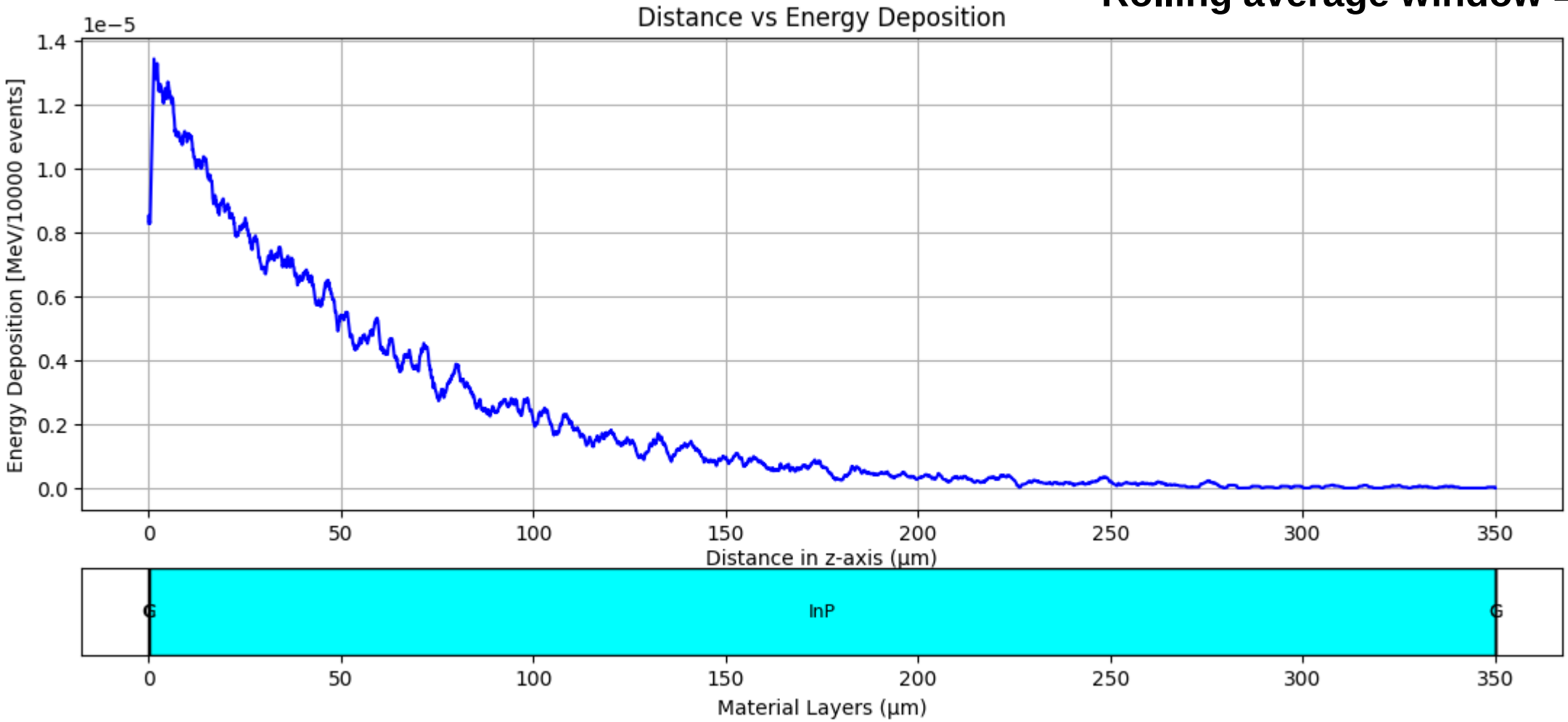


Energy Deposition (in keV/10000 events)



Energy Deposition (in MeV/10000 events)

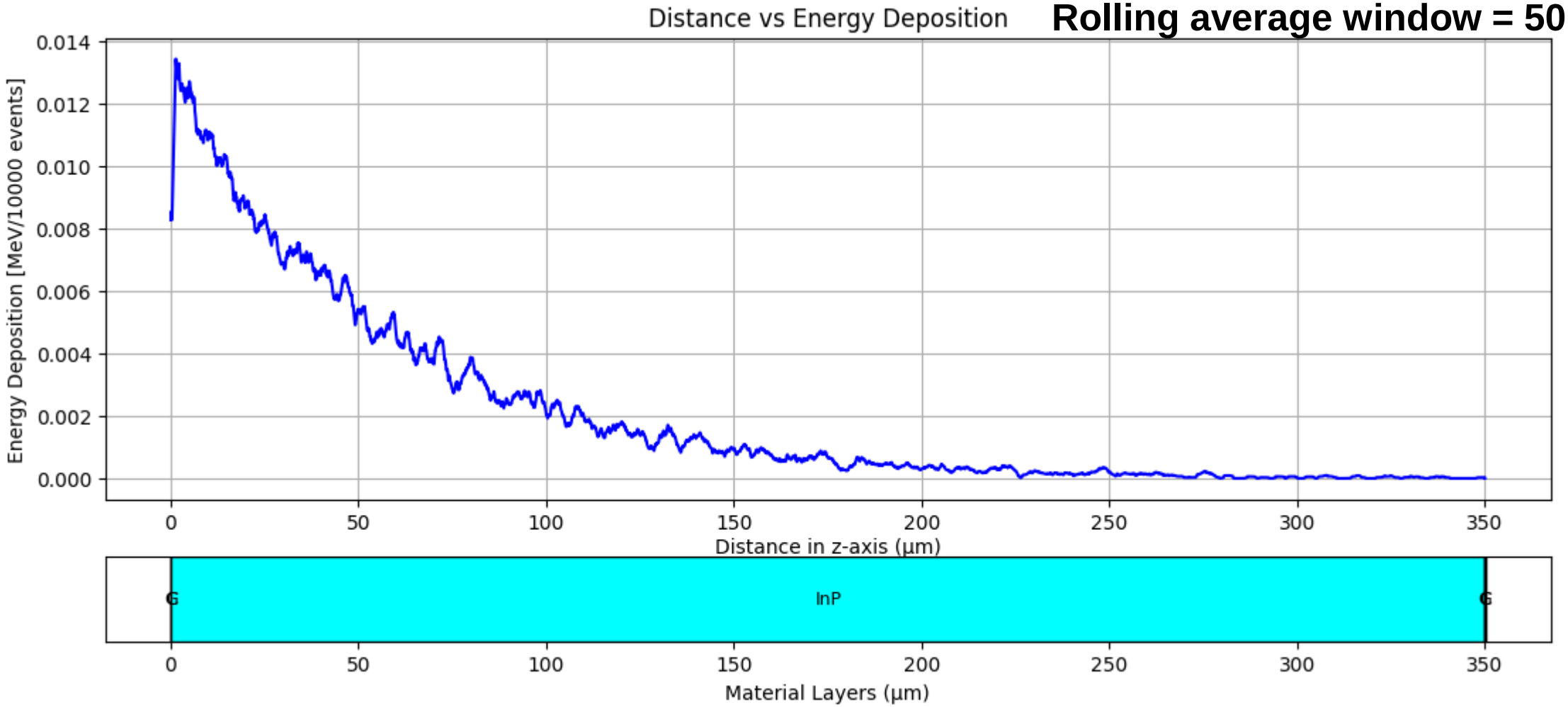
Rolling average window = 50




```
df2['edep_smooth'] = y2.rolling(window=50,center=True).mean()
y2_smooth = df2['edep_smooth']
```

Energy Deposition (in keV/10000 events)

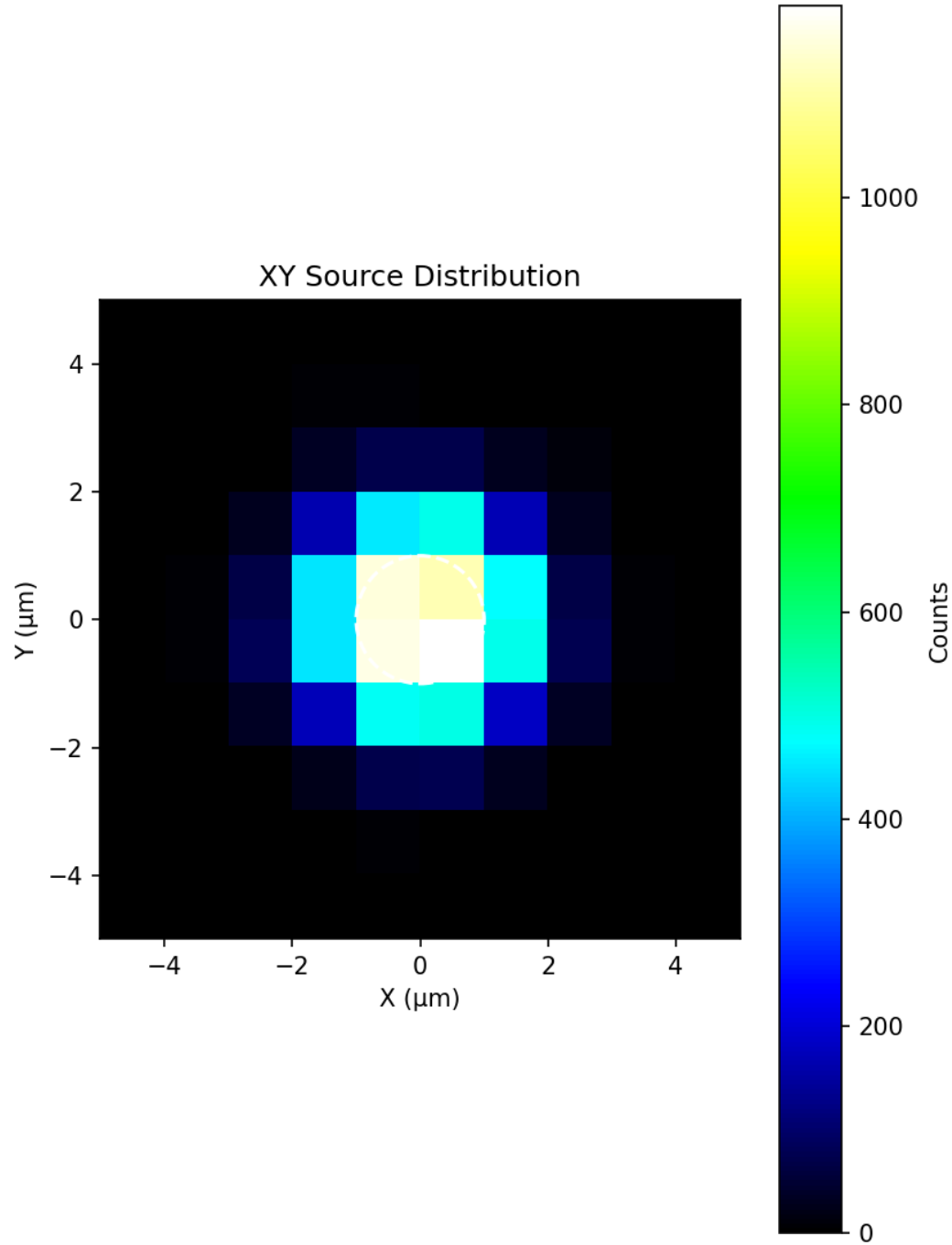
Rolling average window = 50




```
df2['edep_smooth'] = y2.rolling(window=50,center=True).mean()
y2_smooth = df2['edep_smooth']
```

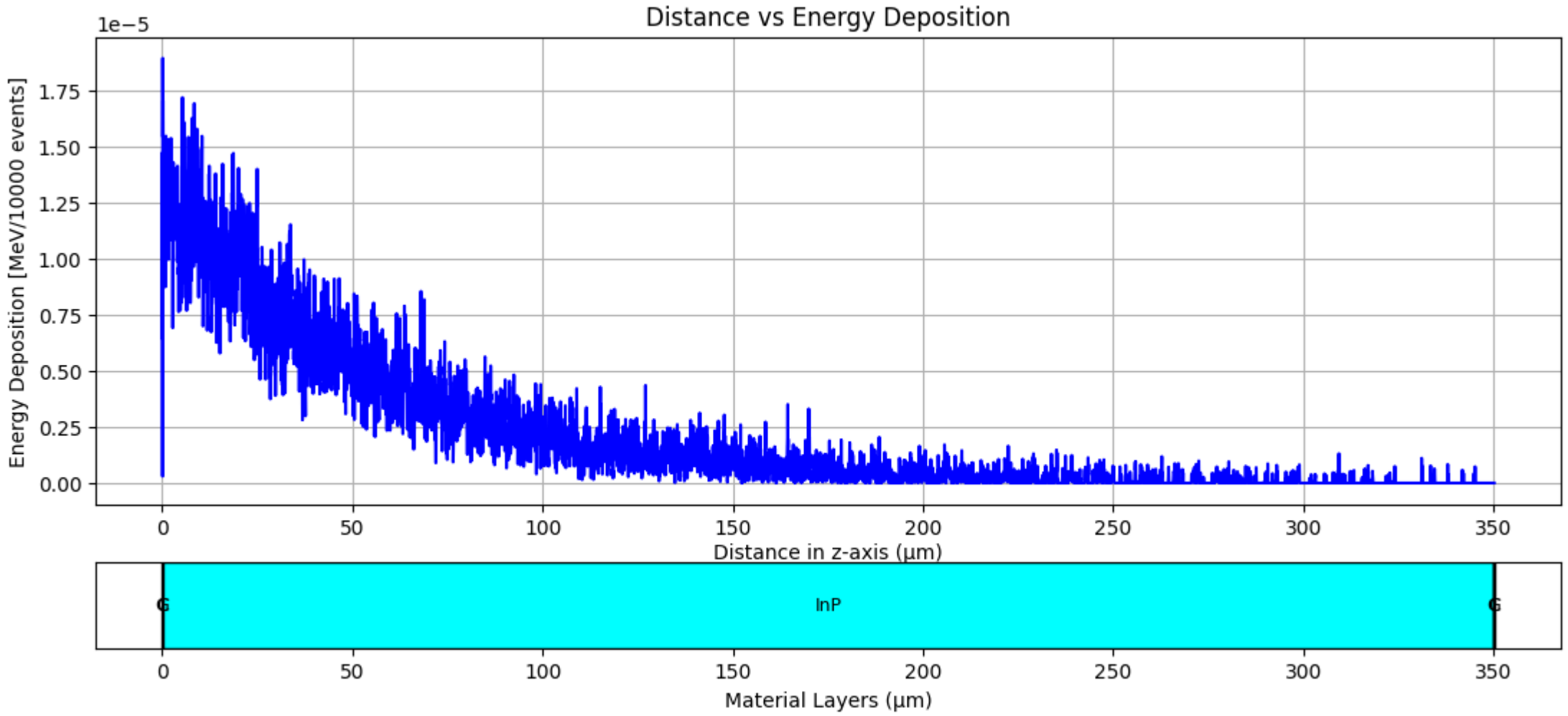
COLLEGE OF ENGINEERING
UNIVERSITY of HAWAI'I at MĀNOA

DLS 2µm Beam

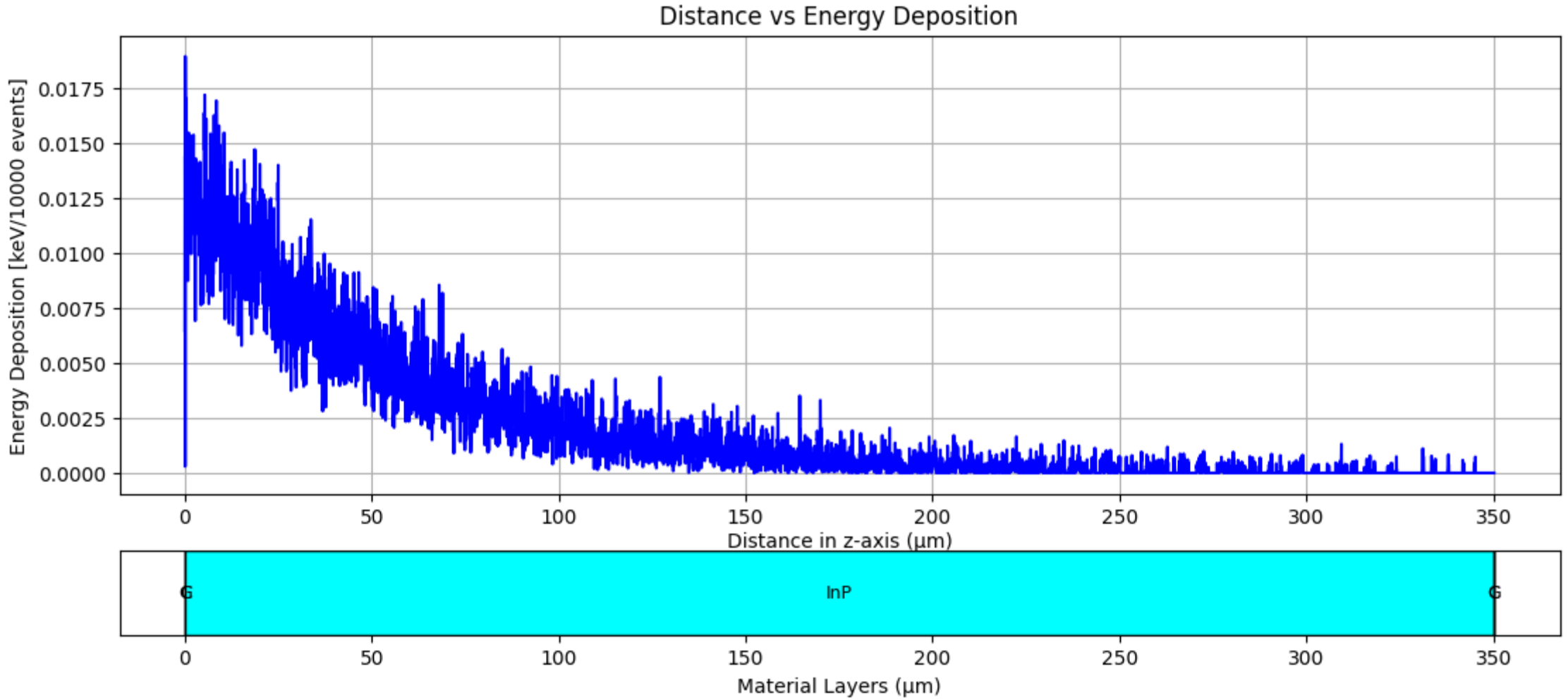

XY Source Distribution
Y (µm)
X (µm)
4
2
0
−2
−4
−4
−2
0
2
4
Counts
1000
800
600
400
200
0

COLLEGE OF ENGINEERING
UNIVERSITY of HAWAI'I at MĀNOA

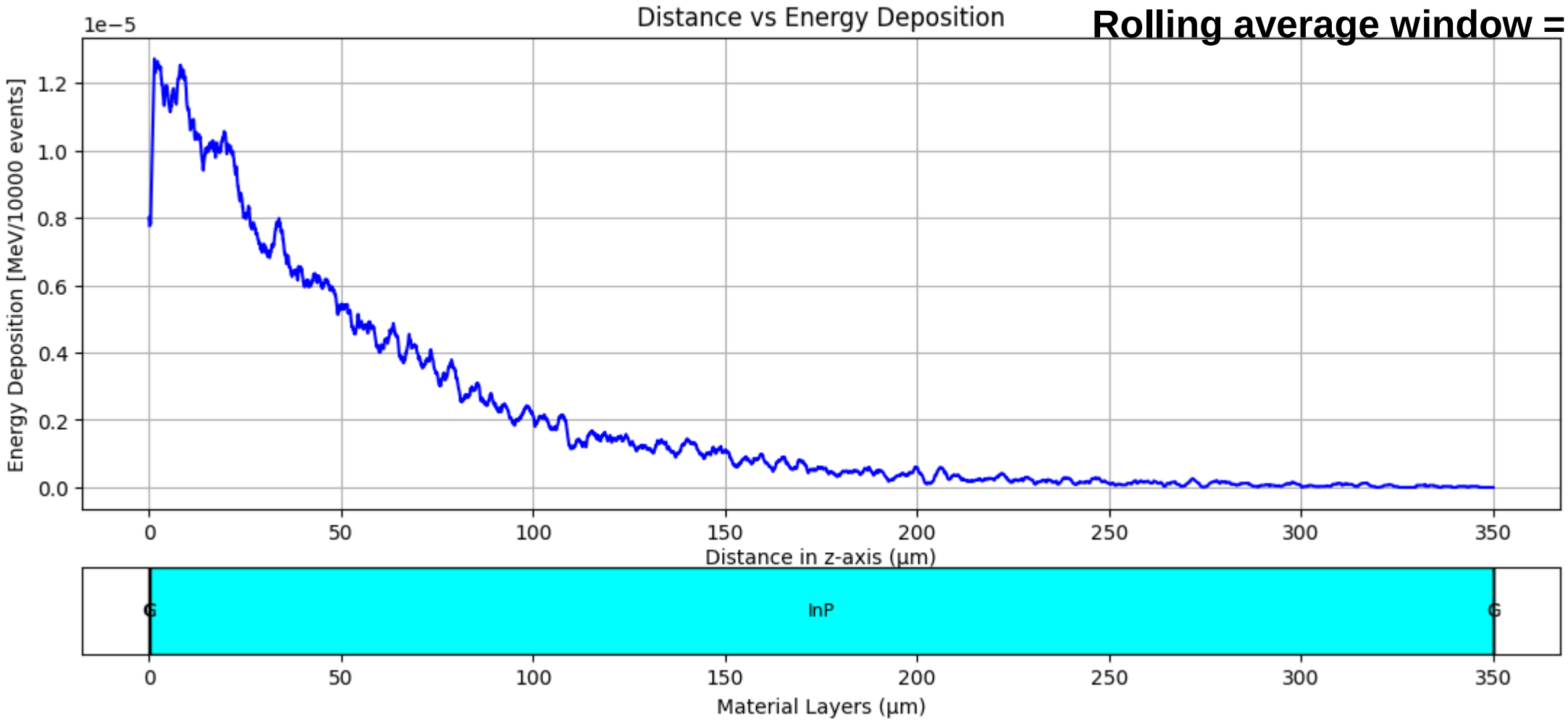
Energy Deposition (in MeV/10000 events)



Energy Deposition (in keV/10000 events)



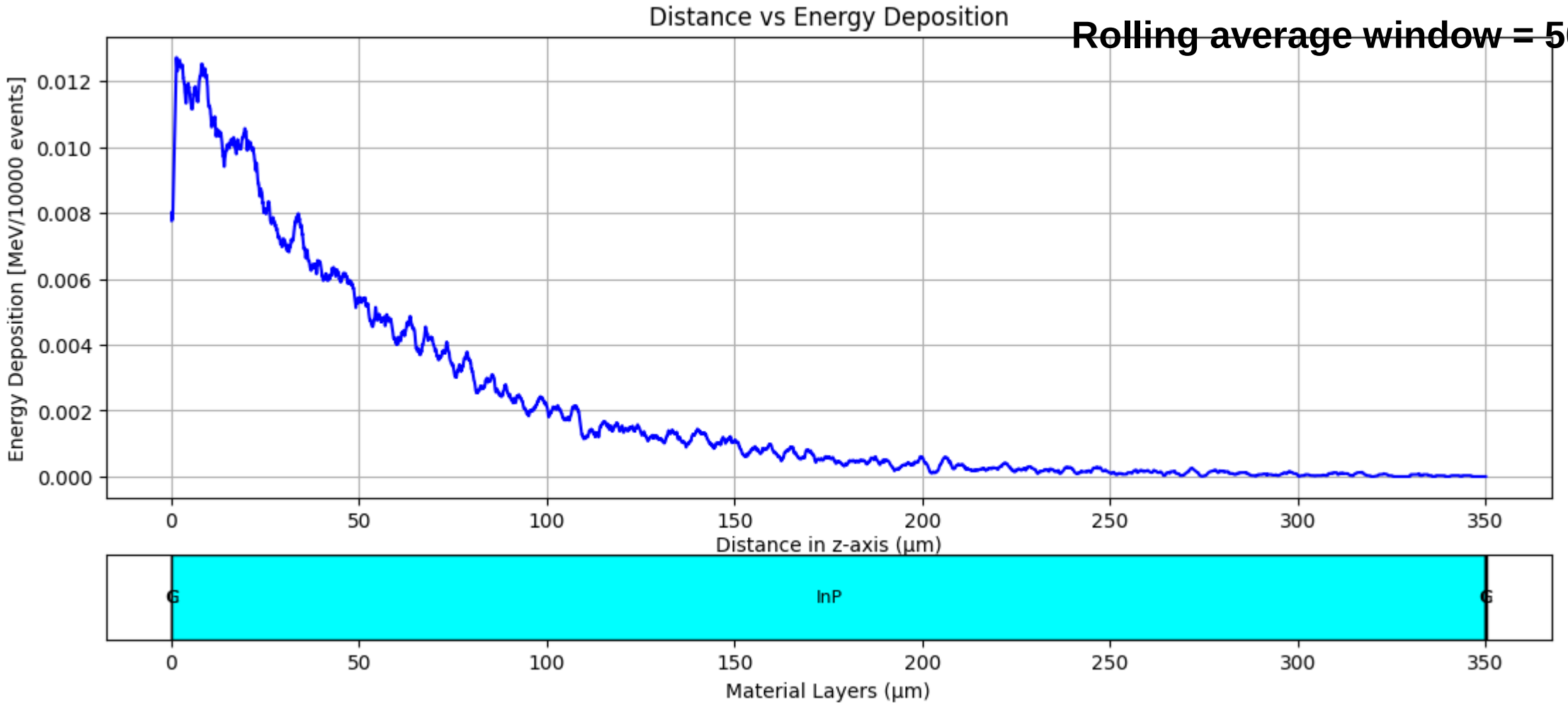
Energy Deposition (in MeV/10000 events)



Rolling average window = 50

```
df2['edep_smooth'] = y2.rolling(window=50,center=True).mean()
y2_smooth = df2['edep_smooth']
```

Energy Deposition (in keV/10000 events)



Rolling average window = 50

```
df2['edep_smooth'] = y2.rolling(window=50,center=True).mean()
y2_smooth = df2['edep_smooth']
```





Backup Slides

SDE Command File

College of Engineering
University of Hawai'i at Mānoa

```
;;; 1.  Clear any previous geometry
(sde:clear)



;;; 2.  Define key dimensions
(define L 5000)    ; length    (x-direction)
(define T  350)    ; thickness (y-direction: y=0 bottom, y=T top)



;;; 3.  Create the InP region
;;;     No doping specified
;;;       y = 0  -> bottom (Cathode)
;;;       y = T  -> top    (Anode)
(sdegeo:create-rectangle
    (position 0  0  0)
    (position L  T  0)
    "InP"
    "InP_body")

;;; 4.  Contacts
;;;     Cathode -> bottom edge  y = 0
;;;     Anode   -> top edge     y = T

(sdegeo:define-contact-set "Cathode" 4 (color:rgb 0 0 1) "##")
(sdegeo:set-current-contact-set "Cathode")
(sdegeo:set-contact (list (car (find-edge-id (position (* L 0.5) 0  0)))) "Cathode")

(sdegeo:define-contact-set "Anode"   4 (color:rgb 1 0 0) "##")
(sdegeo:set-current-contact-set "Anode")
(sdegeo:set-contact (list (car (find-edge-id (position (* L 0.5) T  0)))) "Anode")

;;; 5.  Meshing

;;  -- Global refinement window covering entire device --
(sdedr:define-refeval-window "RefWin.Global" "Rectangle"
    (position 0  0  0)
    (position L  T  0))

(sdedr:define-refinement-size "RefDef.Global" 50 50 5 5)

(sdedr:define-refinement-placement "Place.Global" "RefDef.Global"
    (list "window" "RefWin.Global"))

;;  -- Refinement near Cathode (bottom, y = 0..20 um) --
(sdedr:define-refeval-window "RefWin.Cathode" "Rectangle"
    (position 0  0  0)
    (position L  20  0))

(sdedr:define-refinement-size "RefDef.Cathode" 20 5 2 1)

(sdedr:define-refinement-placement "Place.Cathode" "RefDef.Cathode"
    (list "window" "RefWin.Cathode"))

;;  -- Refinement near Anode (top, y = T-20 to T um) --
(sdedr:define-refeval-window "RefWin.Anode" "Rectangle"
    (position 0  (- T 20)  0)
    (position L   T        0))

(sdedr:define-refinement-size "RefDef.Anode" 20 5 2 1)

(sdedr:define-refinement-placement "Place.Anode" "RefDef.Anode"
    (list "window" "RefWin.Anode"))

;;; 6.  Build mesh and save
(sde:build-mesh "n@node@")

```

Sdevice Command File

COLLEGE OF ENGINEERING
UNIVERSITY of HAWAI'I at MĀNOA

```
File {
  Grid       = "@tdr@"
  Parameters = "@parameter@"
  Plot       = "@tdrdat@"
  Current    = "@plot@"
  Output     = "@log@"
}

Electrode {
  { Name = "Anode"   Voltage = 0.0 }
  { Name = "Cathode" Voltage = 0.0 }
}

Physics {
  Temperature = 300.0
  Fermi
  DefaultParametersFromFile
  EffectiveIntrinsicDensity(noBandGapNarrowing)
  Recombination (
    SRH()
    Radiative
    Auger()
  )
  Mobility (
    DopingDependence(Masetti)
    HighFieldSaturation(CaugheyThomas)
  )
}

Plot {
  Potential eQuasiFermi hQuasiFermi
  eDensity hDensity SpaceCharge
  Current eCurrent hCurrent
  ElectricField
  eMobility hMobility
  eVelocity hVelocity
  ConductionBandEnergy ValenceBandEnergy
  BandGap
}

Math {
  NumberOfThreads = 12
  Wallclock
  Digits = 5
  ErrRef(electron) = 1e4
  ErrRef(hole)     = 1e4
  RHSMin           = 1e-10
  Method    = Blocked
  SubMethod = ILS(set=12)
  ILSrc = "set(12) {
      iterative (gmres(100), tolrel=1e-10, tolunprec=1e-4, maxit=200);
      preconditioning (ilut(1e-08,-1), left);
      ordering (symmetric=nd, nonsymmetric=mpsilst);
      options(verbose=0, refineresidual=10);
  };"
}
```

Sdevice Command File

COLLEGE OF ENGINEERING
UNIVERSITY of HAWAI'I at MĀNOA

```
58 Solve {
59   # --- 1. Initial Equilibrium (Hidden) ---
60   NewCurrentPrefix = "tmp_"
61   Coupled ( Iterations = 100 ) { Poisson }
62   Coupled ( Iterations = 200 ) { Poisson Electron Hole }
63
64   # --- 2. Reverse Bias Sweep (IV_) ---
65   NewCurrentPrefix = ""
66   NewCurrentFile   = "IV_"
67   Quasistationary (
68     InitialStep = 1e-3
69     Increment   = 1.1
70     Decrement   = 2
71     MaxStep     = 1.0
72     MinStep     = 1e-9
73     Goal { Name = "Anode" Voltage = -600.0 }
74   ) { Coupled { Poisson Electron Hole } }
75   Save ( FilePrefix = "_IV_Ramp_" )
76
77   # --- 3. Reset to 0V (Hidden) ---
78   NewCurrentPrefix = "tmp_"
79   Quasistationary (
80     InitialStep = 0.1
81     MaxStep     = 0.1
82     MinStep     = 0.1
83     Goal { Name = "Anode" Voltage = 0.0 }
84   ) { Coupled { Poisson Electron Hole } }
85   Coupled { Poisson }
86   Coupled { Poisson Electron Hole }
87
88   # --- 4. Forward Bias Sweep (IVF_) ---
89   NewCurrentPrefix = ""
90   NewCurrentFile   = "IVF_"
91   # Pin at 0V for a clean origin
92   Quasistationary (
93     InitialStep = 1e-6
94     MaxStep     = 1e-6
95     Goal { Name = "Anode" Voltage = 0.0 }
96   ) { Coupled { Poisson Electron Hole } }
97   # Forward ramp 0V to 5V
98   Quasistationary (
99     InitialStep = 1e-4
100    Increment   = 1.2
101    MaxStep     = 0.02
102    MinStep     = 1e-7
103    Goal { Name = "Anode" Voltage = 600.0 }
104  ) { Coupled { Poisson Electron Hole }
```

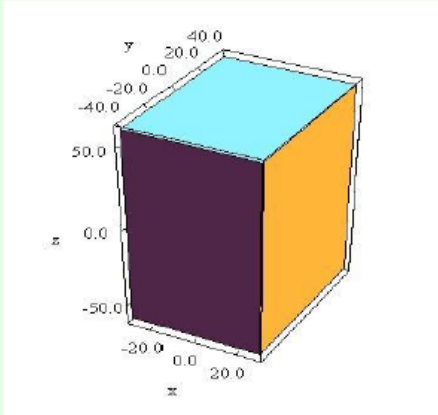
```
105      Plot ( FilePrefix = "_IVF_Ramp_" )
106  }
107  Save ( FilePrefix = "_IVF_Ramp_" )
108
109  # --- 5. Cleanup ---
110  System("rm -f tmp*")
111 }
```

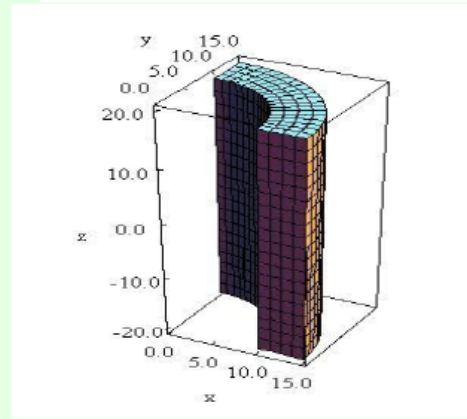
https://geant4.in2p3.fr/IMG/pdf_Lecture-Geometry.pdf

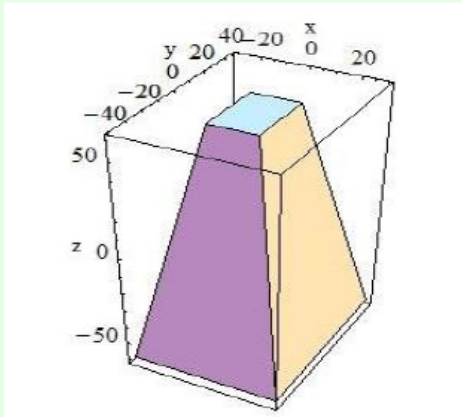
```
G4Box(const G4String& name, // name
      G4double hx,    // x half size
      G4double hy,    // y half size
      G4double hz);   // z half size

G4Tubs(const G4String& name, // name
       G4double rmin,  // inner radius
       G4double rmax,  // outer radius
       G4double hz,    // z-half length
       G4double sphi,  // starting Phi
       G4double dphi); // segment angle

G4Trd(const G4String& name,  // name
      G4double dx1,   // x half size at -dz
      G4double dx2,   // x half size at +dz
      G4double dy1,   // y half size at -dz
      G4double dy2,   // y half size at +dz
      G4double hz);.  // z half size
```







Command	Arguments	Description and restrictions
/gps/pos/type	dist	Sets the source positional distribution type: *Point* [default], *Plane, Beam*, *Surface, Volume*.
/gps/pos/shape	shape	Sets the source shape type, after `/gps/pos/type` has been used. For a Plane this can be *Circle, Annulus*, *Ellipse, Square, Rectangle*. For both Surface or Volume sources this can be *Sphere, Ellipsoid, Cylinder, Para* (parallelepiped).
/gps/pos/centre	X Y Z unit	Sets the centre co-ordinates (X,Y,Z) of the source [default (0,0,0) cm]. The units can only be micron, mm, cm, m or km.
/gps/pos/rot1	$R1_x$ $R1_y$ $R1_z$	Defines the first (x' direction) vector R1 [default (1,0,0)], which does not need to be a unit vector, and is used together with `/gps/pos/rot2` to create the rotation matrix of the shape defined with `/gps/shape`.
/gps/pos/rot2	$R2_x$ $R2_y$ $R2_z$	Defines the second vector R2 in the xy plane [default (0,1,0)], which does not need to be a unit vector, and is used together with `/gps/pos/rot1` to create the rotation matrix of the shape defined with `/gps/shape`.

/gps/pos/sigma_r	sigma unit	Sets the transverse (radial) standard deviation [default 0 cm] of beam position profile. The units can only be micron, mm, cm, m or km.
/gps/pos/sigma_x	sigma unit	Sets the standard deviation [default 0 cm] of beam position profile in x-direction. The units can only be micron, mm, cm, m or km.
/gps/pos/sigma_y	sigma unit	Sets the standard deviation [default 0 cm] of beam position profile in y-direction. The units can only be micron, mm, cm, m or km.
/gps/pos/paralp	alpha unit	Used with a Parallelepiped. The angle [default 0 rad] α formed by the y-axis and the plane joining the centre of the faces parallel to the zx plane at y and +y. The units can only be deg or rad.
/gps/pos/parthe	theta unit	Used with a Parallelepiped. Polar angle [default 0 rad] θ of the line connecting the centre of the face at z to the centre of the face at +z. The units can only be deg or rad.
/gps/pos/parphi	phi unit	Used with a Parallelepiped. The azimuth angle [default 0 rad] ϕ of the line connecting the centre of the face at z with the centre of the face at +z. The units can only be deg or rad.
/gps/pos/confine	name	Allows the user to confine the source to the physical volume *name* [default NULL].

OF ENGINEERING
SITY of HAWAI'I at MĀNOA

Command	Arguments	Description and restrictions
/gps/ang/type	AngDis	Sets the angular distribution type (*iso* [default], *cos, planar, beam1d, beam2d, focused, user*) to either isotropic, cosine-law or user-defined.
/gps/ang/rot1	$AR1_x$ $AR1_y$ $AR1_z$	Defines the first (x' direction) rotation vector AR1 [default (1,0,0)] for the angular distribution and is not necessarily a unit vector. Used with `/gps/ang/rot2` to compute the angular distribution rotation matrix.
/gps/ang/rot2	$AR2_x$ $AR2_y$ $AR2_z$	Defines the second rotation vector AR2 in the xy plane [default (0,1,0)] for the angular distribution, which does not necessarily have to be a unit vector. Used with `/gps/ang/rot2` to compute the angular distribution rotation matrix.
/gps/ang/mintheta	MinTheta unit	Sets a minimum value [default 0 rad] for the θ distribution. Units can be deg or rad.
/gps/ang/maxtheta	MaxTheta unit	Sets a maximum value [default π rad] for the θ distribution. Units can be deg or rad.
/gps/ang/minphi	MinPhi unit	Sets a minimum value [default 0 rad] for the ϕ distribution. Units can be deg or rad.
/gps/ang/maxphi	MaxPhi unit	Sets a maximum value [default 2π rad] for the ϕ distribution. Units can be deg or rad.
/gps/ang/sigma_r	sigma unit	Sets the standard deviation [default 0 rad] of beam directional profile in radial. The units can only be deg or rad.
/gps/ang/sigma_x	sigma unit	Sets the standard deviation [default 0 rad] of beam directional profile in x-direction. The units can only be deg or rad.
/gps/ang/sigma_y	sigma unit	Sets the standard deviation [default 0 rad] of beam directional profile in y-direction. The units can only be deg or rad.