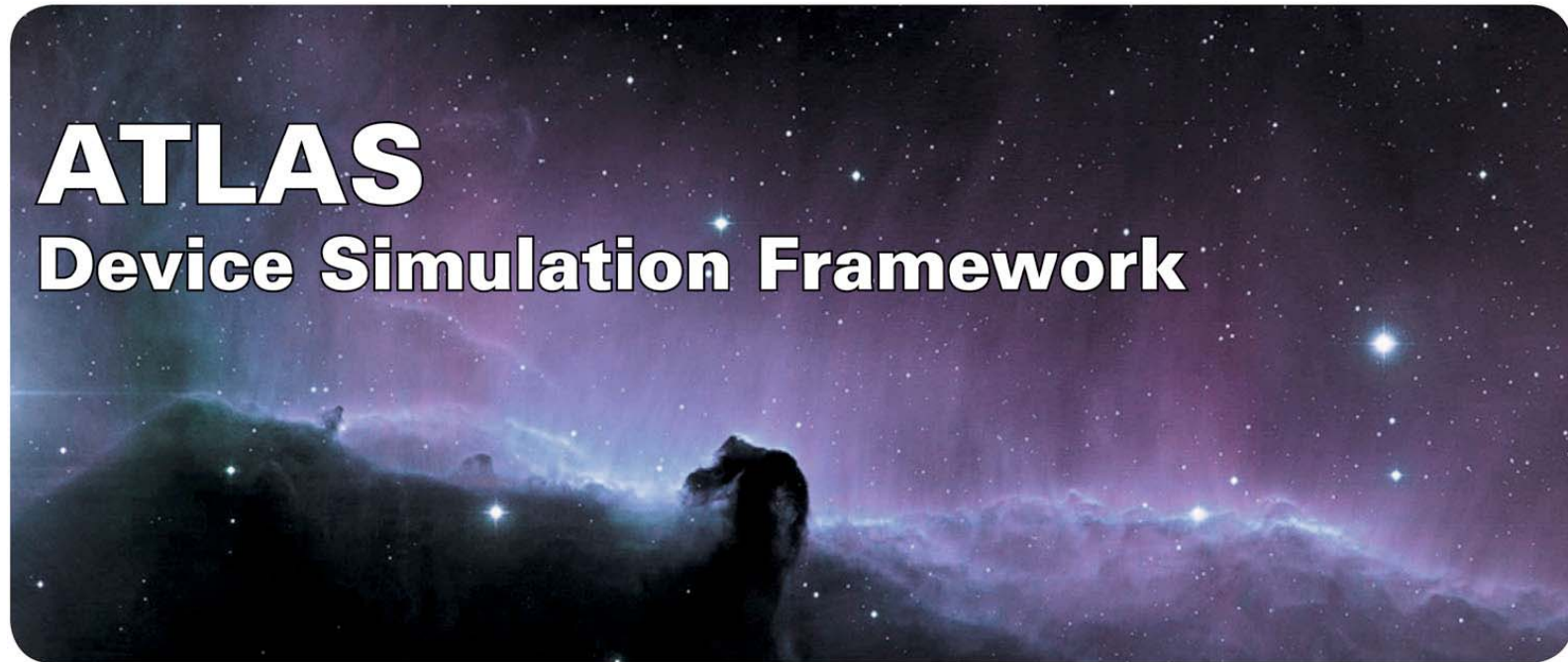


Mocasim



Monte Carlo Transport Parameter Generator



**SILVACO**



## Summary

### Motivation: Reasons for creating Mocasim Simulator

- Outline of the Mocasim simulator
- Output: Sample Results
- Conclusions



## Motivation

### **Modern physical device simulators require both material & bias-dependent transport parameters**

- **PROBLEM**
  - Questionable mobility models once user moves away from common material systems
  - Even if mobility is characterized (often only over a limited range) - users still require additional relationships for more advanced transport models
  - These relationships are completely unavailable for more exotic materials
- **SOLUTION**
  - Mocasim, a new physical transport parameter simulator



## Physical Problem

- More materials
  - Need to model more complex material systems
- Better transport models
  - As device features reduce in size, simple transport models (drift-diffusion) are no longer adequate.
  - Modern device simulators (ATLAS, MERCURY etc.) include more sophisticated models which describe non-stationary transport effects
  - These take the form of hydrodynamic equations which are generated by taking the 1st three moments of the BTE and creating a set of conservation equations
  - Many alternative hydrodynamic models exist, full three valley, single electron gas etc. each with varying numbers of terms. However, all these models require material dependent terms,  $m^*$ ,  $\tau_w$ ,  $\tau_p$  etc., in addition to mobility



# Hydrodynamic Transport Equations

- 1st three moments of the BTE
  - Particle Conservation (current continuity):

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\vec{v}) = 0$$

- Momentum Conservation:

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} + \frac{q\vec{E}}{m_e^*} + \frac{1}{m_e^* n} \nabla (nk_B T) = -\frac{\vec{v}}{\tau_p}$$

- Energy Conservation:

$$\frac{\partial w}{\partial t} + q\vec{v} \cdot \vec{E} + \frac{1}{n} \nabla (n\vec{v}k_B T) = -\frac{(w - w_0)}{\tau_w}$$



## Material Dependent Parameters

- In addition to mobility ( $\mu$ ) Mocasim supplies a variety of material dependent parameters:

Symbol	Parameter
$\mu$	Mobility
V	Velocity
$m^*$	Effective mass
$\tau_p$	Momentum relaxation time
$\tau_w$	Energy relaxation time
$G_{\Gamma,L,X}$	$\Gamma$ , L & X valley distribution fractions
PE	Inter-valley potential energy



## Summary

- Motivation: Reasons for creating Mocasim Simulator

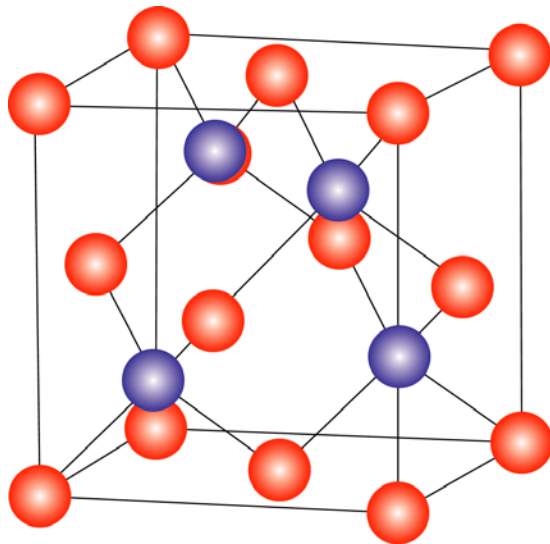
### Outline of the Mocasim Simulator

- Output: Sample Results
- Conclusions

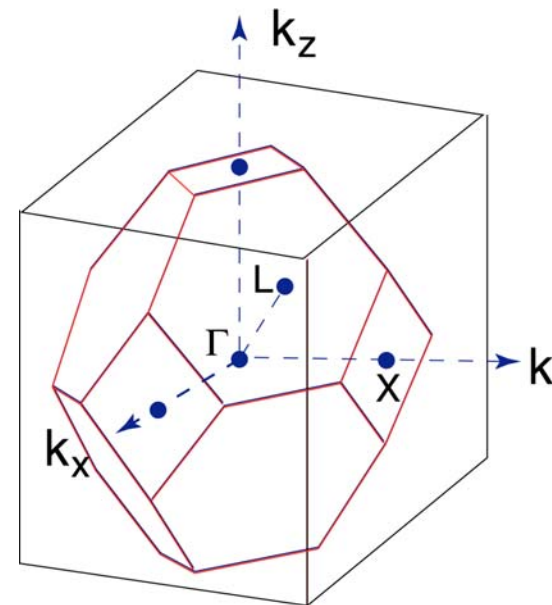


## Mocasim Application

- Mocasim has been designed to generate material dependent parameters for both direct and indirect band gap cubic semiconductors - including group IV and most of the group III-V material systems



Zincblende crystal structure



1<sup>st</sup> Brillouin zone of the FCC lattice





## Operation

- Mocasim is a three-valley ensemble Monte Carlo simulator
- Carriers start with an energy distribution given from equilibrium Boltzmann statistics at the specified lattice temperature
- Electrons undergo acceleration by electric field
- Electrons scatter into different states by mechanisms controlled by the user
- Repeats process (typically 10's of millions of operations) and performs an ensemble average to extract the transport parameters



## Input Variables & Parameters

- Variables for simulator control
  - Doping range
  - Electric field range
- Material parameters
  - Crystal structure properties: density, sound velocity, dielectric constant
  - Conduction band properties: for each band: band gaps, effective masses, non-parabolicity, degeneracy
  - Intravalley phonon parameters: energies for electric and deformation potential scattering
  - Intervalley phonon parameters: energies and deformation potentials for phonon scattering



## Scattering Mechanisms

- No lattice is perfect, rigid or infinite. These imperfections effect the electron's transport
- Assume perturbations small enough not to alter band structure, merely to cause a change in the carriers, from one quantum state to another (scattering)
- These effects are analyzed through 1st order time-dependent perturbation theory
- Rate of change from initial state  $i$  to final state  $f$  given by Fermi's Golden rule:

$$P(k_i, k_f) = \frac{2\pi}{\hbar} |H'|^2 \delta(E_f - E_i)$$



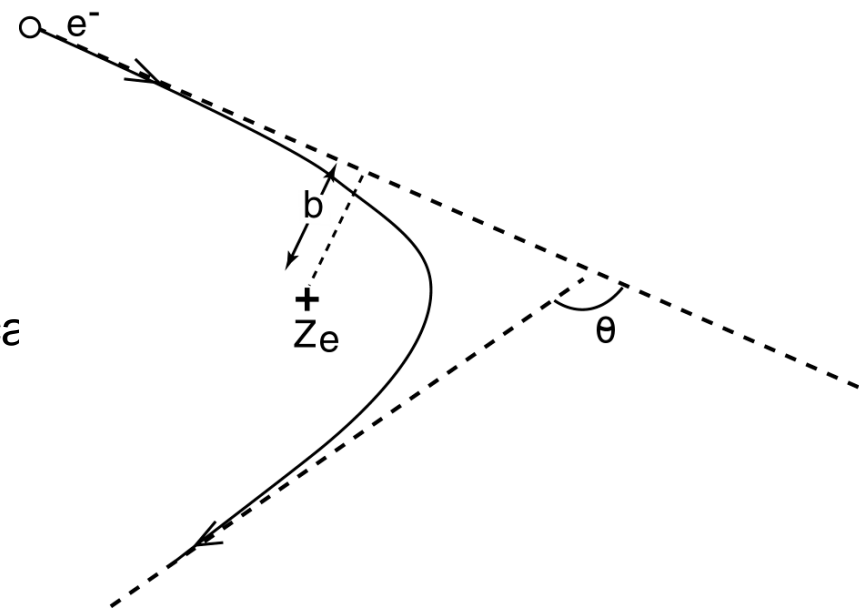
## Scattering Mechanisms (con't)

- Mocasim allows the user to select any number of scattering mechanisms from the following list of predefined algorithms, or define their own by using the Silvaco C-Interpreter
  - Intervalley Phonon Scattering
    - $\Gamma \leftrightarrow X$  transitions
    - $\Gamma \leftrightarrow L$  transitions
    - $L \leftrightarrow L$  transitions
    - $L \leftrightarrow X$  transitions
    - $X \leftrightarrow X$  transitions
  - Intravalley Phonon Scattering
    - Deformation Potential Acoustic
    - Piezoelectric Acoustic
    - Deformation Potential Optical
    - Polar Optical



## Scattering Mechanisms (con't)

- Impurity Scattering
  - Ionized Impurity
  - Neutral Impurity
- User-Defined Scattering
  - Both Intra & Inter-valley Scat



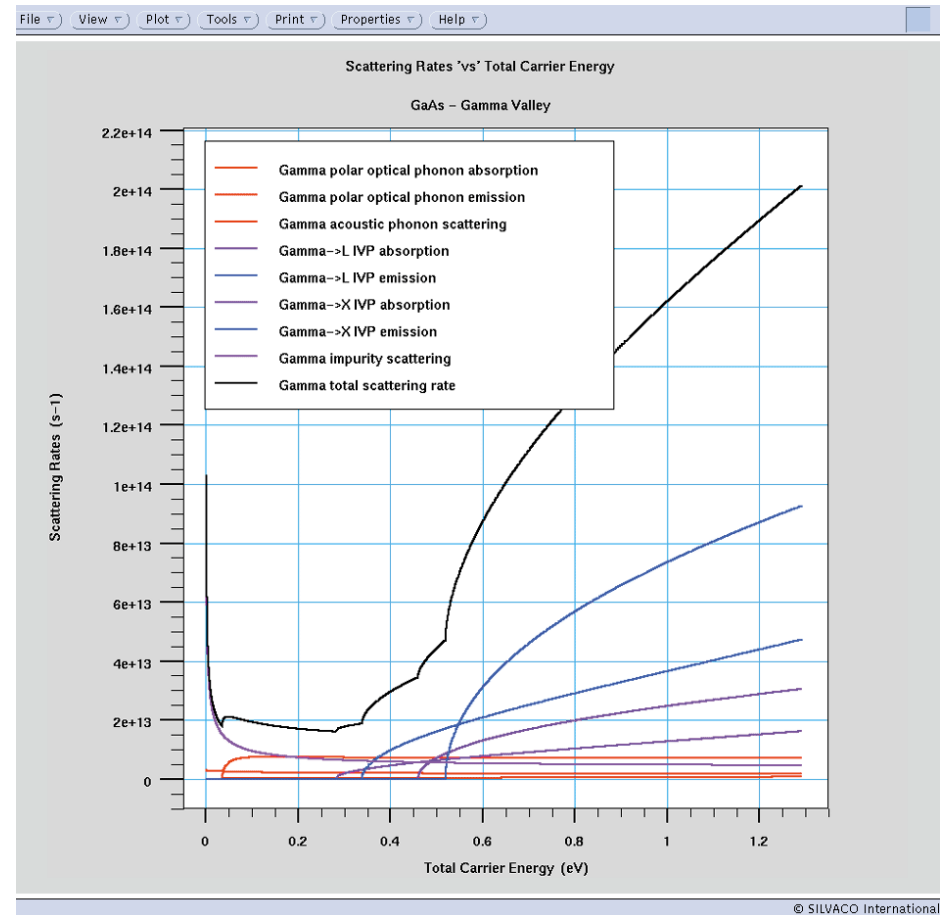
Conwell-Weiskopf model:

- Ionized impurity scattering, where  $e^-$  is the electron,  $Ze^+$  is the impurity,  $b$  the impact parameter and  $\theta$  is the scattering angle



## Scattering Mechanisms (con't)

- The individual scattering rates are then summed to give the total scattering rate for each valley
- The figure opposite shows the individual and cumulative scattering rates for the  $\Gamma$  valley in GaAs as a function of carrier energy





## Silvaco C-Interpreter

- Uses the latest version of the Silvaco C-Interpreter
- III-V material science & devices are an area of active research:
  - Important to be able to incorporate the latest research without waiting for a commercial software update
- Some research is proprietary for competitive advantage. The Silvaco C-Interpreter allows the integration of in-house algorithms into a robust simulation environment
- Minimal impact upon execution speed  
(important for functions that may be called  $10^3$  -  $10^8$  times)
- Includes GUI debugger



## Silvaco C-Interpreter (con't)

```
#include <math.h>

/*****
/* The Gamma -> L scattering rate function */
*****/
double scattering_rate(double EG)
{
    double EL, mag_kL, denom, L, M;

    /* calculate the energy after the scattering */
    /* deltaGL=0.33eV (G lower than L), Ephonon=0.02788eV */
    EL=EG -0.33 +0.02788;
    if (EL<=0.0) return 0.0;

    /* calculate the magnitude of the wavevector after it has scattered */
    mag_kL=2.4139e9*sqrt(EL*(1.0+0.463*EL));

    denom=(1.0 + 1.226*EG)*(1.0 + 0.926*EL);
    L=(1.0 + 0.613*EG)*(1.0 + 0.463*EL) / denom;
    M=0.28382*EG*EL / denom;

    return 3421.1*mag_kL*(1.0 + 0.926*EL)*(2.0*L + 0.66667*M);
}

/*****
```

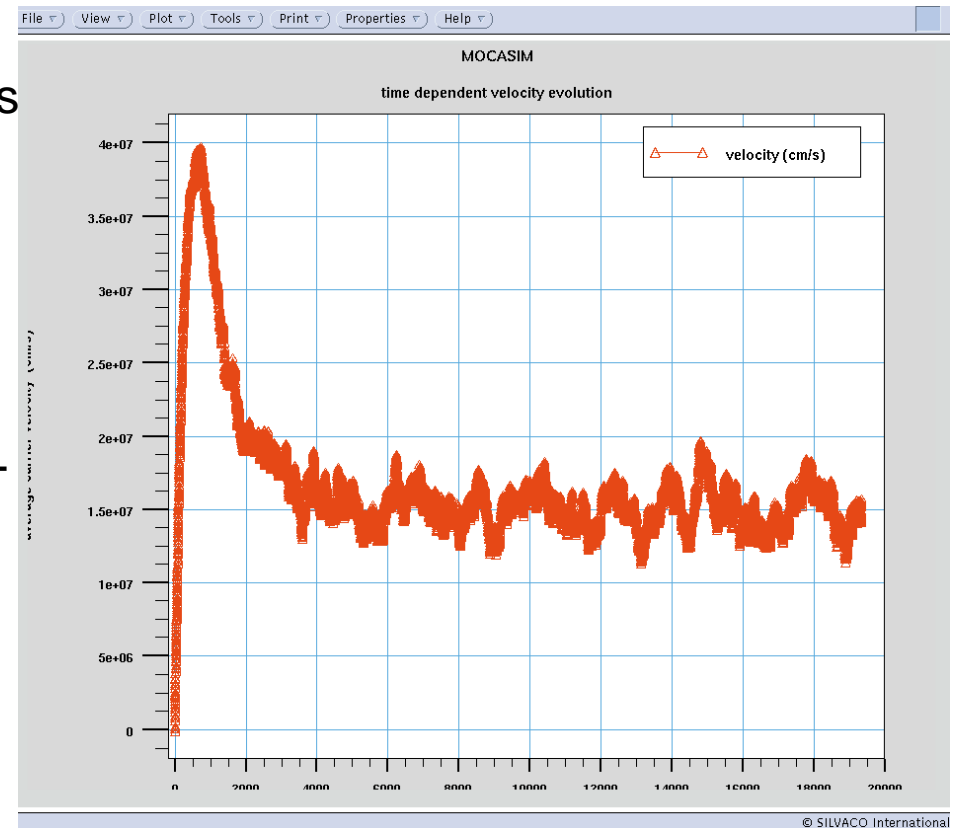
C-Interpreter prototype for the  $\Gamma \leftrightarrow L$  absorption scattering rate





# Simulation

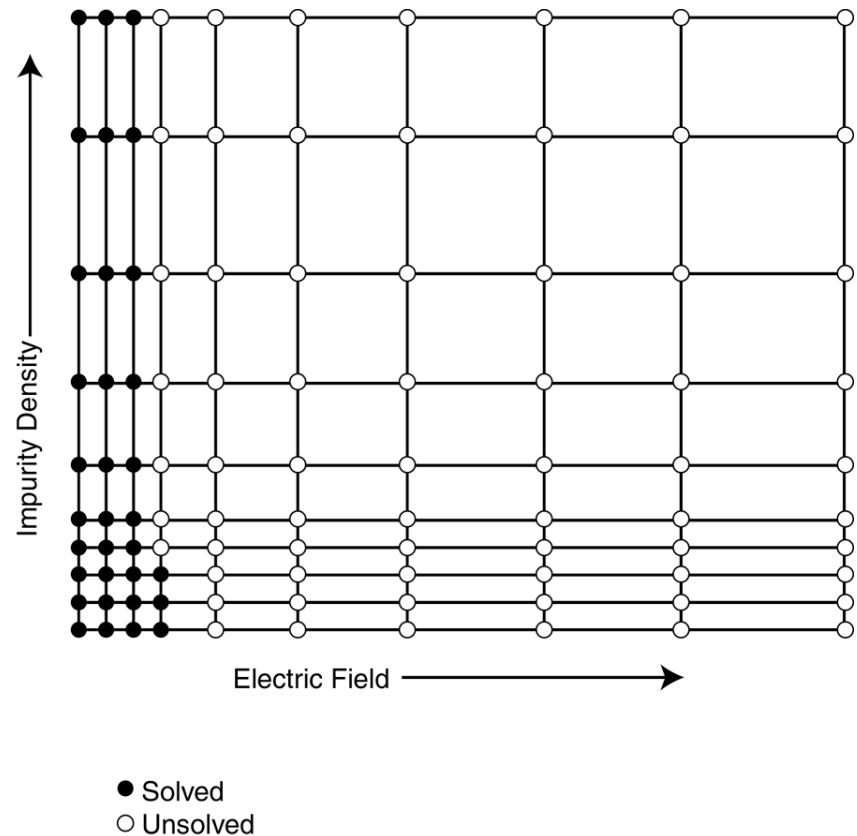
- The carriers are allowed to travel through many time steps
- Once steady-state has been attained, an ensemble average is performed to extract the bulk parameters of interest
- This figure illustrates the burn-in time associated with a simulation as the carriers attain to their final distribution





# Operation

- The user can choose to either simulate a single bias point, perform a sweep in electric field or chose to characterize a full 2D field-doping plane
- The figure opposite shows a nominal 2D field-doping simulation plane. The bias points are generated on a non-uniform grid to provide both accuracy and efficiency





## Summary

- Motivation: Reasons for creating Mocasim Simulator
- Outline of the Mocasim simulator

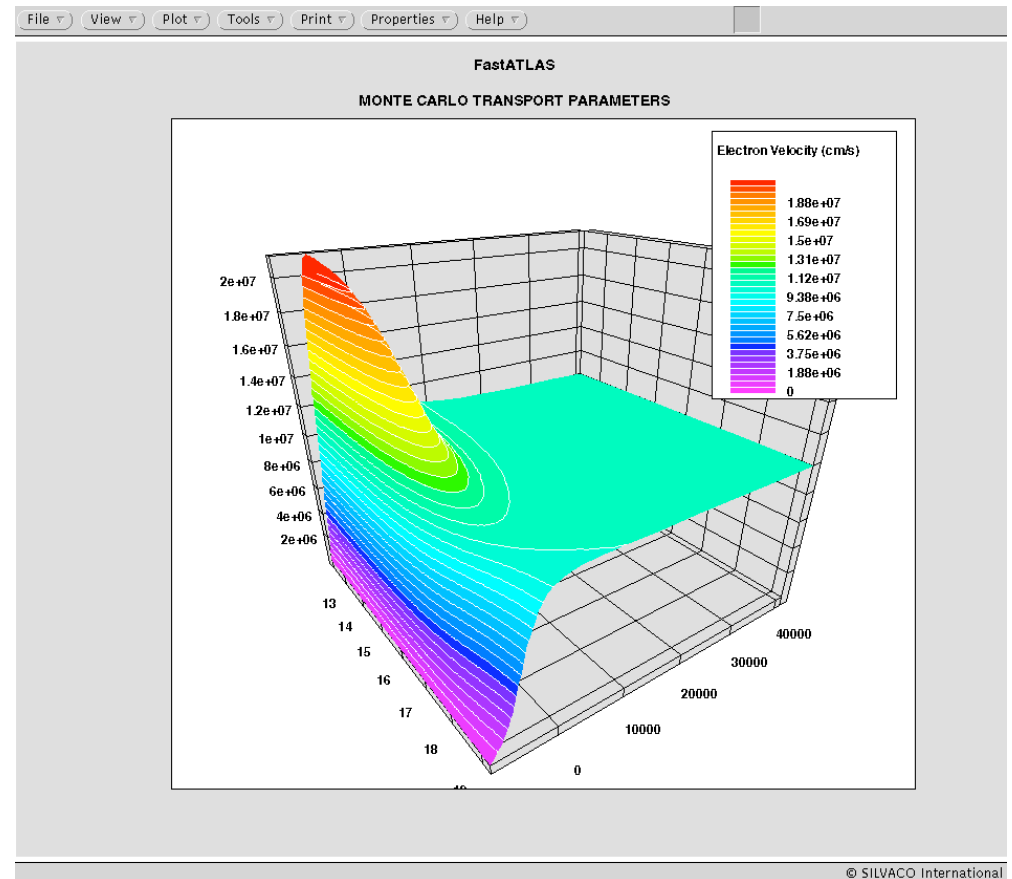
### Output: Sample Results

- Conclusions



# Results

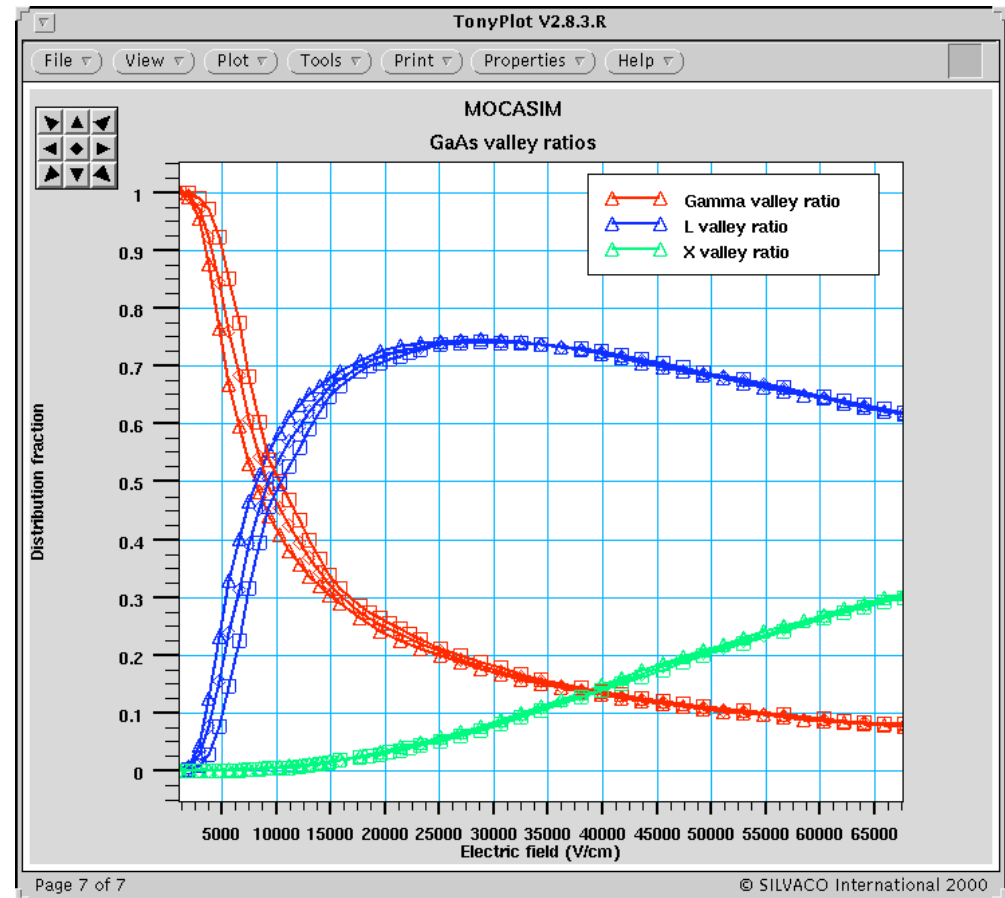
- Once generated, the user can view the results in TonyPlot
- This figure shows the electron velocity for GaAs as a function of applied electric field and impurity density





## Results (con't)

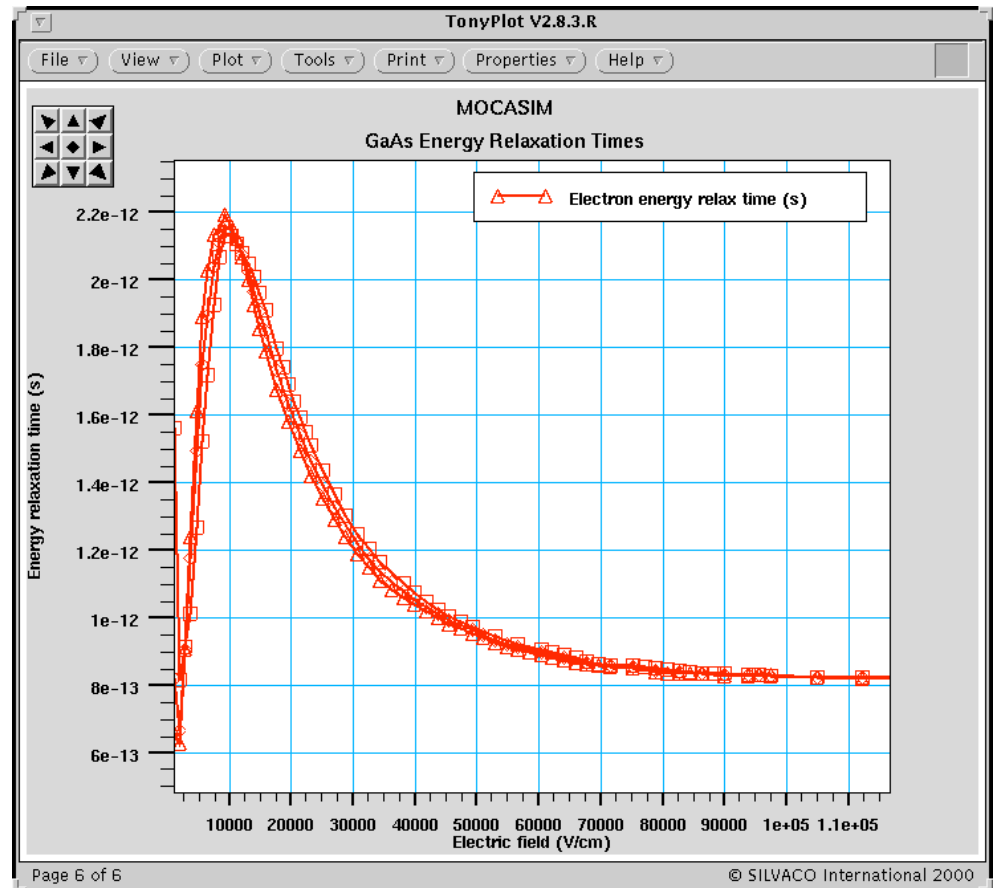
- This figure shows the  $\Gamma$ , L & X valley distribution fractions for GaAs as a function of applied electric field and impurity density





## Results (con't)

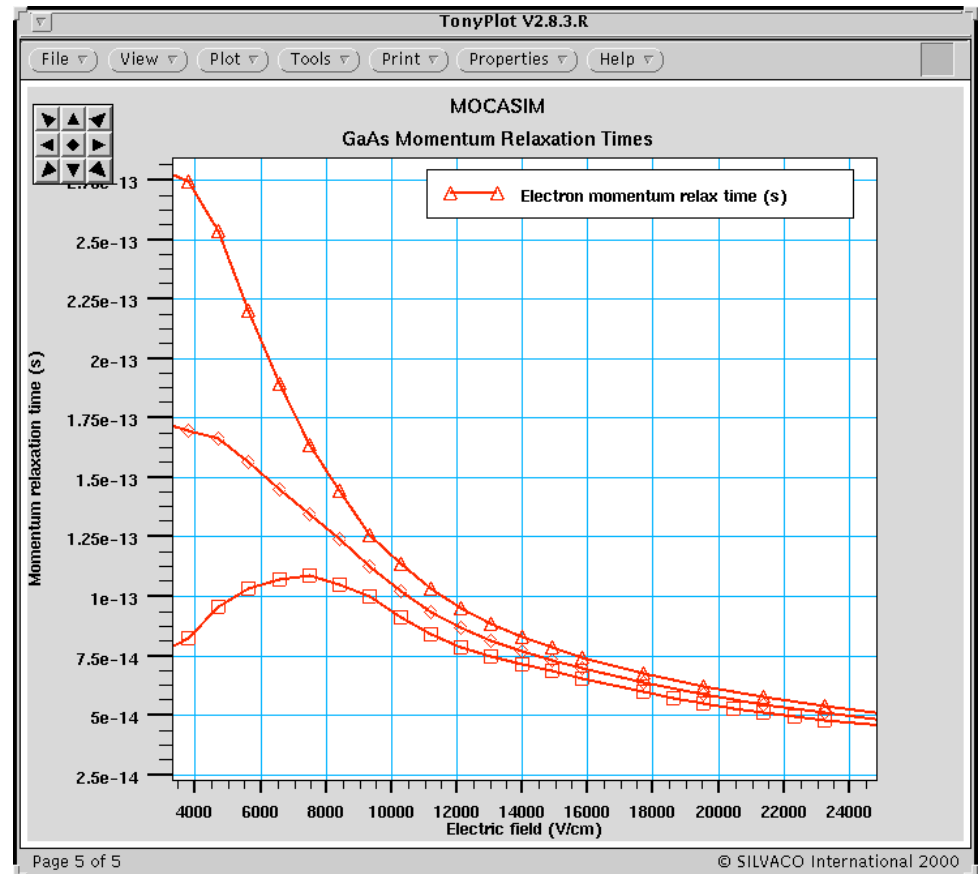
- This figure shows the electron energy relaxation times for GaAs as a function of applied electric field and impurity density





## Results (con't)

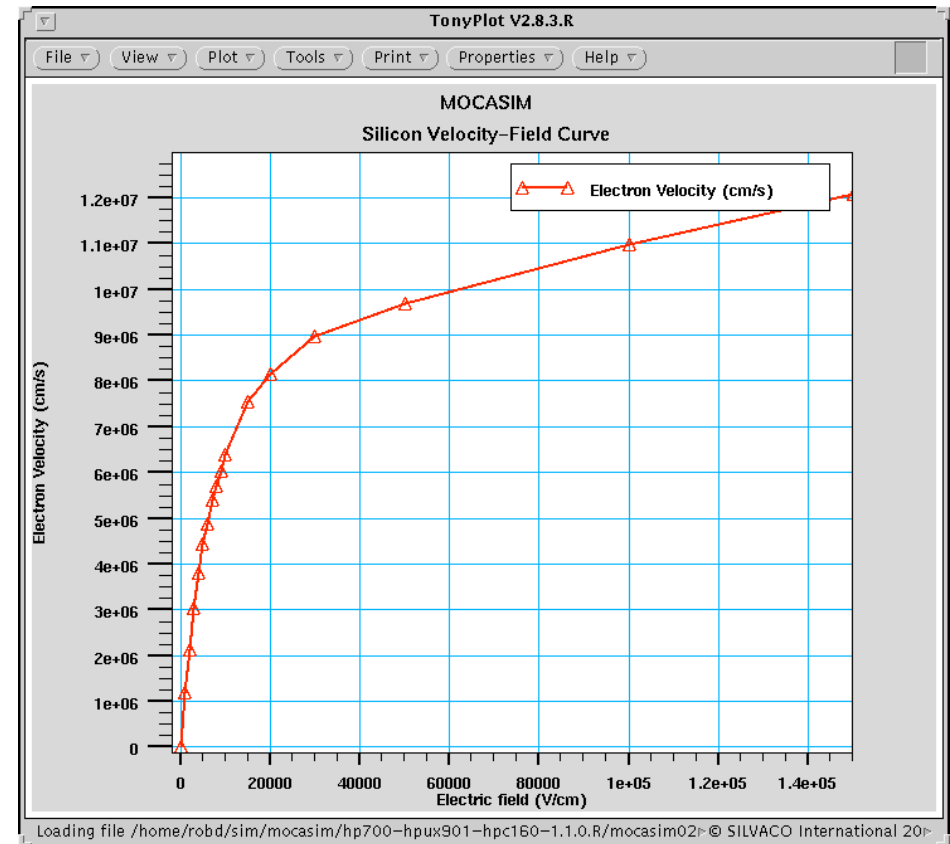
- This figure shows the electron momentum relaxation times for GaAs as a function of applied electric field and impurity density





## Results (con't)

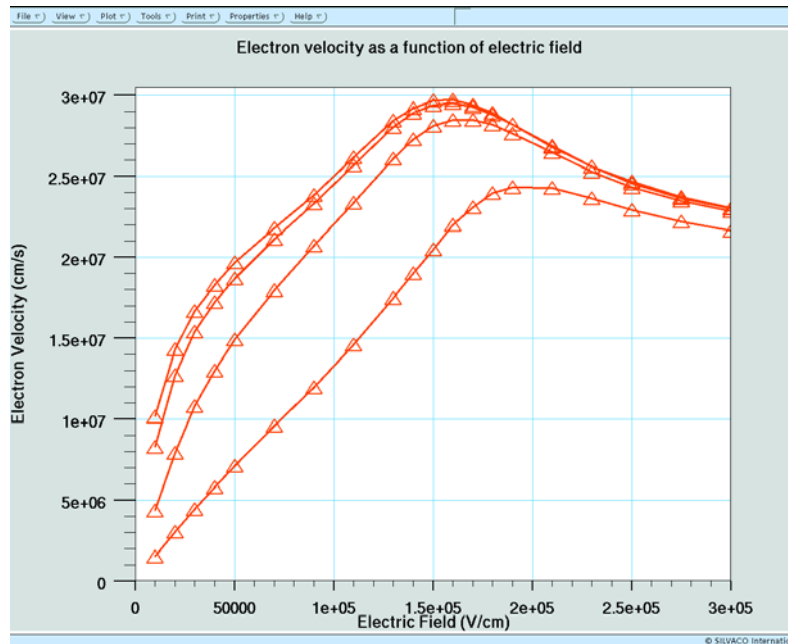
- This figure shows the electron velocity for Silicon as a function of applied electric field







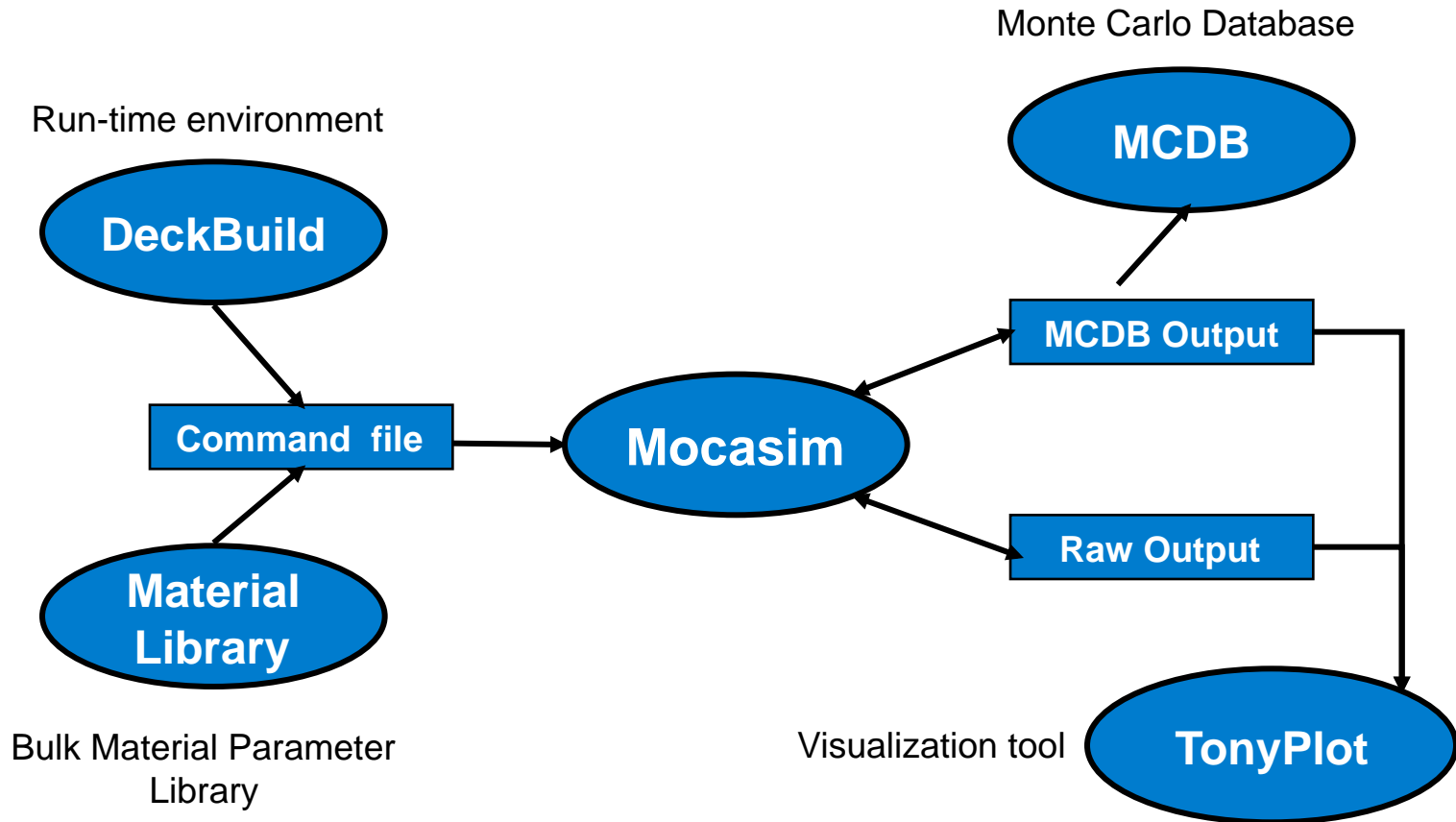
# Empirical Model Development



- The electron velocity in GaN at 300K is shown as a function of electric field at doping of  $10^{15}$ ,  $10^{17}$ ,  $10^{18}$ , and  $10^{19} \text{ cm}^{-3}$
- Fitting these curves to an analytic function provides the empirical model



# Mocasim Overview





## Future Plans

- Impact ionization  
Extend Mocasim to extract the impact ionization coefficients
- Strained layer & interface transport modeling
- Hole transport
- Diffusion coefficient for noise simulation



## Summary

- Motivation: Reasons for creating Mocasim Simulator
- Outline of the Mocasim simulator
- Output: Sample Results

## Conclusions



# Mocasim Simulation Parameters

## Input Parameters

**Electric Field**  
**Impurity Density**  
**Lattice Temperature**

## Output Parameters

**k-vector**  
**Kinetic Energy**  
 **$\Gamma$  -valley fraction**  
**L-valley fraction**  
**X-valley fraction**

## Derived Parameters

**Inter-valley Potential Energy**  
**Total Energy**  
**Effective Mass**  
**Velocity**  
**Mobility**  
**Momentum Relaxation Time**  
**Energy Relaxation Time**



## Conclusion

- Mocasim provides:
  - The user with a method of creating the transport parameters needed for modern device simulation
  - A flexible & robust solution procedure
    - Allows an arbitrary number of scattering mechanisms to be included, selected from a library of pre-defined mechanisms or via the Silvaco C-Interpreter
    - User controls majority of simulation parameters: number of samples, energy ranges, burn-in time etc.