ECE539 - Advanced Theory of Semiconductors and Semiconductor Devices

Numerical Methods and Simulation / Umberto Ravaioli

Monte Carlo Simulation

1 Introduction

The name of Monte Carlo method is usually given to stochastic techniques which use random number generation. In the case of particle transport, the Monte Carlo method is used to solve directly the Boltzmann equation, without making assumptions on the distribution function. The appropriate probability distributions for the parameters involved are obtained from uniform random number sequences. This is achieved through numerical experiments which simulate the actual motion of particles inside a semiconductor. In general, a semiclassical picture is adopted, according to the following approximations:

1. Particles are simulated as classical point–like objects, characterized by definite position and momentum (therefore, the uncertainty principle of quantum mechanics is violated).
2. The effective mass approximation is assumed. This includes the main quantum features related to the periodic crystal potential.
3. The motion of the particles is described as a series of free flights interrupted by scattering events. The trajectories and the momentum variations are calculated using classical mechanics. The scattering rates and the scattering angles are calculated according to quantum mechanical rules.

1.1 Band structure models

The relationship between the momentum $\mathbf{k}$ and the kinetic energy $E(\mathbf{k})$ can be accounted for according to different levels of approximation:

(a) **Parabolic Band**

$$E(\mathbf{k}) = \frac{\hbar^2 |\mathbf{k}|^2}{2m_o^*} \tag{1}$$

where m_o^* is the effective mass at the conduction band minimum (or valence band maximum). The particle velocity is evaluated as

$$\mathbf{v} = \frac{1}{\hbar}\nabla_k E(\mathbf{k}) = \frac{\hbar \mathbf{k}}{m_o^*} \tag{2}$$

which verifies the simple relation for the mechanical momentum

$$\mathbf{p} = \hbar\mathbf{k} = m_o^* \mathbf{v} \tag{3}$$

(b) **Non–parabolic Band**

$$E(\mathbf{k})(1+\alpha E(\mathbf{k})) = \frac{\hbar^2|\mathbf{k}|^2}{2m_o^*} \tag{4}$$

where α is the coefficient of non–parabolicity and has the dimensions of an inverse energy. The solution of the second order equation (4) is

$$E(\mathbf{k}) = \frac{\sqrt{1+\frac{4\alpha\hbar^2|\mathbf{k}|^2}{2m_o^*}}-1}{2\alpha} \tag{5}$$

The velocity is

$$\mathbf{v} = \frac{1}{\hbar}\nabla_k E(\mathbf{k}) = \frac{\hbar\mathbf{k}}{m_o^*}\left(1+4\alpha\frac{\hbar^2|\mathbf{k}|^2}{2m_o^*}\right)^{-1/2} \tag{6}$$

and using (5)

$$\mathbf{v} = \frac{\hbar\mathbf{k}}{m_o^*[1+2\alpha E(\mathbf{k})]} \tag{7}$$

The coefficient of non–parabolicity α is given by

$$\alpha = \frac{(1-\frac{m_o^*}{m_o})^2}{E_{gap}} \tag{8}$$

where m_o is the electron mass in vacuum, and E_{gap} is the energy gap between valence and conduction band.

So far, a spherically symmetric band has been implicitly assumed, with isotropic effective mass. This is normally a good approximation for the Γ valley of $GaAs$, centered about the (0,0,0) point of $\mathbf{k}$–space. In the case of Ge and Si, the bandgap is indirect, therefore in the conduction band there are a number of equivalent valleys, and $E(\mathbf{k})$ is described by an ellipsoid rather than a sphere. Ge has eight valleys with the minimum centered approximately (in units of the reciprocal lattice constant) at the (1,1,1) point of the Brillouin zone, called the L–point, while Si has six valleys with minimum at the (1,0,0) or X–point. In $GaAs$, the first satellite valley is Ge–like and will be called the L–valley in the following. The second satellite valley is Si–like, and will be called the X–valley. In an ellipsoidal valley the mass is not isotropic, therefore it is described by a tensor with components

$$m_{ij}^* = \hbar^2\left(\frac{\partial^2 E}{\partial k_i \partial k_j}\right)^{-1} \tag{9}$$

where i and j correspond to combinations of the coordinates x, y, and z. Cross–terms arise because the axes of the ellipsoids are not going to be parallel to the cartesian axes, in general. For simplicity, one may consider a spherical valley with an average conductivity effective mass, to average the effect of different valley orientation. This is more appropriate for 1–D simulations, but may be acceptable for 2–D simulations, as long as the neglect of anisotropy of the effective mass is not critical.

(c) **Full Band Structure**

The function $E(\mathbf{k})$ is determined numerically with detailed band structure calculations or from experiments. The calculations for the particle motion become fully numerical. The inclusion of band structure in Monte Carlo simulations is discussed in Shichijo and Hess.

The non–parabolic band approximation is probably still the best trade–off for many device applications, since a full band structure approach is very time consuming. In the case of very high field transport, which requires the better physical model of the full band structure, it is also possible to use a hybrid approach which treats the less energetic particles with the non–parabolic band formulation. This is effective for device simulation, since only a small fraction of electrons are likely to be in the high field region (most of the simulated particles are usually in the highly doped contact regions).

1.2 Simulation procedures

Independently of the band–structure model, the following types of Monte Carlo simulations can be performed:

(a) **One–particle Monte Carlo**

The evolution of one single particle is followed in time. If the steady state is reached, according to the ergodic theorem it is possible to use *time averages* for the particle simulated as representative of *ensemble averages* at a give time for all the particles. This method is mostly useful to study bulk properties, like the steady state drift velocity as a function of field, or the field–dependent diffusion coefficient. The direct application to device simulation is necessarily limited to cases in which the potential distribution in space is well known, and it is possible to make reasonable assumptions on the energy of particles injected into the device. For this type of simulation (Monte Carlo post–processing) one electron is injected and the motion is tracked in the domain, until it exits through a contact. Space averages are then made over a large number of such experiments.

(b) **Ensemble Monte Carlo**

The algorithm is essentially identical to the one–particle Monte Carlo, but now a large ensemble of particles is simulated simultaneously. This procedure is obviously a good candidate for super-computation, since one may apply parallelization and vectorization. Also, it is now possible to perform ensemble averages directly, at a given time step, therefore the constraint of steady state for the validity of the results does not apply. This makes the ensemble Monte Carlo suitable for *transient simulations.*

(c) **Self–consistent Ensemble Monte Carlo**

This method couples the ensemble Monte Carlo procedure to Poisson's equation, and is the most suitable for device simulation. Typically, Poisson's equation is solved at fixed intervals (1 to 10 femtoseconds of simulation time) to update the internal field distribution, to reflect the internal redistribution of charge, due to the movement of particles.

2 Random Flight Selection

As already mentioned, the motion of carriers is simulated as a series of free flights interrupted by instantaneous scattering events, representing the carrier–phonon, the carrier–impurity, and the carrier–carrier interactions. After a scattering, the carrier will emerge with a crystal momentum $\mathbf{k}(0)$ at time $t = 0$. The time evolution of the momentum during the flight is

$$\mathbf{k}(t) = \mathbf{k}(0) \pm \frac{q\mathbf{E}(t)}{\hbar} \tag{10}$$

where the plus applies for holes, and the minus for electrons. At each instant there exists a definite total scattering rate $\lambda(\mathbf{k})$, function of the magnitude of the momentum, which is obtained by adding up the scattering rates $\lambda_i(\mathbf{k})$ for all the possible collision mechanisms. The scattering rate indicates the average number of collisions that a particle would suffer in the unit time, if it would maintain the same magnitude of the momentum $\mathbf{k}$, and therefore it has units of inverse seconds. The probability for a carrier to be scattered in the time interval $[t, t + \delta t]$ is $\lambda(\mathbf{k}(t))\delta t$, δt being very small. The probability that the carrier will not scatter in the same interval is therefore $(1 - \lambda(\mathbf{k}(t))\delta t)$. As indicated, the total scattering rate is a function of time, through the momentum dependence. The probability for a carrier to travel without scattering for a time t becomes simply

$$\begin{aligned} P_f(t) &= \prod_i (1 - \lambda(k(t))\delta t_i) & (11)\\ t &= \sum_i \delta t_i & (12) \end{aligned}$$

(11) is easier to handle if the logarithm of both sides is taken

$$\begin{aligned} \ln(P_f(t)) &= \sum_i \ln\left(1 - \lambda(\mathbf{k}(t))\delta t_i\right) \\ &\approx -\sum_i \lambda(\mathbf{k}(t))\delta t_i & (13) \end{aligned}$$

with $1 \gg \lambda(\mathbf{k}(t))\delta t_i$. The summation may be replaced by an integral, and we have

$$P_f(t) = \exp\left(-\int_0^t \lambda(\mathbf{k}(t'))dt'\right) \tag{14}$$

The *density* of probability $p_f(t)$ for a flight to have duration t is simply

$$p_f(t) = P_f(t)\lambda(\mathbf{k}(t)) \tag{15}$$

The density of probability is useful to calculate random distributions of flight times, as necessary in a Monte Carlo simulation. It is not easy to generate directly a distribution of flights according to the given density of probability on a computer. However, it is possible to relate the wanted random distribution to a uniform (pseudo–) random distribution of numbers between 0 and 1, available in most computers through an internal function. The computer generated distribution is pseudo–random because it is actually deterministic and periodic, although the period of the sequence is normally very long. A uniform distribution of random numbers r has a density of probability $p(r) = 1$. The uniform and the wanted random distribution satisfy the integral relation

$$\int_0^t p_f(t')dt' = \int_0^r p(r')dr' = r \tag{16}$$

The integration on the left hand side of (16) is performed using (15)

$$r = 1 - \exp\left(-\int_0^t \lambda(\mathbf{k}(t'))dt'\right) \tag{17}$$

Since r and $(1-r)$ have identical distributions, (17) may be rewritten as

$$-\ln r = \int_0^t \lambda(\mathbf{k}(t'))dt' \tag{18}$$

This is the formula used to calculate the time of flight t from a selected uniform random number r. The integral on the right hand side is trivial only if $\lambda(\mathbf{k}(t))$ is constant, but in general the integral cannot be performed analytically. Since the choice of random flights must be performed over and over again in a Monte Carlo simulation, the solution of (18) is of paramount importance to obtain an efficient implementation of the algorithm. A number of methods have been introduced to solve the integral in (18)

(a) Constant Γ Technique

If the total scattering rate is $\lambda(k(t)) = \Gamma$, where Γ is a constant, then (18) trivially yields

$$-\ln r = \Gamma t \Rightarrow t = -\frac{\ln r}{\Gamma} \tag{19}$$

A virtual scattering process λ_{ss} called *self–scattering* is introduced, such that

$$\lambda(k(t)) + \lambda_{ss} = \Gamma \tag{20}$$

Whenever the self–scattering is selected as the collision mechanism, nothing happens to the particle, which maintains after the scattering the same energy and momentum it had before. Γ is fixed as the largest scattering rate possible in the simulation, to be sure that λ_{ss} is never negative.

The self–scattering is obviously a mathematical trick, with no physical meaning, which does not alter the statistical distribution of the real scattering events. The main advantage of the technique is that the programming becomes very simple. The main disadvantage is that a lot of computer time is spent performing computations related to the self–scattering. Since the total scattering rate is maintained artificially high, the free flights will be much shorter than in reality. It is true that very simple instructions are required to treat the self–scattering event, because the parameters of the particle motion are unchanged, but remember that for every additional collision the computer needs to call twice more the function generating random numbers, the first time to select the scattering mechanism, the second time to restart the free flight time selection procedure. A function call is normally quite expensive.

If the constant Γ technique is adopted, from 75 to 95 % of the computations might be spent just dealing with self–scattering. The method has been popular since the 1970's, since it was seen as the only viable technique for the computer resources of the time (which was an erroneous assumption. Even a straight numerical integration of the right hand side of (18) may be proven to be much more efficient that the constant Γ technique). In large scale simulations the constant Γ technique should be avoided.

(b) Piecewise Γ

The total scattering rates may vary enormously over the necessary energy range. In most applications, the vast majority of simulated particles have an energy associated to low scattering rates, and if the maximum possible scattering rate is taken as a reference for the self–scattering, too much computer time is wasted. The piecewise Γ method divides the energy range into intervals, and the maximum scattering rate within becomes the reference for the self–scattering. Since every energy interval has a different Γ, flights will tend to be longer for the particles whose energy is associated with a smaller Γ, according to (19), which is still used.

The piecewise Γ technique, when the energy intervals are chosen with care, may be highly efficient. A drawback is the need to renormalize the times of flight when Γ changes value within a flight. As an example consider an initial momentum $\mathbf{k}_o = (k_{ox}, k_{oy}, k_{oz})$, in an energy interval with Γ_o. At a value of the momentum magnitude $|\mathbf{k}_1|$ the self–scattering reference becomes Γ_1, and for the given electric field E (constant for simplicity) and the selected random number r, the final momentum magnitude is larger than $|\mathbf{k}_1|$. If $t = 0$ is considered as the initial flight time, we have

$$-\ln r = \Gamma_o t_1 + \Gamma_1 (t - t_1) \tag{21}$$

where t_1 is the time at which Γ changes. The time t_1 is calculated from

$$|\mathbf{k}_1| = \left| \mathbf{k}_o \pm \frac{q\mathbf{E}t_1}{\hbar} \right| \tag{22}$$

with the plus sign for holes and the minus sign for electrons.

The piecewise Γ technique reduces considerably the amount of self–scattering. However, to be efficient, many intervals with different Γ may be required which makes the checking procedure for Γ changes during the flight somewhat cumbersome.

(c) Iterative Γ

The value of Γ is changed at each free flight when using this procedure. If the initial crystal momentum is $\mathbf{k}_o$, an initial value $\Gamma_o = \lambda(\mathbf{k}_o)$ is set, which gives a flight time

$$t_o = -\frac{\ln r}{\Gamma_o} \tag{23}$$

The flight is accepted if the total scattering rate during the flight is lower than Γ_o. Otherwise, a new value $m\Gamma_o$ is selected (a typical value of m is 1.1). If necessary, one keeps multiplying by m, until a satisfactory value of Γ is obtained. The technique is very efficient and fast. However, in some conditions, it may yield unphysical flight time statistics, since the same flight time may be obtained with different pairs of r and Γ. This may induce the computer to generate peaks in the distribution of flights for some particular flight times. Therefore, the technique should be avoided.

(d) Constant Time Technique

The simulation time is subdivided in tiny time intervals, and the maximum scattering rate in the interval is taken as a constant Γ within. Contributions for each constant time interval are added until the equality in (18) is verified

$$-\ln r = \Gamma_o \tau_o + \sum_i \Gamma_i \tau + \Gamma_n \tau_n \tag{24}$$

When a scattering occurs, normally within the constant time interval τ, the final state of the particle is calculated and a new random number is generated to restart the flight. The first portion of the new flight goes from the instant of scattering until the end of the current constant time interval. Here, a new Γ is selected, unless a self–scattering has occurred, in which case the same Γ of the last portion of the previous flight must be used.

The constant time technique is very efficient and solves the integral on the right hand side of (18) exactly in a statistical sense, with very small self–scattering contribution, often negligible. In many applications, it is possible to use the constant time interval τ as a unit for the flight time, so that the scatterings always happen at the end of the interval. If τ is carefully selected, the error can be made very small. In the case of an ensemble Monte Carlo device simulation, the constant time technique

is the method of choice. Since the flights progress synchronously in small increments, programming becomes much easier and a number of costly checking procedures necessary for bookkeeping when dynamics calculations are done for the whole flight, may be avoided. The technique is also very effective for vectorization, since the particles are naturally kept synchronous.

A good reference related to flight time generation is in the paper by Yorston.

3 Flight Dynamics

In a practical implementation of a device Monte Carlo simulation, a grid is generated over the domain and the value of the electric field in the center of every mesh is taken as a constant field value to trace the particle motion inside the mesh. When the particle crosses the boundaries between meshes during a free flight, the flight is subdivided into portions associated to the different fields. Momentum and position are updated using classical equations of motion. For instance, in a uniform field we have

$$\mathbf{k}(t) = \mathbf{k}_o \pm \frac{q\Delta t\mathbf{E}}{\hbar} \tag{25}$$

$$\mathbf{r}(t) = \mathbf{r}_o + \frac{\hbar\mathbf{k}_o\Delta t}{m^*} \pm \frac{q\Delta^2 t\mathbf{E}}{2m^*} \tag{26}$$

As seen in the introduction, the effective mass is a function of energy if the band is not parabolic and this should be taken into account when computing (26. The simplest solution is achieved when using the constant time technique for the free flights, described in section 4.2. At the beginning of each subflight the mass can be recalculated taking into account the energy and the momentum component parallel to the field (using (3.28) for instance), and the mass is then kept constant for the duration of the subflight r.

The distribution of the electric field may be obtained by solving Poisson's equation. The grid used for Poisson's equation may conveniently be the same set up for the Monte Carlo dynamics, for the potential, with the electric field defined on the interleaved mesh (which yields the wanted values on the center of the meshes, often called *cells*). Poisson's equation requires the knowledge of the charge density on the grid points, and this information must be obtained from the distribution of discrete particles inside the device. The choice of a particular grid charge assignment procedure generates always an error in the computed fields. The exact procedure for the field solution would involve the superposition of the potential of the single particles, added to the potential due to the boundary conditions. However, this procedure is *most expensive* and for sure impossible to apply when many thousands of particles are simulated.

The simplest procedure assigns a particle to the closest mesh point. Since a potential mesh point is the midpoint of an interleaved mesh (cell), the number of particles inside an interleaved cell is actually counted and the average charge obtained is assigned to the center point. This procedure is usually called *Nearest Grid Method* (NGM).

A more sophisticated technique, called *Cloud in Cell* (CIC) method, assigns part of the charge of a given particle to a given number of closest grid points (3,4,5, ... points). An excellent source of information is in the book by Hockney and Eastwood.

4 Scattering Mechanisms

The knowledge of scattering mechanisms is essential for the Monte Carlo simulation, since they control the nature of the carrier transport. We have seen already that the total momentum or

energy dependent scattering rates are used to determine the length of the free flights. At the end of the flights the appropriate scattering mechanism must be chosen, to determine the final energy or equivalently the modulus of the momentum, and the scattering angle (which determines the orientation of the momentum after the scattering).

The electron–phonon interactions are essentially inelastic, since a phonon of definite energy is absorbed or emitted by the scattered particle. The most important scattering mechanisms are

- Acoustic phonon
- Polar optical phonon (not present in covalent semiconductors)
- Non–polar optical phonon (for covalent materials and L–valley of $GaAs$)
- Equivalent intervalley phonon
- Non–equivalent intervalley phonon
- Piezoelectric phonon (for very low temperatures)

Other important scattering mechanisms are

- Ionized impurity scattering
- Carrier–carrier scattering (electron–electron, electron–hole, hole–hole scattering. Important for high carrier concentrations)
- Plasmon scattering (due to collective oscillations of the carrier gas)
- Alloy scattering (Important in ternary alloys, e.g. $Al_xGa_{1-x}As$)

The scattering rates of the individual mechanisms are normally stored in tables for the random selection of the scattering type after a free flight.

In the Born approximation, the scattering processes only consist of a transition between two definite momentum states for the carrier involved. One has to make the assumption that the crystal has to be separated into the electron (hole) of interest and the rest of the crystal, the so–called *quasi–particle approximation*. The *transition probability* per unit time for scattering from a state k to a state k', due to a collision represented by a perturbing Hamiltonian H', is given in first order by the *Fermi Golden Rule*

$$S(\mathbf{k},\mathbf{k}') = \frac{2\pi}{\hbar}|\left\langle \mathbf{k}|H'|\mathbf{k}'\right\rangle|^2\delta(E'-E) \tag{27}$$

where $\langle \mathbf{k}|H'|\mathbf{k}'\rangle$ is the matrix element of the transition, in essence an inner product involving the wavefunctions of the initial and final state, and the δ–function expresses the conservation of energy, E and E' being the initial and final energy of the entire gas of electrons and phonons. For transition due to a phonon with wavevector q and frequency ω_q, it is simply

$$E'-E = E(\mathbf{k}')-E(\mathbf{k})\pm\hbar\omega_q \tag{28}$$

$$\mathbf{k}-\mathbf{k}'\pm q = \begin{cases} 0 \\ \mathbf{R} \quad (Umklapp\ process) \end{cases} \tag{29}$$

where $\mathbf{R}$ is a reciprocal latice vector. The transition rate gives the probability per unit time to scatter from an initial state $\mathbf{k}$ into a particular final state $\mathbf{k}'$. The *scattering rate* is the probability to go out of the initial state $\mathbf{k}$ into *any* of the available final states $\mathbf{k}'$. The scattering rate is therefore simply

$$\lambda(\mathbf{k}) = \sum_{k'} S(\mathbf{k}, \mathbf{k}') \tag{30}$$

The summation may be transformed into an integral, and in 3–D we have

$$\lambda(\mathbf{k}) = \frac{V}{8\pi^3} \int dk' \int d\phi \int d\theta S(\mathbf{k}, \mathbf{k}') k'^2 \sin\theta \tag{31}$$

where V is the volume of the crystal, and polar coordinates have been used. The squared matrix element can usually be factored (non–Umklapp processes) as

$$|\langle \mathbf{k}|H'|\mathbf{k}'\rangle|^2 = C_o V(\Delta\mathbf{k}) G(\mathbf{k}, \mathbf{k}') \tag{32}$$

where C_o is a constant, $V(\Delta\mathbf{k})$ contains the $\mathbf{k}$–dependence of the Fourier transform of the interaction potential, and $G(\mathbf{k}, \mathbf{k}')$ is the overlap integral

$$G(\mathbf{k}, \mathbf{k}') = \left| \int u_k(r) u_{k'}(r) dr \right|^2 \tag{33}$$

between the periodic part of the Bloch wavefunctions of the initial and final states. The overlap integral is exactly equal to one for plane waves (bulk) and pure s–state wave functions (parabolic band, i.e. one–band model). When the non–parabolicity of the bands is taken into account, the overlap integral is always less than one and is usually expressed as a function of the non–parabolicity coefficients. The model introduced by Kane approximates the overlap integral as

$$G(\mathbf{k}, \mathbf{k}') = (a_k a_{k'} + c_k c_{k'} \cos\beta) \tag{34}$$

where β is the angle between $\mathbf{k}$ and $\mathbf{k}'$, and

$$a_k = \left[\frac{1 + \alpha E(\mathbf{k})}{1 + 2\alpha E(\mathbf{k})} \right]^{1/2} \tag{35}$$

$$b_k = \left[\frac{\alpha E(\mathbf{k})}{1 + 2\alpha E(\mathbf{k})} \right]^{1/2} \tag{36}$$

A complete detail of the scattering rate calculation is beyond our scope and appropriate references should be consulted.

At the end of the free flight, two random selections must be performed:

1. type of scattering mechanism
2. scattering angle

For the selection of the scattering mechanism, one has to consider in detail the rates for the individual scattering mechanisms $\lambda_1, \lambda_2, \ldots, \lambda_n$, evaluated using the particle energy (or momentum) at the end of the free flight. Calling $\lambda(t_{sc})$ the total scattering rate at the time of scattering, the scattering mechanism is selected by generating a uniform random number $r \in [0, 1]$ and considering the following relations

$$
\begin{array}{lcl}
r < \frac{\lambda_1}{\lambda(t_{sc})} & \Rightarrow & \text{Scattering mechanism 1 is selected} \\
r < \frac{\lambda_1+\lambda_2}{\lambda(t_{sc})} & \Rightarrow & \text{Scattering mechanism 2 is selected} \\
\vdots & & \\
r < \frac{\sum_i^m \lambda_i}{\lambda(t_{sc})} & \Rightarrow & \text{Scattering mechanism m is selected}
\end{array}
$$

In the above procedure, the scattering rates are normalized using the maximum scattering rate. This is particularly efficient when a definite value of the maximum rate is defined throughout the simulation, as in the case of the constant Γ and the piecewise Γ techniques. A table for each normalized partial sum of scattering rates may be calculated and stored at the beginning of the simulation. If the value of $\lambda(t_{sc})$ may actually vary, as in the case of the constant time and iterative Γ techniques, it is better to store the unnormalized partial sums and multiply the random number by $\lambda(t_{sc})$, rather than divide each time all scattering rates by the applicable value of $\lambda(t_{sc})$.

To determine the final state after a scattering, one has first of all to determine the final energy $E(k')$ from

$$E(\mathbf{k}') = E(\mathbf{k}) \pm \hbar\omega_q \pm \Delta E_c \tag{37}$$

where $E(\mathbf{k})$ is the energy at the end of the free flight, $\hbar\omega_q$ is the phonon energy (plus for absorption, minus for emission), and ΔE_c is non zero for intervalley scattering (plus for transition to lower valley, minus for transition to upper valley). The final energy yields directly the absolute value of the momentum, and at this point the random *direction* of the momentum must be determined.

In the simple case, the direction of the momentum after scattering is totally random. This is a good approximation for acoustic scattering (in a parabolic band) and intervalley scattering. It is instructive to analyze the procedure used to determine a totally random direction. Consider a sphere with radius 1, centered at the space location where the scattering occurs. In polar coordinates, two angles need to be determined, in order to identify every point on the sphere (each of the possible momentum directions are obviously identified by a point on the sphere). It is very convenient to align the momentum $\mathbf{k}(t_{sc})$ before scattering along the z–direction so that θ is the deflection angle, and ψ the azimuthal angle.

The probability to select a point P on an element dS of the surface is equal to $\frac{dS}{4\pi}$. Since in spherical coordinates $dS = \sin\theta d\theta d\psi$, we have also

$$\frac{dS}{4\pi} = \sin\theta d\theta d\psi = p(\theta,\psi) d\theta d\psi \tag{38}$$

where $p(\theta,\psi) = \frac{\sin\theta}{4\pi}$ is the density of probability. The separate densities of probability $p(\theta)$ and $p(\psi)$ for the two angles are independent. In fact, we have

$$p(\theta) = \int_0^{2\pi} p(\theta,\psi) d\psi = \frac{\sin\theta}{2} \tag{39}$$

$$p(\psi) = \int_0^{\pi} p(\theta,\psi) d\theta = \frac{1}{2\pi} \tag{40}$$

and $p(\theta,\psi) = p(\theta)p(\psi)$, which proves the independence of θ and ψ. The angles can be selected using uniform random numbers as

$$r_1 = \int_0^{\psi} p(\psi') d\psi' = \frac{\psi}{2\pi} \tag{41}$$

$$r_2 \quad = \quad \int_0^{\theta} p(\theta')d\theta' = \frac{1-\cos\theta}{2} \tag{42}$$

The deflection angle θ is not totally random when the *transition rate* depends on θ. The transition rate itself is proportional to the probability distribution of θ and it is formally convenient to express such probability as a function of $\cos\theta$. For relatively simple angular dependencies, it is possible to invert the integral analogous to the one in (42), to obtain an explicit relationship between $\cos\theta$ and the random number. This is possible, for instance, for ionized impurity scattering and acoustic scattering. The probability distribution of $\cos\theta$ for polar optical scattering may be too difficult to invert explicitly (although some approximations may allow one to do so). One can apply the *rejection technique* to obtain a statistically correct distribution of $\cos\theta$ during the simulation.

Consider the probability $P(\cos\theta) \in [0, P_{max}$ and $\cos\theta \in [-1, 1]$. To apply the rejection technique, two random numbers are selected, $r_1 \in [-1, 1]$ and $r_2 \in [0, P_{max}]$. r_1 represents the randomly selected $\cos\theta$. If $P(r_1) \geq r_2$, the angle is accepted, but if $P(r_1) < r_2$ the angle is rejected. In this way the wanted angular distribution is well reproduced over a large number of events. For examples consult the papers by Fawcett *et al*. and by Jacoboni and Reggiani.

Some Useful References

E. O. Kane, "Band structure of Indium Antimonide," *Journal of the Physics and Chemistry of Solids*, vol. 1, pp. 249–261, 1957.

D. L. Rode, "Low–field electron transport," *Semiconductors and Semimetals*, ed. R. K. Willardson and A. C. Beer, vol. 10, Ch. 1, pp. 1–89, New York: Academic Press, 1975.

R. W. Hockney and J. W. Eastwood, *Computer Simulations Using Particles*, Ch. 10, McGraw–Hill, 1981.

R. M. Yorston, "Free–flight time generation in the Monte Carlo simulation of carrier transport in semiconductors," *Journal of Computational Physics*, vol. 64, pp. 177–194, 1986.

W. Fawcett, A. D. Boardman, and S. Swain, "Monte Carlo determination of electron transport properties in Gallium Arsenide," *Journal of the Physics and Chemistry of Solids*, vol. 31, pp. 1963–1990, 1970.

C. Jacoboni and L. Reggiani, "The Monte Carlo method for the solution of charge transport in semiconductors with application to covalent materials," *Reviews of Modern Physics*, vol. 55, pp. 645–705, 1983.

C. Jacoboni and L. Reggiani, "Bulk hot–electron properties of cubic semiconductors," *Advanced in Physics*, vol. 28, pp. 493–553.

C. Jacoboni and P.Lugli, *The Monte Carlo Method for Semiconductor Device Simulation*, Springer-Verlag, Vienna, 1989.

K. Hess, ed., *Monte Carlo Device Simulation: Full Band and Beyond*, Kluwer Academic Publishers, 1991.