

VILNIAUS UNIVERSITETAS
Kietojo kūno elektronikos katedra
Krūvio pernašos vyksmų skaitinis modeliavimas

Darbas Nr. 4

Elektronų sklaidos puslaidininkyje modeliavimas Monte Karlo metodu

Parengė A. Poškus

2015-01-05

Turinys

| | |
|--|-----------|
| Darbo tikslas | 2 |
| 1. Užduotys | 2 |
| 2. Kontroliniai klausimai | 3 |
| 3. Puslaidininkinių įtaisų skaitinio modeliavimo teorinis pagrindas..... | 5 |
| 3.1. Energijos slėniai. Atvirkštinė gardelė | 5 |
| 3.2. Fononai | 8 |
| 4. Krūvininkų sklaida | 9 |
| 4.1. Kvantinė krūvininkų sklaidos teorija | 9 |
| 4.2. Joninė priemaišinė sklaida | 11 |
| 4.2.1. Debajaus ekranavimo nuotolis | 11 |
| 4.2.2. Joninės priemaišinės sklaidos spartos apskaičiavimas | 12 |
| 4.2.3. Atsitiktinio dydžio (pvz., sklaidos kampo) verčių generavimas, kai yra žinomas jo skirstinys | 13 |
| 4.3. Fononinė sklaida | 15 |
| 4.3.1. Nepolinė sklaida akustiniais fononais | 15 |
| 4.3.2. Kampinių koordinatų reikšmių generavimas, kai sklaida yra izotropinė | 18 |
| 4.3.3. Nepolinė sklaida optiniais fononais | 18 |
| 4.3.4. Tarpslėnė sklaida | 20 |
| 4.3.5. Polinė sklaida optiniais fononais | 22 |
| 5. Viendalelis Monte Karlo metodas..... | 24 |
| 5.1. Viendalelės Monte Karlo programos struktūra | 24 |
| 5.2. Lėkio trukmės skaičiavimas. Dreifo proceso modeliavimas | 25 |
| 5.3. Sklaidos proceso modeliavimas | 26 |
| 5.4. Viendalelio Monte Karlo metodo taikymo pavyzdys: GaAs neigiamojo diferencialinio laidumo modeliavimas | 28 |
| 6. Darbo metodika | 30 |
| Priedas A. Viendalelė Monte Karlo programa MonteCarlo3.dll..... | 38 |
| A.1. Bendros pastabos apie viendalelio Monte Karlo metodo programos sudarymą | 38 |
| A.2. Viendalelės Monte Karlo programos MonteCarlo3.dll struktūra | 40 |
| A.3. Pagrindinių paprogramių antraštės | 44 |
| Priedas B. Bendros žinios apie Monte Karlo modeliavimo laboratorinių darbų atlikimo tvarką .. | 49 |
| Priedas C. Programavimo C kalba pradmenys | 52 |
| C.1. Duomenų tipai | 52 |
| C.2. Operacijos | 54 |
| C.3. Operatoriai | 55 |
| C.4. Kai kurios matematinės funkcijos | 57 |

Darbo tikslas

1. Parašyti C programas, kurios atitinka įvairius elektrono judėjimo puslaidininkyje Monte Karlo modeliavimo etapus. 2. I – V variantai: naudojant parašytą programą, apskaičiuoti tam tikros rūšies sklaidos spartos priklausomybę nuo elektrono energijos. VI – IX variantai: apskaičiuoti elektrono dreifo greičio galio arsenide priklausomybę nuo elektrinio lauko stiprio.

1. Užduotys

I – V variantai – sklaidos spartų skaičiavimas

I – V variantuose, parašius ir sukompiliavus atitinkamą programą, programos SimFit.exe pagrindiniame lange reikia pasirinkti pirmąjį modelį (modelio pavadinimas „Elektrono sklaidos spartų skaičiavimas“). Kiekviename iš šių darbų skaičiuojama tam tikros sklaidos spartos priklausomybė nuo elektrono energijos.

I variantas

*Joninės priemaišinės sklaidos spartos skaičiavimas (paprogramė **rate_ion**)*

Šią paprogramę iškviečia paprogramė **rates** (žr. 27b pav.). Paprogramės **rate_ion** aprašymas pateiktas Priedo A skyrelyje A.3. Joninės priemaišinės sklaidos spartos išraiška pateikta 4.2.2 skyriuje ((4.2.15) formulė).

II variantas

*Akustinės fononinės sklaidos spartos skaičiavimas (paprogramė **rate_acoustic**)*

Šią paprogramę iškviečia paprogramė **rates** (žr. 27b pav.). Paprogramės **rate_acoustic** aprašymas pateiktas Priedo A skyrelyje A.3. Akustinės fononinės sklaidos spartos išraiška pateikta 4.3.1 skyriuje ((4.3.13) formulė).

III variantas

*Optinės nepolinės netarpslėnės sklaidos spartos skaičiavimas (paprogramė **rate_opt_intra**)*

Šią paprogramę iškviečia paprogramė **rates** (žr. 27b pav.). Paprogramės **rate_opt_intra** aprašymas pateiktas Priedo A skyrelyje A.3. Optinės nepolinės netarpslėnės sklaidos spartos išraiška pateikta 4.3.3 skyriuje ((4.3.18) formulė).

IV variantas

*Optinės nepolinės tarpslėnės sklaidos spartos skaičiavimas (paprogramė **rate_opt_inter**)*

Šią paprogramę iškviečia paprogramė **rates** (žr. 27b pav.). Paprogramės **rate_opt_inter** aprašymas pateiktas Priedo A skyrelyje A.3. Optinės nepolinės tarpslėnės sklaidos spartos išraiška pateikta 4.3.4 skyriuje ((4.3.22) formulė).

V variantas

*Optinės polinės (netarpslėnės) sklaidos spartos skaičiavimas (paprogramė **rate_opt_pol**)*

Šią paprogramę iškviečia paprogramė **rates** (žr. 27b pav.). Paprogramės **rate_opt_pol** aprašymas pateiktas Priedo A skyrelyje A.3. Polinės sklaidos optiniais fononais spartos išraiška pateikta 4.3.5 skyriuje ((4.3.30) formulė).

VI – IX variantai – viendalelio Monte Karlo metodo atskirų etapų programavimas

VI – IX variantuose, parašius ir sukompiliavus atitinkamą programą, programos SimFit.exe pagrindiniame lange reikia pasirinkti antrąjį modelį (modelio pavadinimas „Dreifo modeliavimas viendaleliu Monte Karlo metodu“). Kiekviename iš šių variantų skaičiuojama ta pati funkcija: elektrono dreifo greičio galio arsenide priklausomybė nuo elektrinio lauko stiprio.

VI variantas

*Elektrono būsenos po sklaidos skaičiavimas (paprogramė **finalState**)*

Paprogramės **finalState** aprašymas pateiktas Priedo A skyrelyje A.3 (t.p. žr. 27a pav.). Elektrono būsenos po sklaidos skaičiavimo metodika aprašyta 5.3 skyriuje.

VII variantas

*Kiekvieną sklaidos tipą ir kiekvieną pradinį slėnį atitinkančių elektrono energijos pokyčių ir galutinių slėnių nustatymas (paprogramė **scatterTypes**)*

Paprogramės **scatterTypes** aprašymas pateiktas Priedo A skyrelyje A.3 (t.p. žr. 27b pav.). Sklaidos procesų (tipų) ir slėnių numeravimo taisyklės yra aprašytos Priedo A skyrelyje A.1.

VIII variantas

*Sklaidos tipo pasirinkimas (paprogramė **scatter**) ir elektrono laisvojo lėkio trukmės generavimas (paprogramė **flighttime**)*

Paprogramių **scatter** ir **flighttime** aprašymai pateikti Priedo A skyrelyje A.3 (t.p. žr. 27a pav.). Lėkio trukmės generavimo metodika aprašyta 5.2 skyriuje, o sklaidos tipo pasirinkimo metodika aprašyta 5.3 skyriuje.

IX variantas

*Elektrono bangos vektoriaus komponentų laisvojo lėkio pabaigoje skaičiavimas (paprogramė **drift**) ir elektrono energijos skaičiavimas pagal jo bangos vektorius (paprogramė **energy**)*

Paprogramių **drift** ir **energy** aprašymai pateikti Priedo A skyrelyje A.3 (t.p. žr. 27a pav.). Elektrono banginio vektoriaus pokyčio laisvo lėkio metu skaičiavimo metodika aprašyta 5.2 skyriuje, o lygtis energijos atžvilgiu yra (3.1.5).

2. Kontroliniai klausimai

Kontroliniai klausimai, kurie yra bendri visiems variantams:

1. Kas yra atvirkštinė gardelė? Kaip sudaroma pirmoji Brijueno zona?
2. Kas yra energijos slėniai? Kokia yra energijos slėnių padėtis k erdvėje? Neparaboliškumo koeficiento sąvoka.
3. Fermio ir Dirako pasiskirstymo funkcija. Būsenų energinio tankio sąvoka.
4. Kas yra fononai? Fononų klasifikacija. Fononų dispersijos dėsnio bendrasis pavidalas. Bozė ir Einšteino pasiskirstymo funkcija.
5. Sklaidos sąvoka. Bendroji sklaidos iš būsenos k į būseną k' tikimybės per laiko vienetą išraiška (pirmajame trikdymų teorijos artinyje). Delta funkcijos pavidalo daugiklio fizikinė prasmė.
6. Pilnutinės sklaidos spartos $W(k)$ apskaičiavimas, kai yra žinoma sklaidos iš būsenos k į būseną k' sparta.

I varianto kontroliniai klausimai:

7. Kodėl taškinio krūvio elektrostatinio lauko potencialas puslaidininkyje skiriasi nuo potencialo idealiam dielektrike? Paaiškinkite Debajaus ekranavimo nuotolio fizikinę prasmę.
8. Joninės priemaišinės sklaidos spartos išraiška (be išvedimo). Bendrasis šios sklaidos spartos priklausomybės nuo elektrono energijos ir nuo priemaišinių jonų koncentracijos pavidalas, jo aiškinimas.
9. Paaiškinkite joninės priemaišinės sklaidos spartos kitimo kryptį didėjant priemaišinių jonų koncentracijai. Kodėl, didėjant jonų koncentracijai, mažėja elektrono impulso relaksacijos trukmė?

II variantu kontroliniai klausimai:

10. Trikdžio potencialo apibrėžimas akustinės fononinės sklaidos atveju. Akustinės fononinės sklaidos spartos išraiška. Bendrasis šios sklaidos spartos priklausomybės nuo elektrono energijos ir nuo temperatūros pavidalas, jo aiškinimas.
11. Energijos ir impulso tvermės dėsniai fononinės sklaidos atveju. Kodėl akustinę fononinę sklaidą galima laikyti tampriąja sklaida? Apibrėžto dažnio akustinių fononų skaičiaus $n(\omega)$ išraiška.

III variantu kontroliniai klausimai:

12. Trikdžio potencialo apibrėžimas nepolinės optinės fononinės sklaidos atveju. Optinės fononinės sklaidos spartos išraiška.
13. Energijos ir impulso tvermės dėsniai fononinės sklaidos atveju. Kodėl optinę fononinę sklaidą yra netamprioji? Apibrėžto dažnio optinių fononų skaičiaus išraiška.

IV variantu kontroliniai klausimai:

14. Tarpslėnės sklaidos sąvoka. Trikdžio potencialo apibrėžimas nepolinės optinės tarpslėnės fononinės sklaidos atveju.
15. Energijos tvermės dėsnis tarpslėnės sklaidos atveju.
16. Tarpslėnės sklaidos spartos išraiška. Bendrasis šios sklaidos spartos priklausomybės nuo elektrono energijos pavidalas, jo aiškinimas.
17. Paaiškinkite galio arsenido neigiamojo diferencialinio laidumo reiškinį.

V variantu kontroliniai klausimai:

18. Trikdžio potencialo apibrėžimas polinės optinės fononinės sklaidos atveju.
19. Polinės optinės fononinės sklaidos išraiška parabolinių energijos juostų atveju.

VI – IX variantų kontroliniai klausimai:

20. Pagrindiniai elektrono judėjimo modeliavimo Monte Karlo metodu žingsniai. Bendroji elektrono dreifo greičio apskaičiavimo metodika, kai yra taikomas viendalelis Monte Karlo metodas.
21. Elektrono lėkio trukmės generavimo metodika. Virtualiosios sklaidos sąvoka. Elektrono banginio vektoriaus pokytis laisvojo lėkio metu.
22. Sklaidos mechanizmo pasirinkimo metodika.
23. Atsitikinio dydžio verčių generavimas, kai yra žinomas jo skirstinys. Taikymo pavyzdys: polinio ir azimutinio sklaidos kampų θ ir ϕ generavimo metodika, kai sklaida yra izotropinė.
24. Koordinatų transformacijos, kuriose atliekama po kiekvieno sklaidos įvykio, paskirtis ir apibrėžimas.
25. Paaiškinkite galio arsenido neigiamojo diferencialinio laidumo reiškinį.

Literatūra:

1. **Tomizawa K.** Numerical Simulation of Submicron Semiconductor Devices. – Boston, London: Artech House, 1993. – 342 p.
2. **Moglestue C.** Monte Carlo Simulation of Semiconductor Devices. – London: Chapman & Hall, 1993. – 326 p.
3. **Dargys A., Kundrotas J.** Handbook on physical properties of Ge, Si, GaAs and InP. – Vilnius: Science and Encyclopedia Publishers, 1994. – 264 p.

3. Puslaidininkinių įtaisų skaitinio modeliavimo teorinis pagrindas

3.1. Energijos slėniai. Atvirkštinė gardelė

Kaip žinoma iš kietojo kūno elektronikos kurso, idealiojo kristalo vienelektroninės Šrėdingerio lygties

$$\hat{H}^{(0)}\Psi = -\frac{\hbar}{i} \frac{\partial \Psi}{\partial t} \quad (3.1.1)$$

pilnoji ortogonalioji sprendinių sistema – tai Blocho bangos

$$\Psi_k(\mathbf{r}, t) = \psi_k(\mathbf{r}) \exp(-iE_k t / \hbar), \quad (3.1.2a)$$

$$\psi_k(\mathbf{r}) = u_k(\mathbf{r}) \exp(i(\mathbf{k} \cdot \mathbf{r})), \quad (3.1.2b)$$

kur $u_k(\mathbf{r})$ yra periodinė spindulio vektoriaus \mathbf{r} funkcija, kurios periodiškumas yra toks pats, kaip kristalo gardelės periodiškumas, \mathbf{k} yra elektrono bangos vektorius, o E_k yra elektrono energija. Elektrono judėjimo būseną nusako Blocho bangos vektorius \mathbf{k} . Indeksas „ \mathbf{k} “ energijos žymenyje „ E_k “ nurodo, kad elektrono energija yra bangos vektoriaus funkcija. Visuose pagrindiniuose puslaidininkiuose (Ge, Si, GaAs ir InP) elektronų energijos priklausomybė nuo \mathbf{k} yra sudėtinga daugiareikšmė funkcija, kuri turi kelis minimumus ir maksimumus. Šios funkcijos verčių (energijų) visuma, kuri atitinka laisvąsias elektrono būsenas, sudaro vadinamąją **laidumo juostą**, o energijų visuma, kuri atitinka kovalentiniuose cheminiuose ryšiuose dalyvaujančių elektronų būsenas, sudaro **valentinę juostą**. Skirtingų energijos lygmenų užpildą kiekvienoje iš šių juostų nusako vadinamoji **Fermio ir Dirako funkcija**:

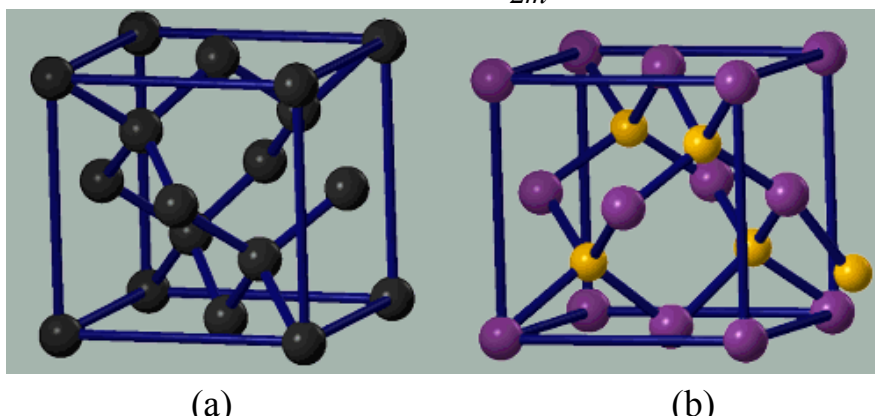
$$f(E) = \frac{1}{1 + \exp\left(\frac{E - E_F}{k_B T}\right)}; \quad (3.1.3)$$

čia E_F yra Fermio energija. Akivaizdu, kad energijos lygmenų užpildą sparčiai mažėja didėjant energijai E . Todėl dažnai galima laikyti, kad visi elektronai yra susitelkę laidumo juostos minimumuose, o visos skylės yra susitelkusios valentinės juostos maksimumuose. Tokiais atvejais pakanka žinoti tik funkcijos E_k pavidalą laidumo juostos minimumų aplinkoje ir valentinės juostos maksimumų aplinkoje. Laidumo juostos minimumai ir valentinės juostos maksimumai vadinami **energijos slėniais**. Žinome, kad ekstremumo aplinkoje bet kuri funkcija yra apytiksliai kvadratinė (t. y. proporcinga argumento nuokrypio nuo minimumo taško kvadratui). Izotropinio slėnio atveju tai reiškia, kad energijos slėnyje izoenerginiai paviršiai yra apytiksliai sferiniai. Tada krūvininko kinetinės energijos priklausomybė nuo bangos vektoriaus yra parabolinė:

$$E_k = \frac{\hbar^2 k^2}{2m^*}; \quad (3.1.4)$$

čia m^* yra krūvininko efektinė masė ekstremumo aplinkoje, o bangos vektorius k atskaitomas nuo ekstremumo taško. Tačiau dažnai tokia aproksimacija yra nepriimtina. Tada izotropinio energijos slėnio neparaboliškumą nusako **neparaboliškumo koeficientas** α :

$$E_k(1 + \alpha E_k) = \frac{\hbar^2 k^2}{2m^*} \equiv \gamma(E_k). \quad (3.1.5)$$



1 pav. Si ir Ge (a) bei GaAs ir InP (b) kristalų gardelių elementarieji narveliai

Energijos slėnių padėtys k erdvėje yra susijusios su kristalo simetrija. Pagrindinių puslaidininkių (Si, Ge, GaAs ir InP) Bravė gardelė yra kubinė paviršiuje centruota, o jos bazę sudaro du atomai. Vienas atomas yra gardelės mazge, o kitas pasislinkęs pirmojo atžvilgiu išilgai narvelio pagrindinės įstrižainės $1/4$ jos ilgio (žr. 1 pav.). Bet kuriai kristalo gardelei k erdvėje galima sudaryti vadinamąją **atvirkštinę gardelę**. Atvirkštinės gardelės primityvūs narvelis sudaromas šitaip. Jeigu tiesioginės gardelės primityviojo narvelio briaunas nusako vektoriai a_1 , a_2 ir a_3 , tada atvirkštinės gardelės primityviojo narvelio briaunas nusako vektoriai

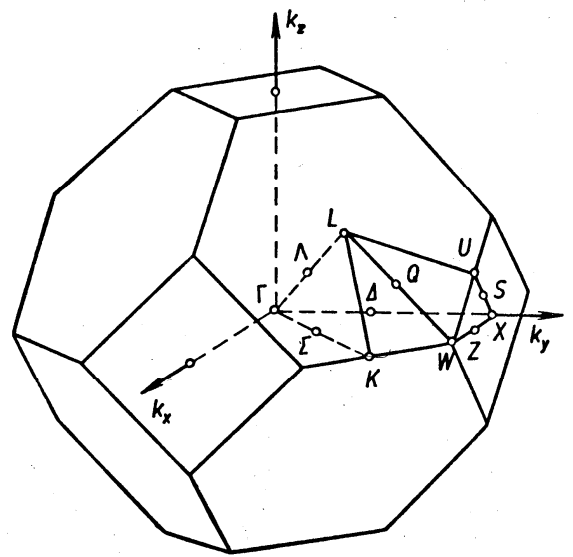
$$b_1 = \frac{2\pi(a_2 \times a_3)}{V_0}, \quad b_2 = \frac{2\pi(a_3 \times a_1)}{V_0}, \quad b_3 = \frac{2\pi(a_1 \times a_2)}{V_0}; \quad (3.1.6)$$

čia V_0 yra tiesioginės gardelės primityviojo narvelio tūris, o laužtiniai skliaustai žymi vektorinę sandaugą. Atvirkštinės gardelės mazgai yra išsidėstę taškuose, kuriuos atitinka šie bangos vektoriai:

$$g = \sum_{i=1}^3 g_i b_i; \quad (3.1.7)$$

čia g_i ($i = 1, 2, 3$) yra bet kokie sveikieji skaičiai. Jeigu tiesioginė gardelė yra kubinė (kaip Ge, Si, GaAs ir InP atvejais), tada atvirkštinė gardelė taip pat yra kubinė. Aukščiau apibrėžtieji vektoriai g – tai atvirkštinės gardelės transliacijos vektoriai (**transliacijos vektoriai** – tai vektorius, kuriuo lygiagrečiai paslinkus begalinę gardelę, galutinė gardelės padėtis yra neatskiriama nuo pradinės). k erdvės taškai, kurie skiriasi atvirkštinės gardelės transliacijos vektoriais, yra lygiaverčiai. Todėl visos fiziškai skirtingos elektrono būsenos yra sutelktos atvirkštinės gardelės pirmojoje Brijueno zonoje. Pirmoji Brijueno zona sudaroma, sujungus atvirkštinės gardelės mazgą $g = 0$ su artimiausiais mazgais ir per kiekvienos tokiu būdu gautos atkarpos vidurį nubrėžus jai statmeną plokštumą. Pirmoji Brijueno zona – tai šių plokštumų gaubiama erdvės sritis. Jos pavidalas Ge, Si, GaAs ir InP atveju yra toks, kaip pavaizduota 2 pav. Be to, 2 pav. pavaizduotos pirmosios Brijueno zonos simetrijos ašys: ketvirtosios eilės $[100]$ (2 pav. ši ašis pažymėta Δ), antrosios eilės ašis $[110]$ (Σ) ir trečiosios eilės ašis $[111]$ (Λ). Brijueno zonos centras ($k = 0$) žymimas Γ . Taškai, kuriuose ašys Δ , Σ ir Λ kerta Brijueno zonos paviršių žymimi atitinkamai X , K ir L (žr. 2 pav.). Atstumas nuo taško Γ iki taško X lygus $2\pi/a$, kur a yra kristalo (tiesioginės) gardelės konstanta, t. y. 1 pav. pavaizduotojo Bravė narvelio briaunos ilgis.

Energijos slėniai yra taškuose Γ , L ir X . Žinyuose dažniausiai pateikiama elektrono energijos priklausomybė nuo banginio vektoriaus absoliutinės vertės, kai Blocho bangos sklaidimo kryptis yra lygiagreti kuriai nors iš dviejų aukščiausios eilės simetrijos ašių: Δ ir Λ . Pvz., 3 pav. pavaizduoti energijos slėniai GaAs atveju. Viršutinės kreivės (su minimumu taške Γ) atitinka laidumo juostą, o apatinės kreivės (su maksimumu taške Γ) – valentinę juostą. Kaip matome, ir laidumo, ir valentinė juosta turi keletą šakų. T. y. laidumo juostoje egzistuoja kelių rūšių elektronai, o valentinėje juostoje egzistuoja kelių rūšių skylės. Tačiau apytiksliuose praktiniuose skaičiavimuose galima kai kurių šakų nepaisyti arba skirtingas šakas apjungti į vieną. Pvz., jeigu skirtingų laidumo juostos šakų energijos skirtumas yra žymiai didesnis už kT , tada, pagal Fermio ir Dirako pasiskirstymo funkciją (3.1.3), didžioji dauguma elektronų užims kvantines būsenas, kurios atitinka žemesniąją šaką, todėl aukštesniųjų šakų galima nepaisyti. Pvz., modeliuojant laisvojo elektrono judėjimą GaAs, dažnai pakanka atsižvelgti tik į mažiausios energijos laidumo juostos šaką, kurios slėniai 3 pav. pažymėti Γ_6 , L_1 ir X_1 .



2 pav. Ge, Si, GaAs ir InP kristalų gardelių pirmoji Brijueno zona

Skaičiuojant kvantinių šuolių iš pradinės būsenos į daugelį galimų galutinių būsenų tikimybes, reikia skaičiuoti integralus galutinių būsenų atžvilgiu. Integruoti įprasta energijos atžvilgiu. Tokių integralų pointegralinėje funkcijoje yra daugiklis, kuris vadinamas **būsenų energiniu tankiu**. Tai yra kvantinių būsenų skaičius, kuris tenka vienetiniam energijos intervalui ir vienetiniam tūriui. Apskaičiuosime būsenų tankio priklausomybę nuo krūvininko energijos E_k . Kvantinėje mechanikoje įrodoma, kad laisvosios dalelės būsenų tankis konfigūracinėje erdvėje yra $1/h^3$; čia h yra Planko konstanta ($h = 2\pi\hbar$). [Čia turimos omenyje tik *erdvinio judėjimo* būsenos, t. y. nepaisoma dalelių sukinio.] Taigi, būsenų skaičius konfigūracinės erdvės tūrio elemente $d^3r d^3p$ yra lygus $d^3r d^3p/h^3$. Kadangi $p = \hbar k$, tai būsenų skaičius spindulių vektorių ir bangos vektorių erdvės tūrio elemente $d^3r d^3k$ yra $d^3r d^3k/(2\pi)^3$. Vadinasi, būsenų skaičius kristalo tūrio vienetė ir bangos vektorių erdvės tūrio elemente d^3k yra $d^3k/(2\pi)^3$. Kadangi energija priklauso tik nuo bangos vektoriaus modulio (bangos skaičiaus) k , tai naudosime sferinio sluoksnio pavidalo tūrio elementą. Jeigu to sluoksnio spindulys yra k , o storis dk , tada jo tūris yra $d^3k = 4\pi k^2 dk$. Būsenų energinis tankis $N(E_k)$ yra lygus būsenų skaičius tame tūrio elemente ir atitinkamo energijos pokyčio dE_k santykiui:

$$N(E_k) \equiv \frac{1}{(2\pi)^3} 4\pi \cdot [k(E_k)]^2 \frac{dk}{dE_k}. \quad (3.1.8)$$

Taigi, skaičiuojant integralą visų galimų kvantinių būsenų atžvilgiu, integravimo kintamojo pakeitimas „ $k \rightarrow E_k$ “ atliekamas pagal tokią taisyklę:

$$k^2 dk \rightarrow 2\pi^2 N(E_k) dE_k. \quad (3.1.9)$$

Išreiškę bangos skaičių k iš (3.1.4) arba (3.1.5) ir įrašę į (3.1.8), gauname, kad parabolinės juostos atveju būsenų tankio priklausomybė nuo energijos yra tokio pavidalo:

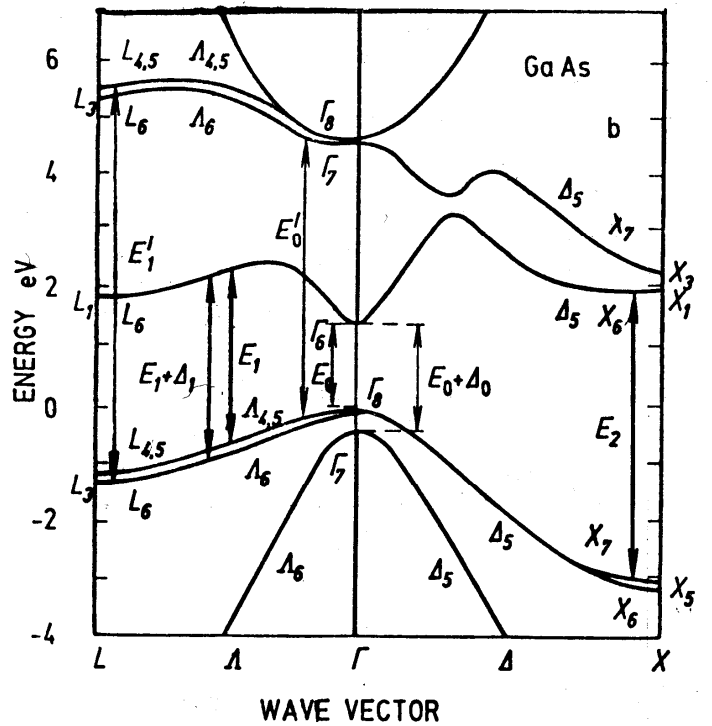
$$N(E_k) = \frac{(2m^*)^{3/2} \sqrt{E_k}}{4\pi^2 \hbar^3}, \quad (3.1.10)$$

o neparabolinės juostos atveju reikia naudoti bendresnę formulę:

$$N(E_k) = \frac{(2m^*)^{3/2}}{4\pi^2 \hbar^3} \sqrt{\gamma(E_k)} \frac{d\gamma(E_k)}{dE_k} = \frac{(2m^*)^{3/2}}{4\pi^2 \hbar^3} (1 + 2\alpha E_k) \sqrt{E_k(1 + \alpha E_k)}, \quad (3.1.11)$$

kur funkcija $\gamma(E_k)$ apibrėžiama (3.1.5) sąryšiu (parabolinės juostos atveju $\gamma(E_k) \equiv E_k$, todėl formulė (3.1.11) virsta formule (3.1.10)). Reikia turėti omenyje, kad (3.1.10) ir (3.1.11) formulės nusako tik *apibrėžtos sukinio projekcijos* kvantinių būsenų tankį. Pilnutinis būsenų skaičius yra du kartus didesnis, nes kiekviename energijos lygmenyje gali būti du elektronai, kurių sukinų projekcijos (orientacijos) yra priešingos. (3.1.10) ir (3.1.11) reiškinuose nėra daugiklio 2, nes vėliau tos formulės bus naudojamos, skaičiuojant sklaidos spartą, o sklaidos metu elektrono sukinio orientacija nesikeičia.

Šalia juostų neparaboliškumo, reikia atsižvelgti ir į juostų anizotropiją. Paraboliniame anizotropiniame energijos slėnyje izoenerginiai paviršiai yra elipsoido formos. Tai reiškia, kad elektrono efektinę masę nusako trys vertės, kurių kiekviena atitinka vieną iš trijų elipsoido pagrindinių ašių: m_x^* , m_y^* ir m_z^* . Tačiau dažnai tokį elipsoidinį minimumą galima formaliai pakeisti



3 pav. GaAs energijos juostų schema

sferiniu (izotropiniu) minimumu, kurio aplinkoje elektrono efektinė masė lygi **būsenų tankio efektinei masei** $m_d^* = (m_x^* m_y^* m_z^*)^{1/3}$. Atitinkamai, energijos išraiškose (3.1.4) ir (3.1.5) bei būsenų tankio išraiškose (3.1.10) ir (3.1.11) vietoj m^* reikia naudoti būsenų tankio efektinę masę.

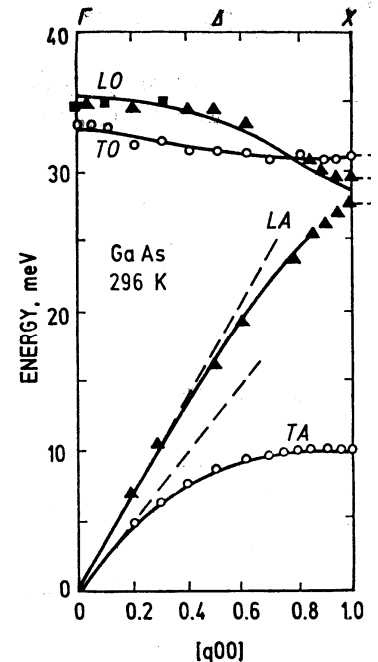
3.2. Fononai

Bet kokią N atomų sistemos virpamąjį judėjimą galima išreikšti $3N$ **normaliuųjų virpesių** (kitaip vadinamų **normaliosiom modom**) suma. Kiekvieną normalųjį virpesį galima laikyti atomų poslinkio plokščiąja banga, kuri sklinda kristalu, ir aprašyti kaip kvantinę kvazidalelę – **fononą**. Fononai, kaip ir kitos kvazidalelės, yra apibūdinami bangos vektoriumi \mathbf{k} , dažniu ω ir energija. Fonono energija yra lygi jo dažnio ir Planko konstantos sandaugai: $\hbar\omega$. Fononai klasifikuojami pagal atomų virpesių kryptį atžvilgiu bangos judėjimo krypties (išilginiai arba skersiniai) ir pagal tai, ar šių virpesių metu elementariojo narvelio masės centras juda (akustiniai fononai), ar nejuda (optiniai fononai). Aišku, kad optiniai fononai egzistuoja tik kristaluose, kurių Bravė narvelyje yra daugiau negu vienas atomas. Pagrindinių puslaidininkių (Si, Ge, GaAs ir InP) Bravė narvelyje yra 8 atomai (žr. 1 pav.). Fonono dažnio priklausomybė nuo jo bangos vektoriaus \mathbf{k} vadinama **dispersijos dėsniu**. Fononų dispersijos dėsnis yra daugiareikšmė funkcija, t. y. jis turi keletą **šakų**. Kaip ir elektrono atveju, visos fiziškai skirtingos fonono bangos vektoriaus vertės yra sutelktos atvirkštinės gardelės pirmojoje Brijeno zonoje, kuri pavaizduota 2 pav. Žinyuose fononų dispersijos dėsnis dažniausiai pateikiamas trim grafikais, kurie nusako fononų energijos priklausomybę nuo banginio vektoriaus absoliutinės vertės, kai bangos sklidimo kryptis yra lygiagreti kuriai nors iš trijų Brijeno zonos simetrijos ašių (2 pav. šios ašys pažymėtos Δ , Σ ir Λ). Pažymėjus atomų skaičių Bravė narvelyje raide p , šakų skaičius bendruoju atveju yra lygus $3p$. Iš jų 3 atitinka akustinius fononus (2 atitinka skersinius, 1 – išilginius), o likusios $3p - 3$ atitinka optinius fononus ($2p - 2$ atitinka skersinius, $p - 1$ – išilginius). Tačiau dažniausiai dėl kristalo simetrijos kai kurios šakos sutampa arba yra "uždraustos". Pvz., aukščiau minėtųjų puslaidininkių dispersijos dėsnis kryptimi $[100]$ (Δ) turi tik keturias šakas (žr. 4 pav.) – TA (skersiniai akustiniai fononai), LA (išilginiai akustiniai fononai), TO (skersiniai optiniai fononai) ir LO (išilginiai optiniai fononai). Visoms akustinėms šakoms būdinga tai, kad taške $k = 0$ fonono energija lygi nuliui, o kai k žymiai mažesnis už atvirkštinės gardelės konstantą, fonono energija proporcinga k (žr. 4 pav.). Visoms optinėms šakoms būdinga tai, kad fonono energija didžiausia taške $k = 0$, o energijos priklausomybė nuo k yra silpnesnė, negu akstinių šakų atveju (žr. 4 pav.). Todėl, skaitmeniškai modeliuojant sklaidą optiniais fononais, dažniausiai laikoma, kad optinio fonono energija yra konstanta, t. y. nepriklauso nuo k (tačiau skirtingų rūšių optinių fononų energijos gali būti skirtingos).

Apibrėžtos rūšies ir apibrėžto bangos vektoriaus fononų vidutinį skaičių nusako **Bozė ir Einšteino pasiskirstymo funkcija**:

$$n(\omega) = \frac{1}{\exp\left(\frac{\hbar\omega}{k_B T}\right) - 1}; \quad (3.2.1)$$

čia ω yra fononų dažnis, o $n(\omega)$ yra duotosios rūšies fononų, kurių dažnis ω , skaičius.



4 pav. Fononų dispersijos dėsnis GaAs kristale kryptimi $[100]$. Ant X ašies atidėtas dydis $ka/(2\pi)$; čia k yra bangos skaičius, o a yra gardelės konstanta

4. Krūvininkų sklaida

4.1. Kvantinė krūvininkų sklaidos teorija

Idealiajame kristale elektrono stacionariausias būsenas nusako Blocho bangos (3.1.2). Tačiau realiųjų kristalų vienelektroninis hamiltonianas \hat{H} skiriasi nuo idealiojo kristalo elektrono hamiltoniano $\hat{H}^{(0)}$. Šis skirtumas vadinamas **trikdžiu**. Hamiltoniano trikdį toliau žymėsime raide \hat{H}' :

$$\hat{H} = \hat{H}^{(0)} + \hat{H}'. \quad (4.1.1)$$

Trikdžio \hat{H}' priežastis gali būti priemaišų atomai, kristalo gardelės virpesiai, elektronų tarpusavio sąveika, išorinis spinduliavimas ir kt. Pasikeitus hamiltonianui, pasikeičia ir jo tikrinės funkcijos, t. y. elektronų banginės funkcijos. Taigi, Blocho bangos, kuriomis aprašoma elektrono būsena, jau nėra *tikslūs* elektrono Šrėdingerio lygties sprendiniai. Tačiau, kadangi Blocho bangos (3.1.2) sudaro pilnąją ortogonaliją sistemą, tai tiksliąją banginę funkciją $\Psi(\mathbf{r}, t)$ galima išreikšti Blocho bangų $\Psi_k(\mathbf{r}, t)$ tiesiniu dariniu:

$$\Psi(\mathbf{r}, t) = \sum_{V_k} c_k(t) \Psi_k(\mathbf{r}, t). \quad (4.1.2a)$$

Čia indeksas “ V_k ” nurodo, kad sumuojama atžvilgiu visų skirtingų kvantinių būsenų. Koeficientų $c_k(t)$ priklausomybė nuo laiko reiškia, kad elektronas, kurio pradinė būsena buvo aprašoma Blocho banga $\Psi_k(\mathbf{r}, t)$, po tam tikro laiko t atsiduria būsenoje, kurią aprašo Blocho bangų su įvairiais \mathbf{k}' darinys:

$$\Psi(\mathbf{r}, t) = \sum_{V_{k'}} c_{k \rightarrow k'}(t) \Psi_{k'}(\mathbf{r}, t). \quad (4.1.2b)$$

Kitais žodžiais, egzistuoja tam tikra tikimybė, kad per laiką t įvyks elektrono kvantinis šuolis iš pradinės būsenos \mathbf{k} į kažkurią iš daugelio galimų galutinių būsenų \mathbf{k}' . Šie kvantiniai šuoliai vadinami elektronų **sklaida**. Turint omenyje banginės funkcijos statistinį aiškinimą ir remiantis formule (4.1.2b), galima teigti, kad šuolio $\mathbf{k} \rightarrow \mathbf{k}'$ tikimybė per laiką t lygi $|c_{k \rightarrow k'}(t)|^2$. Šuolio tikimybė *per laiko vienetą* lygi išvestinei $\left. \frac{d}{dt} |c_{k \rightarrow k'}(t)|^2 \right|_{t=0}$, kurią žymėsime $S(\mathbf{k}, \mathbf{k}')$.

Trikdžių teorija suformuluoja koeficientų $c_{k \rightarrow k'}(t)$ bei jų laikinių išvestinių $S(\mathbf{k}, \mathbf{k}')$ apytikslio skaičiavimo būdą, laikant, kad hamiltoniano trikdys \hat{H}' yra pakankamai “mažas”, ir kartu suformuluoja šio “mažumo” kriterijų. Pagal trikdžių teoriją, koeficientai $c_{k \rightarrow k'}(t)$ skaičiuojami nuosekliųjų artinių metodu. Jeigu trikdys yra harmoninė dažnio funkcija (pvz., $\sim \sin(\omega t)$), tada jį galima išreikšti dviejų tokio pavidalo trikdžių suma:

$$\hat{H}'(\mathbf{r}, t) = U(\mathbf{r}) \exp(\pm i \omega t) \quad (4.1.3)$$

Tokio trikdžio atveju pirmajame trikdžių teorijos artinyje gaunama, kad šuolio iš būsenos \mathbf{k} į būseną \mathbf{k}' tikimybė per laiko vienetą lygi [1, p. 29]

$$S(\mathbf{k}, \mathbf{k}') = \frac{2\pi}{\hbar} |\langle \mathbf{k}' | U | \mathbf{k} \rangle|^2 \delta(E_{k'} - E_k \pm \hbar \omega); \quad (4.1.4)$$

čia $\langle \mathbf{k}' | U | \mathbf{k} \rangle$ yra trikdžio amplitudės $U(\mathbf{r})$ matricos elementas:

$$\langle \mathbf{k}' | U | \mathbf{k} \rangle = \int_V \psi_{k'}^*(\mathbf{r}) U(\mathbf{r}) \psi_k(\mathbf{r}) d^3 \mathbf{r} \quad (4.1.5)$$

(čia integruojama kristalo tūriu V). Šuolio tikimybės išraiškoje (4.1.4) matome, kad yra įmanomi tik šuoliai į būsenas, kurių energija tenkina sąlygą

$$E_{k'} = E_k \mp \hbar \omega \quad (4.1.6)$$

(čia viršutinis ženklas atitinka viršutinį ženklą (4.1.3–4) formulėse). Tai yra energijos tvermės dėsnis: nors elektrono energija pasikeičia, tačiau *pilnutinė* energija, į kurią dar įeina ir trikdžių šaltinio (pvz., elektromagnetinės spinduliuotės arba gardelės virpesių) energija, turi likti nepakitusi.

Todėl elektrono energijos pokytis ($E_{k'} - E_k$) yra priešingas trikdžių šaltinio energijos pokyčiui ($\pm \hbar \omega$).

Skaidos sparta (skaidos tikimybė per laiko vienetą) gaunama, sudėjus šuolių iš duotosios pradinės būsenos \mathbf{k} į visas įmanomas galutines būsenas \mathbf{k}' tikimybes (4.1.4). Vadinasi, skaidos sparta yra lygi

$$W(\mathbf{k}) = \frac{V}{(2\pi)^3} \int S(\mathbf{k}, \mathbf{k}') d^3 \mathbf{k}'. \quad (4.1.12)$$

Šį integralą patogiau skaičiuoti sferinėse koordinatėse (k, θ, ϕ). Kampai θ ir ϕ apibrėžti 5 pav. Kampas θ vadinamas **poliniu kampu**, o kampas ϕ – **azimutiniu kampu**. Kampo θ galimosios vertės yra nuo 0 iki π , o kampo ϕ galimosios vertės yra nuo 0 iki 2π . \mathbf{k} erdvės tūrio elementas sferinėse koordinatėse išreiškiamas šitaip:

$$d^3 \mathbf{k}' = k'^2 \sin \theta dk' d\theta d\phi, \quad (4.1.13)$$

o (4.1.12) integralas išreiškiamas šitaip:

$$W(\mathbf{k}) = \frac{V}{(2\pi)^3} \int_0^{2\pi} d\phi \int_0^\pi d\theta \sin \theta \int_0^\infty dk' \cdot k'^2 S(\mathbf{k}, \mathbf{k}'). \quad (4.1.14)$$

Tačiau elektrono skaidos tikimybė nepriklauso nuo ϕ , todėl integravimas kampo ϕ atžvilgiu pasireiškia tik pastoviu daugikliu 2π :

$$W(\mathbf{k}) = \frac{V}{(2\pi)^2} \int_0^\pi d\theta \sin \theta \int_0^\infty dk' \cdot k'^2 S(\mathbf{k}, \mathbf{k}'). \quad (4.1.15)$$

Irašę (4.1.4) į (4.1.15), gauname

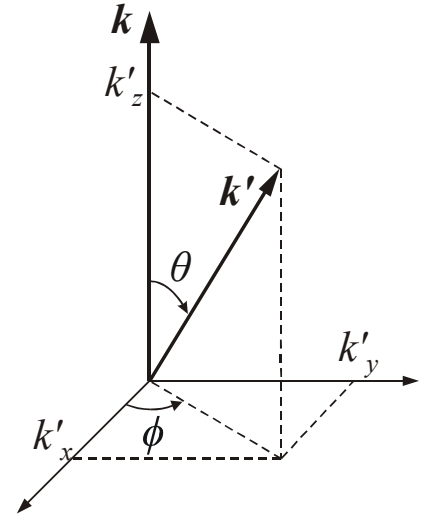
$$W(\mathbf{k}) = \frac{V}{2\pi\hbar} \int_0^\pi d\theta \sin \theta \int_0^\infty dk' \cdot k'^2 |\langle \mathbf{k}' | U | \mathbf{k} \rangle|^2 \delta(E_{k'} - E_k \pm \hbar \omega). \quad (4.1.16)$$

Visas aukščiau užrašytąsias formules galima naudoti ir stacionariojo trikdžio atveju: tada $\omega = 0$. Iš (4.1.6) išplaukia, kad stacionariojo trikdžio atveju elektrono energija nesikeičia; gali pasikeisti tik bangos vektoriaus kryptis. Tokia sklaida vadinama **tampriąja sklaida**.

Nuokrypiai nuo idealaus gardelės periodiškumo (t. y. hamiltoniano trikdžiai) gali būti įvairios prigimtys. Dvi svarbiausios tokių nuokrypių rūšys yra priemaišų atomai ir gardelės virpesiai. Toliau laikysime, kad sklaida gali būti tik šių tipų:

- 1) joninė priemaišinė sklaida (šią sklaidą sąlygoja elektronų kuloninė sąveika su priemaišų jonais);
- 2) nepolinė sklaida akustiniais fononais (šią sklaidą sąlygoja energijos juostų kitimas erdvėje ir laike dėl išilginių akustinių bangų);
- 3) nepolinė sklaida optiniais fononais (šią sklaidą sąlygoja energijos juostų kitimas dėl optinių virpesių; ši sklaida gali būti tarpslėnė);
- 4) polinė sklaida optiniais fononais (šią sklaidą sukelia kintantis elektrinis laukas, kuris atsiranda, kintant gardelės narvelio dipoliniam momentui dėl išilginių optinių virpesių joniniuose puslaidininkuose, pvz., GaAs, InP ir kt.).

Daugumoje praktikoje sutinkamų atvejų pakanka atsižvelgti tik į anksčiau minėtuosius sklaidos tipus [2, p. 80]. Žemiau pateiktas smulkesnis tų sklaidos tipų aprašymas.

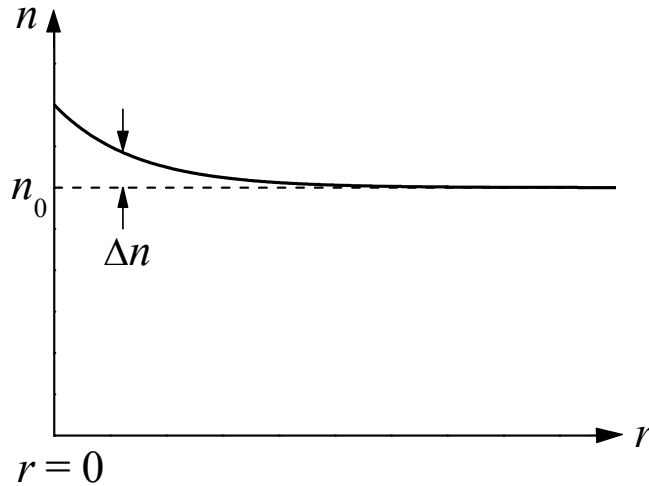


5 pav. Kampų θ ir ϕ prasmė. \mathbf{k} – elektrono bangos vektorius prieš pat sklaidos įvykį, \mathbf{k}' – bangos vektorius po sklaidos

4.2. Joninė priemaišinė sklaida

4.2.1. Debajaus ekranavimo nuotolis

Daugelis puslaidininkinių įtaisų turi specialias sritis, kurios elgiasi kaip krūvininkų rezervuarai. Tų sričių pagrindinė paskirtis – krūvininkų generavimas arba jų pašalinimas dėl rekombinacijos. Tose srityse yra padidinta priemaišų (donorų arba akceptorų) koncentracija, todėl krūvininkai yra stipriai sklaidomi dėl sąveikos su priemaišų jonais. Šiuo atveju vienelektroninio hamiltoniano trikdys yra priemaišos jono elektrostatinio lauko potencialas. Apskaičiuosime šį potencialą.



6 pav. Laisvųjų elektronų tankio padidėjimas aplink teigiamąjį joną

Iš elektrostatikos žinoma, kad taškinio krūvio $+Ze$ potencialas idealioje dielektrikoje yra

$$\varphi(r) = \frac{1}{4\pi\epsilon_0} \frac{Ze}{r}; \quad (4.2.1)$$

čia ϵ_s yra dielektriko dielektrinė skvarba, o r yra atstumas iki krūvininko. Tačiau puslaidininkyje yra laisvųjų krūvininkų, kurie dalinai ekranuoja priemaišos joną. Pvz., aplink teigiamąjį donorų joną padidėja laisvųjų elektronų koncentracija (žr. 6 pav.), todėl to jono efektinis krūvis sumažėja. T. y. aplink donorų joną elektronų koncentracija n tampa didesnė už donorų jonų koncentraciją N_d , todėl atsiranda perteklinis neigiamas krūvis, kurio erdvinis tankis yra lygus $-e\Delta n$; čia $\Delta n \equiv n - N_d$. Vadinasi, pilnutinis erdvinio krūvio tankis (įskaičius dar ir paties teigiamojo jono krūvį) yra

$$\rho(r) = e(Z\delta(r) - \Delta n); \quad (4.2.2)$$

čia Dirako δ funkcija $\delta(r)$ atspindi prielaidą, kad koordinatų pradžios taške $r = 0$ yra taškinis krūvis $+Ze$. Įrašę (4.2.2) į Puasono lygtį

$$\nabla^2 \varphi = -\frac{\rho}{\epsilon_0 \epsilon_s} \quad (4.2.3)$$

ir pasinaudoję operatoriaus ∇^2 (laplasiano) išraiška sferinėmis koordinatėmis, gauname tokią lygtį:

$$\frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{d\varphi}{dr} \right) = -\frac{e}{\epsilon_0 \epsilon_s} (Z\delta(r) - \Delta n(r)). \quad (4.2.4)$$

Aišku, kad šios lygties sprendinys priklauso nuo funkcijos $\Delta n(r)$ pavidalo. Jeigu $\Delta n(r)$ yra pakankamai mažas, tada galima naudoti klasikinę (Bolcmano) pasiskirstymo funkciją, todėl

$$\Delta n = n_0 \exp\left(-\frac{U}{kT}\right) - n_0 = n_0 \exp\left(\frac{e\varphi}{kT}\right) - n_0 \approx \frac{en_0}{kT} \varphi; \quad (4.2.5)$$

čia $U = -e\varphi$ yra elektrono potencinė energija, n_0 yra pusiausviroji elektronų koncentracija toli nuo jono (praktikoje galima laikyti, kad $n_0 \approx N_d$) ir laikoma, kad $|e\varphi| \ll kT$. Įrašę (4.2.5) į (4.2.4), gauname:

$$\frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{d\varphi}{dr} \right) - \frac{1}{\lambda_D^2} \varphi = -\frac{Ze}{\varepsilon_0 \varepsilon_s} \delta(r); \quad (4.2.6)$$

čia λ_D yra vadinamasis **Debajaus ekranavimo nuotolis**:

$$\lambda_D \equiv \sqrt{\frac{\varepsilon_0 \varepsilon_s kT}{e^2 n_0}} \quad (4.2.7)$$

„Ekranavimo nuotolio“ sąvokos prasmė tampa aiški iš (4.2.5) lygties atskirojo sprendinio išraiškos:

$$\varphi(r) = \frac{Ze}{4\pi\varepsilon_0\varepsilon_s r} \exp(-r/\lambda_D) \equiv \frac{Ze}{4\pi\varepsilon_0\varepsilon_s r} \exp(-q_D r). \quad (4.2.8)$$

Čia q_D yra atvirkštinis Dubajaus nuotolis. Taigi, Debajaus ekranavimo nuotolis λ_D – tai atstumas, kuriame priemaišinio jono lauko potencialas sumažėja e kartų, lyginant su to potencialo verte, kuri būtų gauta, jeigu nebūtų ekranavimo. Įrašę į (4.2.7) tipišką kintamųjų vertes ($\varepsilon_s = 11,8$; $kT = 0,0254$ eV; $n_0 = 10^{22} \text{ m}^{-3}$), gauname $\lambda_D = 4 \cdot 10^{-8} \text{ m}$. Puslaidininkio nuskurdintame sluoksnyje laisvųjų krūvininkų yra daug mažiau, todėl Debajaus ekranavimo nuotolis yra daug didesnis.

4.2.2. Joninės priemaišinės sklaidos spartos apskaičiavimas

Dabar galima apskaičiuoti sklaidos, kurią sukelia tokio pavidalo trikdys, spartą. Tuo tikslu įrašome trikdžio išraišką (4.2.8) į trikdžio matricos elemento išraišką (4.1.8):

$$\langle \mathbf{k}' | U | \mathbf{k} \rangle = -\frac{1}{V} \frac{Ze^2}{4\pi\varepsilon_0\varepsilon_s} \int_V e^{-i(\mathbf{k}',r)} \frac{1}{r} e^{-q_D r} e^{i(\mathbf{k},r)} d^3\mathbf{r} = -\frac{1}{V} \frac{Ze^2}{\varepsilon_0\varepsilon_s} \frac{1}{|\mathbf{k}' - \mathbf{k}|^2 + q_D^2}; \quad (4.2.9)$$

čia laikoma, kad juostos yra parabolinės (todėl $\langle \mathbf{k}' | U | \mathbf{k} \rangle = U_{\mathbf{k}' - \mathbf{k}}$). Įrašę (4.2.9) į šuolio $\mathbf{k} \rightarrow \mathbf{k}'$ tikimybės per laiko vienetą išraišką (4.1.4) ir atsižvelgę į tai, kad trikdys (4.2.8) yra stacionarusis ($\omega = 0$), gauname

$$S(\mathbf{k}, \mathbf{k}') = \frac{2\pi}{\hbar} \left(\frac{Ze^2}{V\varepsilon_0\varepsilon_s} \right)^2 \frac{\delta(E_{\mathbf{k}'} - E_{\mathbf{k}})}{(|\mathbf{k}' - \mathbf{k}|^2 + q_D^2)^2}. \quad (4.2.10)$$

Kadangi šiuo atveju elektrono energija sklaidos metu nepakinta, o elektrono bangos skaičius vienareikšmiškai susijęs su energija (žr. (3.1.4)), tai $|\mathbf{k}'| = |\mathbf{k}| = k$ ir

$$|\mathbf{k}' - \mathbf{k}|^2 = 2k^2(1 - \cos\theta). \quad (4.2.11)$$

Be to, kadangi apibrėžto bangos vektoriaus \mathbf{k} elektronas yra banga, kuri pasiskirsčiusi visame kristalo tūryje, tai elektroną gali sklaidyti bet kuris kristale esantis priemaišinis jonas. Vadinas, skaičiuojant pilnutinę sklaidos $\mathbf{k} \rightarrow \mathbf{k}'$ tikimybę, reikia padauginti reiškini (4.2.10) iš visų kristale esančių priemaišinių jonų skaičiaus $N_j V$; čia N_j yra jonų koncentracija. Tokiu būdu gauname:

$$S(\mathbf{k}, \mathbf{k}') = \frac{2\pi}{\hbar} \frac{N_j Z^2 e^4}{V \varepsilon_0^2 \varepsilon_s^2} \frac{\delta(E_{\mathbf{k}'} - E_{\mathbf{k}})}{[2k^2(1 - \cos\theta) + q_D^2]^2}. \quad (4.2.12)$$

Matome, kad, didėjant elektrono bangos skaičiui k (ir energijai), sklaidos sparta mažėja. Taip yra todėl, kad, didėjant k , mažėja pradinės ir galutinės elektrono banginių funkcijų sanklota erdvėje (paprastiau kalbant, tos banginės funkcijos vis labiau skiriasi viena nuo kitos) ir atitinkamai mažėja trikdžio matricos elementas (4.2.9). Pilnutinė sklaidos sparta (įskaitant visas galutines būsenas \mathbf{k}') gaunama įrašius (4.2.12) į (4.1.15):

$$W(\mathbf{k}) = \frac{N_j Z^2 e^4}{2\pi\hbar\varepsilon_0^2\varepsilon_s^2} \int_0^\pi d\theta \cdot \sin\theta \int_0^\infty dk' \cdot k'^2 \frac{\delta(E_{\mathbf{k}'} - E_{\mathbf{k}})}{[2k^2(1 - \cos\theta) + q_D^2]^2}. \quad (4.2.13)$$

Atlikus integravimo kintamojo pakeitimą (3.1.9) ir pasinaudojus Dirako delta funkcijos savybėmis, šis integralas įgyja tokį pavidalą:

$$W(\mathbf{k}) = \frac{\pi N_j Z^2 e^4 N(E_{\mathbf{k}})}{\hbar \varepsilon_0^2 \varepsilon_s^2} \int_0^\pi \frac{\sin\theta d\theta}{[2k^2(1 - \cos\theta) + q_D^2]^2}. \quad (4.2.14)$$

Kadangi $\sin\theta d\theta = -d(\cos\theta)$, tai pastarasis integralas lengvai apskaičiuojamas, atlikus integravimo kintamojo pakeitimą „ $\theta \rightarrow \cos\theta$ “. Tokiu būdu randame:

$$W(\mathbf{k}) = \frac{2\pi N_j Z^2 e^4 N(E_k)}{\hbar \varepsilon_0^2 \varepsilon_s^2} \cdot \frac{1}{q_D^2 (4k^2 + q_D^2)} \quad (4.2.15)$$

Skaidos spartos išraiška (4.2.15) formulė nustoja galioti, kai elektronų koncentracija n_0 yra žymiai mažesnė už jonizuotų priemaišų koncentraciją N_j . Tokia situacija yra galima, pvz., kompensuotame puslaidininkyje, kuriame teigiamųjų priemaišinių jonų (jonizuotų donorų) koncentracija yra apytiksliai lygi neigiamųjų priemaišinių jonų (jonizuotų akceptorų) koncentracijai. Tada efektas toks, lyg elektronai būtų perėję iš donorų į akceptorius, ir laisvųjų elektronų tankis n_0 yra labai mažas. Kai n_0 artėja į nulį, atvirkštinis Debajaus ekranavimo nuotolis q_D (4.2.6) taip pat artėja į nulį, todėl pagal (4.2.15) apskaičiuotoji skaidos sparta artėja į begalybę. Taip yra todėl, kad joninės priemaišinės skaidos spartos išraiška buvo gauta, laikant, kad bet kuriuo laiko momentu elektronas sąveikauja ne daugiau negu su *vienu* jonu. Tačiau kai jonus ekranuojančių elektronų yra mažai, tų jonų elektriniai laukai „persikloja“, todėl minėtoji prielaida nustoja būti teisinga. Siekiant išvengti didelių skaičiavimo klaidų, kai elektronų koncentracija yra maža, galima dirbtinai apriboti joninės skaidos spartą: jeigu pagal (4.2.15) formulę apskaičiuota vertė yra didesnė už 10^{14} s^{-1} , tada tariama, kad joninės skaidos sparta lygi 10^{14} s^{-1} .

7 pav. pavaizduota joninės priemaišinės skaidos spartos GaAs priklausomybė nuo E_k parabolinių juostų atveju (kai būsenų tankį nusako (3.1.10) formulė), laikant, kad $n_0 = N_j$. Gali būti netikėtas tas faktas, kad skaidos sparta mažėja, augant jonų koncentracijai N_j . Tai yra ypač akivaizdu mažų energijų srityje (žr. 7 pav.). Taip yra dėl to, kad, esant mažoms k vertėms, (4.2.15) trupmenos vardiklis yra apytiksliai proporcingas $q_D^4 \sim n_0^2 \approx N_j^2$ (žr. (4.2.6)), o skaitiklis yra proporcingas N_j . Fizikinė šio skaidos spartos mažėjimo priežastis yra ta, kad augant n_0 mažėja Debajaus ekranavimo nuotolis, t. y. mažėja erdvės sritis aplink kiekvieną joną, kurioje pasireiškia jo elektrinis laukas. Tačiau šis skaidos spartos mažėjimas nereiškia, kad, augant priemaišų koncentracijai, auga krūvininkų judris. Mat krūvininkų judris yra proporcingas krūvininkų impulso vidutinei relaksacijos trukmei, o ši priklauso ne vien nuo skaidos spartos, bet ir nuo skaidos krypties skirstinio (t. y. nuo vektorių \mathbf{k}' krypties skirstinio). Išreikšime šį skirstinį.

4.2.3. Atsitiktinio dydžio (pvz., skaidos kampo) verčių generavimas, kai yra žinomas jo skirstinys

Kaip minėta, skaidos kryptį nusako du poliniai kampai θ ir ϕ (žr. 5 pav.). Vadinasi, norint pilnai aprašyti skaidos krypties pasiskirstymą, reikia žinoti du tikimybės tankius: kampo ϕ tikimybės tankį $f_\phi(\phi)$ ir kampo θ tikimybės tankį $f_\theta(\theta)$. Turint tuos tikimybės tankius, galėsime apskaičiuoti skaidos į polinių kampų intervalą $[\theta; \theta + d\theta]$ ir į azimutinių kampų intervalą $[\phi; \phi + d\phi]$ spartas:

$$dW_\theta(\theta) = f_\theta(\theta) d\theta; \quad (4.2.16a)$$

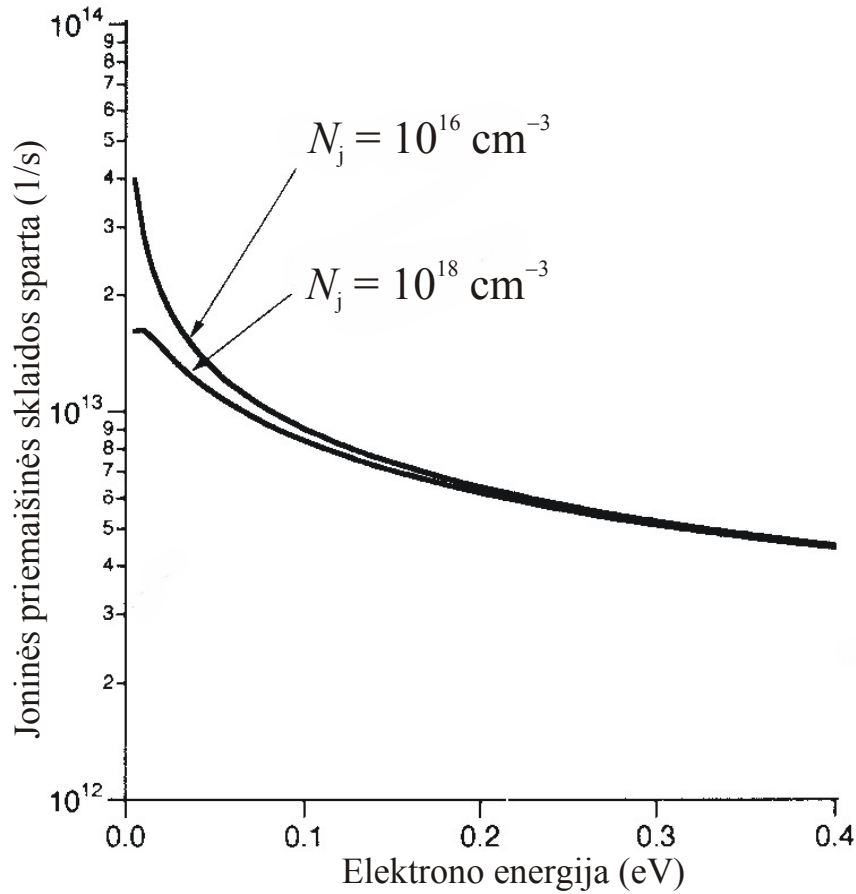
$$dW_\phi(\phi) = f_\phi(\phi) d\phi. \quad (4.2.16b)$$

Dabar suformuluosime bendrąją atsitiktinių skaičių generavimo taisyklę, kuri naudojama Monte Karlo modeliavimo programoje. Tarkime, a yra atsitiktinis dydis (pvz., skaidos kampas), kurio vertę reikia generuoti. Yra žinomas to dydžio tikimybės tankis $f_a(a)$. Apibrėšime to dydžio **pasiskirstymo funkciją** $W_a(a)$: tai yra santykinė dalis įvykių, kuriuose to dydžio vertė yra mažesnė už duotąją vertę a :

$$W_a(a) \equiv \frac{1}{W} \int_{a_{\min}}^a f_a(a) da; \quad (4.2.17a)$$

čia a_{\min} yra mažiausioji įmanoma dydžio a vertė, W yra normavimo daliklis:

$$W \equiv \int_{a_{\min}}^{a_{\max}} f_a(a) da; \quad (4.2.17b)$$



7 pav. Joninės priemaišinės sklaidos spartos priklausomybė nuo elektrono energijos, esant dviem legiravimo laipsniams

čia a_{\max} yra didžiausioji įmanoma dydžio a vertė. Tada dydžio a atsitiktinės vertės turi būti skaičiuojamos pagal lygtį

$$W_a(a) = r; \quad (4.2.18)$$

čia r yra atsitiktinis skaičius, kuris tolygiai pasiskirstęs tarp 0 ir 1. Pritaikysime šią taisyklę kampų θ ir ϕ skaičiavimui. Kadangi visos azimutinio kampo ϕ vertės yra vienodai tikėtinos, šio kampo tikimybės tankis yra konstanta:

$$f_\phi(\phi) = \frac{1}{2\pi}. \quad (4.2.19)$$

Akivaizdu, kad šiuo atveju lygtis (4.2.18) yra tokia:

$$\phi = 2\pi r. \quad (4.2.20)$$

T. y. azimutinio kampo ϕ vertė gaunama, tiesiog padauginus atsitiktinio skaičiaus r , kuris tolygiai pasiskirstęs nuo 0 iki ϕ , reikšmę iš 2π . Polinio kampo θ tikimybės tankis – tai (4.2.14) reiškinio pointegralinis reiškiny (įskaitant ir pastovų daugiklį prieš integralą):

$$f_\theta(\theta) = \frac{\pi N_j Z^2 e^4 N(E_k)}{\hbar \varepsilon_0^2 \varepsilon_s^2} \frac{\sin \theta}{[2k^2(1 - \cos \theta) + q_D^2]^2}. \quad (4.2.21)$$

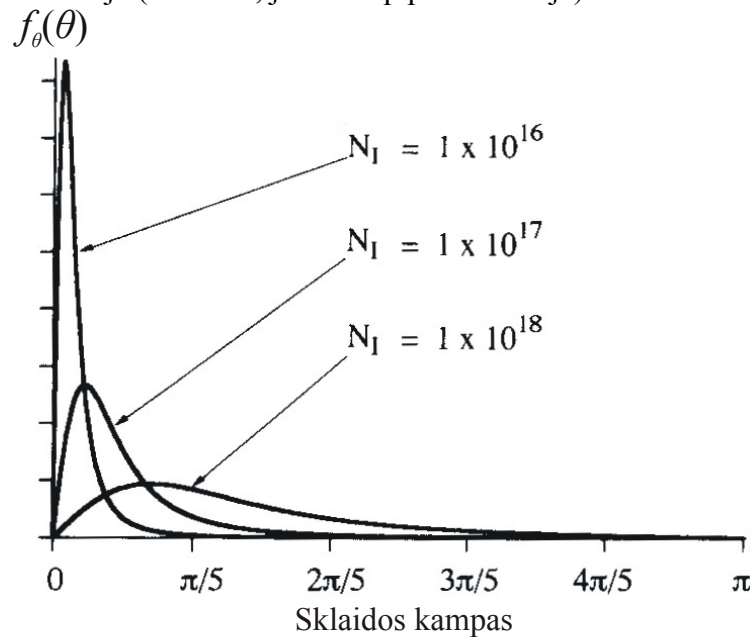
Normavimo daliklis šiuo atveju – tai pilnutinė sklaidos sparta $W(\mathbf{k})$. Vadinas, lygties (4.2.18) kairioji pusė – tai reiškinio (4.2.21) integralas nuo 0 iki θ , padalintas iš $W(\mathbf{k})$. Tokiu būdu gauname:

$$\frac{(1 - \cos \theta) [1 + (2k/q_D)^2]}{2 + (1 - \cos \theta)(2k/q_D)^2} = r; \quad (4.2.22)$$

čia r yra atsitiktinis skaičius, kuris tolygiai pasiskirstęs tarp 0 ir 1 (šis skaičius turi būti generuojamas nepriklausomai nuo atsitiktinio skaičiaus, kuris naudojamas, skaičiuojant kampą ϕ). Lygties (4.2.22) sprendinys yra

$$\cos \theta = 1 - \frac{2r}{1 + (1-r)(2k/q_D)^2}. \quad (4.2.23)$$

Naudojantis polinio tankio tikimybės tankio išraiška (4.2.21), galima paaiškinti, kodėl, augant jonų koncentracijai N_j , judris mažėja, nors sklaidos sparta taip pat mažėja. Taip yra todėl, kad, augant N_j , didėja sklaidos dideliais kampais tikimybė. Tai akivaizdu 8 pav., kuriame pavaizduota funkcija $f_\theta(\theta)$ (čia laikoma, kad $n_0 = N_j$). Matome, kad, esant didesnėms jonų koncentracijoms, kampų θ skirstinys yra platesnis. Vadinasi, nors sklaidos sparta $W(k)$ yra mažesnė, tačiau vidutinis sklaidos įvykių skaičius, kuris reikalingas, kad elektronas „užmirštų“ savo judėjimo istoriją, taip pat yra mažesnis, ir pilnutinis efektas yra tas, kad, padidėjus N_j , vidutinė impulso relaksacijos trukmė sumažėja (vadinasi, judris taip pat sumažėja).



8 pav. Sklaidos kampo θ tikimybės tankis, esant trims legiravimo laipsniams

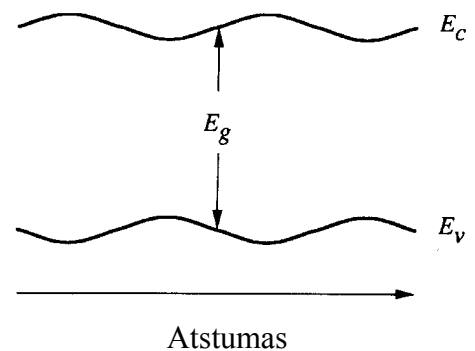
Šiame skyriuje gautosios sklaidos spartos išraiškos galioja tik parabolinių juostų artinyje. Laidumo juostos neparaboliškumas tampa svarbus, tik esant aukštai elektrono energijai. Tada joninės priemaišinės sklaidos vaidmuo yra žymiai mažesnis už kitų rūšių sklaidos vaidmenį [1, p. 235], todėl joninės priemaišinės sklaidos spartos išraiška neparabolinės juostos atveju čia nepateikiama. Tačiau reikia turėti omenyje, kad neparabolinių juostų atveju būsenų tankis $N(E_k)$, kuris įeina į (4.2.15) reiškinį, turi būti skaičiuojamas pagal (3.1.11) formulę.

4.3. Fononinė sklaida

4.3.1. Nėpolinė sklaida akustiniais fononais

Žinome, kad fononai yra kristalo gardelės virpesių kvantai. Gardelės jonų virpėjimas aplink jų pusiausvyros taškus sukelia momentinius energijos juostų pokyčius (žr. 9 pav.), kurie, savo ruožtu, sukelia elektronų sklaidą. Šiame skyriuje aptarsime tik sklaidą, kurią sukelia *nėpoliniai* fononai, t. y. tokie fononai, kurie nėra susiję su kristalo gardelės dipolinio momento pokyčiais.

Elektronų ir fononų sąveika yra vienas iš vyraujančių sklaidos mechanizmų integrinių grandinių komponentuose. Net ir tada, kai temperatūra yra žema (t. y. fononų skaičius yra mažas), egzistuoja didelė savaiminės fononų emisijos tikimybė, nes krūvininkai gali įgyti didelę energiją (dėl stipraus elektrinio lauko) ir paskui dalį tos



9 pav. Energijos juostų kitimas dėl kristalo gardelės deformavimo

energijos prarasti fonono pavidalu.

Kadangi akustinių virpesių metu kaimyninių kristalo gardelės atomai juda viena kryptimi, tai gardelės deformacija atsiranda dėl to, kad kaimyninių narvelių poslinkiai nėra tiksliai vienodi (nors ir labai artimi vienas kitam). T. y. gardelės deformacija yra proporcinga gretimų narvelių atomų poslinkių *skirtumui*. Todėl vienelektroninio hamiltoniano trikdys (sąveikos potencialas), dėl kurio atsiranda sklaida išilginiais akustiniais fononais, yra proporcingas atomų poslinkio iš savo pusiausvyros padėčių *išvestinei* koordinačių atžvilgiu [1, p. 41]:

$$H'(\mathbf{r}, t) = \Xi_d \nabla \cdot \mathbf{u}(\mathbf{r}, t). \quad (4.3.1)$$

Čia $\mathbf{u}(\mathbf{r}, t)$ yra gardelės nuokrypis nuo pusiausvyros padėties taške \mathbf{r} laiko momentu t , o Ξ_d yra proporcingumo koeficientas, kuri vadinama **deformacijos potencialu** (jo matavimo vienetas yra eV). Kuo didesnis deformacijos potencialas, tuo didesnę energijos juostų pokytį sąlygoja duotojo dydžio deformacija. Žymėjimas $\nabla \cdot \mathbf{u}$ reiškia vektoriaus \mathbf{u} divergenciją. Kubinės simetrijos kristaluose elektrono sąveika su akustiniais fononais įmanoma tik tada, kai fononai yra išilginiai. Šalia deformacijos potencialo Ξ_d , nepolinės sklaidos akustiniais fononais tikimybės skaičiavimui reikalingas medžiagos tamprumo koeficientas c_L , kuris lemia išilginių garso bangų greitį v_L [1, p. 41]:

$$\frac{\omega_q}{q} = v_L = \sqrt{\frac{c_L}{\rho}}. \quad (4.3.2)$$

Čia q yra fonono bangos skaičius, ω_q yra fonono dažnis, o ρ yra medžiagos tankis.

Akustinių fononų atveju sklaidos iš būsenos \mathbf{k} į būseną \mathbf{k}' tikimybė per laiko vienetą yra lygi [1, p. 43]

$$S(\mathbf{k}, \mathbf{k}') = \frac{\pi \Xi_d^2 q^2}{\rho \omega_q V} (n_q + \frac{1}{2} \mp \frac{1}{2}) \delta(E_{\mathbf{k}'} - E_{\mathbf{k}} \mp \hbar \omega_q); \quad (4.3.3)$$

čia \mathbf{k}' turi būti toks, kad galiojūt impulso tvermės dėsnis:

$$\mathbf{k}' = \mathbf{k} \pm \mathbf{q}. \quad (4.3.4)$$

\mathbf{q} yra fonono bangos vektorius, o $\hbar \omega_q$ yra fonono energija. (4.3.3) reiškinyje ir tolimesniuose reiškiniuose viršutinis ženklas atitinka fonono sugertį (išnykimą), o apatinis ženklas atitinka fonono emisiją (atsiradimą). Taigi, kiekvienos rūšies fononai sąlygoja du sklaidos vyksmus, kurių tikimybės per laiko vienetą yra skirtingos: fononų sugertis ir fononų emisija. (4.3.3) reiškinyje n_q yra duotosios būsenos fononų vidutinis skaičius (jį nusako Bozė ir Einšteino pasiskirstymo funkcija (3.4.1)). Delta funkcija atspindi energijos tvermės dėsnį: jeigu elektronas sugeria būsenos \mathbf{q} fononą, tada elektrono energija turi padidėti dydžiu $\hbar \omega_q$, o jeigu fononas emituojamas, tada elektrono energija tuo pačiu dydžiu sumažėja:

$$E_{\mathbf{k}'} = E_{\mathbf{k}} \pm \hbar \omega_q. \quad (4.3.5)$$

Parabolinių juostų atveju elektrono energiją galima išreikšti pagal (3.1.4). Įrašę tą išraišką į energijos tvermės dėsnį (4.3.5), gauname:

$$\frac{\hbar^2 k'^2}{2m^*} = \frac{\hbar^2 k^2}{2m^*} \pm \hbar \omega_q. \quad (4.3.6)$$

Įrašę \mathbf{k}' išraišką (4.3.4) į (4.3.6), gauname:

$$\pm \frac{\hbar^2 k q \cos \theta}{m^*} + \frac{\hbar^2 q^2}{2m^*} = \pm \hbar \omega_q. \quad (4.3.7)$$

Iš čia gaunama tokia sklaidos kampo θ kosinuso išraiška:

$$\cos \theta = \frac{1}{2} \left(\mp \frac{q}{k} + \frac{\hbar \omega_q}{E_{\mathbf{k}}} \frac{k}{q} \right). \quad (4.3.8)$$

Kambario temperatūroje akustinių fononų, kurie dalyvauja sklaidoje, energija yra žymiai mažesnė už $k_B T$, todėl elektronų sąveiką su akustiniais fononais galima laikyti tampriąja (t. y. galima laikyti, kad elektrono energija sklaidos metu nekinta), o apibrėžtos būsenos akustinių fononų skaičių galima įvertinti pagal apytikslę formulę $n(\omega) \approx k_B T / \hbar \omega$ (žr. (3.2.1)). Kadangi pastarasis

skaičius yra žymiai didesnis už vieneta, tai $n_q \approx n_q + 1$. Atlikus šiuos pakeitimus ir atsižvelgus į (4.3.2) bei (4.3.7), sklaidos tikimybės išraišką (4.3.3) galima perrašyti šitaip:

$$S(\mathbf{k}, \mathbf{k}') = \frac{\pi \Xi_d^2 k_B T}{\hbar c_L V} \frac{k}{q E_k} \delta\left(\frac{q}{2k} \pm \cos \theta\right); \quad (4.3.9)$$

Be to, prilyginus delta funkcijos argumentą nuliui, gaunamos mažiausioji ir didžiausioji įmanomos q vertės:

$$\begin{aligned} q_{\min} &= 0, \\ q_{\max} &= 2k. \end{aligned} \quad (4.3.10)$$

Pilnutinė akustinės fononinės sklaidos sparta skaičiuojama pagal bendrąją formulę (4.1.12), tačiau šiuo atveju reikia integruoti \mathbf{q} atžvilgiu:

$$W(\mathbf{k}) = \frac{V}{(2\pi)^3} \int S(\mathbf{k}, \mathbf{k}') d^3 \mathbf{k}' = \frac{\Xi_d^2 k_B T}{8\pi^2 \hbar c_L} \frac{k}{E_k} \int \frac{1}{q} \delta\left(\frac{q}{2k} \pm \cos \theta\right) d^3 \mathbf{q}. \quad (4.3.11)$$

Suintegravus, gaunama tokia akustinės fononinės sklaidos spartos išraiška:

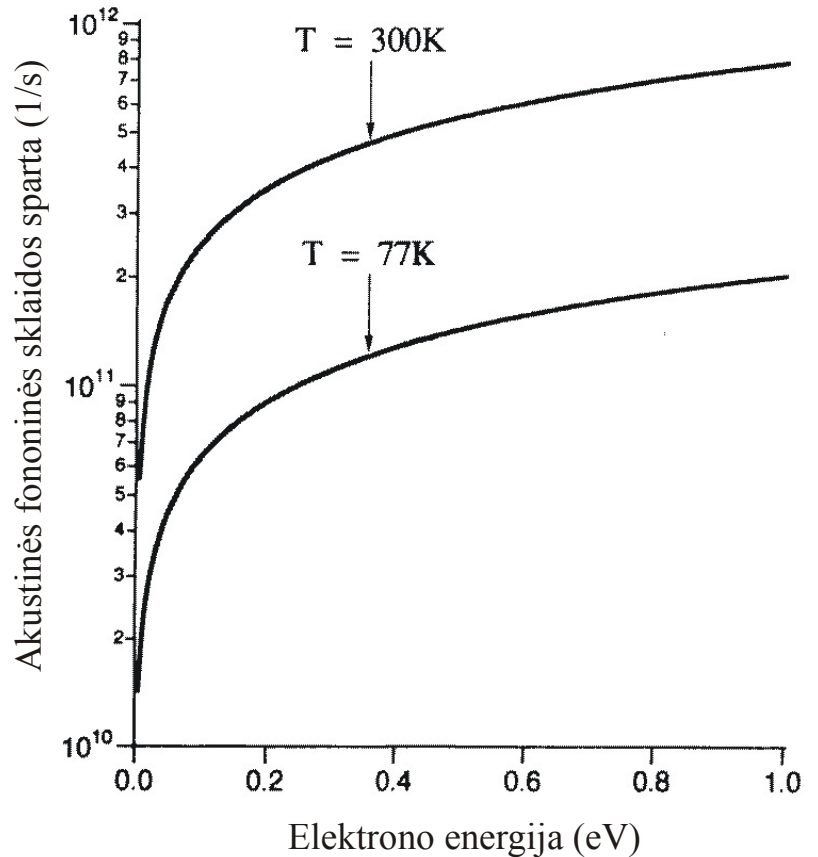
$$W_{\text{par}}(\mathbf{k}) = \frac{2\pi \Xi_d^2 k_B T}{\hbar c_L} N(E_k); \quad (4.3.12)$$

čia indeksas „par“ nusako, kad ši formulė galioja parabolinių juostų atveju. Čia $N(E_k)$ skaičiuojamas pagal (3.1.10) formulę. Neparabolinių juostų atveju [1, p. 236]

$$W(\mathbf{k}) = \frac{2\pi \Xi_d^2 k_B T}{\hbar c_L} N(E_k) \frac{(1 + \alpha E_k)^2 + \frac{1}{3}(\alpha E_k)^2}{(1 + 2\alpha E_k)^2}; \quad (4.3.13)$$

čia α yra neparaboliškumo koeficientas, o būsenų tankis $N(E_k)$ skaičiuojamas pagal (3.1.11) formulę. **Pastaba:** (4.3.13) formulėje ir visose tolesnėse formulėse energija yra atskaitoma nuo slėnio „dugno“, t. y. nuo mažiausios energijos. Bangos vektorius \mathbf{k} taip pat yra atskaitomas nuo vertės, kuri atitinka slėnio dugną.

10 pav. pavaizduota akustinės fononinės sklaidos spartos priklausomybė nuo elektrono energijos GaAs Γ slėnyje, esant dviem temperatūroms. Sklaidos spartos didėjimas, augant elektronų energijai, aiškinamas tuo, kad sklaidos sparta proporcinga būsenų tankiui $N(E_k)$, kuris yra didėjant energijos funkcija. T. y., kuo didesnė elektrono energija, tuo didesnis skaičius būsenų, į kurias jis gali būti išsklaidytas. Todėl ir sklaidos sparta yra didesnė. Sklaidos spartos didėjimas didėjant temperatūrai yra susijęs su tuo, kad didėjant temperatūrai didėja fononų skaičius (ši akustinių fononų skaičiaus didėjimą atspindi daugiklis T , kuris įeina į sklaidos spartos išraiškas (4.3.12) ir (4.3.13)).



10 pav. Akustinės fononinės sklaidos spartos priklausomybė nuo elektrono energijos

4.3.2. Kampinių koordinatų reikšmių generavimas, kai sklaida yra izotropinė

Nepolinė fononinė sklaida yra izotropinė. Tai reiškia, kad, jeigu sklaidos taškas yra sferos centre, tada tikimybė, kad po sklaidos elektronas išlėks pro duotąją tos sferos ploto elementą dS , yra lygi to ploto elemento ir visos sferos ploto santykiui. Naudojant sferines koordinates, sferos ploto elementas išreiškiamas šitaip:

$$dS = r^2 \sin \theta dr d\theta d\phi. \quad (4.3.13a)$$

(žr. 11 pav.). Vadinasi, kai sklaida yra izotropinė, tada sklaidos į tą ploto elementą tikimybė yra lygi

$$dP = \frac{dS}{4\pi r^2} = \frac{1}{4\pi} \sin \theta d\theta d\phi. \quad (4.3.13b)$$

Kampai ϕ ir θ generuojami nepriklausomai vienas nuo kito. Azimutinis sklaidos kampas ϕ generuojamas pagal formulę (4.2.20). Kad išvesti polinio sklaidos kampo θ generavimo formulę, reikia taikyti anksčiau aprašytą bendrą metodą, t. y. visų pirma generuoti atsitiktinį skaičių r , kuris tolygiai pasiskirstęs nuo 0 iki 1, o paskui spręsti (4.2.18) lygtį. Šiuo atveju tos lygties kairiojoje pusėje turi būti funkcija

$$P(\theta) = \int_0^\theta f(\theta') d\theta', \quad (4.3.13b)$$

čia $f(\theta)$ yra kampo θ tikimybės tankis. To tikimybės tankio išraiška gaunama iš (4.3.13b) formulės, taikant bendrąją tikimybės tankio apibrėžtį: tai yra funkcija, kurią padauginus iš $d\theta$, turi būti gauta tikimybė, kad elektronas bus išsklaidytas į kampų intervalą nuo θ iki $\theta + d\theta$. Ši tikimybė gaunama, suintegravus (4.3.13b) reiškinį azimutinio kampo ϕ atžvilgiu nuo 0 iki 2π . Taip gauname:

$$dP = \frac{1}{2} \sin \theta d\theta. \quad (4.3.13c)$$

Tai yra tikimybė, kad elektronas bus išsklaidytas į žiedinę juostą, kuri atitinka polinio kampo intervalą nuo θ iki $\theta + d\theta$ (žr. 11 pav.).

$$f(\theta) \equiv \frac{dP}{d\theta} = \frac{1}{2} \sin \theta. \quad (4.3.13d)$$

Irašius pastarąjį reiškinį į (4.3.13b), gaunamas reiškinys

$$P(\theta) = \frac{1}{2} (1 - \cos \theta). \quad (4.3.13e)$$

Irašius šį reiškinį į (4.2.18) lygties kairiąją pusę ir išreiškus $\cos \theta$, gaunama galutinė formulė, pagal kurią reikia generuoti polinio kampo θ vertes, kai sklaida yra izotropinė:

$$\cos \theta = 1 - 2r; \quad (4.3.14)$$

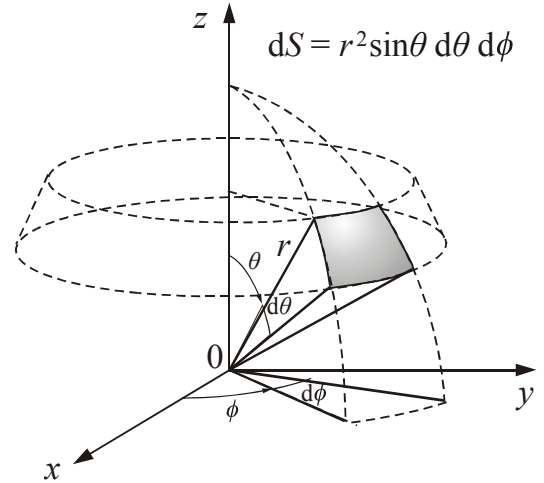
čia r yra atsitiktinis skaičius, kuris tolygiai pasiskirstęs tarp 0 ir 1.

4.3.3. Nepolinė sklaida optiniais fononais

Nors optinių virpesių atveju kaimyniniai kristalo atomai juda priešingomis kryptimis (o akustinių virpesių atveju jie juda ta pačia kryptimi), tačiau sklaida optiniais nepoliniais fononais gali būti analizuojama iš esmės tais pačiais metodais, kaip sklaida akustiniais nepoliniais fononais.

Kadangi optinių virpesių metu kristalo gardelės narvelio atomai juda priešingomis kryptimis, tai kristalo gardelės deformacija yra „sutelkta“ kiekviename narvelyje. Vadinasi, šiuo atveju deformaciją nusako pats atomo nuokrypis nuo pusiausvyros padėties (o ne gretimų narvelių atomų poslinkių skirtumas, kaip akustinių virpesių atveju). Todėl vienelektroninio hamiltoniano trikdys (sąveikos potencialas), dėl kurio atsiranda nepolinė sklaida optiniais fononais, yra tokio pavidalo [1, p. 46]:

$$H'(\mathbf{r}, t) = \mathbf{D}_0 \cdot \mathbf{u}(\mathbf{r}, t), \quad (4.3.15)$$



11 pav. Sferos paviršiaus elementas, į kurį yra išsklaidomas elektronas

čia $\mathbf{u}(\mathbf{r}, t)$ yra gardelės nuokrypis nuo pusiausvyros padėties taške \mathbf{r} laiko momentu t , o D_o yra optinio deformacijos potencialo konstanta (vektorinis dydis, kurio dimensija yra eV/cm). Šiuo atveju, atlikus skaičiavimus, kurie analogiškai praeitame skyriuje atliktiems skaičiavimams, gaunama tokia nepolinės optinės sklaidos iš būsenos \mathbf{k} į būseną \mathbf{k}' tikimybės per laiko vienetą išraiška [1, p. 47]

$$S(\mathbf{k}, \mathbf{k}') = \frac{\pi D_o^2}{\rho \omega_0 V} (n_0 + \frac{1}{2} \mp \frac{1}{2}) \delta \left(\frac{\hbar^2 q^2}{2m^*} \pm \frac{\hbar^2 k q \cos \theta}{m^*} \mp \hbar \omega_0 \right) \quad (4.3.15a)$$

čia n_0 yra optinių fononų skaičius, o ω_0 yra jų dažnis. Nepolinės optinės sklaidos metu elektronas gali arba likti tame pačiame laidumo juostos slėnyje, arba pereiti į kitą slėnį. Kai sklaida yra tame pačiame slėnyje, tada joje dalyvaujančių fononų bangos vektorius yra mažas, o tarpslėnės sklaidos atveju fononų bangos skaičius yra palyginti didelis (I Brijueno zonos matmenų eilės). Optinių fononų energija priklauso nuo jų bangos vektoriaus: mažo bangos vektoriaus (arti I Brijueno zonos centro) optinių fononų energija yra šiek tiek didesnė, negu arti zonos krašto (žr. 4 pav.). Indeksas „0“ parodo, kad yra nagrinėjama sklaida slėnio viduje, kai sklaidą sąlygoja palyginti mažo bangos vektoriaus optiniai fononai (tarpslėnė sklaida bus aptariama kitame skyriuje). Kadangi $\cos \theta$ mažiausia reikšmė yra -1 , o didžiausia reikšmė $+1$, tai iš reikalavimo, kad delta funkcijos argumentas būtų lygus nuliui, išplaukia didžiausio ir mažiausio optinio fonono bangos skaičiaus išraiškos:

$$q_{\min} = k \left| 1 - \left(1 \pm \frac{\hbar \omega_0}{E_k} \right)^{1/2} \right|, \quad (4.3.15b)$$

$$q_{\max} = k \left[1 + \left(1 \pm \frac{\hbar \omega_0}{E_k} \right)^{1/2} \right]. \quad (4.3.15c)$$

Kaip ir nagrinėjant akustinę sklaidą, pilnutinė nepolinės optinės fononinės sklaidos sparta (esant duotai pradinei elektrono būsenai \mathbf{k}) gaunama suintegravus (4.3.15a) reiškinių visų galimų galutinių būsenų atžvilgiu. Kadangi galutinę būseną lemia fonono bangos vektorius \mathbf{q} , tai integruoti galima \mathbf{q} atžvilgiu. Tada parabolinės laidumo juostos atveju gaunama tokia nepolinės sklaidos optiniais fononais tikimybės per laiko vienetą išraiška [1, p. 48]:

$$W_{\text{par}}(\mathbf{k}) = \frac{\pi D_o^2}{\rho \omega_0} (n_0 + \frac{1}{2} \mp \frac{1}{2}) N(E_k \pm \hbar \omega_0), \quad (4.3.16)$$

Dažnio ω_0 ir apibrėžto bangos vektoriaus optinių fononų skaičių nusako Bozė ir Einšteino pasiskirstymo funkcija

$$n_0 = \frac{1}{\exp \left(\frac{\hbar \omega_0}{k_B T} \right) - 1}. \quad (4.3.17)$$

(4.3.15a)–(4.3.16) formulėse viršutinis ženklas atitinka fonono absorbciją, o apatinis – fonono emisiją. Tos formulės galioja tik tada, kai kvantinio šuolio metu elektronas lieka tame pačiame slėnyje. Būsenų tankis $N(E_k)$ apskaičiuojamas pagal (3.1.10) formulę.

Neparabolinės laidumo juostos atveju nepolinės optinės sklaidos slėnyje sparta lygi [1, p. 236]

$$W(\mathbf{k}) = \frac{\pi D_o^2 [n(\omega_0) + \frac{1}{2} \mp \frac{1}{2}]}{\rho \omega_0} N(E_{k'}) F(E_k, E_{k'}), \quad (4.3.18)$$

kur

$$F(E_k, E_{k'}) = \frac{(1 + \alpha E_k)(1 + \alpha E_{k'})}{(1 + 2\alpha E_k)(1 + 2\alpha E_{k'})}. \quad (4.3.19)$$

Čia α yra duotojo slėnio neparaboliškumo koeficientas (žr. (3.1.5)), o $E_{k'} = E_k \pm \hbar \omega$. Formulėje (4.3.18) būsenų tankis $N(E_k)$ apskaičiuojamas pagal (3.1.11) formulę.

Optinių fononų energija yra tos pačios eilės, kaip vidutinė krūvininkų šiluminio judėjimo energija kambario temperatūroje ($k_B T$), todėl optinė fononinė sklaida yra netamprioji (kitaip negu akustinė fononinė sklaida, kurią sąlygoja daug mažesnės energijos akustiniai fononai).

Nepolinė optinė sklaida yra izotropinė (kaip ir nepolinė akustinė sklaida), todėl sklaidos kampai ϕ ir θ turi būti generuojami pagal tas pačias formules, kaip ir nepolinės akustinės sklaidos atveju, t. y. pagal (4.2.20) ir (4.3.14) formules.

4.3.4. Tarpslėnė sklaida

Vykstant nepolinei sklaidai trumpabangiais optiniais arba akustiniais fononais, elektronas gali peršokti į kitą laidumo juostos slėnį. Pvz., elektronas gali peršokti iš Γ slėnio (kuris yra Brijueno zonos centre) į kurį nors iš L slėnių (kurie išsidėstę $\langle 111 \rangle$ kryptimis ant Brijueno zonos krašto) arba į kurį nors iš X slėnių (kurie išsidėstę $\langle 100 \rangle$ kryptimis ant Brijueno zonos krašto). Tokia sklaida vadinama **tarpslėne sklaida**. Kubinės simetrijos puslaidininkiuose, jeigu pradinis slėnis nėra vienas iš L slėnių, nepolinė optinė fononinė sklaida visada yra tarpslėnė. Taip yra dėl vadinamųjų atrankos taisyklių, kurios išplaukia iš kristalo simetrijos ir kurios draudžia netarpslėnę sklaidą Γ ir X slėniuose [1, p. 48]. Kai pradinis slėnis yra L slėnis, tada nepolinė optinė fononinė sklaida gali būti ir tarpslėnė, ir netarpslėnė. Taigi, nagrinėjant kubinės singonijos puslaidininkius, praeito skyriaus formules (nusakančias netarpslėnės optinės fononinės sklaidos spartą) galima taikyti tik tada, kai elektronas yra L slėnyje. Sąveikos potencialo, kuris nusako tarpslėnę sklaidą, pavidalas panašus į (4.3.15) [1, p. 49]:

$$H'(\mathbf{r}, t) = \mathbf{D}_{ij} \cdot \mathbf{u}(\mathbf{r}, t), \quad (4.3.20)$$

kur \mathbf{D}_{ij} yra tarpslėnės deformacijos potencialas, kuris nusako tarpslėnės sklaidos intensyvumą. Atitinkama tarpslėnės sklaidos tikimybė per laiko vienetą parabolinės juostos atveju lygi [1, p. 49]

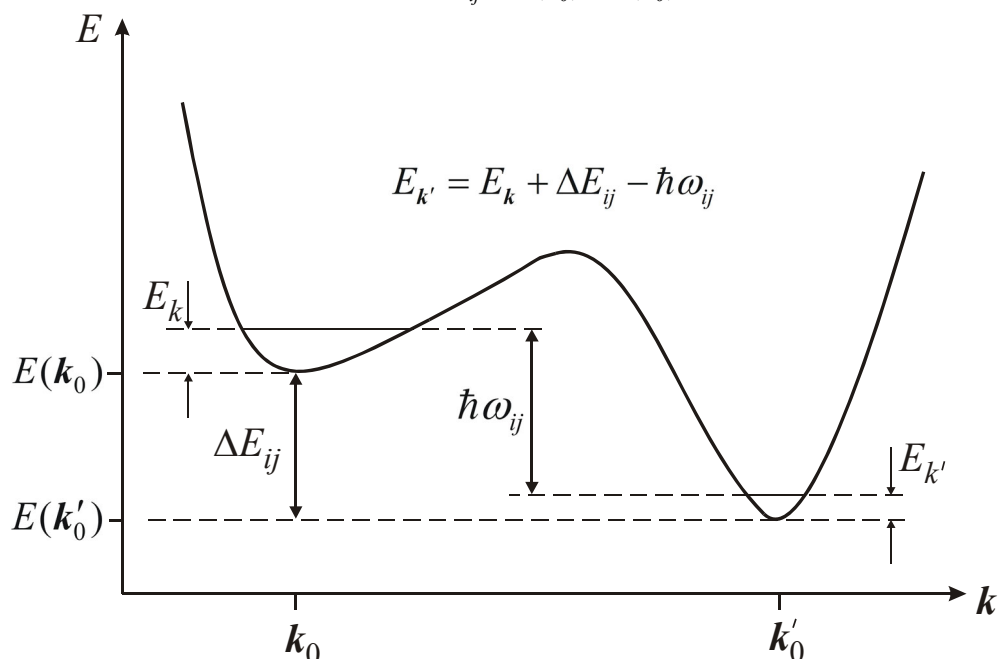
$$W_{\text{par}}(\mathbf{k}) = \frac{\pi D_{ij}^2 Z_j}{\rho \omega_{ij}} [n(\omega_{ij}) + \frac{1}{2} \mp \frac{1}{2}] N_j(E_{k'}), \quad (4.3.21)$$

kur ω_{ij} yra fononų dažnis arti Brijueno zonos krašto, Z_j yra galimų ekvivalenčių galutinių slėnių skaičius, o N_j yra būsenų tankis *galutiniame slėnyje*. $E_{k'}$ yra elektrono energija galutiniame slėnyje. Kadangi ši energija yra atskaitoma nuo *galutinio* slėnio mažiausios energijos, o pradinė energija E_k yra atskaitoma nuo *pradinio* slėnio mažiausios energijos, tai energijos tvermės dėsnis šiuo atveju užrašomas šitaip:

$$E_{k'} = E_k + \Delta E_{ij} \pm \hbar \omega_{ij}, \quad (4.3.21a)$$

kur ΔE_{ij} yra pradinio ir galutinio slėnių mažiausių energijų skirtumas:

$$\Delta E_{ij} = E(\mathbf{k}_0) - E(\mathbf{k}'_0). \quad (4.3.21b)$$



12 pav. Energijos tvermės dėsnis, kai elektronas pereina į mažesnės energijos slėnį ir yra emituojamas fononas, kurio energija $\hbar \omega_{ij}$

Čia \mathbf{k}_0 ir \mathbf{k}'_0 yra bangos vektoriai, atitinkantys pradinio ir galutinio slėnių mažiausios energijos vertes, o $E(\mathbf{k}_0)$ ir $E(\mathbf{k}'_0)$ yra tos energijos vertės, kai energijos atskaitos vertė yra bendra abiem slėniams. Reikia atkreipti dėmesį į tai, kad $N_j(E_{\mathbf{k}'})$ yra *galutinio* slėnio energinis būsenų tankis. Kai juostos yra parabolinės, $N_j(E_{\mathbf{k}'})$ apskaičiuojamas pagal (3.1.10) formulę, o neparabolinių juostų atveju – pagal (3.1.11) formulę. Kadangi elektrono efektinės masės m^* skirtinguose slėniuose yra skirtingos, tai $N_j(E_{\mathbf{k}})$ nėra lygus pradinio slėnio būsenų tankiui $N(E_{\mathbf{k}})$ (žr. būsenų tankio išraišką (3.1.10)). Dažnis ω_{ij} yra šiek tiek mažesnis už dažnį ω_0 , kuris atitinka artimą nuliui optinio fonono banginį vektorius (žr. 4 pav.). Palyginus (4.3.21) su netarpslėnės optinės fononinės sklaidos spartos išraiška (4.3.16), akivaizdu, kad abi formulės yra vienodo pavidalo. Tai ir suprantama, nes jos abi nusako tą patį fizikinį vyksmą – nepolinę optinę fononinę sklaidą. Skirtumas yra tik kiekybinis: skiriasi galutinės elektrono būsenos apibrėžtys (vieni atveju galutinė elektrono būsena atitinka tą patį slėnį, o kitu atveju ji atitinka kitą slėnį). Be to, tarpslėnės sklaidos spartos išraiškoje yra daugiklis Z_j , kurio nėra netarpslėnės sklaidos spartos išraiškoje.

Ekvivalenčių galutinių slėnių skaičius Z_j nustatomas, remiantis šuolio kryptimi ir šiais duomenimis: ekvivalenčių L slėnių skaičius Brijueno zonoje lygus 4, o ekvivalenčių X slėnių skaičius Brijueno zonoje lygus 3 [2, p. 86]. Tačiau kartais laikoma, kad ekvivalenčių X slėnių skaičius lygus 6, t. y. priešinguose Brijueno zonos kraštuose esantys slėniai, kurių padėtys skiriasi atvirkštinės gardelės translacijs vektoriumi, yra laikomi skirtingais ekvivalenčiais slėniais [3]. Naudojant formulę (4.3.21), reikia atkreipti dėmesį, kad deformacijos potencialas D_{ij} būtų apibrėžtas tam pačiam slėnių skaičiui Z_j , kuris įeina į šią formulę [2, p. 96]. Pvz., jeigu žinyne D_{ij} vertė apibrėžta X slėnių skaičiui $Z_j = 6$, o formulėje (4.3.21) naudojamas $Z_j = 3$, tada žinyne pateiktąją D_{ij} vertę reikia padauginti iš $\sqrt{2}$. Šuoliai tarp ekvivalenčių slėnių, kurie yra priešinguose Brijueno zonos kraštuose, vadinami ***g procesais***, o kitų rūšių šuoliai tarp ekvivalenčių slėnių vadinami ***f procesais*** arba ***h procesais***, priklausomai nuo atstumo tarp pradinio ir galutinio slėnių [2, p. 87].

Neparabolinės laidumo juostos atveju tarpslėnės sklaidos sparta lygi [1, p. 236]

$$W(\mathbf{k}) = \frac{\pi D_{ij}^2 [n(\omega_{ij}) + \frac{1}{2} \mp \frac{1}{2}]}{\rho \omega_{ij}} Z_j N_j(E_{\mathbf{k}'}) F_{ij}(E_{\mathbf{k}}, E_{\mathbf{k}'}), \quad (4.3.22)$$

$$F_{ij}(E_{\mathbf{k}}, E_{\mathbf{k}'}) = \frac{(1 + \alpha_i E_{\mathbf{k}})(1 + \alpha_j E_{\mathbf{k}'})}{(1 + 2\alpha_i E_{\mathbf{k}})(1 + 2\alpha_j E_{\mathbf{k}'})}. \quad (4.3.23)$$

Čia α_i ir α_j yra atitinkamai pradinio ir galutinio slėnių neparaboliškumo koeficientai.

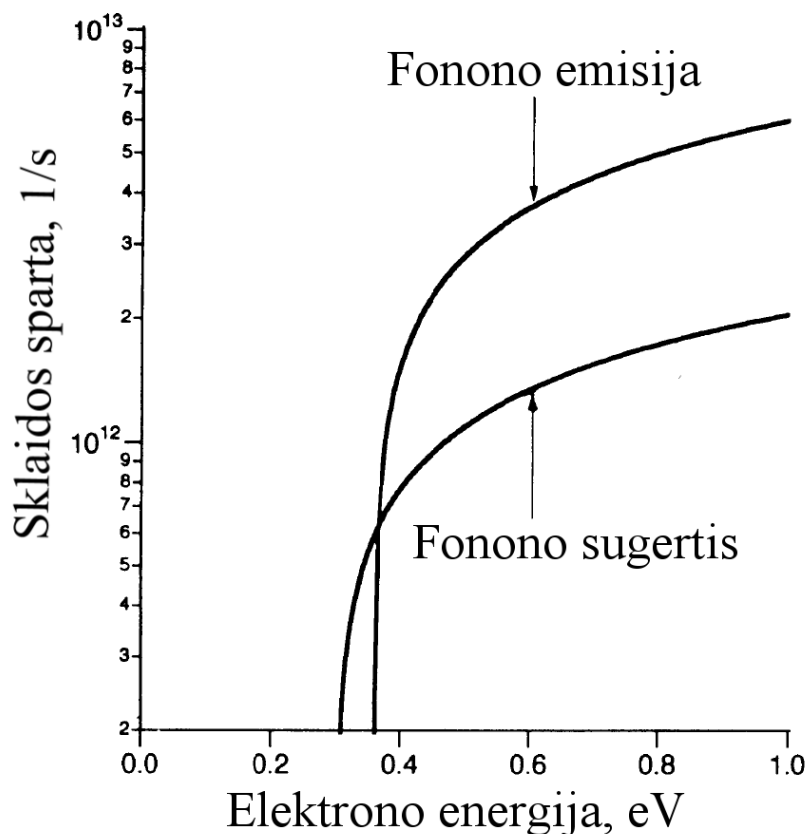
13 pav. parodyta galio arsenido elektronų optinės fononinės sklaidos iš Γ slėnio į L slėnį spartos priklausomybė nuo elektrono energijos. Matome, kad egzistuoja tam tikra mažiausia („slenkstinė“) elektrono energija, nuo kurios tampa galima tokia sklaida. Taip yra todėl, kad galio arsenido Γ slėnio energija yra mažesnė už L slėnio energiją, t. y. (4.3.21a) formulėje $\Delta E_{ij} < 0$. Kitaip sakant $\Delta E_{ji} > 0$ (čia $\Delta E_{ji} \equiv -\Delta E_{ij}$). Slenkstinę elektrono energiją galima apskaičiuoti išreiškus $E_{\mathbf{k}}$ iš (4.3.21a) formulės:

$$E_{\mathbf{k}} = E_{\mathbf{k}'} + \Delta E_{ji} \mp \hbar \omega_{ij}, \quad (4.3.23a)$$

čia, kaip ir anksčiau, viršutinis ženklas reiškia fonono absorbciją, o apatinis reiškia fonono emisiją. Mažiausia galima $E_{\mathbf{k}}$ vertė gaunama įrašius vietoj $E_{\mathbf{k}'}$ nulį:

$$E_{\mathbf{k}_{\min}} = \Delta E_{ji} \mp \hbar \omega_{ij}. \quad (4.3.23a)$$

Tarpslėnės sklaidos atveju elektrono bangos vektoriaus pokytis yra artimas Brijueno zonos spinduliui. Tačiau elektrono judėjimo modeliavimui reikalingas tik bangos vektoriaus nuokrypis nuo \mathbf{k} vertės, atitinkančios galutinio slėnio mažiausią energiją (žr. 3.3 skyrių), kuris yra daug mažesnis už Brijueno zonos matmenis.



13 pav. GaAs elektronų sklaidos iš Γ slėnio į L slėnį spartos priklausomybė nuo elektrono energijos. Kadangi GaAs Γ slėnio energija yra mažesnė už L slėnio energiją, tai šis vyksmas, sugeriant fononą, tampa galimas tik tada, kai elektrono energija Γ slėnyje (atskaičyta nuo to slėnio dugno) yra didesnė už $\Delta E_{ji} - \hbar\omega_{ij}$. Šis vyksmas emituojant fononą yra galimas tik tada, kai elektrono energija $> \Delta E_{ji} + \hbar\omega_{ij}$.

Čia $\Delta E_{ji} = -\Delta E_{ij} > 0$ yra galutinio (L) ir pradinio (Γ) slėnių energijų skirtumas

4.3.5. Polinė sklaida optiniais fononais

Polinę fononinę sklaidą sukelia tik išilginiai fononai [1, p. 49] ir tik joniniuose puslaidininkiuose (pvz., GaAs). Polinė sklaida išilginiais akustiniais fononais (pjezoelektrinė sklaida) svarbi tik labai žemose temperatūrose ir labai grynuose puslaidininkiuose. Kambario temperatūroje pjezoelektrinės sklaidos galima nepaisyti. Polinės optinės fononinės sklaidos atveju vienelektroninio hamiltoniano trikdys yra (4.1.3) pavidalo, o jo amplitudė lygi

$$U(\mathbf{r}) = e \int E_{\parallel}(\mathbf{r}) d\mathbf{l}, \quad (4.3.24)$$

kur E_{\parallel} yra elektrinio lauko stipris, kurį sąlygoja kristalo gardelės elementariojo narvelio dipolinio momento kitimas, sklindant išilginei optinei deformacijos bangai, o l yra koordinatė bangos sklaidimo kryptimi. Šį elektrinį lauką galima susieti su teigiamųjų ir neigiamųjų jonų reliatyviuoju poslinkiu $\mathbf{u} = \mathbf{u}_+ - \mathbf{u}_-$ (čia \mathbf{u}_+ žymi narvelio teigiamojo jono poslinkį, o \mathbf{u}_- – neigiamojo jono poslinkį):

$$E_{\parallel}(\mathbf{r}) = -\frac{N}{V} \frac{e^* u(\mathbf{r})}{\epsilon_0 \epsilon_{\infty}}, \quad (4.3.25)$$

kur N yra jonų porų skaičius, V yra kristalo tūris, ϵ_{∞} yra dielektrinė skvarba optiniame diapazone (kai joninė poliarizacija jau nepasireiškia), $u(\mathbf{r}) = |\mathbf{u}(\mathbf{r})|$, o e^* yra efektinis krūvis [1, p. 51]:

$$e^* = \sqrt{\epsilon_0} \left(\frac{VM}{N} \right)^{1/2} \omega_0 \epsilon_{\infty} \left(\frac{1}{\epsilon_{\infty}} - \frac{1}{\epsilon_s} \right)^{1/2}, \quad (4.3.26)$$

kur M yra teigiamojo ir neigiamojo jonų redukuotoji masė ($1/M = 1/M_+ + 1/M_-$), ω_0 yra optinių fononų dažnis, o ϵ_s yra statinė dielektrinė skvarba. Įrašę (4.3.25) į (4.3.24) ir pasinaudoję tuo, kad

plokščios bangos atveju $u(\mathbf{r}) = u_0 e^{i(\mathbf{q}, \mathbf{r})} = u_0 e^{iq_l}$ (čia \mathbf{q} yra fonono bangos vektorius, o $q = |\mathbf{q}|$), gauname

$$U(\mathbf{r}) = i \frac{N}{V} \frac{e \cdot e^*}{q \varepsilon_0 \varepsilon_\infty} u(\mathbf{r}). \quad (4.3.27)$$

Parabolinės laidumo juostos atveju sklaidos poliniais optiniais fononais tikimybė per laiko vienetą lygi [1, p. 53]:

$$W_{par}(\mathbf{k}) = \frac{e^2 \omega_0}{8\pi \varepsilon_0 \varepsilon_p} \cdot \frac{k}{E_k} [n(\omega_0) + \frac{1}{2} \mp \frac{1}{2}] \ln \left(\frac{q_{\max}}{q_{\min}} \right), \quad (4.3.28)$$

kur ω_0 yra optinių fononų dažnis (fononų energijos ir redukuotosios Planko konstantos santykis), $1/\varepsilon_p = 1/\varepsilon_\infty - 1/\varepsilon_s$, o q_{\max} ir q_{\min} yra fonono bangos skaičiaus q ribinės vertės:

$$q_{\min} = k \left| 1 - \left(1 \pm \frac{\hbar \omega_0}{E_k} \right)^{1/2} \right|, \quad (4.3.29a)$$

$$q_{\max} = k \left[1 + \left(1 \pm \frac{\hbar \omega_0}{E_k} \right)^{1/2} \right] \quad (4.3.29b)$$

(šios formulės išplaukia iš energijos ir impulso tvermės dėsnių). Formulėse (4.3.28) ir (4.3.29) viršutinis ženklas atitinka fonono absorbciją, o apatinis – fonono emisiją.

Neparabolinės laidumo juostos atveju vietoj (4.3.28) gauname [1, p. 236]

$$W(\mathbf{k}) = \frac{e^2 \omega_0 [n(\omega_0) + \frac{1}{2} \mp \frac{1}{2}]}{8\pi \varepsilon_0 \varepsilon_p} \cdot \frac{k}{\gamma(E_k)} \frac{d\gamma(E_{k'})}{dE_{k'}} F_0(E_k, E_{k'}), \quad (4.3.30)$$

kur

$$F_0(E_k, E_{k'}) = \frac{1}{C} \left[A \ln \left| \frac{\sqrt{\gamma(E_k)} + \sqrt{\gamma(E_{k'})}}{\sqrt{\gamma(E_k)} - \sqrt{\gamma(E_{k'})}} \right| + B \right]. \quad (4.3.31)$$

Čia $E_{k'} = E_k \pm \hbar \omega$, funkciją $\gamma(E_k)$ apibrėžia formulė (3.1.5) (kurioje energija E_k atskaitoma nuo minimumo energijos $E(\mathbf{k}_0)$), o koeficientai A , B ir C lygūs

$$A = \{2(1 + \alpha E_k)(1 + \alpha E_{k'}) + \alpha[\gamma(E_k) + \gamma(E_{k'})]\}^2,$$

$$B = -2\alpha \sqrt{\gamma(E_k) \gamma(E_{k'})} \{4(1 + \alpha E_k)(1 + \alpha E_{k'}) + \alpha[\gamma(E_k) + \gamma(E_{k'})]\},$$

$$C = 4(1 + \alpha E_k)(1 + \alpha E_{k'})(1 + 2\alpha E_k)(1 + 2\alpha E_{k'}),$$

5. Viendalelis Monte Karlo metodas

5.1. Viendalelės Monte Karlo programos struktūra

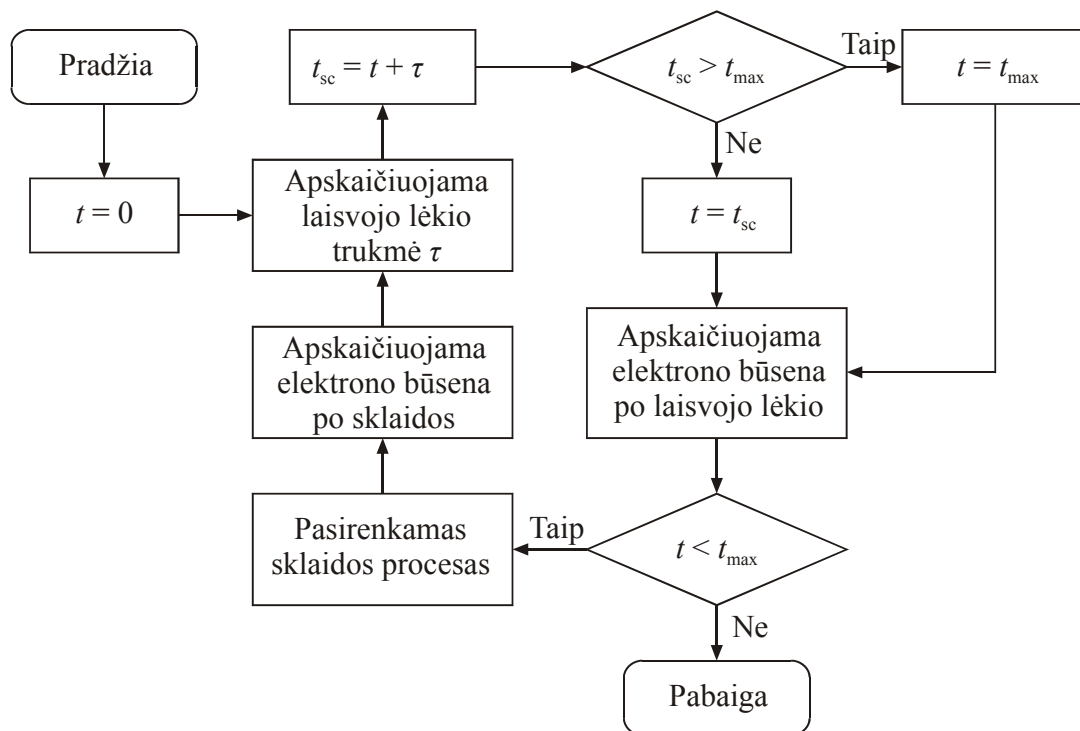
Bet kokios puslaidininkinės struktūros elektrines savybes lemia didelio skaičiaus (milijonų arba daugiau) elektronų arba skylių judėjimas ir tarpusavio sąveika. Tačiau atskirais atvejais tokią daugelio sąveikaujančių dalelių sistemą galima aproksimuoti daugelio vienodų *nepriklausomų* dalelių sistema. Tada sistemos parametrus galima apskaičiuoti, nagrinėjant tik vienos dalelės judėjimą pakankamai ilgame laiko tarpe, o po to apskaičiavus mus dominančio dydžio (pvz., elektrono greičio, energijos, impulso) laikinį vidurkį. Tai yra pagrindinė viendalelio Monte Karlo metodo prielaida.

Bendroji elektrono judėjimo modeliavimo viendaleliu Monte Karlo metodu tvarka yra tokia. Užduodami išorinio elektrinio lauko stipris ir pradinė elektrono būsena (erdvinės koordinatės ir bangos vektorius). Pasirenkamas pilnutinis elektrono judėjimo laikas t_{max} (pilnutinė modeliuojamojo proceso trukmė). Po to cikliškai atliekami šie keturi žingsniai:

1. Apskaičiuojami laisvojo lėkio trukmė τ ir sklaidos momentas $t_{sc} = t + \tau$ (čia t yra einamasis laikas).
2. Apskaičiuojama elektrono būsena laisvojo lėkio pabaigoje.
3. Pasirenkamas sklaidos tipas.
4. Apskaičiuojamas elektrono bangos vektorius po sklaidos.

Šie keturi žingsniai kartojami tol, kol proceso trukmė nepasiekia t_{max} . Taigi, viendalelės Monte Karlo programos supaprastinta struktūrinė schema yra tokia, kaip pavaizduota 14 pav.

Norint rasti elektrono vidutinį dreifinį greitį v , reikia padalyti elektrono nueitą kelią lauko kryptimi (l) iš proceso trukmės. Kelią l galima nustatyti arba tiesiogiai (pagal elektrono koordinatę) arba pagal pilnutinį elektrinio lauko atliktą darbą A : $l = A/(e|\mathcal{E}|)$; čia e yra elektrono krūvis, o $|\mathcal{E}|$ yra išorinio elektrinio lauko stipris. Darbas A randamas, sudėjus visus elektrono energijos pokyčius dreifo metu (t. y. tarpuose tarp visų sklaidos įvykių). Žinant vidutinį dreifinį greitį, judris apskaičiuojamas pagal formulę



14 pav. Viendalelės Monte Karlo programos supaprastinta struktūrinė schema

$$\mu = \frac{v}{|\mathcal{E}|}.$$

Laisvojo lėkio trukmė, sklaidos tipas ir sklaidos kryptis, kurie nustatomi žingsniuose 1, 3 ir 4, yra atsitiktiniai dydžiai. Kiekvienas iš šių dydžių pasirenkamas, naudojant atsitiktinių skaičių generatorių. Tie skaičiai turi būti generuojami taip, kad skirtingų verčių pasikartojimo dažnis atitiktų to dydžio skirstinį.

Aukščiau minėtieji žingsniai 1 ir 2 smulkiau aprašyti 5.2 skyriuje, o žingsniai 3 ir 4 – 5.3 skyriuje.

5.2. Lėkio trukmės skaičiavimas. Dreifo proceso modeliavimas

Laisvojo lėkio trukmė priklauso nuo sklaidos tikimybės per laiko vienetą. Ši tikimybė lygi sumai tikimybių, kurios atitinka skirtingus sklaidos tipus (pvz., priemaišinė arba fononinė sklaida):

$$W(E_k) = \sum_{j=1}^n W_j(E_k). \quad (5.2.1)$$

Čia indeksas j nusako sklaidos rūšį, o E_k yra elektrono, kurio bangos vektorius \mathbf{k} , energija. Tikimybė per laiko vienetą, kad elektronas, kurio pradinis bangos vektorius yra \mathbf{k} , judės be susidūrimų laiką τ , o po to bus išsklaidytas, yra lygi

$$P(\tau) = W(E_k) \exp \left[- \int_0^{\tau} W(E_k) dt \right]. \quad (5.2.2)$$

Pagal bendrąją atsitiktinių skaičių generavimo taisyklę (žr. 4.2 skyrių), lėkio trukmė τ turėtų būti skaičiuojama tokiu būdu: santykis $P(\tau)/W(E_k)$ prilyginamas atsitiktiniam skaičiui, kuris tolygiai pasiskirstęs tarp 0 ir 1, o po to pagal (5.2.2) apskaičiuojama trukmė τ . Tačiau tam reikėtų apskaičiuoti integralą laužtiniuose skliaustuose. Šio integralo skaičiavimas yra sudėtingas, nes \mathbf{k} (vadinasi, ir E_k) kinta laike, o $W(E_k)$ priklausomybė nuo E_k yra sudėtinga.

Sunkumų, kurie susiję su minėtojo integralo skaičiavimu, išvengiama tokiu būdu. Apibrėžiamas naujas sklaidos tipas – „virtualioji sklaida“ (*self-scattering*), kurios metu elektrono būseną nepasikeičia ir kurios tikimybė per laiko vienetą $W_0(E_k)$ yra parinkta taip, kad naujoji pilnutinė sklaidos tikimybė per laiko vienetą nepriklausytų nuo energijos [1, p. 81]:

$$W_0(E_k) = \Gamma - \sum_{j=1}^n W_j(E_k) \quad \text{arba} \quad \Gamma = \sum_{j=0}^n W_j(E_k), \quad (5.2.3)$$

kur Γ yra konstanta, kuri nusako naująją pilnutinę sklaidos tikimybę per laiko vienetą. Tada formulė (5.2.2) tampa daug paprastesnė:

$$P(\tau) = \Gamma e^{-\Gamma \tau}. \quad (5.2.4)$$

Atitinkamai, lėkio trukmė skaičiuojama pagal formulę

$$\tau = - \frac{\ln(r_1)}{\Gamma}, \quad (5.2.5)$$

kur r_1 yra atsitiktinis skaičius, kuris tolygiai pasiskirstęs tarp 0 ir 1.

Konstantos Γ vertė turi būti parinkta taip, kad visame tiriamajame energijų intervale ji būtų didesnė už tikrąją pilnąją tikimybę (5.2.1). Tačiau Γ vertė turi būti kuo mažesnė, kad virtualiosios sklaidos įvykių dažnis būtų kuo mažesnis.

Laikysime, kad išorinis elektrinis laukas \mathcal{E} , kuris veikia elektronus, erdvėje kinta žymiai lėčiau negu kristalo tarpatominis laukas. Tada elektrono bangos vektoriaus pokytis per laisvojo lėkio trukmę τ yra lygus

$$\Delta \mathbf{k} = - \frac{e\mathcal{E}}{\hbar} \tau. \quad (5.2.6)$$

5.3. Sklaidos proceso modeliavimas

Norint teisingai aprašyti sklaidos įtaką elektrono judėjimui, reikia žinoti diferencialinio sklaidos skerspjūvio priklausomybę nuo sklaidos kampo ir nuo elektrono būsenos. Tam reikia žinoti sklaidos tipą. Tarkime, kad egzistuoja n sklaidos tipų (neįskaitant virtualiosios sklaidos, kuri minėta 5.2 skyriuje). Tada sklaidos tipą elektrono lėkio pabaigoje galima pasirinkti, naudojant atsitiktinių skaičių generatorių ir funkcijas $\Lambda_j(E_k)$, kurios apibrėžiamos šitaip [1, p. 84]:

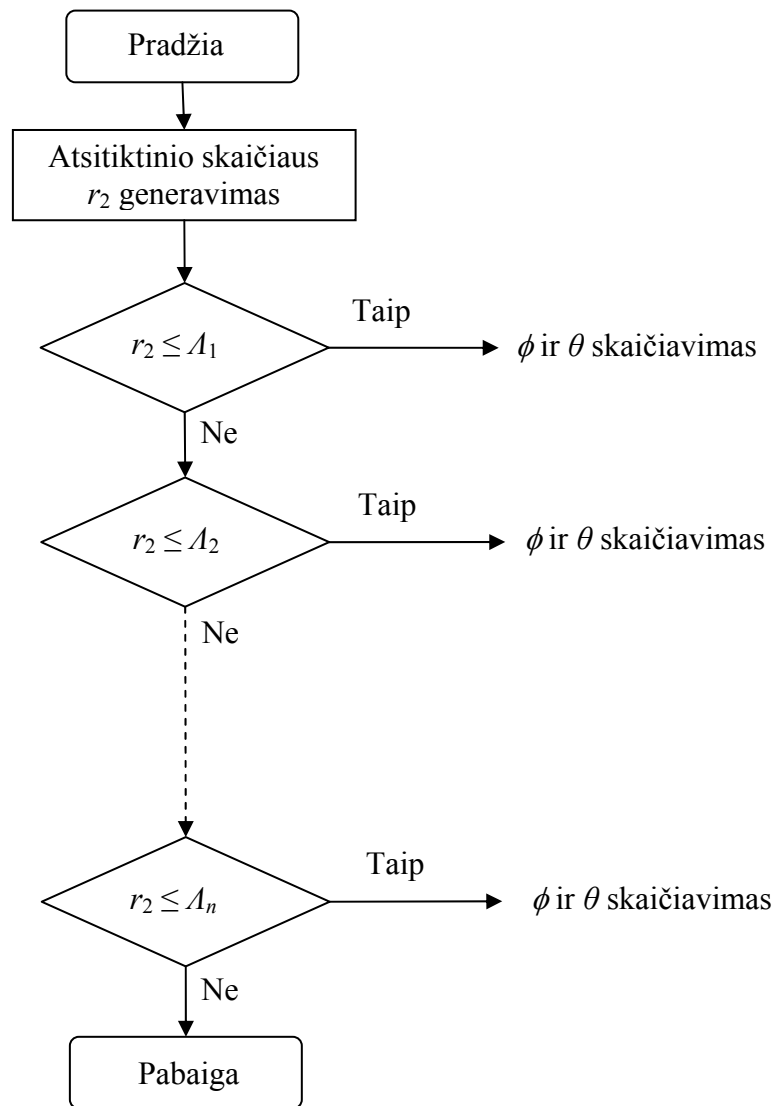
$$\Lambda_j(E_k) = \frac{\sum_{i=1}^j W_i(E_k)}{\Gamma} \quad (j = 1, 2, \dots, n) \quad (5.3.1)$$

(konstanta Γ apibrėžta 5.2 skyriuje). j -tojo tipo sklaida pasirenkama tuo atveju, jeigu atsitiktinis skaičius r_2 , kuris tolygiai pasiskirstęs tarp 0 ir 1, priklauso intervalui

$$\Lambda_{j-1}(E_k) < r_2 \leq \Lambda_j(E_k) \quad (j = 1, 2, \dots, n) \quad (5.3.2)$$

(čia laikoma, kad $\Lambda_0 = 0$). Jeigu $r_2 > \Lambda_n$, pasirenkama virtualioji sklaida, kurios metu elektrono judėjimo būseną nepakinta, t. y. sklaidos faktiškai nėra. Paprogramės, kuris pasirenka sklaidos tipą, struktūrinė schema galėtų būti tokia, kaip pavaizduota 15 pav.

Pasirinkus sklaidos tipą, reikia apskaičiuoti elektrono bangos vektorių \mathbf{k}' po sklaidos. Šio vektoriaus absoliutinė vertė randama, naudojantis energijos tvermės dėsniu ir (5.1.4) arba (5.1.5) formule. Joninės sklaidos atveju elektrono energija nepakinta (nes jono kuloninis laukas yra



15 pav. Sklaidos tipo pasirinkimo struktūrinė schema.

stacionarus), o fononinės sklaidos atveju elektrono energija pakinta dydžiu, kuris lygus arba priešingas fonono energijai $\hbar\omega_0$, priklausomai nuo to, ar fononas buvo absorbuotas, ar emituotas:

$$E_{k'} = E_k \pm \hbar\omega_0. \quad (5.3.3)$$

Elektrono momentinis greitis (t. y. elektrono Blocho bangos grupinis greitis) lygus

$$\mathbf{v} = \frac{1}{\hbar} \frac{dE_k}{d\mathbf{k}}. \quad (5.3.4)$$

Pagal (5.1.5), vektorių \mathbf{k}' absoliutinė vertė lygi

$$k' = \frac{1}{\hbar} \sqrt{2m^* E_{k'} (1 + \alpha E_{k'})}, \quad (5.3.5)$$

kur $E(\mathbf{k}_0')$ žymi galutinio slėnio minimumo tašką.

Be absoliutinės vertės k' , reikia žinoti ir vektorių \mathbf{k}' kryptį. Sklaidos azimutinis kampas ϕ , kurį nusako vektorių \mathbf{k}' projekcija į statmeną \mathbf{k} kryptiai plokštumą (žr. 5 pav.), yra tolygiai pasiskirstęs tarp 0 ir 2π , t. y. jis skaičiuojamas pagal (4.2.20) formulę:

$$\phi = 2\pi r_3; \quad (5.3.6)$$

čia r_3 yra atsitiktinis skaičius, kurio galimos vertės tolygiai pasiskirsčiusios intervale nuo 0 iki 1. Sklaidos polinis kampas (t. y. kampas tarp vektorių \mathbf{k} ir \mathbf{k}') priklauso nuo sklaidos tipo. Kai elektroną išsklaido jonas, sklaidos polinis kampas θ randamas pagal formulę (4.2.23):

$$\cos \theta = 1 - \frac{2r_4}{1 + (1 - r_4) \left(\frac{2k}{q_D} \right)^2}, \quad (5.3.7)$$

o kai elektroną išsklaido polinis optinis fononas, kampas θ skaičiuojamas pagal formules [1, p. 55]

$$\cos \theta = \frac{1 + f - (1 + 2f)^{r_4}}{f}, \quad (5.3.8a)$$

$$f = \frac{2\sqrt{E_k E_{k'}}}{(\sqrt{E_k} - \sqrt{E_{k'}})^2}. \quad (5.3.8b)$$

Čia r_4 yra atsitiktinis skaičius, kurio vertės tolygiai pasiskirsčiusios intervale nuo 0 iki 1.

Reikia turėti omenyje, kad formulėse (5.3.6), (5.3.7) ir (5.3.8a) kampų θ ir ϕ atskaitos kryptys yra susietos su vektorių \mathbf{k} kryptimi (žr. 5 pav.). Tačiau \mathbf{k} kryptis nuolat kinta dėl sklaidos. Kadangi modeliavimo metu siekiama rasti elektrono judėjimo kryptį laboratorinėje (nejudančioje) atskaitos sistemoje, tai po kiekvieno sklaidos įvykio elektrono bangos vektorių komponentes reikia perskaičiuoti, atliekant tokią koordinačių transformaciją [1, p. 93]:

$$\begin{pmatrix} k'_x \\ k'_y \\ k'_z \end{pmatrix} = \begin{pmatrix} \frac{k_y}{\sqrt{k_x^2 + k_y^2}} & \frac{k_x k_z}{k \sqrt{k_x^2 + k_y^2}} & \frac{k_x}{k} \\ \frac{-k_x}{\sqrt{k_x^2 + k_y^2}} & \frac{k_y k_z}{k \sqrt{k_x^2 + k_y^2}} & \frac{k_y}{k} \\ 0 & \frac{-\sqrt{k_x^2 + k_y^2}}{k} & \frac{k_z}{k} \end{pmatrix} \begin{pmatrix} k' \sin \theta \cos \phi \\ k' \sin \theta \sin \phi \\ k' \cos \theta \end{pmatrix}. \quad (5.3.9)$$

Čia dydžiai k_x , k_y ir k_z yra pradinio bangos vektorių \mathbf{k} Dekarto komponentės laboratorinėje atskaitos sistemoje, dydžiai $k' \sin \theta \cos \phi$, $k' \sin \theta \sin \phi$ ir $k' \cos \theta$ – tai galutinio bangos vektorių \mathbf{k}' Dekarto komponentės koordinačių sistemoje, kuri pavaizduota 5 pav. ($k' = |\mathbf{k}'|$), o dydžiai k_x' , k_y' and k_z' – tai bangos vektorių \mathbf{k}' Dekarto komponentės laboratorinėje atskaitos sistemoje (t. y. jos skiriasi nuo Dekarto komponentių, kurios pavaizduotos 5 pav., nors žymėjimai yra tie patys).

Vykstant nepolinei fononinei sklaidai (kuri gali būti tarpslėnė), vektorių \mathbf{k}' yra pasiskirstęs izotropiškai [1, p. 45 ir p. 48]. T. y. kampas θ tarp \mathbf{k}' ir bet kokios fiksuotos krypties erdvėje turi būti skaičiuojamas pagal formulę

$$\cos \theta = 1 - 2r_4. \quad (5.3.10)$$

Šiuo atveju kampus ϕ ir θ patogiau atskaityti nuo laboratorinės Dekarto koordinatų sistemos, kurioje apibrėžtos elektrono koordinatės, ašių X ir Z . Tada

$$\begin{aligned} k'_x &= k' \sin \theta \cos \phi, \\ k'_y &= k' \sin \theta \sin \phi, \\ k'_z &= k' \cos \theta. \end{aligned} \quad (5.3.11)$$

Taigi, nepolinės sklaidos optiniais arba akustiniais fononais atveju koordinatų transformacijos atlikti nereikia.

5.4. Viendalelio Monte Karlo metodo taikymo pavyzdys: GaAs neigiamąjo diferencialinio laidumo modeliavimas

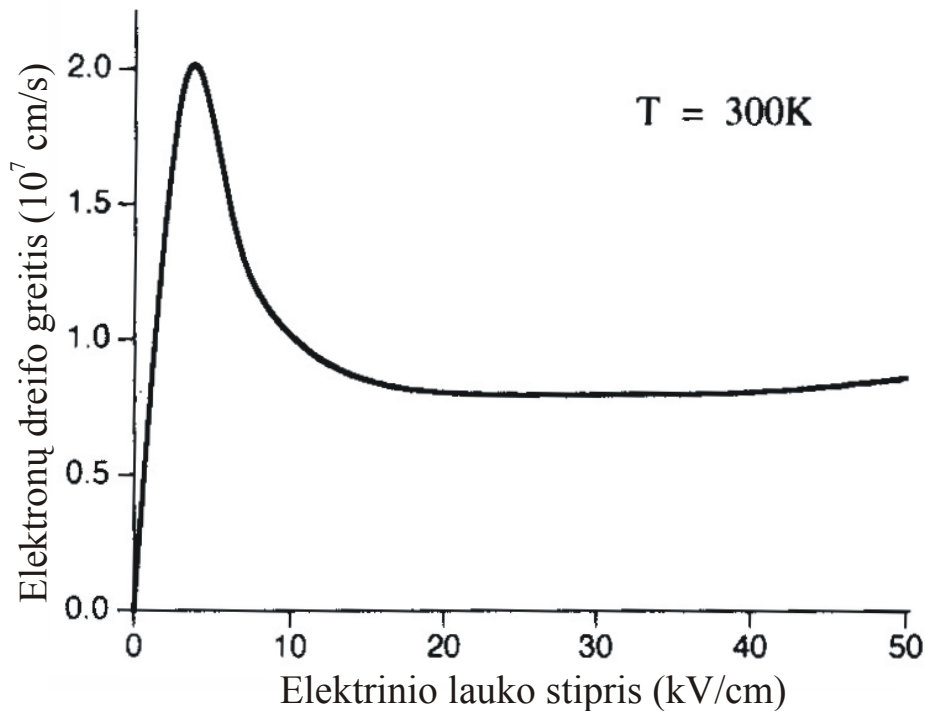
Elektronų dreifo greičio priklausomybė nuo elektrinio lauko stiprio dažnai turi maksimumą (žr. 16 pav.). Dreifo greičio sumažėjimas, kai elektrinis laukas viršija tam tikrą slenkstinę vertę, yra vadinamas **neigiamuoju diferencialiniu laidumu** (NDL). Minėtasis slenkstinis elektrinis laukas dažniausiai būna artimas 4 kV/cm. Galio arsenido neigiamąjo diferencialinio laidumo priežastis yra elektronų pernaša iš Γ slėnio, kuriame jų efektinė masė yra maža ($0,063m_0$, kur m_0 yra elektrono rimties masė vakuume), į L slėnį, kuriame jų efektinė masė yra didelė ($0,23m_0$). Žinome, kad elektronų judris yra lygus

$$\mu = \frac{e\tau}{m^*}; \quad (5.4.1)$$

čia τ yra elektronų impulso relaksacijos trukmė, o m^* yra elektronų efektinė masė. Vadinasi, elektronų efektinės masės padidėjimas yra susijęs su jų judrio sumažėjimu. Tuo pačiu sumažėja ir dreifo greitis, kuris lygus

$$v = \mu \mathcal{E}; \quad (5.4.2)$$

čia \mathcal{E} yra elektrinio lauko stipris.



16 pav. Elektronų dreifo greičio galio arsenide priklausomybė nuo elektrinio lauko stiprio 300 K temperatūroje, kai jonizuotų priemaišų koncentracija lygi 10^{14} cm^{-3} (iš [1]).

Elektronų pernaša iš Γ slėnio į L slėnį vyksta dėl to, kad elektrinis laukas „įkaitina“ elektronus. Kitaip sakant, elektronų netvarkiojo judėjimo vidutinė kinetinė energija stipriame elektriniame lauke gali žymiai viršyti jų šiluminio judėjimo energiją ($k_B T$). Atitinkamai, pagal Bolcmano pasiskirstymo funkciją, padidėja ir tikimybė aptikti elektronus aukštesniuose energijos lygmenyse, kurių užpilda normaliomis sąlygomis yra praktiškai lygi nuliui. Kai elektrono vidutinė kinetinė energija tampa tos pačios eilės, kaip L ir Γ slėnių energijų skirtumas (GaAs atveju – maždaug 0,3 eV), tampa pastebimas L slėnio užpildos augimas ir elektronų dreifo greičio mažėjimas.

Kadangi būsenų tankis yra proporcingas $(m^*)^{3/2}$ (žr. (3.1.11)), tai būsenų tankis L slėnyje yra daug didesnis už būsenų tankį Γ slėnyje. Todėl, esant pakankamai aukštai elektronų energijai, *dauguma* elektronų pereina į L slėnį (nors Γ slėnio energija yra mažesnė). Tolimesnis elektronų dreifo greičio ir vidutinės energijos augimas, stiprėjant elektriniam laukui, yra žymiai lėtesnis, nes L slėnyje yra žymiai didesnė optinės fononinės sklaidos, emituojant fononą, tikimybė. Ši sklaida neleidžia žymiai išaugti elektronų energijai (sumažina energijos relaksacijos trukmę), t. y. sulėtina tolimesnį elektronų „kaitimą“.

6. Darbo metodika

Bendros žinios apie modeliavimo tvarką ir naudojamą pagalbines programas yra pateiktos Prieduose A ir B. Šiame skyriuje yra konkretūs nurodymai apie tai, kaip dirbti su tomis programomis.

Darbas sudarytas iš trijų etapų:

- 1) Programos rašymas.
- 2) Programos kompiliavimas ir sintaksės klaidų taisymas.
- 3) Programos testavimas ir skaičiavimo klaidų taisymas.

Programa rašoma naudojant kompiliatorių **Microsoft Visual Studio 2010 Professional**. Kiekvienam iš šio darbo variantų egzistuoja Visual Studio projektas. Kiekvieno projekto failai yra atskirame aplanke, kurio vardas yra tokio pavidalo:

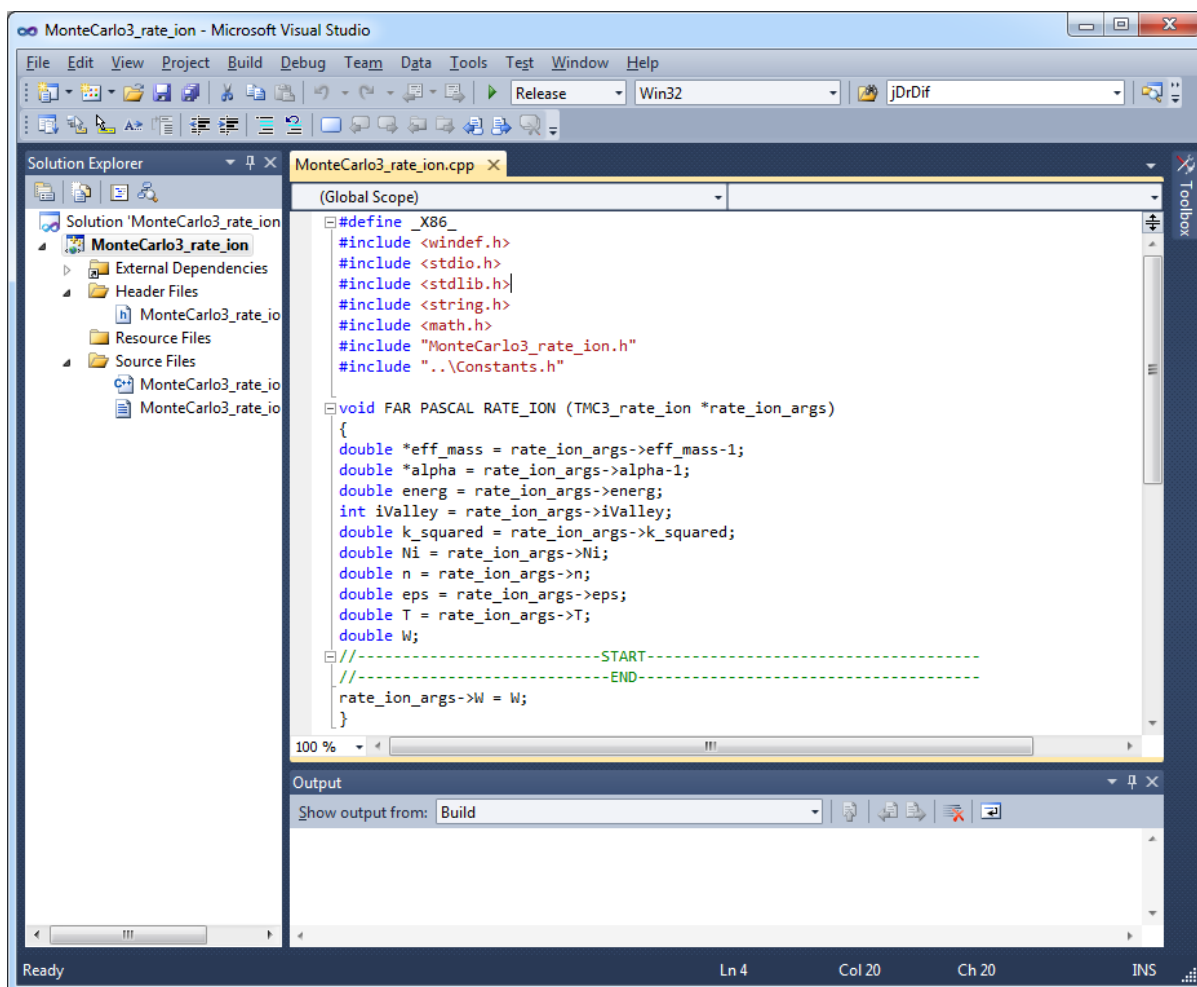
C:\Modeliav\Programos\MonteCarlo3_<funkcijos_pavadinimas>

Pvz., I varianto (kuriame reikia parašyti paprogramę `rate_ion`) Visual Studio projekto failai yra aplanke C:\Modeliav\Programos\MonteCarlo3_ `rate_ion`.

Kad atidaryti atitinkamą projektą, reikia iš Windows Explorer (arba naudojant Visual Studio meniu komandą „File/Open“) atidaryti failą, kurio vardo plėtinys yra `vcxproj` arba `sln` (pvz., I variante reikia atidaryti vieną iš failų `MonteCarlo3_rate_ion.vcxproj` arba `MonteCarlo3_rate_ion.sln`, kurie yra minėtame aplanke). Standartinis atidaryto projekto failo vaizdas yra toks, kaip parodyta 17 pav. Pagrindinės lango dalys yra:

- 1) Programos tekstas,
- 2) Projekto naršyklė („Solution Explorer“), kurioje galima pasirinkti bet kurį šio projekto failą,
- 3) Pranešimų sritis („Output“), kurioje, pvz., pasirodo pranešimai apie sintaksės klaidas.

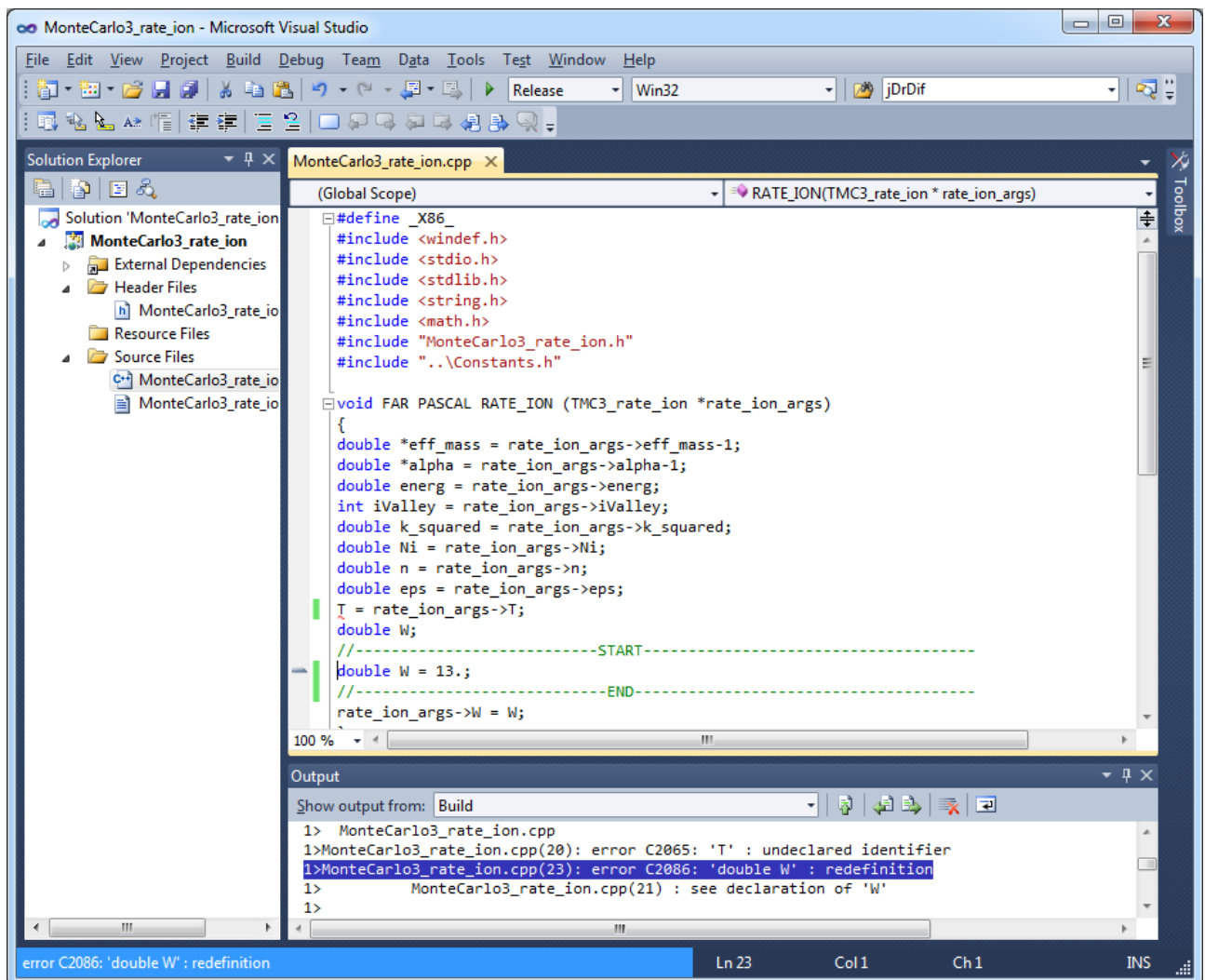
Jeigu kurios nors iš šių dalių nesimato, ją galima atidaryti naudojant meniu „View“.



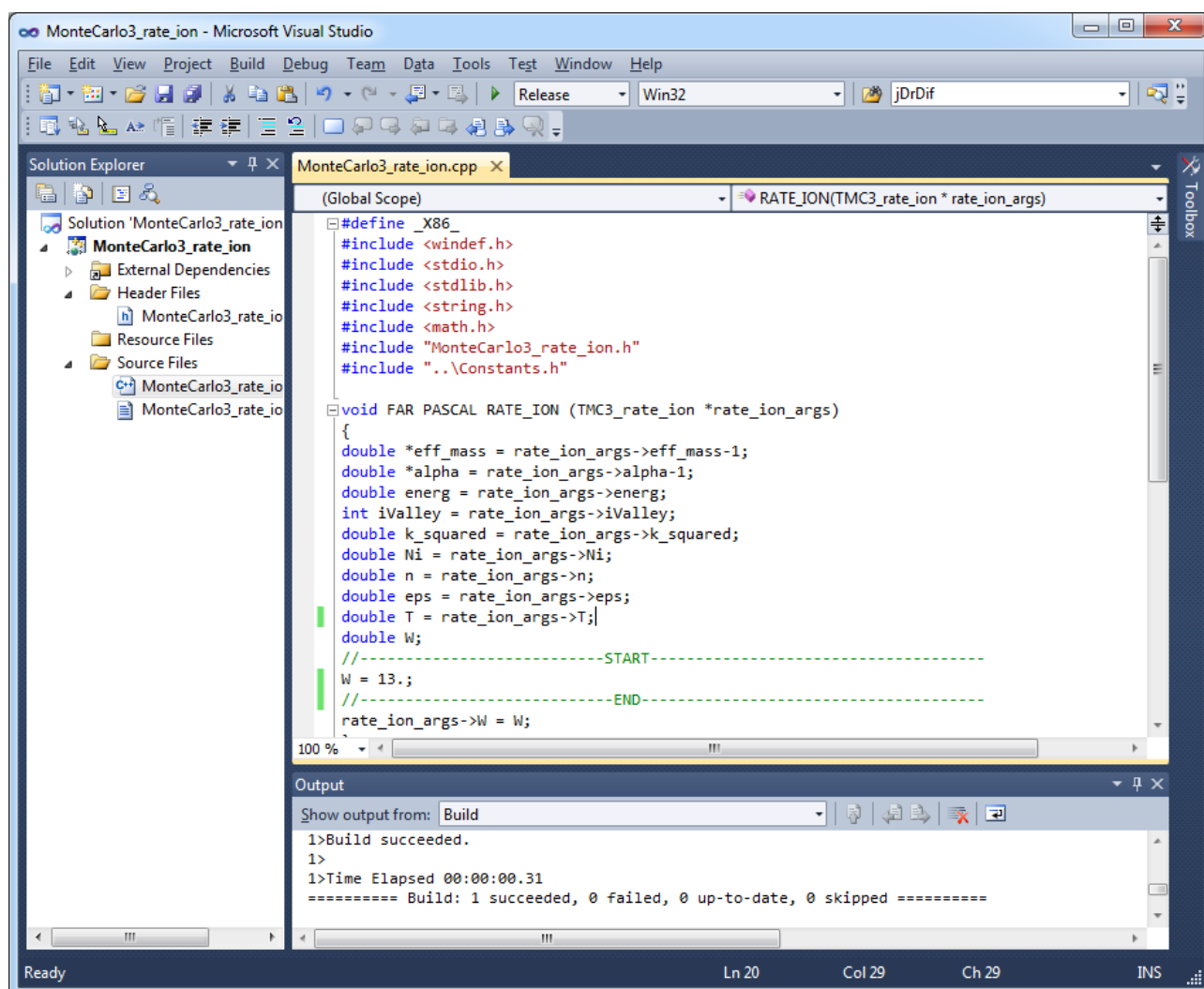
17 pav. Visual Studio 2010 langas atidarius projektą

Parašius programos tekstą, ją reikia sukompiliuoti. Tam reikia įvykdyti meniu komandą „Build / Build Solution“. Tada Visual Studio bandys sugeneruoti vykdomąjį failą, kuriame yra procesoriaus instrukcijos, atitinkančios parašytą programą (projektą, kurie naudojami šiame laboratoriniame darbe, nuostatose yra užduota, kad turi būti sukurtas .DLL failas, kuris turi būti patalpintas į aplanką C:\Modeliav\SimFit\Models). Tačiau vykdomasis failas bus sukurtas tik tuo atveju, jeigu programos tekste nėra sintaksės klaidų. Priešingu atveju lange „Output“ pasirodys pranešimai apie visas aptiktas klaidas. Dukart spustelėjus kairįjį pelės mygtuką ant kurio nors iš tų pranešimų, pelės žymeklis bus automatiškai patalpintas į atitinkamą programos teksto eilutę (žr. 18 pav.). Kai kurios sintaksės klaidos automatiškai nurodomos programos tekste raudonu vingiuotu pabraukimu (pvz., 18 pav. yra pabrauktas kintamojo vardas „T“, prieš kurį trūksta tipo specifikacijos „double“). Jeigu klaidų nėra, tada, įvykdžius meniu komandą „Build / Build Solution“, lange „Output“ pasirodo pranešimas „Build succeeded“ (žr. 19 pav.). Tai reiškia, kad vykdomasis failas buvo sėkmingai sukurtas. Tačiau tai dar nereiškia, kad programoje nėra kitokio tipo – skaičiavimo – klaidų. Pvz., nors 19 pav. programos sintaksė yra teisinga, tačiau akivaizdu, kad programa yra neteisinga, nes ji visada apskaičiuos tą pačią joninės priemaišinės sklaidos spartos reikšmę (13 s^{-1}). Tokias klaidas galima aptikti pagal skaičiavimo rezultatus (jeigu jie yra akivaizdžiai neteisingi).

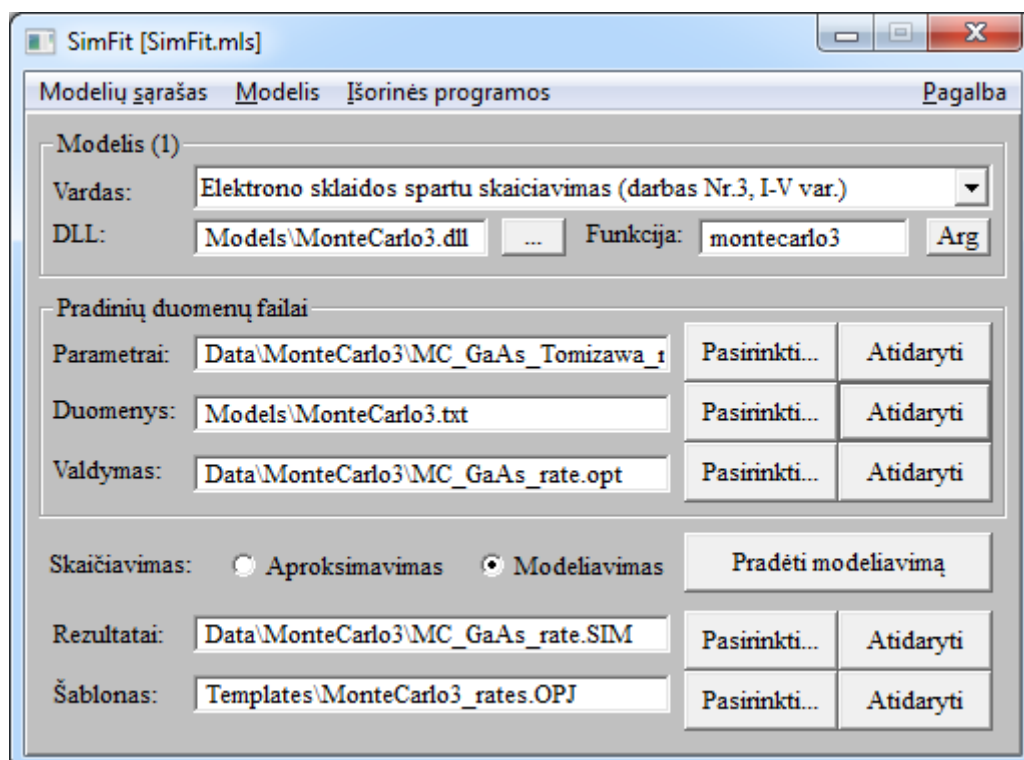
Kad patikrinti parašytą programą, reikia startuoti failą SimFit.exe, kuris yra aplanke C:\Modeliav\SimFit. Programos SimFit.exe langas yra parodytas 20 pav. Turi būti pažymėtas žymimasis laukelis „Modeliavimas“ (kaip 20 pav.). Jeigu atliekamas I – V variantas, tada modelių sąrašė „Vardas“ reikia pasirinkti pirmąją eilutę, o jeigu atliekamas VI – IX variantas, tada reikia pasirinkti antrąją eilutę. Paskui reikia atidaryti modeliuojamos sistemos parametrų failą, kurio vardas nurodyta lauke „Parametrai“. Kad atidaryti tą failą, reikia spustelėti atitinkamą mygtuką „Atidaryti“.



18 pav. Visual Studio 2010 langas įvykdžius meniu komandą „Build/Build Solution“. Yra pranešimas apie dvi klaidas: nurodytas kintamojo T tipas (prieš to kintamojo vardą turi būti „double“), o kintamasis W apibrėžtas du kartus (kad ištaisyti pastarąją klaidą, reikia ištrinti „double“ nurodytoje eilutėje)



19 pav. Visual Studio 2010 langas įvykdžius meniu komandą „Build / Build Solution“, kai nėra sintaksės klaidų



20 pav. Programos SimFit.exe pagrindinis langas

Parametrų failo fragmento pavyzdys yra pateiktas 21 pav. Failas sudarytas iš trijų blokų, kurie atskirti tuščiomis eilutėmis: medžiagos parametrų blokas, kintamųjų vardų blokas ir aproksimavimo parametrų blokas. Medžiagos parametrų bloke kiekviena eilutė yra sudaryta iš šių elementų: 1) parametro vertė, 2) parametro vardas, 3) parametro vertės išvedimo formatas, 4) komentarai. Eilutės, kuriose yra tik komentarai, prasideda komentaro simboliu – šauktuku '!'. Jeigu atliekamas I – V variantas, tada medžiagos parametrų bloke reikia surasti eilutes, kuriose yra „ijungiami“ arba „išjungiami“ sklaidos fizikiniai mechanizmai ir palikti „ijungtą“ tik vieną sklaidos mechanizmą (tą, kuris atitinka parašytą paprogramę). Jeigu kuris nors sklaidos tipas yra išjungtas, tada atitinkamo parametro vertė lygi 0. Atitinkami parametrų vardai yra:

| | |
|-------------------|--|
| „ion“ | (joninė priemaišinė sklaida); |
| „opt_pol“ | (polinė sklaida optiniais fononais); |
| „ac“ | (nepolinė sklaida akustiniais fononais); |
| „opt_intra“ | (nepolinė netarpslėnė sklaida optiniais fononais); |
| „inter1 – inter7“ | (nepolinė tarpslėnė sklaida optiniais fononais); |

Jeigu atliekamas I, II, III arba V variantas, tada parametras, kuris atitinka reikalingą sklaidos tipą, reikia prilyginti vienetui, o visus kitus minėtus parametrus reikia prilyginti 0. Jeigu atliekamas IV variantas, tada reikia prilyginti vienetui parametras inter1, o visus kitus minėtus parametrus prilyginti nuliui. Jeigu atliekamas VI – IX variantas, tada parametrai „ion“, „opt_pol“, „ac“ ir „inter1“ turi būti lygūs 1, parametras „inter3“ turi būti lygus 3, o visi kiti minėti parametrai turi būti lygūs nuliui. Įvedus visų parametrų reikšmes, reikia įrašyti parametrų failą (tam naudojama „Notepad“ meniu komanda „File / Save“ arba klavišų derinys „Ctrl + S“).

```

MC_GaAs_Tomizawa_rates.par - Notepad
File Edit Format View Help
-2e5      field      f10.2      (4) Elektrinio lauko stiprumas, v/m
300       T          f8.2       (5) Temperatūra, K
1.e+22    Ni          1pe9.1     (6) Jonizuotų gaudyklių tankis, 1/m^3 (jeigu <= 0,
! apskaičiuojamas pagal gaudyklių parametrus)
5000      nsteps    f8.2       (7) Energijos verčių skaičius sklaidos spartų lentelėje
! (energija keičiama nuo -gamma/nsteps iki -gamma,
! o jeigu Gamma>0 - nuo 2/nsteps iki 2 ev).

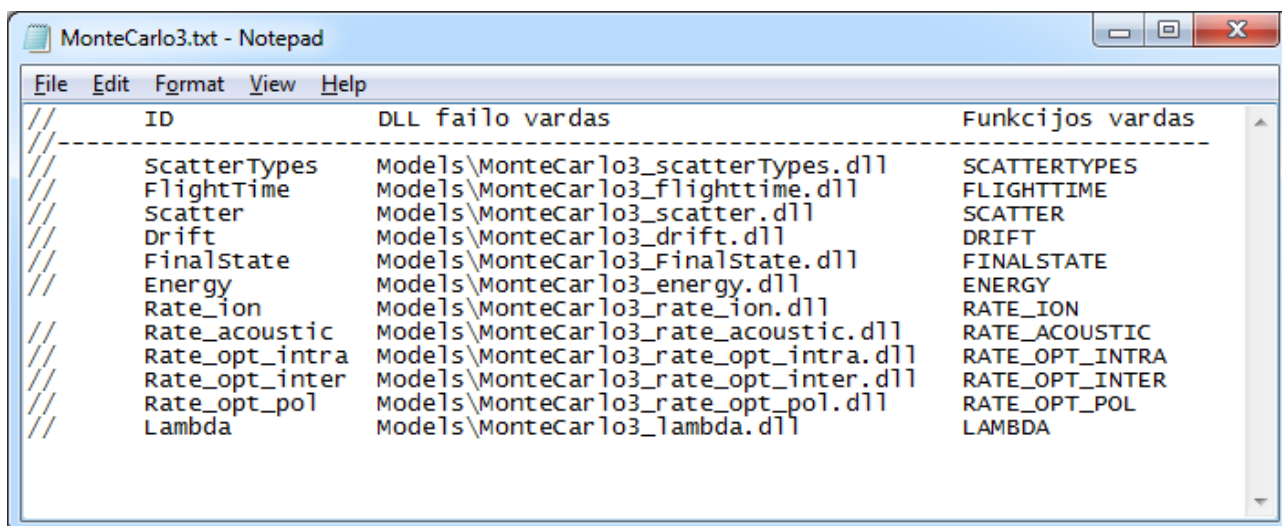
0.        E_G       f7.3       (8) G minimumo energija, ev
0.29      E_L       f7.3       (9) L minimumo energija, ev
0.52      E_X       f7.3       (10) X minimumo energija, ev
0.067     meff_G/m    f7.3       (11) Efektinės masės ir masės vakuume santykis G minimumo aplinkoje
0.35      meff_L/m  f7.3       (12) Efektinės masės ir masės vakuume santykis L minimumo aplinkoje
0.43      meff_X/m  f7.3       (13) Efektinės masės ir masės vakuume santykis X minimumo aplinkoje
0.6113    alphaG     f7.3       (14) G slenčio neparabolikumo koeficientas, 1/ev (is žinyno)
0.2465    alphaL     f7.3       (15) L slenčio neparabolikumo koeficientas, 1/ev (is žinyno)
0.36      alphaX     f7.3       (16) X slenčio neparabolikumo koeficientas, 1/ev (is žinyno)
0.64      alphaG     f7.3       (14) G slenčio neparabolikumo koeficientas, 1/ev (is žinyno)
0.46      alphaL     f7.3       (15) L slenčio neparabolikumo koeficientas, 1/ev (is žinyno)
0.20      alphaX     f7.3       (16) X slenčio neparabolikumo koeficientas, 1/ev
1         ion       f2.0       (17) Ar egzistuoja sklaida jonizuotom priemaisom?
0         opt_pol    f2.0       (18) Ar egzistuoja polinė sklaida optiniais fononais?
0         ac         f2.0       (19) Ar egzistuoja nepolinė sklaida akustiniais fononais?
0         opt_intra  f2.0       (20) Ar egzistuoja nepolinė optinė fononinė sklaida L slėnyje?
0.03536   E_p_opt_G     f8.4       (21) Polinių optinių fononų energija G slėniui, ev
0.03536   E_p_opt_L     f8.4       (22) Polinių optinių fononų energija L slėniui, ev
0.0343    E_p_opt_X  f8.4       (23) Polinių optinių fononų energija X slėniui, ev
12.9      eps       f7.3       (24) Statinė dielektrinė skvarba
10.92     epsinf    f7.3       (25) Aukštadažnė dielektrinė skvarba
7         Ksi_d_G    f7.3       (26) Akustinis deformacijos potencialas G slėniui, ev
7         Ksi_d_L    f7.3       (27) Akustinis deformacijos potencialas L slėniui, ev
7         Ksi_d_X    f7.3       (28) Akustinis deformacijos potencialas X slėniui, ev
5.24e3    vel_L     f9.1       (29) Garso greitis, m/s
5360      ro        f7.4       (30) Tankis, kg/m^3
1e11      D_opt     1pe10.2    (31) Optinės sklaidos L slėnyje deformacijos potencialas, ev/m
0.03      E_opt_intra f8.4       (32) Optinės sklaidos L slėnyje fononų energija, ev
! Tarpslėnės sklaidos tipai: 1 - GL, 2 - GX, 3 - LL,
! Ekvivalenčių slėnių skaičiai: G - 1, L - 4, X - 6.
0         inter1    f3.0       (33) Pirmojo tarpslėnės sklaidos proceso tipo
1e11      D_inter1   1pe10.2    (34) Pirmojo tarpslėnės sklaidos proceso deformacijos potencialas, ev/m
0.03      E_inter1   f8.4       (35) Pirmojo tarpslėnės sklaidos proceso fononų energija, ev
0         inter2    f3.0       (36) Antrojo tarpslėnės sklaidos proceso tipo
1e11      D_inter2   1pe10.2    (37) Antrojo tarpslėnės sklaidos proceso deformacijos potencialas, ev/m
0.026     E_inter2   f8.4       (38) Antrojo tarpslėnės sklaidos proceso fononų energija, ev
0         inter3    f3.0       (39) Trečiojo tarpslėnės sklaidos proceso tipo
1e11      D_inter3   1pe10.2    (40) Trečiojo tarpslėnės sklaidos proceso deformacijos potencialas, ev/m
0.03      E_inter3   f8.4       (41) Trečiojo tarpslėnės sklaidos proceso fononų energija, ev
0         inter4    f3.0       (42) Ketvirtąjo tarpslėnės sklaidos proceso tipo
0.9e11    D_inter4   1pe10.2    (43) Ketvirtąjo tarpslėnės sklaidos proceso deformacijos potencialas, ev/m
0.026     E_inter4   f8.4       (44) Ketvirtąjo tarpslėnės sklaidos proceso fononų energija, ev
0         inter5    f3.0       (45) Penktojo tarpslėnės sklaidos proceso tipo
0.9e11    D_inter5   1pe10.2    (46) Penktojo tarpslėnės sklaidos proceso deformacijos potencialas, ev/m
0.026     E_inter5   f8.4       (47) Penktojo tarpslėnės sklaidos proceso fononų energija, ev
0         inter6    f3.0       (48) Šeštojo tarpslėnės sklaidos proceso tipo
1e11      D_inter6   1pe10.2    (49) Šeštojo tarpslėnės sklaidos proceso deformacijos potencialas, ev/m
0.026     E_inter6   f8.4       (50) Šeštojo tarpslėnės sklaidos proceso fononų energija, ev
0         inter7    f3.0       (51) Septintojo tarpslėnės sklaidos proceso tipo
1e11      D_inter7   1pe10.2    (52) Septintojo tarpslėnės sklaidos proceso deformacijos potencialas, ev/m
0.026     E_inter7   f8.4       (53) Septintojo tarpslėnės sklaidos proceso fononų energija, ev

```

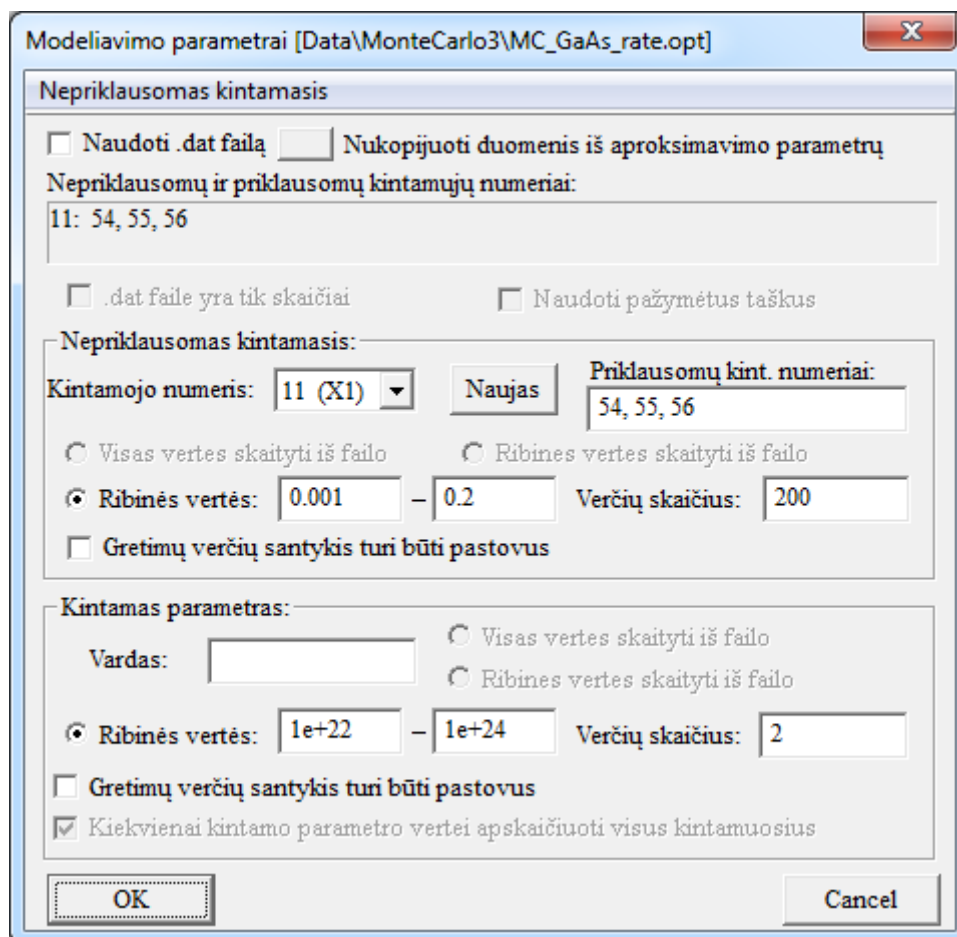
21 pav. Modeliuojamos sistemos parametrų failo fragmento pavyzdys

Įvedus medžiagos parametrus, reikia atidaryti failą MonteCarlo3.txt. Pvz., tą failą galima atidaryti spustelėjus mygtuką „Atidaryti“, kuris yra programos SimFit.exe pagrindiniame lange šalia to failo vardo (žr. 20 pav.). To failo turinio pavyzdys yra 22 pav. Kiekvienoje to failo eilutėje yra funkcijos vardas ir atitinkamo DLL failo vardas. Jeigu eilutė prasideda komentaro simboliu – dviem įstrižais brūkšniais ‘//’ – tada modeliavimo metu pagrindinė programa MonteCarlo3.dll nenaudos atitinkamo DLL failo, o vietoj jo iškvies vidinę paprogramę (kuri yra faile MonteCarlo3.dll). Todėl, norint išbandyti savo DLL failą, reikia ištrinti komentaro simbolį atitinkamoje eilutėje ir įrašyti failą MonteCarlo3.txt. Pvz. 22 pav. atveju yra išbandomas DLL failas MonteCarlo3_rate_ion.dll, kuris apskaičiuoja joninės priemaišinės sklaidos spartą.

Paskui reikia atidaryti valdymo parametrų failą, kurio vardas nurodytas programos SimFit.exe lauke „Valdymas“ (žr. 20 pav.). Kad atidaryti tą failą, reikia spustelėti mygtuką „Atidaryti“, kuris yra šalia to lauko. Tada atsidaro programos SimFit valdymo parametrų dialogo langas, kurio pavyzdys pateiktas 23 pav.



22 pav. Failo MonteCarlo3.txt pavyzdys



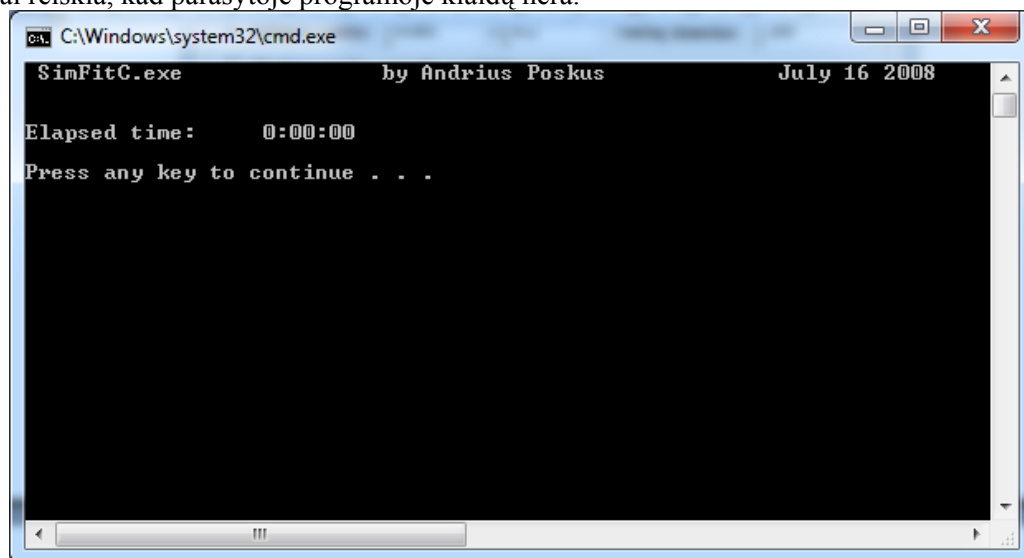
23 pav. Programos SimFit valdymo parametrų dialogo langas

Valdymo parametrų grupėje „Nepriklausomas kintamasis“ yra pasirenkamas kintamasis, kuris atlieka apskaičiuojamų funkcijų argumento vaidmenį (jis vadinamas „nepriklausomuoju kintamuoju“) ir, be to, nurodomos funkcijos, kurias reikia apskaičiuoti (t. y. „priklausomieji kintamieji“). Visi kintamieji identifikuojami jų numeriais, kurie nurodyti modeliuojamos sistemos parametrų faile (jis buvo minėtas anksčiau). Pvz., tame faile elektrinio lauko stipris pažymėtas numeriu 4, vidutinis elektrono dreifo greitis pažymėtas numeriu 8, elektrono energija pažymėta numeriu 11, pilnutinė sklaidos sparta pažymėta numeriu 54, sklaidos sparta sugeriant fononą pažymėta numeriu 55, o sklaidos sparta emituojant fononą – numeriu 56. Todėl, naudojant tokius numerius kaip parodyta 23 pav., programa apskaičiuos tris sklaidos spartas: pilnutinę, sugeriant fononą ir emituojant fononą, esant įvairioms elektrono energijos vėrtėms. Parametrų grupės „Nepriklausomas kintamasis“ laukuose „Ribinės vėrtės“ įvedamos didžiausia ir mažiausia nepriklausomojo kintamojo vėrtės, o lauke „Verčių skaičius“ įvedamas nepriklausomojo kintamojo verčių skaičius. Pvz., 23 pav. atveju elektrono energija būtų keičiama nuo 0,001 eV iki 0,2 eV kas 0,001 eV (iš viso – 200 energijos verčių). **Pastabos: 1)** Modeliuojant tampriąją sklaidą, t. y. joninę priemaišinę sklaidą arba nepolinę sklaidą akustiniais fononais, galima ištrinti numerius „55“ ir „56“, nes šiuo atveju tos dvi funkcijos būtų tapačios lygios nuliui. **2)** Parametrų grupėje „Kintamas parametras“ galima analogiškai apibrėžti dar vieną kintamąjį, t. y. antrą funkcijos argumentą, tačiau šiame darbe to nereikia, todėl tos grupės laukas „Vardas“ turi būti tuščias (tada valdymo parametrų reikšmės, kurios nurodytos po tuo lauku, nėra naudojamos).

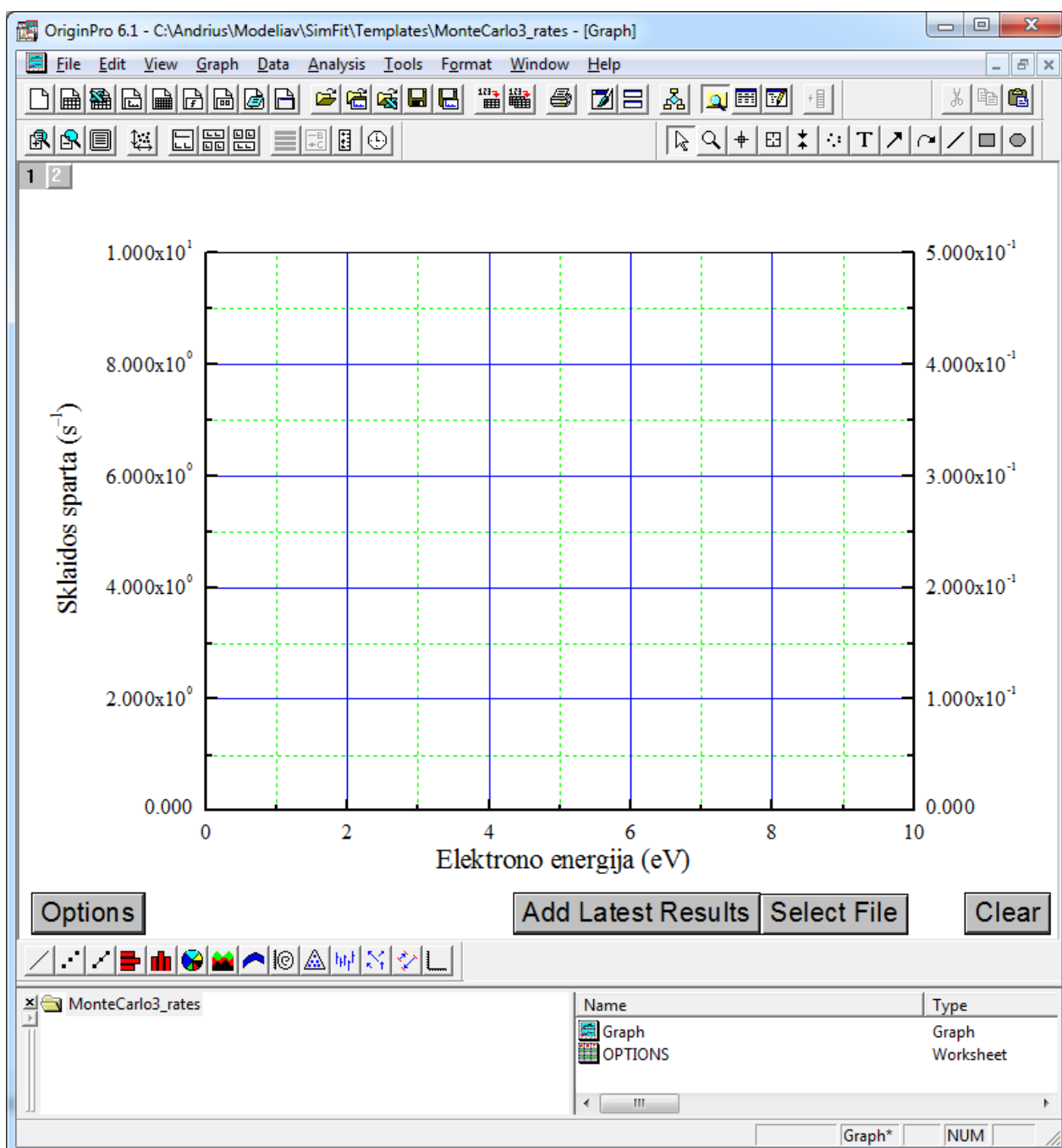
Įvedus valdymo parametrus, reikia spustelėti valdymo parametrų dialogo lango mygtuką „Gera!“, o paskui spustelėti programos „SimFit“ pagrindinio lango mygtuką „Pradėti modeliavimą“. Tada atsidaro programos SimFitC.exe langas (žr. 24 pav.). Jeigu programa sėkmingai baigia modeliavimo procesą, pasirodo užrašas „Press any key to continue...“. Tada reikia spustelėti bet kurį klaviatūros klavišą, kad tas langas užsidarytų. Uždarius programos SimFitC.exe langą, programos SimFit.exe pagrindinio lango lauke „Rezultatai“ atsiranda rezultatų failų vardas. Tai yra teksto (ASCII) failas, kurį galima atidaryti su bet kuriuo teksto redaktoriumi (pvz., spustelėjus mygtuką „Atidaryti“, kuris yra šalia to failo vardo, jis bus atidarytas naudojant redaktorių „Notepad“). Tai yra failas su dviem arba daugiau skaičių stulpelių: pirmajame stulpelyje nepriklausomoji kintamojo vėrtės, o kituose stulpeliuose – priklausomųjų kintamųjų vėrtės.

Modeliavimo metu gali išryškėti įvairios programavimo klaidos, dėl kurių programa SimFitC.exe gali būti automatiškai nutraukta nebaigus modeliavimo. Tokių klaidų pavyzdžiai – dalyba iš nulio, šaknis iš neigiamo skaičiaus, neteisingas masyvo indeksas ir pan. (tada programos SimFitC.exe lange atsiranda standartiniai pranešimai apie klaidas). Gali būti, kad programa nebus nutraukta, tačiau skaičiavimo trukmė bus neįprastai ilga arba rezultatai bus akivaizdžiai neteisingi (pvz., visos apskaičiuotos funkcijų vėrtės lygios nuliui). Tada reikia grįžti į Visual Studio aplinką ir pabandyti surasti ir ištaisyti tas klaidas. Tam galima pasinaudoti Visual Studio programų derinimo galimybėmis (menu „Debug“).

Jeigu programa sėkmingai sugeneruoja rezultatų failą ir rezultatai nėra akivaizdžiai neteisingi, tada juos reikia palyginti su rezultatais, kurie gaunami naudojant vidinę funkciją (priklausančią failui MonteCarlo3.dll). Tam reikia vėl atidaryti failą MonteCarlo3.txt ir įterpti komentaro simbolį „//“ prieš parašytos paprogramės vardą (pvz., faile, kuris pavaizduotas 22 pav., reikia įterpti „//“ prieš „Rate_ion“. Tada reikia įrašyti failą MonteCarlo3.txt ir vėl spustelėti mygtuką „Pradėti modeliavimą“. Tada programa atliks modeliavimą naudodama vidinę paprogramę (kurioje nėra klaidų). Jeigu abiem būdais gauti rezultatai sutampa, tai reiškia, kad parašytoje programoje klaidų nėra.



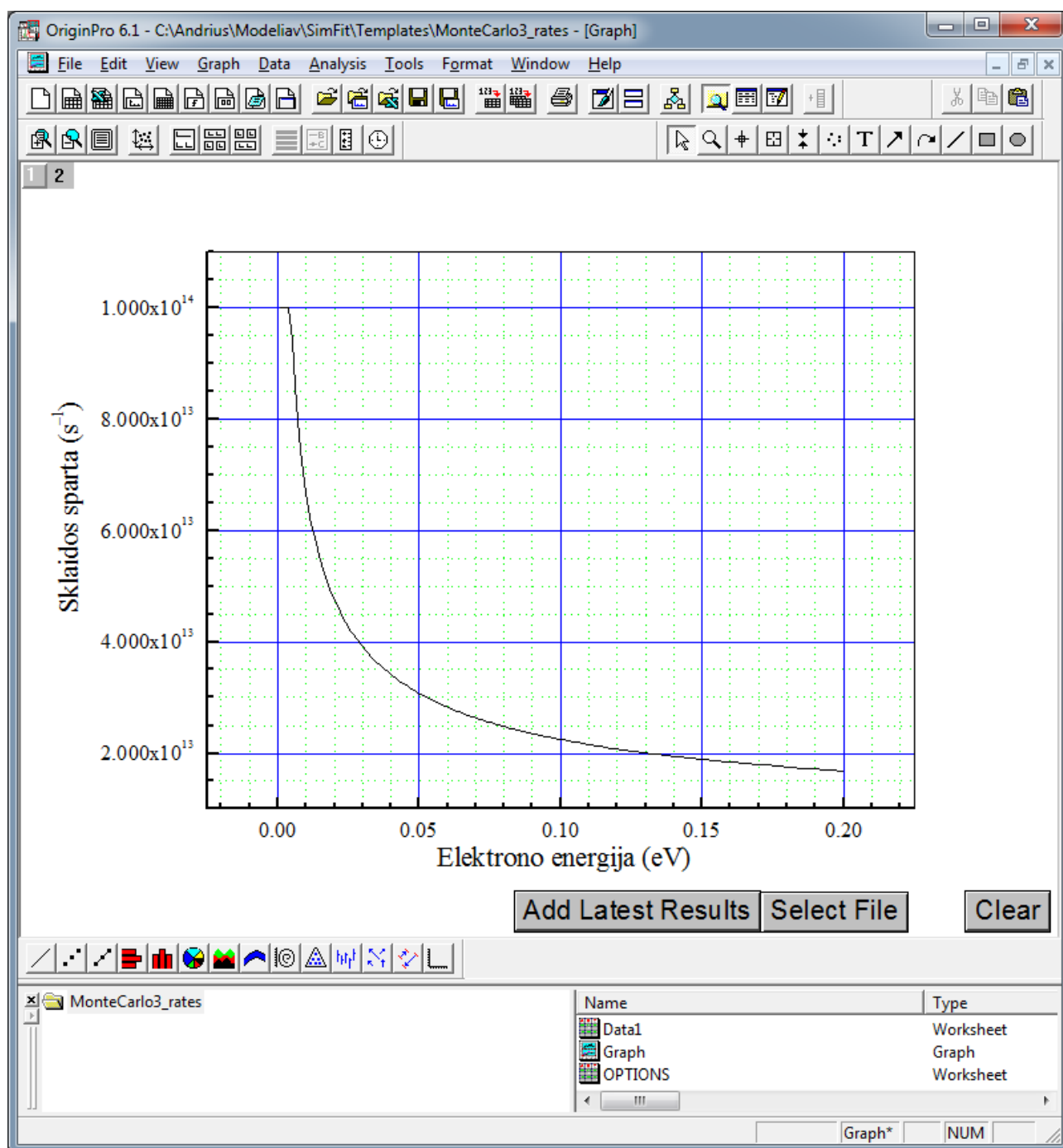
24 pav. Programos SimFitC.exe langas



25 pav. Origin failas („šablonas“), kuris pritaikytas atvaizduoti programos SimFit modeliavimo rezultatus

Į programų paketą SimFit įeina specialūs programos Origin projektai, kurie pritaikyti atvaizduoti modeliavimo rezultatus. To failo vardas yra programos SimFit lauke „Šablonas“ (žr. 20 pav.). Jis atidaromas spustelėjus mygtuką „Atidaryti“, kuris yra šalia to lauko, arba naudojant programos Origin meniu komandą „File / Open“. Atidaryto Origin šablono pavyzdys yra pavaizduotas 25 pav. Spustelėjus šio Origin projekto mygtuką „Add Latest Results“, jo grafiko lange atsiras kreivės, kurios atitinka paskutinio modeliavimo duomenis (anksčiau buvusios kreivės nebus ištrintos). Tokio grafiko pavyzdys yra pateiktas 26 pav.

Pastaba: Su mygtukais, kurie matomi 25 pav., yra susietos programos, kurios parašytos Origin Labtalk programavimo kalba. Kadangi programa SimFit buvo parašyta 2000 m. ir nuo to laiko nebuvo tobulinama, tos programos teisingai veikia tik naudojant Origin 6.x ir Origin 5.x. Su naujesnėmis Origin versijomis jos neveikia. Pvz., naudojant Origin 8.x nepavyktų įkelti SimFit rezultatų failą į Origin projektą. Todėl, jeigu yra naudojama nauja Origin versija, duomenis reikia įkelti naudojant Origin meniu komandą „File / Import / Single ASCII“. Šiuo atveju, prieš įkeliant duomenis, reikia ištrinti antraštines eilutes iš rezultatų failo.



26 pav. Programos SimFit rezultatų grafiko pavyzdys

Galutiniame darbo apraše turi būti grafikas, kuriame pavaizduotos apskaičiuotos priklausomybės, ir parašytos programos tekstas.

Priedas A. Viendalelė Monte Karlo programa MonteCarlo3.dll

A.1. Bendros pastabos apie viendalelio Monte Karlo metodo programos sudarymą

Su kiekvienu sklaidos tipu turi būti susietas jo identifikatorius (numeris). Pvz.,

- 1 – joninė priemaišinė sklaida,
- 2 – polinė fononinė sklaida,
- 3 – sklaida akustiniais fononais,
- 4 – sklaida optiniais fononais L slėnyje,
- 5 – sklaida tarp Γ ir L slėnių,
- 6 – sklaida tarp Γ ir X slėnių,
- 7 – sklaida tarp skirtingų L slėnių,
- 8 – sklaida tarp L ir X slėnių,
- 9 – sklaida tarp skirtingų X slėnių.

Jeigu sklaidoje dalyvauja optiniai fononai (procesai 2 ir 4 – 9), tada sklaida yra netamprioji, t. y. elektrono energijos pokytis priklauso nuo to, ar fononas sugeriamas, ar emituojamas. Vadinasi, kiekvienas iš šių procesų tikrovėje atitinka dviejų galimų procesų porą, kurioje vienas procesas atitinka fonono sugertį, o kitas – fonono emisiją. Taigi, jeigu vyksta 1 – 9 tipo sklaida, tada iš viso egzistuoja $2 + 7 \cdot 2 = 16$ sklaidos tipų.

Lankstesniam sistemos aprašymui turėtų būti numatyta galimybė užduoti, kurie sklaidos procesai vyksta, o kurie – ne. Tai galima pasiekti, pvz., kiekvienam iš minėtųjų sklaidos tipų pradinių duomenų faile priskyrus loginį kintamąjį, kurio vertė 1 reiškia, kad šis procesas vyksta, o vertė 0 reiškia, kad šis procesas nevyksta. Pagal šiuos kintamuosius programa turėtų nustatyti vykstančių („aktyviųjų“) procesų skaičių **nScatter**, sunumeruoti juos (**iScatter** = 1, 2, ..., **nScatter**) ir kiekvienam iš jų apskaičiuoti visus dydžius (koeficientus), kurie reikalingi, skaičiuojant to proceso spartą. Šie koeficientai bendroju atveju priklauso nuo pradinio slėnio. Vadinasi, kiekvienos rūšies koeficientai sudaro dvimatį masyvą **Coef[iValley][iScatter]** (**iValley** = 1, 2, 3); čia pirmasis indeksas (**iValley**) nusako pradinį slėnį (pvz., 1 atitinka Γ slėnį, 2 – L slėnį, 3 – X slėnį). Koeficientų **Coef** rūšys yra šios:

- Vykstančio („aktyviojo“) proceso identifikatorius (sklaidos tipas) **iScat**. Pvz., galima būtų naudoti aukščiau užrašytus numerius. Jeigu procese dalyvauja optiniai fononai, tada fonono emisiją galima susitarti žymėti teigiamu identifikatoriumi, o fonono sugertį – neigiamu identifikatoriumi. Pvz., 2 galėtų reikšti polinę fononinę sklaidą, išspinduliuojant fononą, o -2 – polinę fononinę sklaidą, sugeriant fononą.
- Elektrono energijos pokytis, vykstant duotosios rūšies sklaidai, **energyIncr**. Procesų 1 ir 3 atveju šis pokytis lygus nuliui (akustinę fononinę sklaidą galima apytiksliai laikyti tampriaja), o procesų 2 ir 4 – 11 atveju jis yra lygus arba priešingas fonono energijai: $\pm \hbar\omega$. Reikia turėti omenyje, kad visų rūšių sklaidos procesų sparta priklauso nuo *skirtumo tarp elektrono energijos ir slėnio energijos minimumo* (žr. sklaidos spartų išraiškas). Todėl skaičiavimų metu elektrono energiją racionaliausia visada atskaityti nuo slėnio, kuriame jis yra (arba kuriame jis atsiduria po sklaidos), energijos minimumo. Todėl procesų 5, 6 ir 8 atveju prie $\pm \hbar\omega$ dar reikia pridėti pradinio ir galutinio slėnių energijų skirtumą.
- Galutinis slėnis, kuriame atsiduria elektronas, įvykus duotosios rūšies sklaidai, **iFinalValley**. Sklaidos 1 – 4 atveju galutinis slėnis lygus pradiniam slėniui (t. y. **iValley**), o sklaidos 5 – 11 atveju galutinis slėnis vienareikšmiškai priklauso nuo sklaidos tipo **iScat** ir pradinio slėnio **iValley**. Pvz., jeigu **iScat** = ± 5 , o **iValley** = 2, tada **iFinalValley** = 1 (šuoelis $L \rightarrow \Gamma$). Jeigu **iScat** = ± 5 , o **iValley** = 1, tada **iFinalValley** = 2 (šuoelis $\Gamma \rightarrow L$).

Pastaba: Siekiant palengvinti programos skaitymą ir derinimą, pagrindinėms fizikinėms ir matematinėms konstantoms reikėtų priskirti lengvai suprantamus žymėjimus. Visos reikalingos konstantos su žymėjimų pavyzdžiais yra pateiktos žemiau. Medžiagos parametrus taip pat patartina priskirti lengvai suprantamus vardus, pvz., **eff_mass**[3].

Pagrindinės konstantos, kurios reikalingos, skaičiuojant sklaidos spartą

| Prasmė | Žymėjimas | Vertė | Kintamasis |
|--|----------------------|---------------------------------|---------------|
| Skaičius "pi" | π | 3.14159265358979324 | pi |
| Elektrono krūvis | e | $1.6021892 \cdot 10^{-19}$ C | elch |
| Elektrinė konstanta | ϵ_0 | $8.8541878 \cdot 10^{-12}$ F/m | eps0 |
| Bolcmano konstanta | k | $1.380658 \cdot 10^{-23}$ J/K | boltz |
| Elektronvoltais išreikšta Bolcmano konstanta | k / e | $8.6173218 \cdot 10^{-5}$ eV/K | boltz_el |
| Elektrono masė vakuume | m | $9.1093897 \cdot 10^{-31}$ kg | elmass |
| Planko konstanta | h | $6.626176 \cdot 10^{-34}$ J·s | planck |
| Elektronvoltais išreikšta Planko konstanta | h / e | $4.1357013 \cdot 10^{-15}$ eV·s | planck_el |
| Redukuotoji Planko konstanta | $\hbar = h / (2\pi)$ | $1.0545886 \cdot 10^{-34}$ J·s | planck_red |
| Elektronvoltais išreikšta redukuotoji Planko konstanta | \hbar / e | $6.582173 \cdot 10^{-16}$ eV·s | planck_red_el |
| Šviesos greitis vakuume | c | 299792458 m/s | vel_light |

Elektronvoltais išreikštos Bolcmano ir Planko konstantos naudingos tada, kai programa operuoja elektronvoltais išreikštomis elektronų ir fononų energijomis.

Be medžiagos parametrų ir temperatūros, pradinį duomenų faile turi būti pateikti keli modeliavimo proceso valdymo parametrai. Šie parametrai yra:

- Elektrinio lauko stiprio ribinės vertės ir verčių skaičius. Šie parametrai reikalingi todėl, kad modeliavimo tikslas – ištirti elektrono dreifinio greičio ir vidutinės energijos priklausomybę nuo elektrinio lauko stiprio. Taigi, visus žemiau aprašytus skaičiavimus reikės pakartoti kelioms elektrinio lauko stiprio vertėms.
- Pilnutinis sklaidos įvykių skaičius, esant kiekvienam lauko stipriui. Šis skaičius nusako, po kelinto sklaidos įvykio programa turi nutraukti elektrono judėjimo modeliavimą ir apskaičiuoti elektrono vidutinį greitį, vidutinę energiją bei judrį. Siekiant sumažinti atsitiktines paklaidas, šis skaičius turi būti kuo didesnis (ne mažiau už 10^6).
- Atsitiktinių skaičių generatoriaus inicializavimo parametras (angliškas terminas "seed"). Šis parametras naudingas, patikrinant, kokią įtaką turi atsitiktinės paklaidos (t. y. ar pakankamai didelis sklaidos įvykių skaičius). Jeigu, naudojant skirtingas šio parametro vertes, gaunami beveik vienodi rezultatai, tai reiškia, kad sklaidos įvykių skaičius yra pakankamas.
- Didžiausia elektrono energija ***E_{max}***. Šis parametras nusako energijos intervalą, kuriam sudaroma sklaidos spartų lentelė (žr. žemiau). Kambario temperatūroje šis parametras neturėtų viršyti kelių elektronvoltų, pvz., 3 eV. Jeigu yra abejonių dėl šio parametro vertės, skaičiavimus reikia pakartoti, esant kelioms šio parametro vertėms, ir palyginti rezultatus: jeigu rezultatai nesiskiria, tai reiškia, kad didžiausia elektrono energija yra pakankamai didelė. Pastaba: kai elektrono energijos vertė tampa didesnė už draustinės juostos plotį, tampa įmanoma smūginė jonizacija, kurios metu elektronas išlaisvina dar vieną elektroną iš valentinės juostos. Kadangi šio vyksmo metu laisvųjų elektronų skaičius padidėja vienetu, tai viendalelis Monte Karlo metodas šiuo atveju netinka.
- Elektrono energijos verčių skaičius ***nSteps***. Šis skaičius lemia intervalą tarp energijos verčių, sudarant sklaidos spartų lentelę (žr. žemiau): šis intervalas lygus ***E_{max} / nSteps***.

Siekiant sumažinti skaičiavimų trukmę, prieš elektrono judėjimo modeliavimą reikia sudaryti 5.3 skyriuje apibrėžtų funkcijų $A_j(E)$ verčių lentelę ir judėjimo modeliavimo metu imti $A_j(E)$ vertes iš šios lentelės ($A_j(E)$ – tai visų sklaidos procesų su numeriais nuo 1 iki j spartų suma, padalinta iš pilnutinės sklaidos spartos, į kurią įeina virtualiosios sklaidos sparta). Tada nereikia po kiekvieno sklaidos įvykio skaičiuoti iš naujo visas sklaidos spartas, ir pilnutinė skaičiavimų trukmė sumažėja kelis kartus. Naudojant šią sklaidos spartų lentelę, padidėja skaičiavimų paklaidos (nes diskretizuojama elektrono energija), tačiau šias paklaidas visada galima sumažinti, optimizavus energijos verčių skaičių ir intervalą tarp jų. Funkcijų $A_j(E)$ verčių lentelė yra trimatis masyvas, kurio pirmieji du indeksai nusako pradinį slėnį ir sklaidos tipo numerį (t. y. aukščiau minėtieji ***iValley*** ir ***iScatter***), o trečiasis indeksas nusako tam tikrą fiksuotą elektrono energijos vertę:

$\Lambda[iValley][iScatter][iStep]$ ($iValley = 1, 2, 3$; $iScatter = 1, 2, \dots, nScatter$; $iStep = 1, 2, \dots, nSteps$). Čia $iStep$ yra energijos intervalo („žingsnio“) numeris; $nSteps$ yra tokių intervalų pilnutinis skaičius. Elektronų energijos vertės, kurios atitinka kiekvieną indekso $iStep$ reikšmę ($E_{el}[iStep]$) turi būti parinktos taip, kad, iš vienos pusės, intervalas tarp jų būtų kuo mažesnis (siekiama padidinti tikslumą), o iš kitos pusės, kad elektronų energija niekada netaptų mažesnė už $E_{el}[1] = E_{max} / nSteps$ arba didesnė už $E_{el}[nSteps] = E_{max}$. Todėl $E_{el}[1]$ turėtų būti žymiai mažesnė už elektronų vidutinę šiluminio judėjimo energiją kT , o E_{max} turėtų būti lygi keliems elektronvoltams. Pvz., galima naudoti vertes $E_{max} = 3$ eV ir $nSteps = 3000$. Reikia turėti omenyje, kad visos šios energijos vertės atskaitomos nuo atitinkamo slėnio dugno, t. y., norint nusakyti elektronų tikrąją energiją, reikia pridėti atitinkamo slėnio dugno energiją.

Skaičiuojant masyvo $\Lambda[3][nScatter][nSteps]$ elementus, randama ir konstanta Γ , kuri reikalinga funkcijų $A_j(E)$ normavimui (žr. formulę (5.3.1)) bei lėkio trukmės skaičiavimui (žr. (5.2.5)): ši konstanta yra lygi didžiausiai pilnutinei sklaidos spartai tiriamajame energijų intervale. Ši konstanta turi būti apibrėžta kiekvienam slėniui atskirai, t. y. reikalingas trijų elementų masyvas $\Gamma[iValley]$ ($iValley = 1, 2, 3$).

A.2. Viendalelės Monte Karlo programos MonteCarlo3.dll struktūra

Programos MonteCarlo3.dll, kuri modeliuoja elektronų judėjimą vienalyčiame elektriniame lauke, struktūrinė schema pavaizduota 27a–b pav. Žemiau pateiktas pagrindinių programos etapų aprašymas.

Paruošiamieji veiksmai:

1. Medžiagos parametrai, valdymo parametrai ir pradinė elektronų būseną skaitomi iš pradinių duomenų failų (žr. paprogramių *readParms* ir *readInitState* aprašymus A.3 skyriuje).
2. Pagal pradinius duomenis nustatomi aktyviųjų sklaidos tipų skaičius, kiekvieno sklaidos tipo identifikatoriai bei kiekvieną sklaidos tipą ir kiekvieną pradinį slėnį atitinkantys elektronų energijos pokyčiai ir galutiniai slėniai (žr. *scatterTypes*).
3. Apskaičiuojama funkcijų A_j verčių lentelė.
4. Inicializuojamas atsitiktinių skaičių generatorius.

Pats svarbiausias paruošiamasis žingsnis, prieš pradedant modeliuoti elektronų judėjimą duotajame elektriniame lauke – tai funkcijų A_j verčių lentelės sudarymas (žr. formulę (5.3.1) ir A.1 skyriaus paskutinės dvi pastraipas). Tuo tikslu atliekamas dviejų lygių ciklas – atžvilgiu slėnio $iValley = 1, 2, 3$ ir energijos vertės numerio $iStep = 1, 2, \dots, nSteps$. Esant kiekvienai indeksui $iValley$ ir $iStep$ porai, iškviečiama sklaidos spartų skaičiavimo paprogramė (žr. paprogramės *rates* aprašymą), kuri apskaičiuoja $nScatter$ konstantų A_j ($j = 1, \dots, nScatter$). Šios vertės priskiriamos trimačio masyvo elementams $\Lambda(iValley, iScatter, iStep)$ (čia $iScatter = 1, 2, \dots, nScatter$).

Kaip matyti formulėje (5.3.1), į konstantų A_n apibrėžimą įeina normavimo daliklis Γ . Šis daliklis turėtų būti lygus didžiausiai pilnutinei sklaidos spartai duotajame energijų intervale, kai elektronas yra duotajame slėnyje. Taigi, normavimo dalikliai Γ sudaro trijų elementų masyvą $\Gamma[iValley]$ ($iValley = 1, 2, 3$). Šis masyvas turėtų būti naudojamas kaip paprogramės *rates* įėjimo parametras. Tačiau, kadangi iš pradžių normavimo dalikliai yra nežinomi, tai visų pirma dydžiai $\Lambda[iValley][iScatter][iStep]$ apskaičiuojami, laikant, kad visi trys normavimo dalikliai $\Gamma[iValley]$ yra lygūs 1, ir kiekvienai $iValley$ reikšmei randama didžiausioji Λ reikšmė. Šie trys skaičiai ir yra ieškomieji $\Gamma[iValley]$. Tokiu būdu radus normavimo daliklius Γ , visi masyvo Λ elementai padalinami iš atitinkamų normavimo konstantų, t. y. atliekamas pakeitimas

$$\Lambda[iValley][iScatter][iStep] = \Lambda[iValley][iScatter][iStep] / \Gamma[iValley].$$

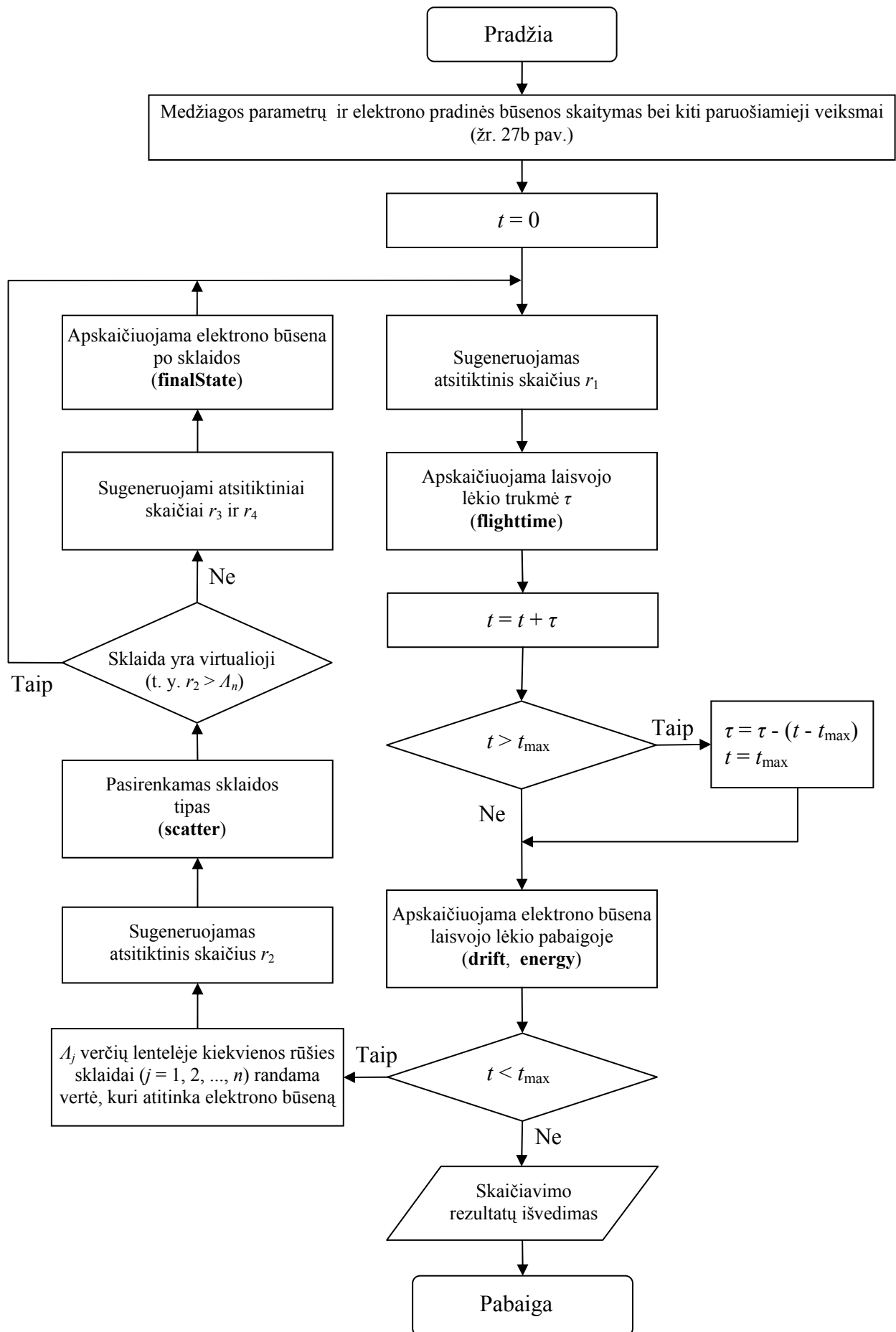
Atlikus šiuos paruošiamuosius veiksmus, pradedamas ciklas atžvilgiu elektrinio lauko stiprio verčių. Kiekvienam elektrinio lauko stipriui sumodeliuojamas duotasis sklaidos įvykių skaičius. T. y. atliekamas antrojo lygio ciklas, kurio kintamasis yra sklaidos įvykio numeris. Prieš pradedant šį ciklą, elektroną reikia "patalpinti" į pradinį slėnį, o jo banginiam vektoriui ir koordinatėms reikia priskirti pradines vertes. Be to, prieš pradedant šį ciklą, laikas ir sklaidos įvykių skaičius turi būti prilyginti nuliui. Šio ciklo kiekvienas žingsnis prasideda dreifu ir baigiasi sklaidos įvykiu. Kiekviename šio ciklo žingsnyje (t. y. tarp dviejų sklaidos įvykių) programa atlieka šiuos veiksmus:

5. Naudojant atsitiktinį skaičių r_1 (žr. 5.2 skyrių), sugeneruojama laisvojo lėkio trukmė (žr. *flighttime*).
6. Apskaičiuojamos elektrono bangos vektoriaus komponentės laisvojo lėkio pabaigoje (žr. formulę (5.2.6) ir *drift*).
7. Apskaičiuojami elektrono energija (žr. *energy*) ir šie dydžiai:
 - 1) pilnutinis laikas,
 - 2) elektrono energijos integralas laiko atžvilgiu (reikalingas vidutinės energijos skaičiavimui),
 - 3) pilnutinis elektrinio lauko atliktas darbas (reikalingas dreifinio greičio skaičiavimui).
8. Kiekvienos rūšies sklaidai funkcijų A_j verčių lentelėje randama vertė, kuri geriausiai atitinka einamąją elektrono būseną (t. y. slėnį ir energiją). Tokiu būdu suformuojamas vienmatis masyvas, kurio elementai nusako A_j vertes (žr. *lambda*).
9. Naudojant atsitiktinį skaičių r_2 (žr. 5.3 skyrių), pasirenkamas sklaidos tipas (žr. *scatter*) ir sklaidos įvykių skaičius padidinamas vienetu (net jeigu pasirinktoji sklaida yra virtualioji).
10. Jeigu pasirinktasis procesas nėra virtualusis (t. y. jeigu r_2 nevirsija A_n , kur n yra aktyviųjų procesų skaičius), tada apskaičiuojamos elektrono energija po sklaidos (5.3.3), bangos vektoriaus absoliutinė vertė po sklaidos (5.3.5), ir, naudojant atsitiktinius skaičius r_3 ir r_4 (žr. 5.3 skyrių), apskaičiuojama bangos vektoriaus kryptis (žr. *finalState*).

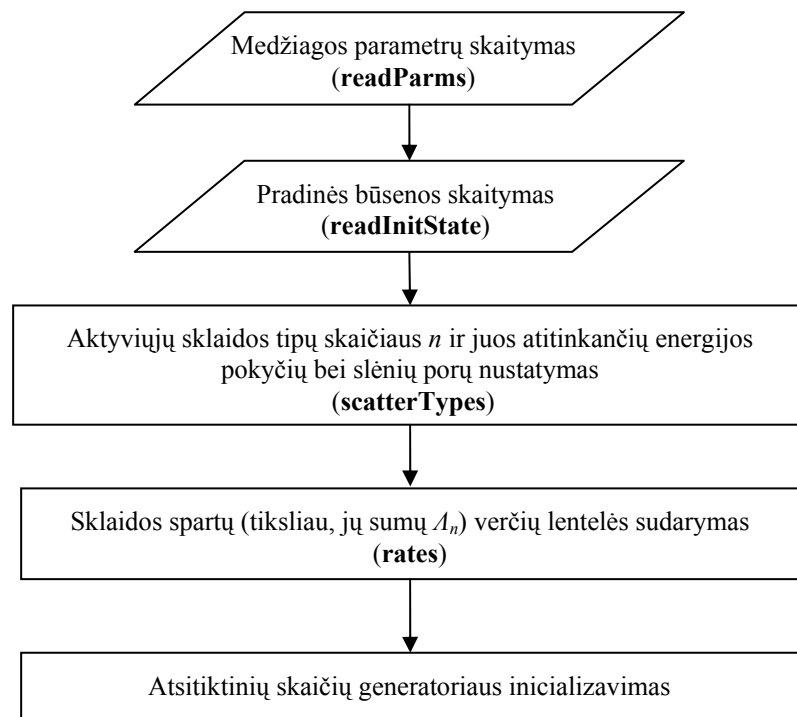
Šis ciklas baigiamas, kai pilnutinė modeliuojamojo proceso trukmė arba pilnutinis sklaidos įvykių skaičius trukmė pasiekia duotąją didžiausiąją reikšmę. Tada programa apskaičiuoja elektrono vidutinę energiją, dreifinį greitį ir judrį, įrašo šiuos duomenis į failą (paprogramė *output*), pakeičia elektrinio lauko stiprį ir vėl pakartoja tuos pačius veiksmus. Skaičiavimų rezultatas – ASCII failas su keturiais skaičių stulpeliais:

- 1) elektrinio lauko stipris,
- 2) elektrono dreifinis greitis,
- 3) elektrono judris,
- 4) elektrono vidutinė energija.

A.3 skyriuje pateiktos paprogramių, kurios naudojamos aukščiau aprašytiems veiksams, antraštės ir paaiškinti jų argumentai.



27a pav. Viendalelės Monte Karlo programos MonteCarlo3.dll struktūrinė schema.



27b pav. Paruošiamieji veiksmai, vykdant viendalelę Monte Karlo programą MonteCarlo3.dll.

A.3. Pagrindinių paprogramių antraštės

Paprogramė **readParms** skaito medžiagos parametrus ir valdymo parametrus iš pradinių duomenų failo.

Paprogramės readParms argumentai (dydžiai, kurie skaitomi iš failo):

| | |
|------------------|---|
| field_min | – mažiausias elektrinio lauko stipris (V/m) |
| field_max | – didžiausias elektrinio lauko stipris (V/m) |
| nField | – elektrinio lauko stiprio verčių skaičius |
| nScat | – didžiausias sklaidos įvykių skaičius |
| seed | – atsitiktinių skaičių generatoriaus inicializavimo parametras |
| E _{max} | – didžiausia elektrono energija (atžvilgiu slėnio dugno), eV |
| nSteps | – energijos verčių skaičius sklaidos spartų lentelėje |
| ion | – ar vyksta sklaida jonizuotom priemaisom? (1 – taip, 0 – ne) |
| opt_pol | – ar vyksta polinė sklaida optiniais fononais? |
| ac | – ar vyksta nepolinė sklaida akustiniais fononais? |
| opt_intra | – ar vyksta nepolinė optinė fononinė sklaida L slėnyje? |
| iPair[1...7] | – septynių elementų masyvas, kurio vertės nusako slėnių poras, kurios dalyvauja tarpslėnėje sklaidoje. Galimos vertės: 1 – sklaida tarp tarp Γ ir L slėnių, 2 – sklaida tarp tarp Γ ir X slėnių, 3 – sklaida tarp skirtingų L slėnių, 4 – sklaida tarp L ir X slėnių, 5 – sklaida tarp skirtingų X slėnių, 6 – sklaida tarp dviejų X slėnių, kurie yra priešinguose Brijueno zonos kraštuose, 7 – sklaida tarp dviejų gretimų X slėnių. Jeigu kuris nors šio masyvo elementas yra mažesnis už 1 arba didesnis už 7, tada laikoma, kad atitinkamas procesas nevyksta, t. y. programa ignoruoja atitinkamas sklaidos parametrų D_inter ir E_inter vertes (žr. žemiau). |
| T | – temperatūra, K |
| Ni | – jonizuotų priemaišų tankis, m^{-3} |
| E[1,2,3] | – trijų elementų masyvas, kurio elementų vertės nusako slėnių energijas, eV (pirmasis elementas atitinka Γ slėnį, antrasis – L slėnį, trečiasis – X slėnį) |
| eff_mass[1,2,3] | – trijų elementų masyvas, kurio elementų vertės nusako elektrono efektinės masės ir masės vakuume santykius kiekviename slėnyje |
| alpha[1,2,3] | – slėnių neparaboliškumo koeficientai, $1/eV$ |
| E_opt_pol[1,2,3] | – polinių optinių fononų energijos kiekvienam slėniui, eV |
| eps | – statinė dielektrinė skvarba |
| epsinf | – aukštadažnė dielektrinė skvarba |
| Ksi_d[1,2,3] | – akustiniai deformacijos potencialai kiekvienam slėniui, eV |
| vel_L | – išilginių garso bangų greitis, m/s |
| ro | – tankis, kg/m^3 |
| D_opt_intra | – optinės sklaidos L slėnyje deformacijos potencialas, eV/m |
| E_opt_intra | – optinės sklaidos L slėnyje fononų energija, eV |
| D_inter[1...7] | – tarpslėnės sklaidos procesų, kuriuos nusako masyvo iPair atitinkami elementai, deformacijos potencialai, eV/m |
| E_inter[1...7] | – tarpslėnės sklaidos procesų fononų energijos, eV |

Paprogramė **readInitState** skaito pradinę elektrono būseną iš failo „MonteCarloInit.txt“:

readInitState(iValley0, kx0, ky0, kz0, x0, y0, z0)

Argumentai (dydžiai, kurie skaitomi iš failo):

iValley0 – pradinis slėnis (1, 2 arba 3)

kx0 – pradinė bangos vektoriaus Dekarto dedamoji X kryptimi, t. y. elektrinio lauko kryptimi (matavimo vienetas – m^{-1})

ky0 – pradinė bangos vektoriaus Dekarto dedamoji Y kryptimi

kz0 – pradinė bangos vektoriaus Dekarto dedamoji Z kryptimi

x0 – pradinė x koordinatė (m)

y0 – pradinė y koordinatė

z0 – pradinė z koordinatė

Elektrono bangos vektorius, kaip ir elektrono energija, visada atskaitomas nuo slėnio, kuriame yra elektronas, dugno. Todėl parametrų kx0, ky0 ir kz0 vertės turėtų būti žymiai mažesnės už Brijueno zonos matmenis. Tačiau pradinės būsenos pasirinkimas neturi didelės reikšmės, nes po kelių dešimčių sklaidos įvykių elektronas "užmiršta" savo pradinę būseną. Pvz., bet kuriuos du iš parametrų kx0, ky0 ir kz0 galima prilyginti nuliui, o trečiąjį apskaičiuoti, remiantis sąlyga, kad elektrono energija lygi vidutinei šiluminio judėjimo energijai kT .

Paprogramė **scatterTypes** nustato aktyviųjų sklaidos tipų skaičių, kiekvieno sklaidos tipo identifikatorių bei kiekvieną sklaidos tipą ir kiekvieną pradinį slėnį atitinkančius elektrono energijos pokyčius ir galutinius slėnius:

scatterTypes (ion, opt_pol, ac, opt_intra, E, E_opt_pol, E_opt_intra, E_inter, iPair,
nScatter, iScat, energyIncr, iFinalValley)

Pirmieji 9 argumentai (nuo ion iki iPair) yra paprogramės readParms apskaičiuotieji dydžiai (šiuo atveju jie naudojami kaip "įėjimo parametrai"). Paskutiniųjų keturių argumentų ("išėjimo parametrų") vertės apskaičiuoja paprogramė scatterTypes:

nScatter – vykstančiųjų ("aktyviųjų") sklaidos tipų skaičius,

iScat – vienmatis masyvas, kurio ilgis nScatter ir kurio elementai nusako aktyviųjų procesų tipus (galimos vertės – 1, ± 2 , 3, ± 4 , ..., ± 9),

energyIncr – dvimatis masyvas, kurio pirmasis indeksas nusako pradinį slėnį (nuo 1 iki 3), antrasis indeksas nusako aktyviojo sklaidos tipo numerį (nuo 1 iki nScatter), o kiekvieno elemento vertė nusako elektrono energijos pokytį, vykstant duotosios rūšies sklaidai iš duotojo pradinio slėnio,

iFinalValley – dvimatis masyvas, kurio indeksai turi tą pačią prasmę, kaip energyIncr indeksai, o kiekvieno elemento vertė nusako galutinį slėnį po sklaidos.

Paprogramė **rates** apskaičiuoja kiekvieno aktyviojo sklaidos proceso spartą, esant duotajai elektrono energijai duotajame slėnyje:

rates (nScatter, iScat, energyIncr, iFinalValley, iPair,
T, Ni, eff_mass, E_opt_pol, eps, epsinf, Ksi_d, vel_L, ro, E_opt_intra, D_opt_intra,
D_inter, E_inter, alpha, Gam, iValley, energ, k_abs, k_squared, aLambda, W)

Įėjimo parametrai:

Pirmieji 4 argumentai yra paprogramės scatterTypes apskaičiuotieji dydžiai,

argumentus Nr. 5–19 (nuo iPair iki alpha) nuskaitė paprogramė readParms,

Gam – trijų elementų masyvas, kuris nusako sklaidos spartų normavimo konstantos Γ vertes kiekvienam slėniui (šie dydžiai apskaičiuojami pagrindinėje programoje),

iValley – slėnis, kuriame yra elektronas prieš pat sklaidą (galimos vertės – 1, 2, 3),

energ – nuo slėnio dugno atskaityta elektrono energija,

k_abs – elektrono bangos vektoriaus modulis (bangos skaičiaus),

k_squared – elektrono bangos vektoriaus kvadratas.

Išėjimo parametrai:

aLambda – vienmatis masyvas, kurio ilgis lygus aktyviųjų sklaidos tipų skaičiui (nScatter) ir kurio n -tasis elementas nusako procesų su numeriais nuo 1 iki n spartų sumą einamajai elektrono būsenai,
W – kiekvieno aktyviojo sklaidos proceso sparta einamojoje būsenoje.

Kiekvienos rūšies sklaida apskaičiuojama atskiroje paprogramėje, kurią iškviečia paprogramė **rates**. Kiekviena iš tų paprogramių turi vienintelį išėjimo parametą – duotosios rūšies sklaidos sparta W (t. y. tikimybė per laiko vienetą). Žemiau yra pateiktos tų paprogramių antraštės:

Paprogramė **rate_ion** apskaičiuoja joninės priemaišinės sklaidos spartą:

rate_ion(eff_mass, alpha, energ, iValley, k_squared, Ni, n, eps, T, W)

Čia „n“ yra laisvųjų elektronų koncentracija (paprogramėje **rates** kintamajam „n“ priskiriama priemaišinių jonų koncentracijos vertė, t. y. $n = Ni$).

Paprogramė **rate_acoustic** apskaičiuoja akustinės fononinės sklaidos spartą:

rate_acoustic(eff_mass, alpha, energ, iValley ro, vel_L, Ksi_d, T, W)

Paprogramė **rate_opt_intra** apskaičiuoja optinės nepolinės (ir netarpslėnės) sklaidos L slėnyje spartą (kai optinė nepolinė sklaida vyksta Γ arba X slėnyje, tada ji visada yra tarpslėnė, todėl jos spartą skaičiuoja kita paprogramė):

rate_opt_intra (eff_mass, alpha, energ, iValley, finalEnergy, E_opt_intra, D_opt_intra, isAbs, ro, T, W)

Čia argumentas „isAbs“ nusako, ar optinis fononas yra sugeriamas, ar emituojamas. Jeigu $isAbs = 1$, tada optinis fononas yra sugeriamas (elektrono energija padidėja), o jeigu $isAbs = 0$, tada optinis fononas yra emituojamas (elektrono energija sumažėja). Argumentas „finalEnergy“ nusako galutinę elektrono energiją (atskaitytą nuo slėnio dugno).

Paprogramė **rate_opt_inter** apskaičiuoja optinės nepolinės tarpslėnės sklaidos spartą:

rate_opt_inter (eff_mass, alpha, energ, iValley, finalEnergy, finalValley, E_inter, D_inter, isAbs, ro, T, W)

Čia argumentas „finalValley“ nusako galutinį slėnį, o „finalEnergy“ nusako galutinę elektrono energiją (atskaitytą nuo galutinio slėnio dugno)

Paprogramė **rate_opt_pol** apskaičiuoja optinės polinės (netarpslėnės) sklaidos spartą:

rate_opt_pol (eff_mass, alpha, energ, iValley, finalEnergy, E_opt_pol, eps, epsinf, isAbs, T, W)

Paprogramė **lambda** iš sklaidos spartų sumų lentelės, kurią nusako trimatis masyvas **Lambda** (žr. A.1 skyriaus paskutinės dvi pastraipas ir A.2 skyrių), išrenka **nScatter** verčių, kurios atitinka einamąją elektrono energiją ir einamąjį slėnį:

lambda(lambda_table, nScatter, nSteps, de, iValley, energ, aLambda)

Iėjimo parametrai:

lambda_table – trimatis masyvas **Lambda[iValley][iScatter][iStep]** su visomis sklaidos spartų sumų vertėmis,

nScatter – aktyviųjų sklaidos tipų skaičius,

nSteps – elektrono energijos verčių („diskretizavimo žingsnių“) skaičius,

de – gretimų energijos verčių skirtumas („diskretizavimo žingsnis“),

iValley – elektrono energijos slėnio numeris,

energ – elektrono energija (atskaitoma nuo slėnio dugno).

Išėjimo parametrai:

aLambda – nScatter ilgio vienmatis masyvas, kurio elementai lygūs apytikslėms funkcijų A_n vertėms (jos paimtos iš verčių lentelės **Lambda[iValley][iScatter][iStep]**).

Paprogramė **scatter** pasirenka sklaidos tipą:

scatter(nScatter, aLambda, r2, iScatter)

Įėjimo parametrai:

nScatter – aktyviųjų sklaidos tipų skaičius,

aLambda – nScatter ilgio vienmatis masyvas, kurio elementai lygūs apytikslėms funkcijų A_n vertėms (jos paimtos iš verčių lentelės **Lambda[iValley][iScatter][iStep]**),

r2 – atsitiktinis skaičius, kuris tolygiai pasiskirstęs intervale nuo 0 iki 1.

Išėjimo parametras:

iScatter – sklaidos įvykio tipas. Galimos vertės – nuo 0 iki nScatter. Jeigu iScatter = 0, tai reiškia, kad sklaida yra virtualioji, t. y. sklaidos faktiškai nėra.

Paprogramė **finalState** apskaičiuoja elektrono būseną (slėnį, energiją ir bangos vektorių) iš karto po sklaidos:

finalState (iScat, energyIncr, iFinalValley, eff_mass, alpha, n, T, eps, iScatter, r3, r4, iValley, energ, k_abs, k_squared, kx, ky, kz)

Įėjimo parametrai:

argumentai nuo Nr. 1 (iScat) iki Nr. 8 (eps) aprašyti aukščiau. Vienintelis pakeitimas yra tas, kad eff_mass[1,2,3] yra tikrosios efektinės masės (kg), o ne santykinės masės;

iScatter – paskutiniojo sklaidos įvykio rūšies eilės numeris.

r3 ir r4 – atsitiktiniai skaičiai, kurie tolygiai pasiskirstę intervale nuo 0 iki 1,

Išėjimo parametrai:

iValley – slėnis, kuriame elektronas atsidūrė po sklaidos,

energ – elektrono energija šio slėnio dugno atžvilgiu,

k_abs – elektrono bangos vektoriaus modulis po sklaidos,

k_squared – elektrono bangos vektoriaus kvadratas po sklaidos,

kx, ky, kz – elektrono bangos vektoriaus komponentės galutinio slėnio dugno atžvilgiu.

Paprogramė **flighttime** sugeneruoja elektrono laisvojo lėkio trukmę:

flighttime (Gam, iValley, r1, tau)

Įėjimo parametrai:

Gam – trijų elementų masyvas, kurį sudaro pagrindinėje programoje apskaičiuoti dalikliai Γ (žr. A.2 skyrių ir A.1 skyriaus pabaigą),

iValley – slėnis, kuriame yra elektronas.

r1 – atsitiktinis skaičius, kuris tolygiai pasiskirstęs intervale nuo 0 iki 1,

Išėjimo parametras:

tau – laisvojo lėkio trukmė.

Paprogramė **drift** apskaičiuoja elektrono bangos vektorių laisvojo lėkio pabaigoje (prieš pat kitą sklaidos įvykį):

`drift(eff_mass, iValley, tau, field, x, y, z, kx, ky, kz, k_abs, k_squared)`

Įėjimo parametrai:

`eff_mass` – efektinės masės kiekviename slėnyje (kg),

`iValley` – slėnis,

`tau` – lėkio trukmė,

`field` – elektrinio lauko stipris,

`ky`, `kz` – elektrono bangos vektoriaus Dekarto dedamosios statmena laukui kryptimi (laukas nukreiptas X ašies kryptimi).

Įėjimo/išėjimo parametrai:

`x`, `y`, `z` – elektrono Dekarto koordinatės,

`kx` – elektrono banginio vektoriaus x dedamoji (dedamosios `ky` ir `kz` laisvojo lėkio metu nekinta, nes elektrinis laukas nukreiptas X kryptimi)

Išėjimo parametrai:

`k_abs` – bangos vektoriaus modulis,

`k_squared` – bangos vektoriaus kvadratas.

Paprogramė **energy** apskaičiuoja elektrono energiją laisvojo lėkio pabaigoje (prieš pat kitą sklaidos įvykį):

`energy(eff_mass, alpha, iValley, k_squared, energ)`

Įėjimo parametrai:

`eff_mass` – elektrono efektinės masės kiekviename slėnyje (kg),

`alpha` – kiekvieno slėnio neparaboliškumo koeficientai,

`iValley` – slėnis,

`k_squared` – elektrono bangos vektoriaus kvadratas (atžvilgiu slėnio dugno),

Išėjimo parametras:

`energ` – elektrono energija.

Paprogramė **output** įrašo skaičiavimų rezultatus į tekstinį (ASCII) failą:

`output(field, drift_velocity, miu, mean_energy)`

Argumentai (dydžiai, kurie rašomi į failą):

`field` – elektrinio lauko stipris,

`drift_velocity` – elektrono dreifinis greitis,

`miu` – elektrono judris,

`mean_energy` – elektrono vidutinė energija.

Priedas B. Bendros žinios apie Monte Karlo modeliavimo laboratorinių darbų atlikimo tvarką

Monte Karlo modeliavimo laboratoriniai darbai atliekami, naudojant modeliavimo ir netiesinio aproksimavimo sistemą **SimFit**. Šią sistemą sudaro dvi programos:

- 1) aproksimavimo modelių redaktorius **SimFit.exe**;
- 2) modelių skaičiavimo bei aproksimavimo programa **SimFitC.exe**.

Aproksimavimui **SimFit** naudoja vartotojo sudarytuose DLL failuose esančias funkcijas, o duomenų grafiniam atvaizdavimui naudoja programą **Origin**.

Galimi du **SimFit** darbo režimai – aproksimavimo ir modeliavimo. Aproksimavimo atveju **SimFit** apskaičiuoja teorinio modelio, kurį nusako duotoji funkcija, parametrų vertes, kurios atitinka mažiausią teorinio skaičiavimo duomenų nuokrypį nuo duotųjų matavimo duomenų. Modeliavimo atveju **SimFit** apskaičiuoja teorinio modelio priklausomųjų kintamųjų vertes, kurios atitinka duotąsias modelio parametrų ir nepriklausomųjų kintamųjų (argumentų) vertes. Pastarosios vertės gali būti skaitomos iš duotojo failo arba gali būti apskaičiuotos, naudojantis vartotojo pateiktomis ribinėmis vertėmis.

SimFit.exe sudaro modelių sąrašą, kiekvienam modeliui sudaro sąrašą su failų, kuriuose saugomi skaičiavimo parametrai, vardais, pateikia grafinį interfeisą kai kurių valdymo parametrų modifikavimui. Kiekvienam modeliui vienas iš tų parametrų yra DLL failo su vartotojo sudaryta programa vardas (pvz., **MonteCarlo3.dll**). **SimFit.exe** iškviečia programą **SimFitC.exe**, kuri atlieka visus skaičiavimus (tame tarpe – ir aproksimavimą).

Programa **SimFitC.exe** skaito parametrų vertes iš failų, kurių sąrašą sudarė **SimFit.exe**, ir periodiškai iškviečia vartotojo programą (pvz., **MonteCarlo3.dll**), naudodama iš anksto apibrėžtą jos argumento verčių seką.

Vartotojo DLL programa (pvz., **MonteCarlo3.dll**) apskaičiuoja duotojo priklausomojo kintamojo (pvz., elektrono dreifo greičio arba judrio arba sklaidos spartos) vertę, esant duotai argumento (pvz., elektrinio lauko stiprio arba elektrono energijos) vertei ir duotoms pastoviujų parametrų (pvz., medžiagos parametrų) vertėms.

Kai kurios paprogramės, kurias iškviečia programa **MonteCarlo3.dll**, turi du variantus: "vidinė" paprogramė (ji įeina į pačią programą **MonteCarlo3.dll**) ir "išorinė" paprogramė (ji yra atskirame DLL faile). Nustatymui, kurią iš tų funkcijų naudoti, programa **MonteCarlo3.dll** naudoja failą **"MonteCarlo3.txt"**. Tame faile yra minėtųjų paprogramių ir atitinkamų DLL failų vardai (po vieną vardą eilutėje). Jeigu eilutė prasideda komentaro simboliu "//", tuomet atitinkamus skaičiavimus atlieka vidinė paprogramė, o priešingu atveju tuos skaičiavimus atlieka išorinė paprogramė, kurios ieškoma atitinkamame DLL faile.

Analogišką tekstinį failą „MonteCarlo5.txt“ naudoja ir daugiadalelė Monte Karlo programa „MonteCarlo5.dll“.

Failas „MonteCarlo3.txt“ turi būti tame pačiame kataloge, kaip failas MonteCarlo3.dll, o failas „MonteCarlo5.txt“ turi būti tame pačiame kataloge, kaip failas MonteCarlo5.dll. Žemiau yra failo „MonteCarlo3.txt“ turinio pavyzdys.

| // ID | DLL failo vardas | Funkcijos vardas |
|-------------------|---------------------------------------|------------------|
| ----- | | |
| // ScatterTypes | Models\MonteCarlo3_scatterTypes.dll | SCATTERTYPES |
| // FlightTime | Models\MonteCarlo3_flighttime.dll | FLIGHTTIME |
| // Scatter | Models\MonteCarlo3_scatter.dll | SCATTER |
| // Drift | Models\MonteCarlo3_drift.dll | DRIFT |
| // FinalState | Models\MonteCarlo3_FinalState.dll | FINALSTATE |
| // Energy | Models\MonteCarlo3_energy.dll | ENERGY |
| // Rate_ion | Models\MonteCarlo3_rate_ion.dll | RATE_ION |
| // Rate_acoustic | Models\MonteCarlo3_rate_acoustic.dll | RATE_ACOUSTIC |
| Rate_opt_intra | Models\MonteCarlo3_rate_opt_intra.dll | RATE_OPT_INTRA |
| // Rate_opt_inter | Models\MonteCarlo3_rate_opt_inter.dll | RATE_OPT_INTER |
| // Rate_opt_pol | Models\MonteCarlo3_rate_opt_pol.dll | RATE_OPT_POL |
| // Lambda | Models\MonteCarlo3_lambda.dll | LAMBDA |

Tie DLL failai rašomi programavimo kalba C++ ir kompiliuojami kompiliatoriumi Microsoft Visual Studio 2010 Professional. Tačiau DLL failus galima rašyti ir bet kuria kita kalba, naudojant bet kurią kitą kompiliatorių, kuris gali generuoti DLL failus: vienas iš DLL panaudojimo privalumų yra tas, kad jų programavimo kalba gali skirtis nuo juos iškviečiančios programos programavimo kalbos (pvz., anksčiau minėtosios programos „MonteCarlo3.dll“ ir „MonteCarlo5.dll“ parašytos FORTRAN-90 kalba).

C++ kalba yra papildyta C kalba. Esminė C++ kalbos ypatybė yra ta, kad programose, kurios parašytos ta kalba, galima naudoti vadinamąsias „klases“. Tačiau aptariamuoju atveju klasės nenaudojamos. T. y. minėtosios paprogramės rašomos C kalba, naudojant kai kurias neesmines C++ kalbos ypatybes.

Kiekvienam iš minėtųjų failų yra sudarytas atskiras Visual Studio projektas ir yra parašytas „šablonas“ – .cpp failas, kuriame yra paprogramės antraštė ir jos argumentų apibrėžtys. Žemiau yra paprogramės DRIFT, kurią iškviečia programa MonteCarlo3.dll, šablonas:

```
#define _X86_
#include <windef.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "MonteCarlo3_drift.h"
#include "..\Constants.h"

void FAR PASCAL DRIFT (TMC3_drift *drift_args)
{
double *eff_mass = drift_args->eff_mass-1;
int iValley = drift_args->iValley;
double tau = drift_args->tau;
double field = drift_args->field;
double x = drift_args->x;
double y = drift_args->y;
double z = drift_args->z;
double kx = drift_args->kx;
double ky = drift_args->ky;
```

```

double kz = drift_args->kz;
double k_abs = drift_args->k_abs;
double k_squared = drift_args->k_squared;
//-----START-----
//-----END-----
drift_args->x = x;
drift_args->y = y;
drift_args->z = z;
drift_args->kx = kx;
drift_args->k_abs = k_abs;
drift_args->k_squared = k_squared;
}

```

Programos dalis, kurią reikia parašyti, yra tarp komentarų eilučių su žodžiais „START“ ir „END“ (C++ kalboje komentarų eilutės pradedamos simboliu „/“). Prieš eilutę su žodžiu „START“ yra visų argumentų apibrėžtys.

Visų argumentų prasmė yra paaiškinta Priede A. Be to, ten nurodyta, kurie argumentai yra „įėjimo“, o kurie – „išėjimo“. Įėjimo argumentai yra tie, kurių vertės yra žinomos iš anksto. Juos apskaičiuoja pagrindinė programa (pvz., MonteCarlo3.dll), prieš iškviisdama duotąją paprogramę. „Išėjimo“ argumentai yra tie, kurių vertes turi apskaičiuoti duotoji paprogramė. Skaičiavimams naudojamos formulės, kurios pateiktos šio darbo aprašo teorinėje dalyje.

Visos minėtosios paprogramės naudoja antraštinį failą „Constants.h“, kuriame yra apibrėžti kintamieji, kurių vertės lygios pagrindinių fizikinių konstantų vertėms. Todėl visose formulėse vietoj minėtųjų konstantų galima naudoti jų vardus (tie vardai paaiškinti lentelėje, kuri yra Priedo A skyrelyje A.1). Pvz., vietoj Planko konstantos galima rašyti „planck“.

Rašant programas, reikia turėti omenyje, kad programoje MonteCarlo3.dll energija visur išreikšta elektronvoltais (visi kiti dydžiai išreikšti SI sistemos vienetais). Todėl, jeigu vienas iš duotosios paprogramės įėjimo parametrų yra energija, o skaičiavimus ketinama atlikti SI sistemoje, tada, prieš pradedant skaičiavimus, tą energiją reikia išreikšti džauliais. Analogiškai, jeigu paprogramėje skaičiavimai atliekami SI sistemoje, o vienas iš išėjimo parametrų yra energija, tada, prieš išeinant iš paprogramės, tą energiją reikia išreikšti elektronvoltais.

Skaičiuojant sklaidos spartą, pradinis slėnis („iValley“) skaitomas iš failo „MonteCarloInit.txt“, kuris turi būti tame pačiame kataloge, kaip ir programa SimFitC.exe. Atliekant modeliavimą viendaleliu Monte Karlo metodu, pradinė elektrono būseną skaitoma iš to paties failo. Atliekant modeliavimą daugiadaleliu Monte Karlo metodu, minėtasis failas nėra naudojamas, nes dalelių pradinės būsenos yra generuojamos, naudojant atsitiktinių skaičių generatorių. Skaičiuojant sklaidos spartą, iš failo MonteCarloInit.txt skaitomas tik slėnio numeris (jis nurodytas pirmojoje minėtojo failo eilutėje).

Priedas C. Programavimo C kalba pradmenys

C.1. Duomenų tipai

Pagal C kalbos taisykles, kiekvieno kintamojo apibrėžtis prasideda jo tipo vardu, po kurio yra kintamojo vardas. Po kintamojo vardo gali būti lygybės ženklas ir to kintamojo pradinė vertė (konstanta arba aritmetinis reiškiny). Galimieji duomenų tipai yra šie:

`int` – sveikasis skaičius su ženklu (4 baitai; galimos reikšmės – nuo $-2^{31} + 1$ iki $2^{31} - 1$),
`unsigned` – sveikasis skaičius be ženklo (4 baitai; galimos reikšmės – nuo 0 iki $2^{32} - 1$),
`float` – slankaus taško skaičius (4 baitai, tikslumas – 7 reikšminiai skaitmenys),
`double` – dvigubo tikslumo slankaus taško skaičius (8 baitai, tikslumas – 15 reikšminių skaitmenų).

Specialus kintamojo tipas yra vadinamieji **masyvai**. Masyvas yra atminties sritis, kurioje saugomi keli skaičiai. Masyvo apibrėžtis ypatinga tuo, kad joje reikia nurodyti masyvo dydį. Masyvo dydis nurodomas tarp laužtinių skliaustų. Pvz., žemiau apibrėžtas vienmatis slankaus taško skaičių masyvas, kurio dydis lygus 3:

```
double eff_mass[3];
```

(C kalboje visi operatoriai baigiami kabliataškiu). Į atskirus masyvo elementus taip pat kreipiamasi, naudojant laužtinius skliaustus, tarp kurių nurodomas elemento indeksas (eilės numeris duotajame masyve). Numeruojama, pradedant nuo 0. Pvz., aukščiau apibrėžtojo masyvo atveju galimos indekso vertės yra 0, 1 ir 2.

Kitas specialus kintamojo tipas yra vadinamosios **rodyklės**. Rodyklė – tai kintamasis, kurio prasmė yra kito kintamojo adresas. Rodyklė apibrėžiama, prieš jos vardą įterpiant žvaigždutę. Pvz., žemiau apibrėžta rodyklė į slankaus taško kintamąjį:

```
double *pointer;
```

Šiame pavyzdyje rodyklės `pointer` vertė yra neapibrėžta. Žemiau tai rodyklei priskiriamas masyvo `eff_mass[3]` pirmojo elemento adresas:

```
pointer = eff_mass;
```

Masyvo vardas yra specialaus tipo rodyklė. Kaip matome pastarajame pavyzdyje, masyvo vardą galima naudoti kaip rodyklę į to masyvo pirmąjį elementą. Prie bet kurios rodyklės pridėjus sveikąjį skaičių, gaunama rodyklė į kitą to paties tipo kintamąjį, kurio adresas skiriasi nuo pradinio adreso dydžiu, kuris lygus to sveikąjo skaičiaus ir to tipo kintamojo dydžio sandaugai. Pvz., jeigu rodyklė `pointer` apibrėžta taip, kaip aukščiau, tada `pointer+1` nusako rodyklę į masyvo `eff_mass` antrąjį elementą (t. y. į `eff_mass[1]`). Kintamojo, į kurį nukreipta rodyklė, vertę galima sužinoti, naudojant „deadresavimo“ operatorių „*“. Pvz., pastarajame pavyzdyje `*pointer` arba `*eff_mass` yra lygus `eff_mass[0]`, o `*(pointer+1)` arba `*(eff_mass+1)` yra lygus `eff_mass[1]`.

Iki šiol buvo kalbama tik apie vienmačius masyvus. Dvimatį masyvą galima apibrėžti, pvz., taip:

```
int a[10][20];
```

Tokia apibrėžtis rodo, kad dvimatis masyvas `a` turi 10 eilučių, o kiekviena eilutė turi po 20 elementų. Kaip žinome, masyvo vardas – tai rodyklė į jo pirmąjį elementą. Dvimačio masyvo elementai kompiuterio atmintyje išdėstyti eilutėmis (t. y., pereinant nuo vieno elemento prie gretimo, pirmiausia kinta antrasis indeksas). Todėl, naudojant aukščiau pateiktąją masyvo `a` apibrėžtį, užrašymas „`*(a + 19)`“ yra tapatus „`a[0][19]`“, o užrašymas „`*(a + 20)`“ yra tapatus „`a[1][0]`“. Dvimačio masyvo atveju savo vardus turi ir jo eilutės. Minėtame pavyzdyje – tai yra vienmačiai masyvai `a[0]`, `a[1]`, `a[2]`, ..., `a[9]`. Vėl pasinaudoję tuo, kad masyvo vardas

yra rodyklė į jo pirmąjį elementą, gauname, kad užrašymas „`*a[0]`“ yra tapatus „`a[0][0]`“, užrašymas „`*a[1]`“ yra tapatus „`a[1][0]`“ ir t.t.

Dar vienas specialusis kintamojo tipas yra vadinamosios *struktūros*. Struktūra nuo masyvo skiriasi tuo, kad jos elementai gali būti skirtingų tipų. Minėtųjų paprogramių, kurias reikės parašyti, visi argumentai yra tam tikrų struktūrų laukai. Kiekviena struktūra apibrėžiama, naudojant operatorių „`typedef struct`“. Pvz., žemiau apibrėžta struktūra `TMC3_drift`, kurios laukai yra paprogramės DRIFT argumentai.

```
typedef struct
{
    double *eff_mass;
    int iValley;
    double tau;
    double field;
    double *x;
    double *y;
    double *z;
    double *kx;
    double ky;
    double kz;
    double *k_abs;
    double *k_squared;
} TMC3_drift;
```

Apibrėžus struktūrą, jos vardą galima naudoti kaip kintamojo tipą kintamųjų apibrėžtyse. Pvz., žemiau apibrėžtas kintamasis `tempStruct`, kurio tipas yra `TMC3_drift`:

```
TMC3_drift tempStruct;
```

Atskirų struktūros laukų vertes galima sužinoti, naudojant taško operatorių. Pvz., jeigu kintamasis `tempStruct` apibrėžtas taip, kaip aukščiau, tada yra leistini tokie operatoriai:

```
tempStruct.eff_mass[0] = 1.;
tempStruct.field = 0.;
```

Turint rodyklę į struktūrą, jos atskirų laukų vertes galima sužinoti, naudojant operatorių „`->`“. Pvz., jeigu `ptr` yra rodyklė į aukščiau apibrėžtą tipo `TMC3_drift` struktūrą, tada minėtuosius du operatorius reikia rašyti taip:

```
ptr->eff_mass[0] = 1.;
ptr->field = 0.;
```

C kalboje kintamųjų apibrėžtys turi būti pačioje programos pradžioje (prieš visas aritmetines operacijas). Tačiau C++ kalba leidžia talpinti kintamųjų apibrėžtis bet kurioje programos vietoje. Reikia turėti omenyje, kad kintamasis, kuris apibrėžtas sudėtinio operatoriaus viduje (t. y. tarp riestinių skliaustų „`{`“ ir „`}`“), yra „matomas“ tik to operatoriaus viduje. Pvz., žemiau pateiktajame pavyzdyje kintamojo `i` reikšmė sudėtinio operatoriaus viduje yra 2, o to paties kintamojo reikšmė, išėjus iš to operatoriaus, yra lygi 1:

```
int i = 1;
{
    ...
    int i = 2;
    ...
}
```

C.2. Operacijos

Pagrindiniai operacijų tipai:

- aritmetinės operacijos,
- palyginimo operacijos,
- loginės operacijos,
- priskyrimo operacijos,
- sekos operacija (kablelis).

Aritmetinės operacijos:

- sudėtis (+),
- atimtis (−),
- dalyba (/),
- daugyba (*).

Palyginimo operacijos:

- “lygu” (==),
- “nelygu” (!=),
- “mažiau” (<),
- “mažiau arba lygu” (<=),
- “daugiau” (>),
- “daugiau arba lygu” (>=).

Visų palyginimo operacijų rezultatas yra `int` tipo skaičius: jeigu duotoji sąlyga galioja, tada rezultatas yra vienetas, o priešingu atveju – nulis. Pvz., reiškinio `2==2` rezultatas yra 1, o reiškinio `1>2` rezultatas yra 0. Todėl palyginimo operacijas galima naudoti reiškiniuose kartu su aritmetinėmis operacijomis (tačiau reikia turėti omenyje, kad palyginimo operacijų prioritetas yra mažesnis, negu aritmetinių operacijų).

Loginės operacijos:

- “ir” (&&),
- “arba” (||),
- neigimas (!).

Loginių operacijų argumentai („operandai“) gali būti bet kokie reiškiniai, kurių rezultatas yra skaičius. Visi skaičiai, kurie skiriasi nuo nulio, interpretuojami kaip „tiesa“ (t. y. kaip vienetas). Loginių operacijų reikšmės skaičiuojamos iš kairės į dešinę ir nutrūksta, kai tik galima vienareikšmiškai nustatyti rezultatą. Pvz., jeigu operacija „arba“ atliekama su dviem reiškiniais, kurių pirmasis skiriasi nuo nulio, tada programa nebando skaičiuoti antrojo reiškinio, nes nuo jo reikšmės galutinis rezultatas nepriklauso.

Priskyrimo operacijos:

- dešinėsios pusės reikšmės priskyrimas kariajai pusei (=),
- kairėsios pusės padidinimas dydžiu, kuris lygus dešiniajai pusei (+=),
- kairėsios pusės sumažinimas dydžiu, kuris lygus dešiniajai pusei (−=),
- kairėsios pusės pakeitimas, dauginant ją iš dydžio, kuris yra dešiniojoje pusėje (*=),
- kairėsios pusės pakeitimas, dalijant ją iš dydžio, kuris yra dešiniojoje pusėje (/=),
- dydžio padidinimas vienetu (++), pvz., `i++`
- dydžio sumažinimas vienetu (−−), pvz., `i−−`.

Operacija „kablelis“ vartojama reiškinių tarpusavio sujungimui. Reiškinių, kurie atskirti vienas nuo kito kableliu, sąrašas yra interpretuojamas kaip vienas reiškinys ir skaičiuojamas iš kairės į dešinę. Toks reiškinys visada įgauna paskutiniojo skaičiavimo reikšmę.

C.3. Operatoriai

Pagrindiniai operatorių tipai:

- sąlygos operatoriai,
- ciklo operatorius,
- funkcijos iškviatimo operatoriai.

Visi operatoriai baigiami kabliataškiu. Žemiau pateiktose operatorių apibrėžtyse žodis „operatorius“ gali reikšti ir vieną operatorių, ir sudėtinį operatorių, t. y. kelis operatorius, kurie surašyti tarp riestinių skliaustų „{“ ir „}“. Po uždarančiojo skliausto „}“ kabliataškis nerašomas.

Sąlygos operatoriai yra keturių rūšių:

1. `if` (loginis reiškinys)
operatorius
2. `if` (loginis reiškinys)
operatorius_1
`else`
operatorius_2
3. loginis reiškinys ? reiškinys_1 : reiškinys_2
4. operatorius `switch ... case` (ši operatorių dažniausiai galima pakeisti vienu iš aukščiau minėtųjų sąlygos operatorių, todėl čia jis neaprašomas).

Pvz., žemiau pateiktame operatoriaus `if ... else` pavyzdyje kintamajam `a` priskiriamas didesnis iš dviejų skaičių `b` ir `c`:

```
if (b > c) a = b;
else a = c;
```

Tą patį rezultatą galima gauti ir šitaip:

```
a = (b > c) ? b : c;
```

Ciklo operatoriaus veikimas pasireiškia tuo, kad duotasis operatorius (kuris gali būti sudėtinis) vykdomas kelis kartus iš eilės, kol yra tenkinama tam tikra sąlyga. Tas operatorius kartais vadinamas „ciklo kūnu“ (siekiant jį atskirti nuo „ciklo antraštės“), o vienas to operatoriaus skaičiavimas vadinamas „iteracija“. Ciklo operatoriai yra trijų rūšių:

1. `while` (loginis reiškinys)
operatorius;
2. `do`
operatorius_1
`while` (loginis reiškinys)
3. `for` (inicializavimas; loginis reiškinys; nauja reikšmė)
operatorius;

„while“ ciklas vykdomas tol, kol loginis reiškiny yra nelygus nuliui, o jeigu tas reiškiny jau yra lygus nuliui prieš pradėdant vykdyti ciklą, tada ciklas iš viso nevykdomas. „do“ ciklas skiriasi nuo „while“ ciklo tik tuo, kad loginio reiškiny reikšmė tikrinama ne prieš operatoriaus vykdymą, o po jo. Taigi, „do“ ciklo atveju operatorius visada įvykdomas bent vieną kartą. „for“ ciklas skiriasi nuo „while“ ciklo tuo, kad „for“ cikle numatytos dar dvi operacijos: inicializavimo operacija, kuri įvykdoma pačioje „for“ ciklo pradžioje, ir naujos reikšmės skaičiavimo operacija, kuri įvykdoma kiekvieną kartą, apskaičiavus operatorių (t. y. po kiekvienos iteracijos). Žemiau pateiktas „for“ ciklas, kuris skaičiuoja sveikąjį skaičiaus n faktorialą $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$:

```
int i, fact;
for (i = 1, fact = 1; i <= n; i++) fact *= i;
```

Šiuo atveju ciklo inicializavimas pasireiškia tuo, kad kintamiesiems i ir $fact$ priskiriama reikšmė 1. Atlikus operatorių „ $fact *= i$ “, kintamasis i padidinamas vienetu (operacija „ $i++$ “). Prieš atliekant operaciją „ $fact *= i$ “, tikrinama sąlyga „ $i <= n$ “ (jeigu ta sąlyga negalioja, tada operacijos „ $fact *= i$ “ ir „ $i++$ “ nevykdomos).

„for“ ciklo antraštėje inicializavimo ir naujos reikšmės skaičiavimo operacijos nėra būtinos. Pvz., žemiau pateiktas „for“ ciklo pavyzdys, kuris skiriasi nuo ankstesniojo pavyzdžio tik tuo, kad kintamųjų inicializavimas atliekamas prieš ciklo vykdymą, o naujos reikšmės skaičiavimas „prijungtas“ prie ciklo kūno, kurį šiuo atveju sudaro du operatoriai:

```
int i = 1, fact = 1;
for (; i <= n;) {
    fact *= i;
    i++;
}
```

Tokio pavidalo „for“ ciklas yra tapatus „while“ ciklui (t. y. pastarajame pavyzdyje ciklo antraštė galėtų būti „while ($i <= n$)“).

Aukščiau aprašytieji ciklo operatoriai priklauso bendresnei „valdymo operatorių“ kategorijai. Valdymo operatoriai – tai operatoriai, kurie valdo operatorių atlikimo tvarką (seką). Kiti valdymo operatoriai yra „break“, „continue“ ir „goto“. Jeigu, vykdant ciklo operatorių, aptinkamas operatorius „break“, tada išeinama iš duotojo ciklo operatoriaus. Jeigu, vykdant ciklo operatorių, aptinkamas operatorius „continue“, tada praleidžiama visa likusioji ciklo kūno dalis ir pereinama prie kitos iteracijos. Operatorius „goto“ perduoda valdymą programos eilutei, kuri prasideda duotąja žyme. **Žymė** – tai bet kokia simbolių seka, kuri užsibaigia dvitaškiu. Žymė (be dvitaškio) turi būti nurodyta po „goto“. Pvz., žemiau pateiktame pavyzdyje operatorius „goto“ naudojamas skaičiaus n faktorialo skaičiavimui:

```
int i = 1, fact = 1;
if (n <= 0) goto END;
START:
fact *= i;
i++;
if (i <= n) goto START;
END:
...
```

C.4. Kai kurios matematinės funkcijos

Toliau išvardytos kelios standartinės funkcijos. Kiekvienos funkcijos apibrėžtyje yra nurodyti jos vardas, argumentų tipai ir skaičiuojamos reikšmės tipas (tačiau, išskviečiant funkciją, tipų nurodyti nereikia, pvz., „ $y = \sin(x)$ “). Užrašas „void“ reiškia, kad funkcija negrąžina jokios reikšmės (tokios funkcijos negali būti naudojamos aritmetiniuose reiškiniuose ir priskyrimo operacijose).

```
double acos(double x) – parametro  $x$  arkksinusas (galimoji vertė – nuo 0 iki  $\pi$ ).
double asin(double x) – parametro  $x$  arksinusas (galimoji vertė – nuo  $-\pi/2$  iki  $+\pi/2$ ).
double atan(double x) – parametro  $x$  arktangentas (galimoji vertė – nuo  $-\pi/2$  iki  $+\pi/2$ ).
double atan2(double y, double x) – santykio  $y/x$  arktangentas (galimoji vertė – nuo 0 iki  $\pi$ ).
double ceil(double x) – mažiausias sveikasis skaičius, kuris yra nemažesnis už  $x$  (t.p. žr. floor()).
double cos(double x) – parametro  $x$  kosinusas ( $x$  turi būti išreikštas radianais).
double exp(double x) – parametro  $x$  eksponentė ( $e^x$ ).
double fabs(double x) – parametro  $x$  modulis.
double floor(double x) – didžiausias sveikasis skaičius, kuris neviršija  $x$  (t.p. žr. ceil()).
double fmod(double x, double y) – skaičiaus  $x$  dalybos iš skaičiaus  $y$  liekana.
double log(double x) – skaičiaus  $x$  natūralusis logaritmas.
double log10(double x) – skaičiaus  $x$  dešimtainis logaritmas.
double pow(double x, double y) – skaičiaus  $x$  kėlimas laipsniu  $y$  ( $x^y$ ).
double sin(double x) – parametro  $x$  sinusas ( $x$  turi būti išreikštas radianais).
double sqrt(double x) – kvadratinė šaknis iš  $x$ .
double tan(double x) – parametro  $x$  tangentas ( $x$  turi būti išreikštas radianais).
```

```
int rand() – generuoja atsitiktinį sveikąjį skaičių, kurio vertė yra nuo 0 iki RAND_MAX. Konstanta RAND_MAX apibrėžta standartiniame antraštiniame faile „stdlib.h“ (RAND_MAX dažniausiai lygus 32767).
void srand(unsigned random) – inicializuoja atsitiktinių skaičių generatorių (žr. rand()). Atsitiktinių skaičių seka, kurią generuoja rand(), priklauso tik nuo parametro random.
```