

**VENTSPILS AUGSTSKOLA
INFORMĀCIJAS TEHNOLOĢIJU FAKULTĀTE**

BAKALAURA DARBS

**DIFŪZIJAS ĪPATNĪBU MODELĒŠANA
DIVDIMENSIONĀLĀ VIDĒ AR IESLĒGUMIEM**

Autors

Ventspils Augstskolas
Informācijas tehnoloģiju fakultātes
bakalaura studiju programmas „Datorzinātnes”
3. kursa students
Vitālijs Daņilevičs
Matr.nr. 2009020066

(paraksts)

Fakultātes dekāns

asoc.prof., Dr. math. Gaļina Hilķeviča

(paraksts)

Zinātniskais vadītājs

asoc.prof., Dr. habil. phys. J. R. Kalniņš
(ieņemamais amats, zinātniskais nosaukums, vārds, uzvārds)

(paraksts)

Recenzents

(ieņemamais amats, zinātniskais nosaukums, vārds, uzvārds)

(paraksts)

Ventspils
2012

ANOTĀCIJA

Darba nosaukums: *Difūzijas īpatnību modelēšana divdimensionālā vidē ar ieslēgumiem*

Darba autors: Vitālijs Daņilevičs

Darba vadītājs: Dr. habil. phys. J. R. Kalniņš

Darba apjoms: 57 lpp., 28 attēli

Atslēgas vārdi: DIFŪZIJAS EFEKTĪVAIS KOEFIECIENTS, DIFŪZIJA, BRAUNA KUSTĪBA

Darbā tiek pētīts difūzijas process divdimensionālā heterogēnā vidē ar riņķveida ieslēgumiem. Izveidots simulācijas modelis efektīvā difūzijas koeficienta noteikšanai. Simulācijas paātrināšanai izmantota *first passage* metode. Izveidots interfeiss programmai, kas ļauj viegli mainīt modeļa parametrus un sekot aprēķina gaitai. Testa simulācijas rezultāti salīdzināti ar teoriju. Gūts apstiprinājums teorijas lietošanai gadījumos, kad ieslēgumi ir ķērāj centri ar atgriezenisku reakciju.

Darba teorētiskajā daļā ir 2 nodaļas. Pirmajā nodaļā tiek apskatīts difūzijas process, punktveida defekti un Brauna daļiņu kustība. Otrajā nodaļā tiek aprakstīta difūzijas efektīva koeficienta aprēķināšanas teorētiskās formulas, molekulas gadījuma kustības teorija, *first passage* metode.

Darba praktiskajā daļā tiek aprakstīta simulācijas programmas izveide, izmantotās bibliotēkas un algoritmi. Doti simulācijas rezultāti un to salīdzinājums ar teoriju.

ANNOTATION

Title: *Modeling of diffusion properties in 2-dimensional media with inclusions*

Author: Vitaly Danilevich

Superintendent: Dr. habil. phys. Yu. R. Kalnin

Volume: 57 pages, 28 images

Keywords: EFFECTIVE DIFFUSION COEFFICIENT, DIFFUSION, BROWNIAN MOTION

The paper examines the process of diffusion in two-dimensional heterogeneous environment with circular inclusions. The simulation model of effective diffusion coefficient is created. First passage method is used to accelerate simulation. The program interface has been created, which allows you to easily change the model parameters and to follow the progress of the calculation. The test simulation results are compared with the theory. The theory gained approval for use in cases where the inclusion is the catcher with reversible reaction centre.

The theoretical part of the paper contains 2 chapters. The first chapter discusses the diffusion process, point defects and Brownian motion. The second chapter describes theoretical formulas for the calculation of effective diffusion coefficient, molecular motion theory and first passage method.

The practical part of the paper describes the development of simulation programs, libraries and algorithms used. The simulation results and their comparison with the theory are described.

АННОТАЦИЯ

Название работы: *Моделирование свойств диффузии в 2-мерной среде со включениями*

Автор работы: Виталий Данилевич

Руководитель работы: Dr. habil. phys. Ю. Р. Калныньш

Размер: 57 стр., 28 изображений.

Ключевые слова: ЭФФЕКТИВНЫЙ КОЭФФИЦИЕНТ ДИФфуЗИИ, ДИФфуЗИЯ, БРОУНОВСКОЕ ДВИЖЕНИЕ

В работе исследован процесс диффузии в двумерной гетерогенной среде с круговыми включениями. Создана имитационная модель для определения эффективного коэффициента диффузии. Для ускорения моделирования использован метод первого прохождения. Создан интерфейс программы, который позволяет легко изменять параметры модели и следить за ходом расчета. Результаты тестов моделирования сравниваются с теорией. Автор подтвердил теорию для использования в тех случаях, когда включения действуют как ловушки с обратимой реакцией.

Теоретическая часть работы содержит 2 главы. В первой главе рассматривается процесс диффузии, точечных дефектов и броуновское движение. Во второй главе описываются теоретические формулы для расчета эффективного коэффициента диффузии, молекулярная теория движения и первый метод прохождения.

Практическая часть работы описывает разработку программ моделирования, использованных библиотек и алгоритмов. Результаты моделирования и их сравнение с теорией описаны в работе.

Saturs

IEVADS.....	6
1. DIFŪZIJA.....	7
1.1. Difūzijas process.....	7
1.2. Brauna kustība.....	8
1.3. Punktteida defekti.....	10
2. TEORĒTISKĀS FORMULAS.....	12
2.1. Difūzijas koeficients.....	12
2.2. Standarta Maksvela-Garneta pieeja.....	14
2.3. MG vienādojuma trūkumi.....	16
2.4. Efektīvais difūzijas koeficients.....	23
2.5. Molekulas gadījuma kustība.....	27
2.6. Pirmās pārejas metode (First-Passage method).....	38
3. PROGRAMMAS IZSTRĀDE.....	40
3.1. Grafiskās vides izstrāde.....	41
3.2. Galvenais algoritms.....	42
3.3. Konsoles izvads uz <i>TextBox</i>	45
3.4. Grafiku zīmēšanas bibliotēka.....	46
3.5. Eksperimenta rezultātu saglabāšana PDF formātā.....	47
3.6. Mersenne Twister algoritms.....	49
3.7. Programmas instalatora izveide.....	50
3.8. Modelēšana.....	52
SECINĀJUMI UN PRIEKŠLIKUMI.....	55
IZMANTOTĀS LITERATŪRAS UN AVOTU SARAKSTS.....	56

IEVADS

Darbā aplūkotā tēma ir aktuāla pētot difūzijas procesu heterogēnās vidēs gan ar absolūti necaurlaidīgiem otrās fāzes ieslēgumiem, gan ieslēgumiem ar daļēju caurlaidību, gan arī modelējot difūziju vidē ar ķērāj centriem. Tēma ir aktuāla materiālu zinātnē, porainas vides ķīmijā un fizikā, biomedicīnā, signālu teorijā u.c.[4-10].

Darbs veltīts difūzijas efektīvā koeficienta modelēšanai. Darba mērķis ir pētīt ar simulācijas palīdzību daļiņas efektīvo difūziju divdimensionālā heterogēnā vidē ar riņķveida ieslēgumiem.

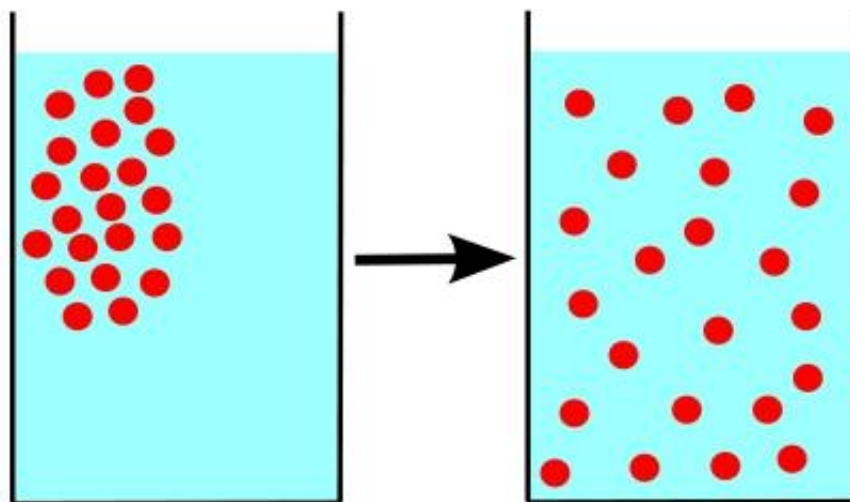
Darba uzdevumi ir: iepazīties ar efektīvā difūzijas koeficienta aprēķināšanas teoriju vidējā lauka ietvaros; izveidot efektīvā difūzijas koeficienta simulācijas modeli; izveidot interfeisu programmai, kas ļauj viegli mainīt modeļa parametrus un sekot aprēķina gaitai un noformēt datu izvadi ērtā grafiskā vidē; veikt testa simulācijas un salīdzināt iegūtos rezultātus ar teoriju.

Programmu plānots izstrādāt valodā C++ bez grafiskās vides, kura ātri izrēķinās rezultātu no sākuma nosacījumiem. Kā arī uztaisīt alternatīvu programmu, kura attēlos procesu grafiski. Otru programmu plānoju taisīt valodā C#.

1. DIFŪZIJA

1.1. Difūzijas process

Difūzija no latīņu valodas – izplatīšanās, izplūšana, izkliede, mijiedarbība. Difūzijas process apraksta daļiņu izplatīšanos no vairāk koncentrētiem daļiņu reģioniem uz mazāk apdzīvotiem daļiņu reģioniem.



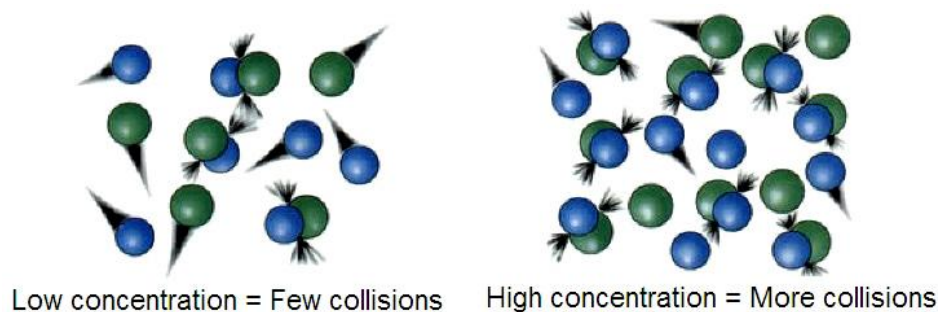
1.1. att. Difūzijas piemērs [20]

Piemērā var redzēt, ka kādas vielas daļiņas sākumā ir koncentrētas stūrī un ar laiku tās izplatās ūdens glāzē.

Ar jēdzienu difūziju pamatā tiek apzīmēti šādi procesi – vielas elementārdaļiņu izplatīšanās, kā arī kustības vai siltuma daudzuma izplatīšanās. Šajā fizikālajā procesā vielas molekulas savstarpēji mijiedarbojas un pakļaujas Brauna kustības principiem. Homogēna (viendabīgā) vidē difūzija ir izotops fizikāls process, kas nozīmē, ka vielā difūzijas īpašības visos virzienos notiek pēc viena likuma. Bet heterogēnā (neviendabīgā) vidē difūzija ir izotops process – dažādos virzienos vielas molekulas izplatās pēc atšķirīgiem likumiem.

Difūzijas likumiem ir pakļautas ne tikai šķidru vielu daļiņas, bet arī cietvielu kristāla daļiņas. Starpmezglu atomi un piejaukuma starpmezglu atomi ir spējīgi difundēt.

Gāzveida vielas difundē visātrāk, šķidrumi lēnāk, bet cietvielās vislēnāk. Gāzveida vielas difundē ātrāk, ja temperatūra ir augstāka. Difūzija norisinās lēnāk, ja vielas molekulu sadursmju skaits ir liels.



1.2. att. Molekulu sadursmju skaits [32]

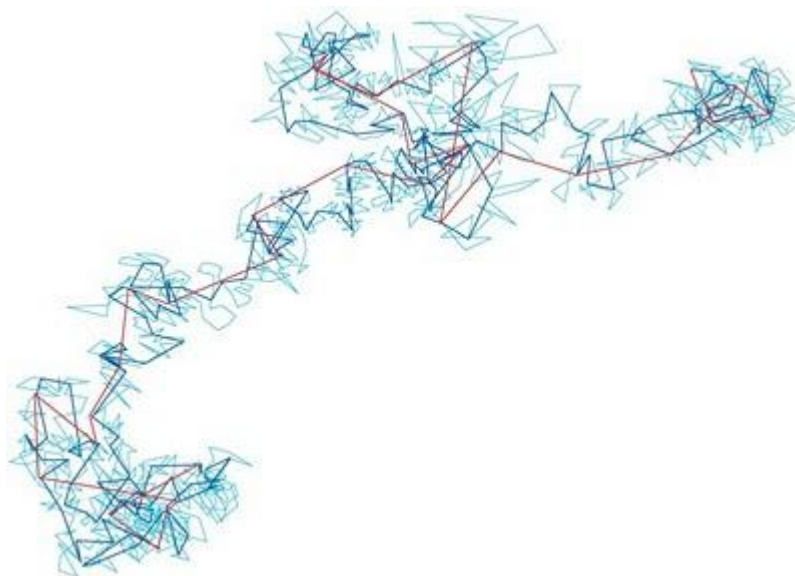
Parasti difūziju procesu saprot, kā vielas elementārdaļiņu izplatīšanās, kā arī kustības vai siltuma daudzuma izplatīšanās, bet difūzija apraksta arī siltuma vadāmību, berzi starp diviem materiāliem, vielu caurlaidības īpašības u.c.

Difūzijas norises laiks ir atkarīgs no daudziem faktoriem. Piemēram, metāla stieņa gadījumā siltuma difūzija norit ļoti ātri, bet, ja stienis ir izgatavots no sintētiska materiāla siltuma, difūzija noritēs lēni. Vielu molekulu difūzija parasti norit vēl lēnāk, piemēram, ja cukura gabaliņu ievieto glāzē ar ūdeni, un ūdeni nemaisīt, paies vairākas nedēļas līdz šķidrums kļūs viendabīgs. Vēl lēnāk norisinās vienas cietas vielas difūzija otrā cietā vielā. Piemēram, ja vara stienis būs pārklāts ar zeltu notiks vara un zelta difūzija, bet pie normāliem apstākļiem (istabas temperatūra un atmosfēras spiediens) zelta saturošais slānis sasniegs vairāku mikrometru slāni tikai pēc vairākiem tūkstošiem gadu. [20] [24] [26]

1.2. Brauna kustība.

Angļu botāniķis Roberts Brauns šo parādību pirmoreiz novēroja 1827. gadā, aplūkojot mikroskopā ūdenī iejauktas augu sporas. Tagad parasti izmanto gumiguta krāsas daļiņas, kuras ūdenī nešķīst. Šīs daļiņas kustas haotiski, un kustība nekad neapsīkst. Paaugstinot temperatūru, kustības intensitāte pieaug.

Brauna kustību var novērot gāzē. Gaisā esošo putekliņu vai dūmu kustība arī ir Brauna kustība.



1.3. att. Brauna kustības piemērs, pie dažādiem soļu garumiem

Mūsdienās Brauna kustības jēdzienu izmanto daudz plašākā nozīmē. Piemēram, Brauna kustība ir jūtīgu mēraparātu rādītāju drebēšana. Tās cēlonis ir aparātu un apkārtējās vides atomu siltumkustība.

Brauna kustība - ar aci vai mikroskopu novērojama šķidrumā (vai gāzē) iejauktu nelielu daļiņu siltumkustība. Šīs daļiņas kustas haotiski. Brauna kustība arī izskaidro tādu fizikālu parādību, kā difūziju.

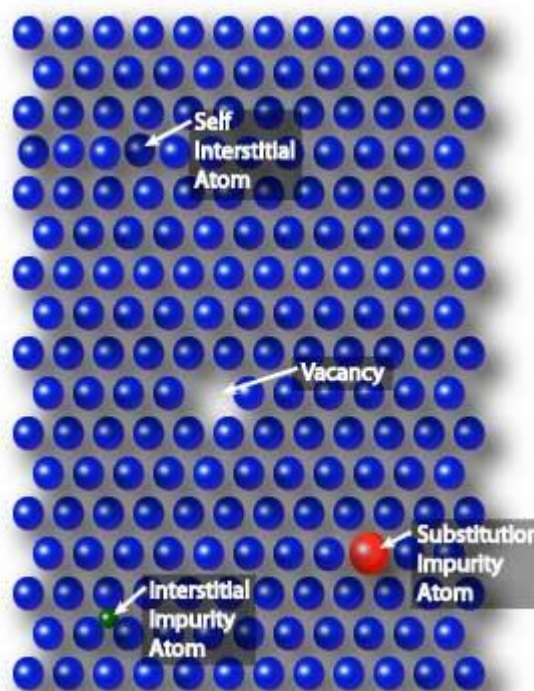
Brauna kustību var izskaidrot, pamatojoties uz molekulāri kinētisko teoriju. Brauna kustības cēlonis ir tas, ka molekulu triecieni pret daļiņu nekompensē cits citu. Molekulām haotiski kustoties, impulsi, ko saņem daļiņa, piemēram, no kreisās un labās puses nav vienādi. Tāpēc rezultējošs spiediena spēks nav nulle un tas arī rada Brauna kustību.

Kā gāzē, tā arī šķidrumā spiedienam ir kaut kāda vidējā vērtība. Taču vienmēr pastāv nelielas gadījuma atšķirības no vidējās vērtības. Jo mazāks ir ķermeņa laukums, jo ievērojamākas ir tā spiediena spēka relatīvās izmaiņas, kuras darbojas uz doto laukumu. Piemēram, ja laukuma izmēri ir dažu molekulu diametru lielumā, tad uz to darbojošais spēks mainās lēcienveidā no nulles līdz kādai vērtībai brīdī, kad šajā laukumā trāpa molekula.

Brauna kustības molekulāri kinētisko teoriju izveidoja Alberts Einšteins 1905. gadā. Franču fiziķis Žans Perrēns pilnīgi pabeidza izveidot molekulāri kinētisko teoriju, izstrādādams Brauna kustības teoriju un to eksperimentāli apstiprinādams. Tika pierādīts, ka Brauna daļiņas piedalās šķidrumu un gāzu molekulu haotiskajā kustībā. [25]

1.3. Punktteida defekti.

Punktteida defekts, ja atoms ir pazudis vai ir neregulāra vietā režģa struktūrā. Punktteida defekti ietver sevī starpmezglu atomus, starpmezglu piemaisījumu atomus, aizstājošus atomus un vakances. Starpmezglu atoms ir papildus atoms, kas aizpilda starpmezglu tukšumu kristāliskā struktūrā. Aizstājoši starpmezglu atomi parādās tikai zemas koncentrācijas metāliem, jo tie izkropļo, sarežģī režģa struktūru.



1.4. att. Punktteida defekti [22]

Aizstājošo piemaisījumu atoms ir cita tipa atoms nekā lielākā daļa vielas atomu, kurš ir aizvietojis vienu vielas atomu režģī. Aizstājošo piemaisījumu atoms parasti ir līdzīgs pēc izmēra (atšķiras apmēram par 15 %) citiem vielas atomiem. Piemērs aizstājošo piemaisījumu atoms ir cinka atoms misiņā. Misiņā, cinka atomi ar rādiuss 0.133 nm aizstāj dažus vara atomus, kura rādiuss ir 0.128 nm.

Starpmezglu piemaisījuma atomi ir daudz mazāki nekā atomi vielas matricā. Starpmezglu piemaisījumu atomu iederas tukšā vietā starp vielas atomu režģa struktūrā. Piemērs starpmezglu piemaisījuma atomiem ir oglekļa atomi, kas tiek pievienoti pie

dzelzs, lai izveidotos tērauds. Oglekļa atomi ar rādiuss 0,071 nm, labi iederas brīvās vietās starp lielākiem (0.124 nm) dzelzs atomiem.

Vakances ir tukšie laukumi, kur atoms atrastos, bet trūkst. Tie bieži parādās pie augstām temperatūrām, kad atomi sāk nejauši mainīt savas pozīcijas, atstājot brīvu vietu režģī. Vairumā gadījumu difūzija (atomu pārvietošanās, kustība) var notikt dēļ vakancēm.
[22]

2. TEORĒTISKĀS FORMULAS

2.1. Difūzijas koeficients

Caurlaidības īpašību pētīšanai nehomogēniem materiāliem sākās ar Maksvela-Garneta pētījumiem. Saliktus materiālus parasti modelē kombinējot posmus no galvenā (matrica), kuru raksturo difūzijas koeficients no zondes daļiņas D_2 , un sfēriskie ieslēgumi, kurus raksturo daļiņu difūzijas koeficients D_1 rādiuss r_0 un ieslēguma un režģa laukumu attiecība Φ . Šie divi difūzijas koeficienti ir izteikti ar soļa garumu l un laiku starp diviem veiksmīgiem soļiem τ :

$$D_i = \frac{l_i^2}{2d\tau_i}, \quad (2.1.)$$

kur

d - apzīmē telpas dimensiju (1, 2 vai 3),

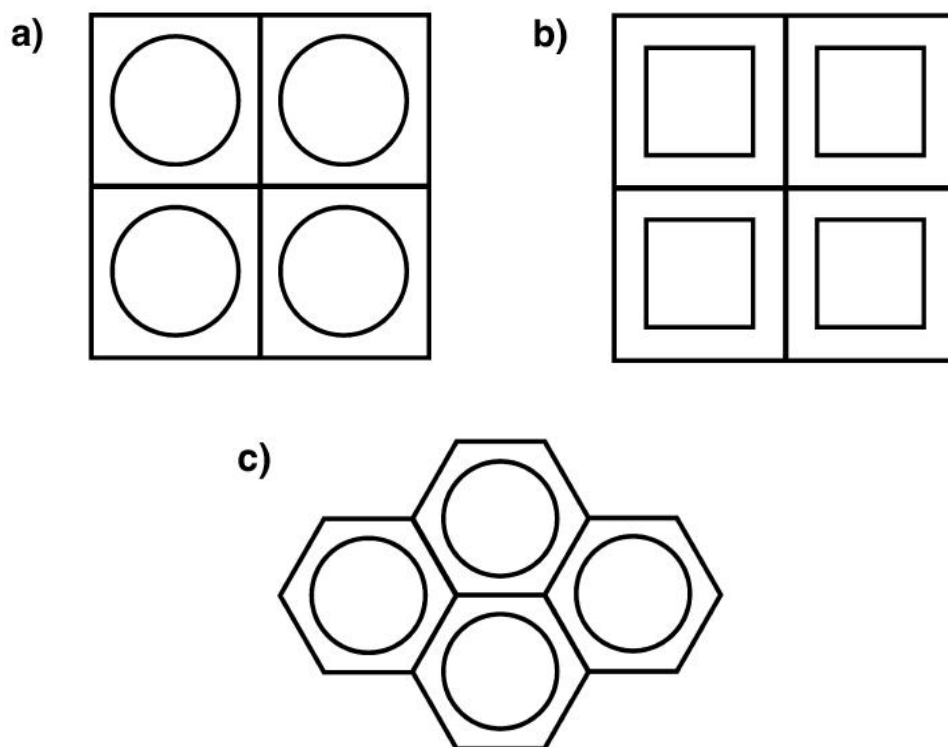
$i = 1$ (ārpus ieslēguma), 2 (ieslēgumā),

τ - laiks starp diviem veiksmīgiem soļiem,

l - soļa garums.

[1,1.lpp]

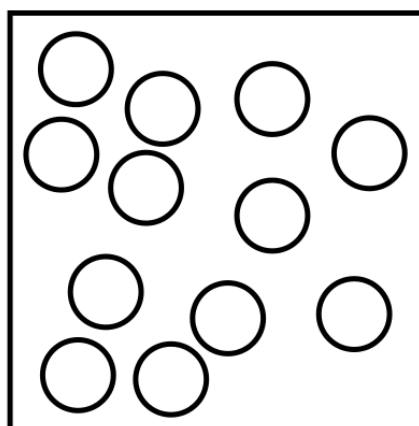
Ir dažādi režģu un ieslēgumu veidi, piemēram, režģi ar periodiskiem ieslēgumiem:



2.1. att. Režģu piemēri [1,6.lpp]

Dažādi režģu un ieslēgumu piemēri: sfēriski un kvadrātveida ieslēgumi kvadrātveida režģos (a, b), sfēriski ieslēgumi sešstūrveida režģī 1.2 attēls (c) piemērs. [1,6.lpp]

Kā arī pastāv ieslēgumi ar neperiodisku sadalījumu:



2.2. att. Neperiodisks režģis ar apaļiem ieslēgumiem

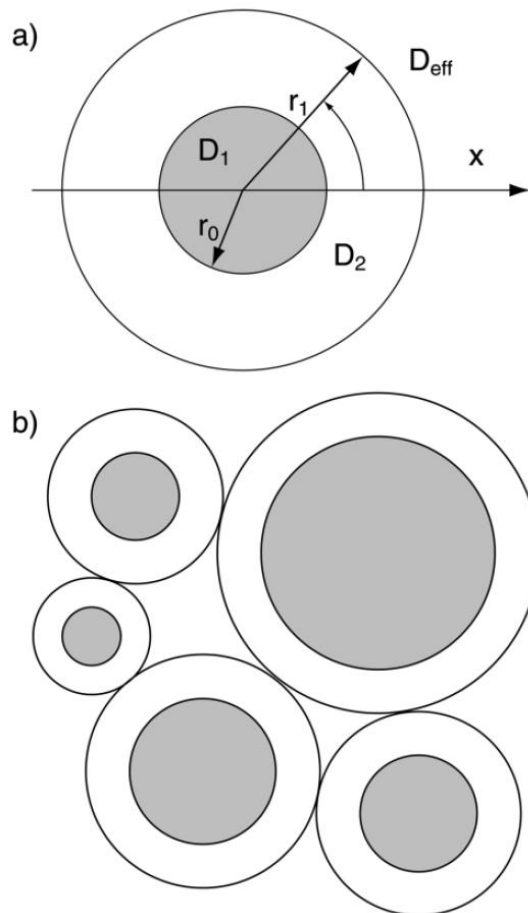
Režģu sadalījums ir atkarīgs no pamatvielas un ieslēgumiem.

Taisot programmu izmantošu kvadrātisku režģi ar sfēriskiem ieslēgumiem(a).

2.2. Standarta Maksvela-Garneta pieeja

Eksperimentāli matrica ar iekļāvumiem ir aprakstīta ar *efektīvu difūzijas koeficientu*, D_{eff} , kurš ir funkcija no D_1 , D_2 un Φ . Lai to noteiktu var izmantot elektrostatiskos analogus. Ir makroskopiski homogēns materiāls ar difūzijas koeficientu D_{eff} . Pēc sākotnējās MG vienādojuma atvasināšanas iedomājamies, ka daļiņu koncentrācijai c ir vidējs gradients g (līdzīgi kā viendabīgā elektriskā laukā). Tad mēs ievietojam vielā sfēriskus ieslēgumus ar rādiusu r_0 , kuri ir iekļauti lodveida čaulā pamatmateriālā (matricā) ar rādiusu r_1 un pieņemsim, ka ieslēgumi nemaina koncentrācijas lauku ārpusē, t.i., pie $r \geq r_1$. Rādiusi r_0 un r_1 apmierina ieslēgumu laukuma vienādojumu

$$\Phi = \frac{r_0^3}{r_1^3} \cdot [1,1.lpp]$$



2.3. att. (a) Shematisks attēlojums, matrica ar ieslēgumiem pamat čaulā.

(b) Gadījums ar dažāda izmēra ieslēgumiem. [1,2.lpp]

Efektīvais difūzijas koeficients var būt izteikts ar šo vienādojumu:

$$\Delta c_{r,\vartheta}(r, \vartheta) = 0, \quad (2.2.)$$

kur

r, ϑ – koordinātes,

ϑ – leņķis starp \vec{r} un ārējo gradientu \vec{g} .

Piemērots risinājums vienādojumam (2.2.) saskan ar:

$$c_1(r, \vartheta) = Ar \cos \vartheta, \quad (0 < r \leq r_0), \quad (2.3.)$$

$$c_2(r, \vartheta) = \left(Br + \frac{E}{r^2} \right) \cos \vartheta, \quad (0 < r \leq r_1), \quad (2.4.)$$

$$c_{eff}(r, \vartheta) = -gr \cos \vartheta, \quad (r_1 < r), \quad (2.5.)$$

kur,

$c_i(r, \vartheta)$ - vietējā daļiņu koncentrācija ieslēgumos ($i = 1$) un pamatvielā ($i = 2$).

Vienādojumi ar nezināmajiem mainīgajiem A, B, E un g rodas no plūsmu un daļiņu koncentrācijas robežu nosacījumiem:

$$c_1(r_0, \vartheta) = c_2(r_0, \vartheta) \quad (2.6.)$$

$$D_1 \frac{\partial c_1(r, \vartheta)}{\partial r} \Big|_{r=r_0} = D_2 \frac{\partial c_2(r, \vartheta)}{\partial r} \Big|_{r=r_0} \quad (2.7.)$$

$$c_2(r_1, \vartheta) = c_{eff}(r_1, \vartheta) \quad (2.8.)$$

$$D_2 \frac{\partial c_2(r, \vartheta)}{\partial r} \Big|_{r=r_1} = D_{eff} \frac{\partial c_{eff}(r, \vartheta)}{\partial r} \Big|_{r=r_1} \quad (2.9.)$$

No vienādojumiem (2.3.) – (2.9.) iegūstam vairākus vienādojumus:

$$r_0^3 A - r_0^3 B - E = 0 \quad (2.10.)$$

$$D_1 r_0^3 A - D_2 r_0^3 B + 2D_2 E = 0 \quad (2.11.)$$

$$r_1^3 B + E + r_1^3 g = 0 \quad (2.12.)$$

$$D_2 r_1^3 B + D_{eff} r_1^3 g - 2D_2 E = 0 \quad (2.13.)$$

Izmantojot iepriekšminētos vienādojumus iegūstam MG vienādojumu:

$$D_{eff} = D_2 \left[1 + \frac{3(D_1 - D_2)\Phi}{D_1 + 2D_2 - (D_1 - D_2)\Phi} \right] \quad (2.14.)$$

Kopumā telpas dimensijas var apzīmēt ar d ($d = 1, 2, 3$). Un pārveidojot iegūstam šādu vienādojumu:

$$D_{eff} = D_2 \left[1 + \frac{d(D_1 - D_2)\Phi}{D_1 + (d-1)D_2 - (D_1 - D_2)\Phi} \right], \quad (2.15.)$$

kur

D_1 – difūzijas koeficients pamat vidē,

D_2 – difūzijas koeficients ieslēgumā,

d - apzīmē telpas dimensiju (1, 2 vai 3),

Φ - un ieslēguma un režģa laukumu attiecība.

Šī formula atbilst ne tikai periodiskiem vienāda izmēra sfēriskiem ieslēgumiem, bet arī gadījuma ieslēgumiem ar dažādiem r_0^i , ja nosacījums:

$$\Phi = \frac{(r_0^i)^d}{(r_1^i)^d}, \text{ kur } d \text{ ir telpas dimensijas. Tomēr, jautājums, kad apjoma frakcijas sāk}$$

‘konkurēt’ un vienādojums (2.15.) vairs nav derīgs, paliek atvērts un varētu tikt atrisināts analītiski (skatīt piemēru grāmatā [12]) vai ar tiešo datora simulāciju. Tāpēc, ka risinot analītiski, tiek apskatīti daudzu daļiņu efekti.

2.3. MG vienādojuma trūkumi

Apskatīsim tagad vairākus gadījumus, kad pieņemtais MG vienādojums (2.15.) dod nepareizus rezultātus.

1. Sāksim ar situāciju, kad ieslēgums ir pilnīgi necaurlaidīgs, tas ir, kad daļiņa atstarojas uz r_0 , kas var būt aprakstīt stāvokli:

$$\left. \frac{\partial c_2(r, \vartheta)}{\partial r} \right|_{r=r_0} = 0$$

(jāņem vērā arī robežgadījums $\frac{D_1}{D_2} \rightarrow 0$ vienādojumā (2.7.)).

Šajā situācijā vienādojums (2.15.) atgriež:

$$D_{eff} = D_2 \left[1 - \frac{2\Phi}{1 + \Phi} \right], (d = 2) \quad (2.16.)$$

[1,2.lpp]

$$D_{eff} = D_2 \left[1 - \frac{3\Phi}{2 + \Phi} \right], (d = 3) \quad (2.17.)$$

Šādas attiecības ir sastopamas reakcijas ātruma teorijā [11].

Tas pats rezultāts var būt iegūts no vienādojumiem (2.3.-2.9.) ievietojot tajos $c_1(r, \vartheta) = 0$, kā arī $A = 0$.

Taču vienādojumi (2.16.) un (2.17.) atgriež nepareizas koncentrācijas atkarības (skatīt diskusiju grāmatā [13]). Pareizas Φ atkarības ir šajos vienādojumos:

$$D_{eff} = \frac{D_2}{1 - \Phi} \left[1 - \frac{3\Phi}{2 + \Phi} \right], (d = 3), \quad (2.18.)$$

$$D_{eff} = \frac{D_2}{1 - \Phi} \left[1 - \frac{2\Phi}{1 + \Phi} \right], (d = 2). \quad (2.19.)$$

Šī kļūda rodas izmantojot vienādojumu (2.8.). Faktiski difundējošo daļiņu koncentrācija matricā nevar būt vienāda šim efektīvā vidē, tāpēc ka vēlāk daļiņas izvietoja visas sistēmas izmērā, un to vidējai koncentrācijai jābūt $1 - \Phi$. No tā seko, ka vienādojums (2.8.) ir jāizlabo uz:

$$c_2(r_1, \vartheta) = k_1 c_{eff}(r_1, \vartheta), \quad (2.20.)$$

tas ir, patiesībā tur ir koncentrācijas lēciens kodolā (ieslēgumā) – čaulā (matricā) robežā, $r=r_1$. Jautājums ir kā atrast koeficientu k_1 ? Mēs piedāvājam iegūt to izmantojot

daudz lielāku tuvinājumu, no vienādojuma vidēju līdzsvara daļiņu koncentrāciju sistēmā (kopējs daļiņu skaits izdalīts ar kopēju tilpumu).

$$c_{eff} = c_1\Phi + c_2(1 - \Phi). \quad (2.21.)$$

Šajā gadījumā necaurlaidīgus ieslēgumus ($c_I=0$) iegūstam no vienādojuma (2.21.):

$$c_{eff} = c_2(1 - \Phi), \quad k_1 = \frac{1}{1 - \Phi}. \quad (2.22.)$$

2. Cits ierobežojums izmantojot vienādojumu (2.15.) iziet no Maksvella robežgadījuma $c_1(r_0, \vartheta) = c_2(r_0, \vartheta)$ vienādojumā (2.6.). Patiesībā vienādojums (2.15.) var būt patiesš pie dažādiem difūzijas koeficientiem, matricā un ieslēgumā, $D_I \neq D_2$, tikai tad, kad ātrumi ieslēgumā un pamat vidē sakrīt

$$\frac{l_1}{\tau_1} = \frac{l_2}{\tau_2}.$$

Parasti ir nepieciešams ievest izmērāmu koeficientu k , kurš saista c_I un c_2 :

$$c_1(r_0, \vartheta) = kc_2(r_0, \vartheta). \quad (2.23.)$$

Lai uzlabotu MG vienādojumu izmantojot augstākminētos efektīvā difūzijas koeficienta standarta atvasinājumus, varam izmantot vienādojumus (2.20.) un (2.23.), oriģinālo Maksvela vienādojumu (2.6.) un (2.8.) vietā.

Veicot pārveidojumus iegūstam:

$$D_{eff} = D_2 k_1 \left[1 + \frac{d(D_1 k - D_2)\Phi}{kD_1 + (d-1)D_2 - (kD_1 - D_2)\Phi} \right]. \quad (2.24.)$$

Viendimensionālā gadījumā šis vienādojums sakrīt ar tiešu risinājumu [14]. Divi koeficienti k un k_I ir saistīti ar līdzsvara koncentrāciju ieslēgumos un matricā un vielas sadrumstalotību Φ saskaņā ar vienādojumiem (2.20), (2.21), (2.22):

$$k = \frac{c_1}{c_2} \quad (2.25.)$$

un

$$c_2 = k_1 c_{eff}. \quad (2.26.)$$

No vienādojuma (2.21.) iegūstam vielas atkarības no koeficienta k_1 :

$$k_1 = \frac{1}{1 - \Phi + \frac{c_1}{c_2} \Phi}. \quad (2.27.)$$

Izmantojot vienādojumu (2.27.) iegūstam citu attiecību vienādojuma (2.24.) vietā:

$$D_{eff} = \frac{D_2}{1 - \Phi + \frac{c_1}{c_2} \Phi} \left[1 + \frac{d \left(D_1 \frac{c_1}{c_2} - D_2 \right) \Phi}{(d-1)D_2 + \frac{c_1}{c_2} D_1 - \left(\frac{c_1}{c_2} D_1 - D_2 \right) \Phi} \right] \quad (2.28.)$$

Līdzīgi MG teorijai šis vienādojums atspoguļo abus ierobežojošus gadījumus, kad Φ tiecas uz nulli un vienotību. Vienādojums (2.28.) pamat rezultāts mūsu teorijai. Der atgādināt, ka c_1 un c_2 ir vidējās difundējošo daļiņu koncentrācijas divās fāzēs – ieslēgumos un matricā.

Ir ērti izteikt attiecību c_1/c_2 vienādojumā (2.28.) ar matricas un ieslēgumu kinētiskiem parametriem. Līdzsvara stāvoklī, stabilā situācijā daļiņu plūsmas ieslēgumos un ārā no tiem ir vienādas:

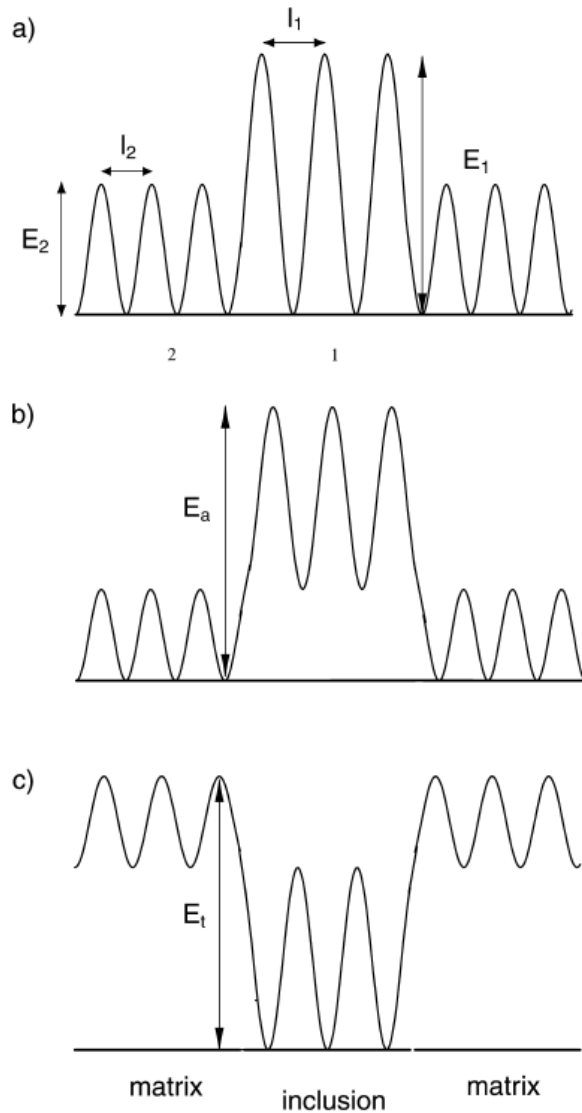
$$c_1 \frac{l_1}{\tau_1} = c_2 \frac{l_2}{\tau_2}. \quad (2.29.)$$

Jāatceras, ka difūzijas koeficienti D_1 un D_2 ir definēti vienādojumā (2.1.)

Attēlā 2.3., kur parādīta difundējošo daļiņu potenciālā enerģija vairākas svarīgas situācijas. Tiek modelēts daļiņas daļējs ieslodzījuma brīdis ieslēgumā (potenciālā enerģija arī) un (daļēja) atstarošanās no ieslēguma dēļ enerģētiskās barjeras. Lai aprakstītu šādas situācijas ir nepieciešami caurlaidības koeficienti p_1 no ieslēguma matricā un p_2 no matricas ieslēgumos. Pamat gadījumā iegūstam:

$$\frac{c_1}{c_2} = \frac{l_2 p_2 \tau_1}{l_1 p_1 \tau_2}. \quad (2.30.)$$

[1,3.lpp]



2.4. att. [1,4.lpp] Dažādos gadījumos, kad enerģijās barjeras starp matricu un ieslēgumiem, l_i , E_i , ir soļa attālumā un difūzijas aktivācijas enerģijas divās fāzēs, $i=1, 2$: (a) ieslēgums ar difūzijas koeficientu ieslēgumos mazāku nekā matricā; tur nav enerģētiskās barjeras starp tiem; (b) enerģijas barjera E_a , varbūtība iekļūt ieslēgumā, $p_2 \leq 1$; (c) daļēja daļiņu ieslēgšana ieslēgumos, $p_1 \leq 1$. Enerģija nekļūt ieslēgtam ir E_t .

Daļiņas varbūtība iziet potenciālās barjeras gadījumā (laika vienībā) ir nosakāma pēc aktivācijas enerģijas E_{12} [15]:

$$p_2 = \tau_2 * \exp\left(\frac{-E_{12}}{kT}\right). \quad (2.31.)$$

Tas pats ir patiesi gadījumā daļiņai atlecot no ieslēguma ar varbūtību p_1 . Vienādojumi (2.28.) un (2.30.) ļauj aprakstīt daudzus difūzijas kontrolētus procesus kompleksā vidē ar varbūtībām mobilās daļiņas ieslēgt un atlaist.

Tagad mēs salīdzināsim rezultātus ar iepriekšējām teorijām. Viendimensionālā gadījumā vienādojums (2.28.) dod tiešu rezultātu periodiskai ieslēgumu sadalīšanai Kronig-Penny modelī ar daļiņu atstarošanas (vienādojums (2.4.) [14]), kas rakstās šādi:

$$D_{eff} = \left[\frac{\Phi}{D_1 k} + \frac{1-\Phi}{D_2} \right]^{-1} * \frac{1}{1-\Phi + \Phi k}, \quad (2.32.)$$

kur

$$k = c_1/c_2.$$

Savukārt trīsdimensionālā gadījumā mūsu D_{eff} vienādojums (2.28.) sakrīt ar iegūto rezultātu atsaucē [16] šajā ieslēgumu izplatības gadījumā izmantojot matemātiski ļoti sarežģītu neatkārtojamu termodinamikas formālismu ar ķīmiskām īpašībām. Analītiski rezultāti atsaucē [17] apliecina arī atšķirīgu kofaktoru klātbūtni:

$$\left(1 - \Phi + \frac{c_1}{c_2} \Phi \right)^{-1} \text{ ievadot } D_{eff}.$$

Gadījumā kad ir pilnīga daļiņu atstarošanās no ieslēgumiem; $D_1=0$, $\frac{c_1}{c_2} = 0$, vienādojums (2.28.) pārveidojas vienādojumos (2.18.) un (2.19.), kā minēts iepriekš. Maza apjoma ieslēgumi, kad $\Phi \leq 1$, pārveidojas šādi:

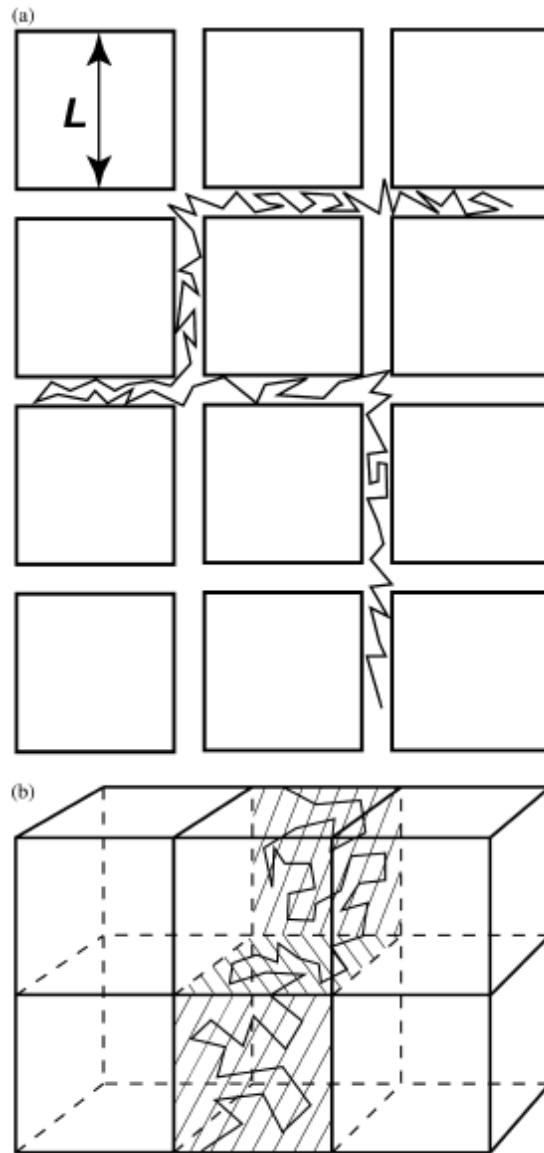
$$D_{eff} = D_2 \left[1 - \frac{\Phi}{2} \right], (d = 3), \quad (2.33.)$$

$$D_{eff} = D_2 [1 - \Phi], (d = 2). \quad (2.34.)$$

Vienādojums (2.33.) tika iegūts agrāk atsaucē [18] tā kā vienādojums (2.34.) bija iegūts atsaucē [13] izmantojot efektīvu vidējo tuvināšanu. Izteiksme (2.33.) arī bija iegūta aprēķinot efektīvo pašdifūzijas konstanti mobilajās sugās izmantojot risinājumu no [19].

Tagad apskatīsim dažus D_{eff} ierobežojošus gadījumus. Necaurīdīgu ieslēgumu gadījumā (pilnīgi atstarojošas daļiņas, $c_1/c_2 \rightarrow 0$), vienādojums (2.28.) ir vienkāršots:

$$D_{eff} = \frac{D_2(d-1)}{d-1+\Phi}. \quad (2.35.)$$



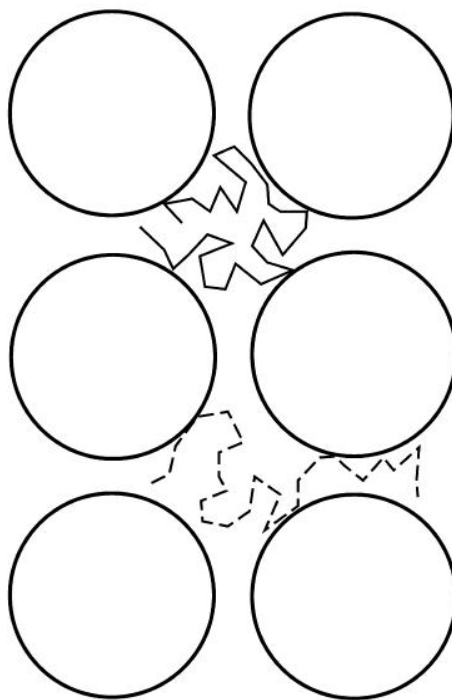
2.5. att. 2-D (3-D) difūzijas transformācija uz 1-D, 2-D difūziju, kā kvadrātiski (kubiski) ieslēgumi, (a) un (b) [1,5.lpp]

Kad $\Phi \rightarrow 1$ ir 2-D vai 3-D gadījumi, $D_{eff}=D_2/2$ un $D_2/3$, attiecīgi, kamēr MG vienādojums rādīs nulles efektīvu difūzijas koeficientu. Šis rezultāts nozīmē, ka ierobežojošos gadījumos daļiņu difūzija paliek viendimensionāla vai divdimensionāla kā parādīts attēlā 2.5. kvadrātveida un kubiskiem ieslēgumiem.

Lai uzsvērtu D_{eff} potenciālās barjeras lomu, pieņemsim spēcīgu ierobežojošu gadījumu, kad $c_2/c_1 \rightarrow 0$, no vienādojuma (2.28.) iegūstam vienādojumu:

$$D_{eff} = \frac{D_2}{k\Phi} * \frac{1+d-\Phi}{1-\Phi}, \quad (2.36.)$$

kas pierāda, D_{eff} ir neatkarīgs no difūzijas ieslēgumos. Tie paši rezultāti rodas ierobežojošā gadījumā, kad $D_2/D_1 \rightarrow 0$. [1,4.lpp]

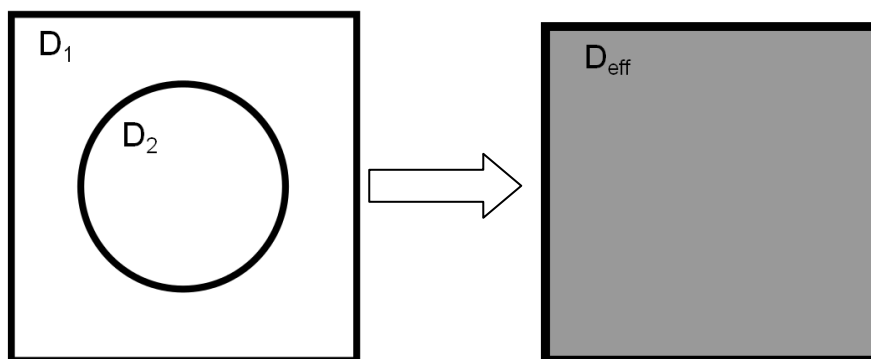


2.6. att. Ieslodzītas daļiņas [1,6.lpp]

2.4. Efektīvais difūzijas koeficients

Efektīvo difūzijas koeficientu aprēķina, lai noteiktu vielas caurlaidības spējas, vadāmību pretestību, piemēram, sūknējot naftu, var aprēķināt difūzijas efektīvo koeficientu.

Efektīvā difūzijas koeficienta aprēķināšana:



2.7. att. Difūzijas efektīvais koeficients

Efektīvā difūzijas koeficienta aprēķināšanas formula, kad daļiņai pastāv iespēja šķērsot robežu

$$D_{eff} = \left(\frac{D_1}{\left(1 - \Phi + \frac{\Phi}{k}\right)} \right) \left(1 + \frac{(d(D_2 - kD_1)\Phi)}{D_2 + kD_1 - (D_2 - kD_1)\Phi} \right), \quad (2.37.)$$

kur

D_1 – difūzijas koeficients pamat vidē,

D_2 – difūzijas koeficients ieslēgumā,

d – apzīmē telpas dimensiju (1, 2 vai 3),

Φ – ieslēguma un režģa laukumu attiecība,

k – varbūtības koeficients šķērsot robežu.

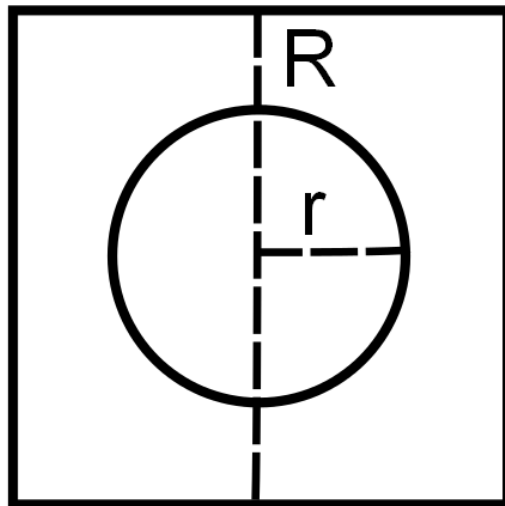
Efektīvā difūzijas koeficienta aprēķināšanas formula, kad daļiņas varbūtība šķērsot robežu ir nulle

$$D_{eff} = \frac{D_1}{1 + \Phi}, \quad (2.38.)$$

kur

D_1 – difūzijas koeficients pamat vidē,

Φ – ieslēguma un režģa laukumu attiecība.



2.8. att. Kvadrātisks režģis ar apaļu ieslēgumu

Lai aprēķinātu efektīvo difūzijas koeficientu ir nepieciešams zināt parametrus. Jāzina ieslēguma rādiuss, režģa izmērs.

Tad var aprēķināt k vērtību. Varbūtības koeficients šķērsot robežu

$$k = \frac{d_2 \left(\frac{l_2}{t_2} \right)}{d_1 \left(\frac{l_1}{t_1} \right)}, \quad (2.39.)$$

kur

d_1 – varbūtība šķērsot režģa robežu,

d_2 – varbūtība šķērsot ieslēguma robežu,

l_1 – soļa garums pamat vidē,

l_2 – soļa garums ieslēgumā,

t_1 – laiks starp diviem veiksmīgiem soļiem pamat vidē,

t_2 – laiks starp diviem veiksmīgiem soļiem ieslēgumā.

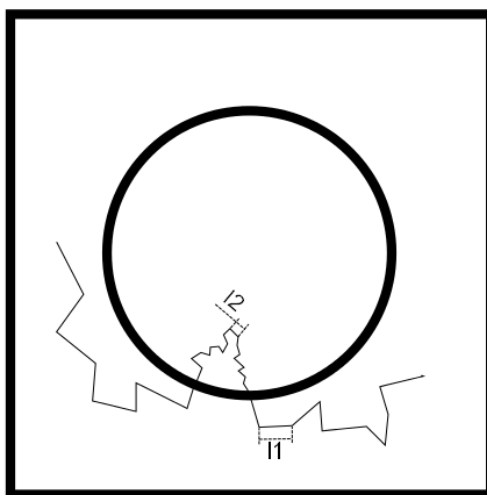
Ieslēguma un režģa laukumu attiecība

$$\Phi = \frac{\pi r^2}{R^2}, \quad (2.40.)$$

kur

r – ieslēguma rādiuss,

R – attālums starp defektiem.



2.9. att. Režģa un ieslēguma molekulas soļa garums

Vēl ir jāzina molekulas soļa garums l_1 un l_2 . Molekulas soļa garums ārpus ieslēguma l_1 un ieslēgumā l_2 .

Dotie parametri:

$$R = 10;$$

$$r = 2;$$

$$l_1 = 0,2;$$

$$t_1 = 1;$$

$$l_2 = 0,1;$$

$$t_2 = 1;$$

Izmantojot formulu (2.1) aprēķinām D_1

$$D_1 = \frac{0,2 * 0,2}{2 * 2 * 1} = \frac{0,04}{4} = 0,01$$

Izmantojot formulu (2.1.) aprēķinām D_2

$$D_2 = \frac{0,1 * 0,1}{2 * 2 * 1} = \frac{0,01}{4} = 0,0025$$

Izmantojot formulu (2.40.) aprēķinām Φ

$$\Phi = \frac{\pi 2^2}{10^2} = \frac{12,5663706}{100} = 0,125663706$$

Izmantojot formulu (2.39.) aprēķinām k

$$k = \frac{1\left(\frac{0,1}{1}\right)}{1\left(\frac{0,2}{1}\right)} = 0,5$$

Tagad var aprēķināt difūzijas efektīvo koeficientu izmantojot formulu (2.37.)

$$\begin{aligned} D_{eff} &= \left(\frac{0,01}{1,125663706} \right) \left(1 + \frac{-0,001193805207}{0,00875} \right) = \\ &= 0,00888364788 * 0,8635651192 = 0,007671608440423027296 \end{aligned}$$

Tātad difūzijas efektīvais koeficients pie šiem parametriem ir
0,007671608440423027296.

2.5. Molekulas gadījuma kustība.

Daudzām parādībām un norisēm dabā sabiedrībā, ekonomikā un tehnikā piemīt gadījuma raksturs. Var minēt gāzes molekulu kustību, vielas radioaktīvo sabrukšanu, telefona zvanu pienākšanu telefona centrālē, akciju tirgus indeksu svārstības u.c.

Teoriju, kura izskaidro molekulas kustības gadījuma raksturu – gadījuma klejošanu (random walking- angl.) – radīja M. Smoluhovskis un A. Einšteins. Nosaukumus gadījuma klejošana un Brauna kustība bieži lieto kā sinonīmus. Brauna kustība ir diskrēta stohastiska procesa piemērs. Skotu botāniķis R. Brauns 1827. gadā atklāja, ka augu sporas ūdenī kustās haotiski. Ūdens temperatūrai paaugstinoties, šī kustība paātrinās.

Apskatīsim vienkāršo gadījumu – viendimensiju gadījuma klejošanu. Pieņemsim, ka sākuma momentā daļiņa (molekula, atoms), kas veic gadījuma klejošanu, atrodas punktā ar koordināti $x_0=0$. Pieņemsim arī, ka ik pēc laika sprāža τ daļiņa veic lēcieni pa labi vai pa kreisi, ar vienādu lēciena garumu l . Stohastiska procesa rezultātā daļiņa no viena un tā paša sākuma stāvokļa ar noteiktu varbūtību var nonākt dažādos gala stāvokļos (viena lēciena gadījumā koordināte x vai nu palielināsies vai samazināsies par l). Daļiņas koordināti pēc i -tā lēciena apzīmēsim ar x_i . Acīmredzot daļiņas koordināte x_{i+1} pēc $i+1$ lēciena būs

$$x_{i+1} = x_i + l\gamma, \quad (2.41.)$$

kur

γ – gadījuma skaitlis, kas ar vienādu varbūtību pieņem vērtību -1 vai 1.

Tālāk izpētīsim kā mainās daļiņas koordināte x_n atkarībā no lēcienu skaita n un kādas īpašības piemīt gadījuma klejošanai. [3,68.lpp] [3,67.lpp]

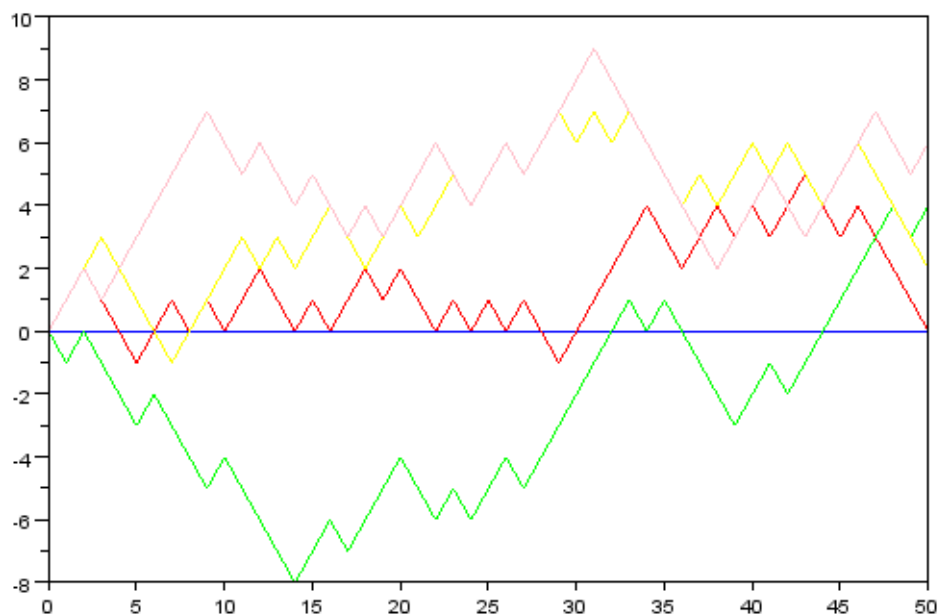
Ir izveidota SciLab funkcija:

```
function xs=walk1(n)
  xs=zeros(1,n)
  xs(1)=0
  for i=1:n
    p=rand()
    if (p <= 0.5)
      xs(i+1)=xs(i)+1
    elseif(p > 0.5)
      xs(i+1)=xs(i)-1
    end
  end
endfunction
timet=50;
xs=walk1(timet)
```

```

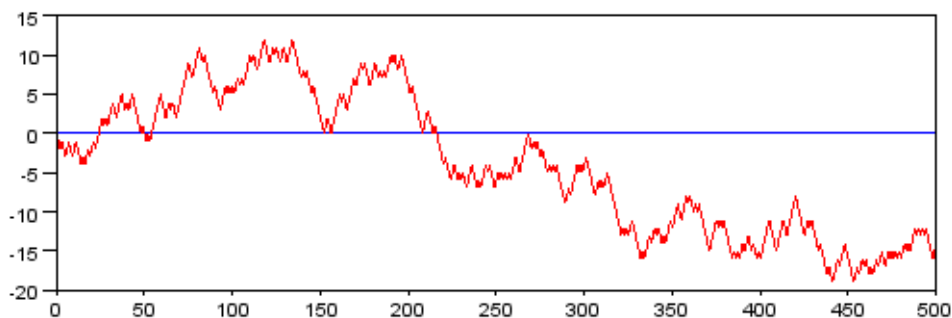
plot(0:timet,0,'b-')
x = 0 : 1 : timet;
plot2d(x, xs, style=color("red"))
xs = walk1(timet)
plot2d(x, xs, style=color("green"))
xs = walk1(timet)
plot2d(x, xs, style=color("yellow"))
xs = walk1(timet)
plot2d(x, xs, style=color("pink"))

```



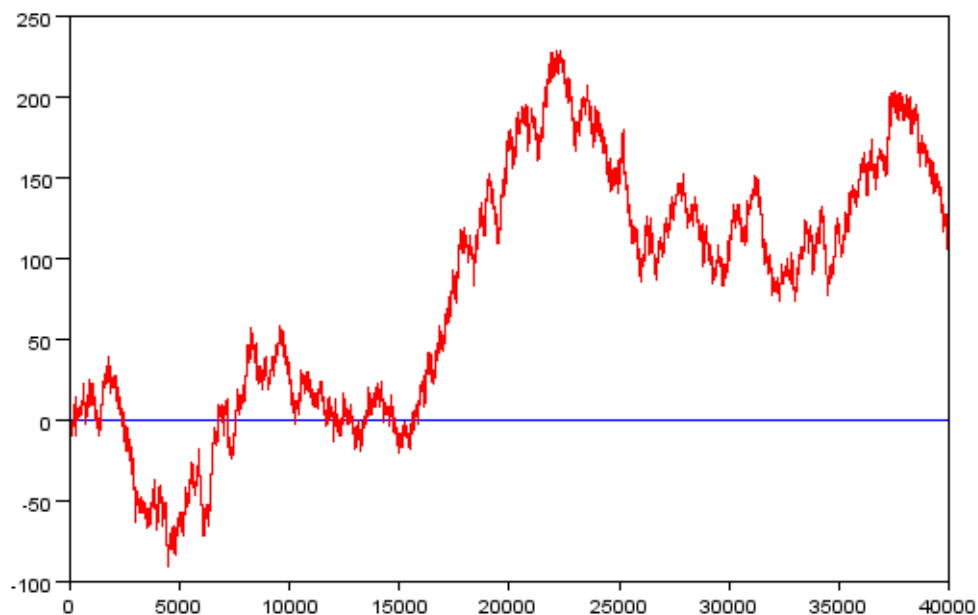
2.10. att. x_n kā funkcija no lēcienu skaita, ja $TIME = 50$, $l = 1$

Attēlā 2.10. x_n vērtības ir saliktas uz vertikālās ass. No kāda stāvokļa x_i nākamā solī ar varbūtību $\frac{1}{2}$ x_{i+1} vērtība būs $x_i + l$ vai $x_i - l$.



2.11. att. x_n kā funkcija no lēcienu skaita, ja $TIME = 500$, $l = 1$

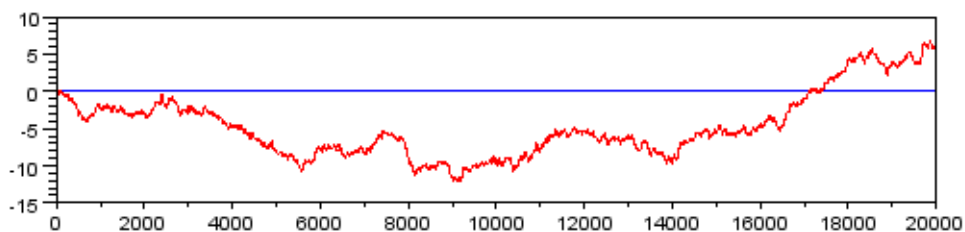
No attēla 2.11. redzam, ka x_n svārstās ap sākuma vērtību $x_0=0$. [3,69.lpp] [3,68.lpp]



2.12. att. x_n kā funkcija no lēcienu skaita, ja $TIME = 40000$, $l = 1$

Ja $TIME = 40000$, novērojama noviržu no nulles vērtības palielināšanās.

Attēlos 2.10. - 2.12.tika parādīti rezultāti atsevišķai gadījuma klejošanas realizācijai. Gadījuma klejošanas īpašību izpētei nosaka vidējās koordinātes vērtības x_{vid} (kā funkcija no klejošanas soļu skaita n) no vairākām neatkarīgām gadījuma klejošanas realizācijām. Aprēķinot vidējo x_{vid} no 200 neatkarīgām x_n realizācijām iegūstam grafiku(att. 2.13.). Var novērot, ka palielinot realizāciju skaitu, samazinās novirzes absolūtā vērtība. [3,70.lpp]



2.13. att. x_{vid} grafiks (x_n vidējā vērtība no 200 neatkarīgām realizācijām)

Apskatīsim viendimensijas gadījuma klejošanas teoriju. Noteiksim vidējo attālumu $\langle x(n) \rangle$, ko noiet daļiņa veicot n lēcienus un vidējo kvadrātisko novirzi

$\langle (x(n) - \langle x(n) \rangle)^2 \rangle$. Uzdevumu nedaudz vispārināsim, pieņemot, ka daļiņas lēciena varbūtība pozitīvā x ass virzienā ir p (tad $1 - p$ būs lēciena varbūtība pretēji x ass virzienam). Apzīmēsim ar $P_n(x)$ varbūtību, ka daļiņa atrodas punktā x laikā $t = n\tau$. Apzīmējot ar n_+ daļiņas lēcienu skaitu pozitīvā x ass virzienā un ar n_- daļiņas lēcienu skaitu pretēji x ass pozitīvajam virzienam ($n = n_+ + n_-$), varbūtību $P_n(x)$ var aprēķināt pēc formulas

$$P_n(x) = \frac{n!}{n_+!n_-!} p^{n_+} (1-p)^{n_-}, \quad (2.42.)$$

kur

$$x = n_+ l - n_- l.$$

Aplūkotais uzdevums ir identisks uzdevumam: noteikt varbūtību, ka metot monētu n reizes, n_+ reizes parādīsies ģērbonis un n_- reizes cipars. Uzdevuma atrisinājums dod Bernulli diskretā gadījuma lieluma sadalījuma formulu (2.42.). Atgādināsim, ka faktoriāla attiecība izteiksmē (2.42.) parāda, kāds ir atšķirīgu kombināciju skaits, ja, veicot n metienus, n_+ gadījumos parādīsies ģērbonis un n_- gadījumos cipars. Vienas šādas kombinācijas varbūtība ir $p^{n_+} (1-p)^{n_-}$.

No formulas (2.42.) varam atrast n lēcienos noietu vidējo attālumu $\langle x(n) \rangle$. Pēc vidējā lieluma definīcijas, ja zināma gadījuma lieluma sadalījuma funkcija $P_n(x)$, iegūstam

$$\langle x(n) \rangle = \sum_{n_+=0}^n P_n(x) x = \sum_{n_+=0}^n \frac{n!}{n_+!n_-!} p^{n_+} (1-p)^{n_-} (n_+ - n_-) l, \quad (2.43.)$$

[3,71.lpp]

Lai aprēķinātu (2.43.), atvasināsim Ņūtona binoma formulu pēc p

$$(p + q)^n = \sum_{n_+=0}^n \frac{n!}{n_+!n_-!} p^{n_+} q^{n_-}, \quad (2.44.)$$

Rezultātā iegūsim

$$n(p + q)^{n-1} = \sum_{n_+=1}^n \frac{n!}{n_+!n_-!} n_+ p^{n_+-1} q^{n_-}, \quad (2.45.)$$

Reizinot vienādojuma (2.45.) abas puses ar p , iegūsim izteiksmi

$$np(p+q)^{n-1} = \sum_{n_+=1}^n \frac{n!}{n_+!n_-!} n_+ p^{n_+} q^{n_-}, \quad (2.46.)$$

no kurienes gadījumā, kad $q = 1 - p$, seko

$$\sum_{n_+=0}^n \frac{n!}{n_+!n_-!} n_+ p^{n_+} (1-p)^{n_-} = np, \quad (2.47.)$$

Līdzīgi, atvasinot (2.44.) pēc q , iegūsim

$$\sum_{n_+=0}^n \frac{n!}{n_+!n_-!} n_- p^{n_+} (1-p)^{n_-} = np. \quad (2.48.)$$

No (2.43.), (2.47.) un (2.48.) iegūsim, ka

$$\langle x(n) \rangle = nl(p-q). \quad (2.49.)$$

Gadījumā, kad lēciena varbūtība abos x ass virzienos ir vienāda, ievietojot (2.49.) $p=q=1/2$, iegūsim, ka vidējais noietais attālums ir vienāds ar nulli:

$$\langle x(n) \rangle = 0. \quad (2.50.)$$

Simulācijā pie palielināta realizāciju skaita (att. 2.13.) parāda, ka x_{vid} svārstās ap vērtību 0. Šo rezultātu varēja paredzēt, jo vidēji lēcienu skaits pozitīvā un negatīvā x ass virzienā ir vienāds, tāpēc arī daļiņa „vidēji” paliek uz vietas.

Vidējo kvadrātisko novirzi nosakām pēc formulas

$$\langle (x(n) - \langle x(n) \rangle)^2 \rangle = \langle x^2(n) \rangle - \langle x(n) \rangle^2 = \langle x^2(n) \rangle. \quad (2.51.)$$

[3,72.lpp]

Te izmantojam jau iegūto rezultātu (2.50.). Pēc definīcijas

$$\langle x^2(n) \rangle = \sum_{n_+=0}^n P_n(x) x^2 = \sum_{n_+=0}^n \frac{n!}{n_+!n_-!} p^{n_+} (1-p)^{n_-} [(n_+ - n_-)l]^2. \quad (2.52.)$$

Tagad atvasināsim Ņūtona binoma formulu divas reizes pēc p .

Rezultātā iegūstam $\langle x^2(n) \rangle = 4pql^2$, no kurienes pie nosacījuma, ka $p=q=1/2$, seko

$$\langle x^2(n) \rangle = nl^2. \quad (2.53.)$$

Apkoposim iegūtos viendimensiju gadījuma klejošanas rezultātus: pie nosacījuma, ka $p=q=1/2$, vidējais noietais attālums ir nulle, vidējā kvadrātiskā novirze ir proporcionāla lēcienu skaitam un standartnovirzes $\sigma_x(n)$ izteiksme ir

$$\sigma_x(n) = \sqrt{\langle x^2(n) \rangle} = \sqrt{nl}. \quad (2.54.)$$

Veicot n lēcienu, daļiņa samērā lēni, pieaugot lēcienu skaitam, attālinās no sākumpunkta (proporcionāli kvadrātsaknei no lēcienu skaita). Šī ir raksturīgākā gadījuma klejošanas iezīme.

Izteiksmi (2.53.), ievērojot, ka laiks t ir $t = \tau n$, varam pārrakstīt izskatā

$$\langle x^2(t) \rangle = 2 \frac{l^2}{2\tau} t = 2Dt, \quad (2.55.)$$

kur

D – ir difūzijas koeficients

Difūzijas teorijā ir pazīstama formula $\langle x^2(t) \rangle = 2Dt$, kas saista vidējo kvadrātisko novirzi ar makroskopisku konstanti D , ko sauc par difūzijas koeficientu.

No (2.55.) izrietošā izteiksme

$$D = \frac{l^2}{2\tau} \quad (2.56.)$$

[3,73.lpp]

Šī izteiksme var tikt uzskatīta kā difūzijas koeficienta D mikroskopiska definīcija, izmantojot gadījuma klejošanas parametrus.

Var parādīt, ka divdimensiju un trīsdimensiju gadījuma difūzijas formulu (2.55.) un (2.56.) vietā attiecīgi būs

$$\langle r^2(t) \rangle = 4Dt, \quad D = \frac{l^2}{4\tau} \quad (2.57.)$$

un

$$\langle r^2(t) \rangle = 6Dt, \quad D = \frac{l^2}{6\tau}. \quad (2.58.)$$

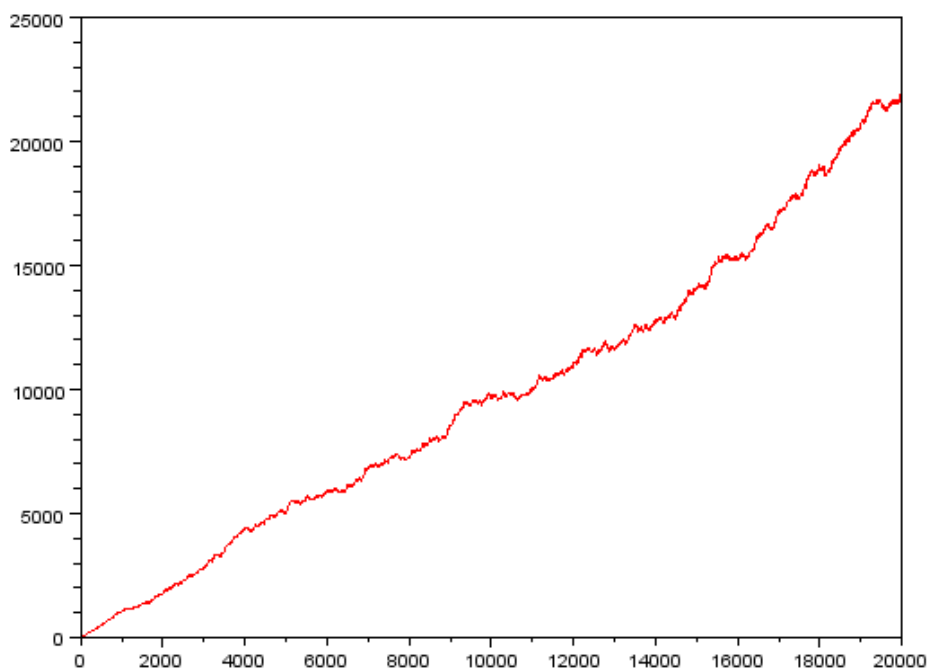
Kur $r(t)$ molekulas gadījuma nobīdes vektors, kurš atbilst laika momentam t , t.i.,

$r(t) = (x(t), y(t))$ divdimensiju gadījumā un $r(t) = (x(t), y(t), z(t))$ trīsdimensiju gadījumā.

Visās dimensijās saglabājas formula (2.53.)

$$\langle r^2(n) \rangle = nl^2. \quad (2.59.)$$

Atzīmēto likumsakarību pārbaudei ar grafika palīdzību noteiksim $\langle x^2(t) \rangle$ vidējo, ņemot 200 atsevišķas realizācijas. Rezultāts ir parādīts attēlā 2.14 un 2.15. [3,74.lpp]



2.14. att. Vidējā kvadrātiskā novirze kā laika funkcija



2.15. att. $\frac{\langle x^2(t) \rangle}{t}$ (vidējais no 200 realizācijām)

Salīdzinot (2.13.) ar attēlos (2.14.), (2.15.) redzamajiem rezultātiem, redzam, ka teorētiskie rezultāti un simulācijas rezultāti saskan. Atzīmēsim, ka modelī pieņemtās vērtības $l=1$ un $\tau=1$ dod $D=1/2$ (no (2.56.)), un tādā gadījumā $\frac{\langle x^2(t) \rangle}{t} = 1$.

Lai iegūtu ticamus rezultātus, nepieciešams vidējo ņemt no pietiekami liela realizāciju skaita.[3,75.lpp]

Pārlietu palielināt realizāciju skaitu arī nav saprātīgi, jo jāņem vērā gadījumu skaitļu ģeneratoru kvalitāte un citas modelēšanas kļūdas.

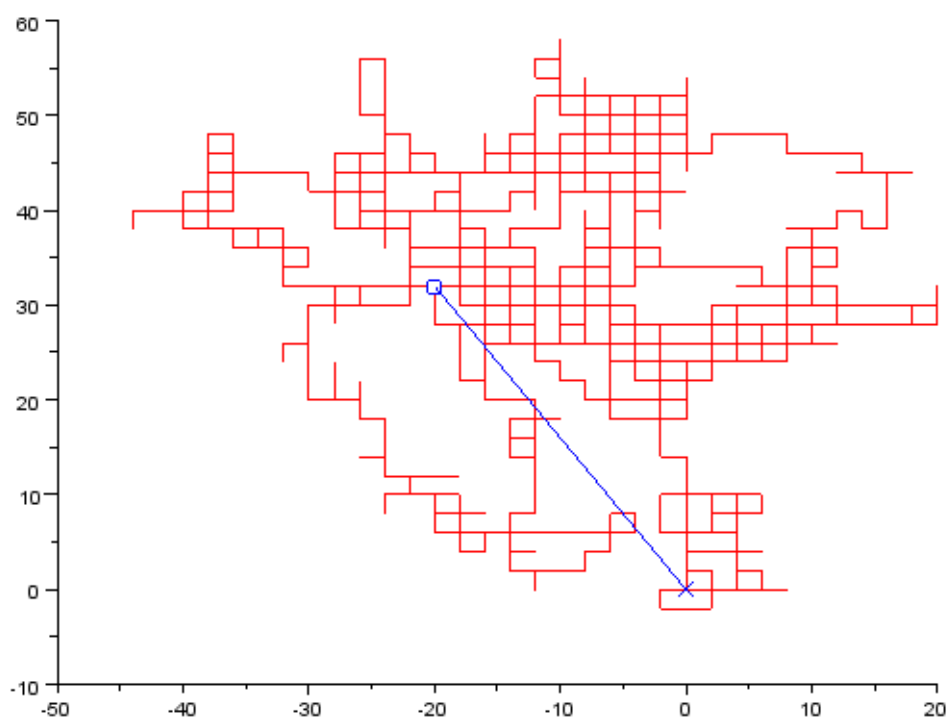
Tagad apskatīsim divdimensiju gadījuma klejošanas modeli, pieņemot, ka daļiņa ar vienādu varbūtību izvēlas vienu no četriem iespējamiem virzieniem: lēcieni ar garumu l pa labi vai pa kreisi x ass virzienā un lēcieni uz augšu vai leju y ass virzienā. Aprakstīto gadījuma klejošanu sauc par *klejošanu pa kvadrāta režģi*. [3,76.lpp]

SciLab kods:

```
function [xs, ys]=walk2Dlattice(n, x0, y0)

xs = zeros(1, n)
ys = zeros(1, n)
l = 2
xs(1) = x0
ys(1) = y0
for i = 1:n
    p = rand()
    if(p <= 0.25) //pa labi
        xs(i+1) = xs(i) + 1
        ys(i+1) = ys(i)
    elseif(p <= 0.5 & p > 0.25) //pa kreisi
        xs(i+1) = xs(i) - 1
        ys(i+1) = ys(i)
    elseif(p <= 0.75 & p > 0.5) //uz augsu
        xs(i+1) = xs(i)
        ys(i+1) = ys(i) + 1
    elseif(p <= 1 & p > 0.75) //uz leju
        xs(i+1) = xs(i)
        ys(i+1) = ys(i) - 1
    end
end
endfunction
x0 = 0;
y0 = 0;
timet = 1200;
[xs, ys] = walk2Dlattice(timet, x0, y0);
plot2d(xs, ys, style=color("red"));
plot(x0, y0, 'x');
plot(xs(timet+1), ys(timet+1), 'o');
x=linspace(x0,xs(timet+1),2);
y=linspace(y0,ys(timet+1),2);
plot(x,y);
```

Tipiska divdimensiju gadījuma trajektorija kvadrāta režģī ir parādīta attēlā 2.16., kur ar krustiņu ir apzīmēts sākuma punkts, ar kvadrātiņu gala punkts, kurš ir savienots ar sākuma punktu.[3,76.lpp]



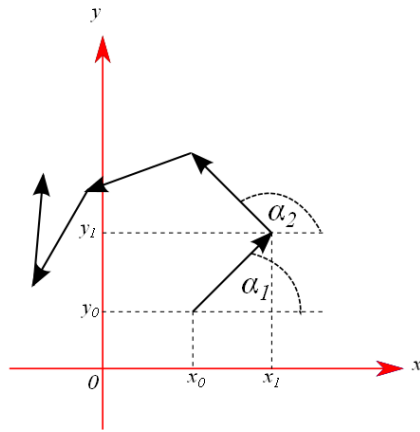
2.16. att. Gadījuma klejošanas ceļš, ja $TIME = 1200$, $l = 2$

Gadījuma klejošana pa režģi labi apraksta vakances vai starpmezglu atoma kustību uz kristāla virsmas ar vienkāršu kubisku struktūru.

Atšķirīga situācija, salīdzinot ar iepriekšējo gadījumu, veidojas, ja aplūkojam molekulas kustību gāzē. Te ir jāņem vērā, ka molekula pēc kārtējās sadursmes var kustēties jebkurā virzienā (vispārīgā gadījumā var uzdot arī varbūtības sadalījumu leņķim starp ātruma virzieniem pirms un pēc sadursmes).

Apzīmēsim ar α_n leņķi, ko veido molekulas n -tā lēciena ātruma vektors ar x ass pozitīvo virzienu (att 2.17.).

Apskatīsim vienkāršāku gadījumu, kur α_n ir gadījuma lielums ar vienmērīgu sadalījumu intervālā $(0, 2\pi)$ un lēciena garums ir konstants lielums l . [3,77.lpp]



2.17. att. Molekulas kustība gāzē

Ja ar γ_n apzīmējam gadījuma skaitli ar vienmērīgu sadalījumu intervālā no 0 līdz 1, kas atbilst n -tām lēcienam, tad $\alpha_n = 2\pi\gamma_n$.

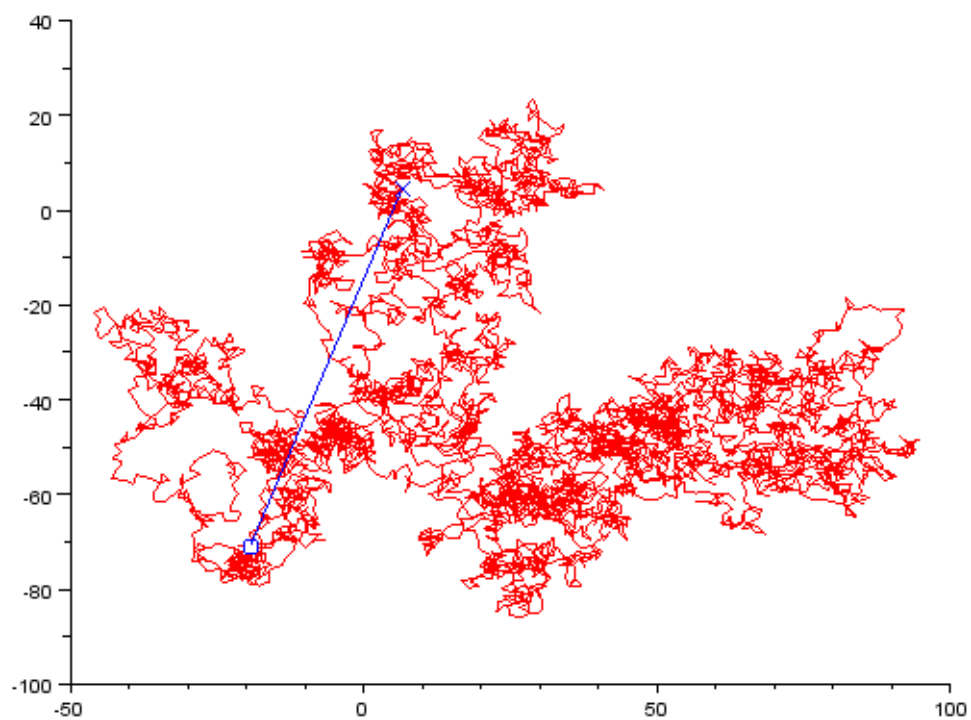
Ja molekulas sākuma stāvokļa koordinātes ir (x_0, y_0) , $n+1$ solī tās koordinātes būs

$$x_{n+1} = x_n + l * \cos(\alpha_{n+1}), \quad y_{n+1} = y_n + l * \sin(\alpha_{n+1}). \quad (2.60.)$$

[3,78.lpp]

SciLab kods:

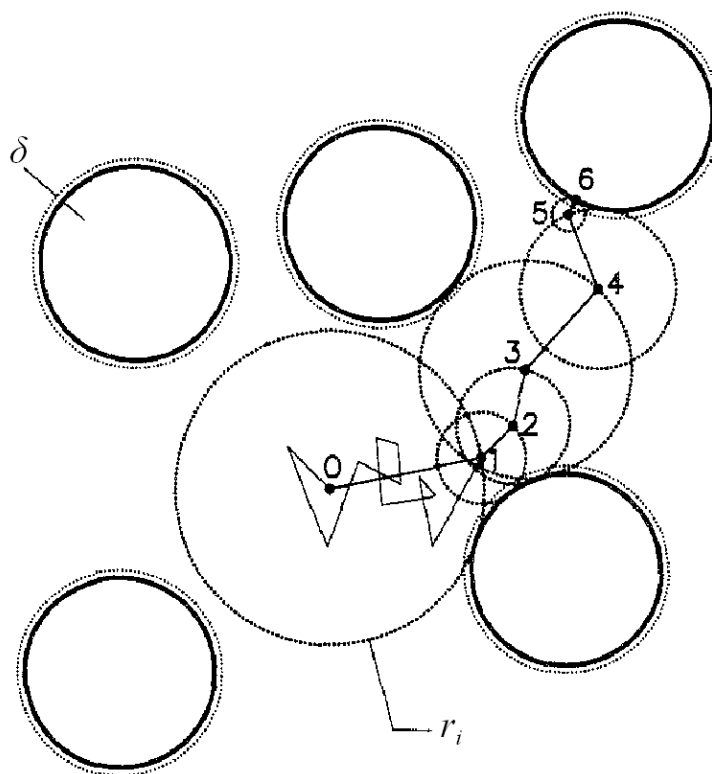
```
function [xs, ys]=walk2(n, x0, y0)
    xs = zeros(1, n)
    ys = zeros(1, n)
    l = 2
    PI = 3.141592
    xs(1) = x0
    ys(1) = y0
    for i = 1:n
        p = rand()
        alfa = 2*PI*p
        xs(i+1) = xs(i) + l * cos(alfa)
        ys(i+1) = ys(i) + l * sin(alfa)
    end
endfunction
x0 = 6.5;
y0 = 4.5;
time = 5000;
[xs, ys] = walk2(time, x0, y0);
plot2d(xs, ys, style=color("red"));
plot(xs(time+1), ys(time+1), 'o');
plot(x0, y0, 'x');
x=linspace(x0,xs(time+1),2);
y=linspace(y0,ys(time+1),2);
plot(x,y);
```



2.18. att. Molekulas gadījuma klejošana, ja $l = 2$, $n = 5000$, $x_0 = 6.5$, $y_0 = 4.5$

2.6. Pirmās pārejas metode (First-Passage method)

Pirmās pārejas algoritms ir domāts, lai paātrinātu programmas izpildes laiku un ātrāk iegūtu rezultātu.



2.19. att. Divdimensiju shematisks pirmās pārejas metodes attēlojums, kur δ – čaulas biezums, r_i – koncentriskās sfēras rādiuss

Izlasses daļiņas gadījumu kustībai nav nepieciešams modelēt katru soli. Tā vietā izveidojam koncentrisku sfēru ar rādiusu r_i , kurš nepārklājas ne ar vienu ieslēgumu; nākamo pozīciju uzskatītas par koncentriskās sfēras virsmu (izvēlās pēc nejaušības principa). Šis process ir atkārtots uzskaitot r_i^2 tik līdz daļiņa sasniedz ieslēgumu (t.i., daļiņa paliek δ attālumā no ieslēguma). [2,2.lpp]

3. PROGRAMMAS IZSTRĀDE

Programma izstrādāta programmēšanas valodās C# un C++. Programmas versija C# valodā ir ar grafisko vidi un ar papildus iespējām saglabāt un noformēt rezultātus. C++ versija darbojas ļoti ātri un ir bez grafiskās vides.

C# programmas versija ir izstrādāta izmantojot Microsoft Visual Studio 2010 Ultimate ar versiju 10.0.30319.1. Projekta tips Windows Forms Application.

Izpildot programmu parādās galvenais logs, kurā ir lauki, kuros jāievada parametri:

- attālums starp defektiem – R ,
- defekta (ieslēguma) rādiuss – r ,
- lēciena laiks ārpusē – t_1 ,
- brīvā ceļa garums – l_1 ,
- daļiņu skaits – n ,
- maksimālais soļu skaits – $time$,
- varbūtība iziet cauri robežai $(1 - 2) - d_1$,
- varbūtība iziet cauri robežai $(2 - 1) - d_2$,
- brīvā ceļa garums defektā – l_2 ,
- lēciena laiks defektā – t_2 .

No šiem parametriem programma izrēķina:

- difūzijas efektīvo koeficientu,
- daļiņu koncentrāciju pamat vidē / daļiņu koncentrāciju ieslēgumos,
- vidējo kvadrātisko novirzi,
- laiku ieslēgumos / difūzijas laiku,
- difūziju / difūziju teorētiski,
- cik reižu daļiņa ienāk ieslēgumos,
- kļūdu rēķinot difūzijas efektīvo koeficientu.

Kā arī programma uzzīmē grafisku difūzijas attēlojumu, kurā var redzēt kā notikusi difūzija pie ievadītiem parametriem. Ir iespējams saglabāt iegūtas difūzijas attēlu kā *BMP* failu. Ir iespējams mainīt krāsas, izvēloties vienu no iespējamajiem variantiem.

Vēl programma uzzīmē grafikus vidējai kvadrātiskai novirzei, ja, protams, tā nav vienāda ar nulli.

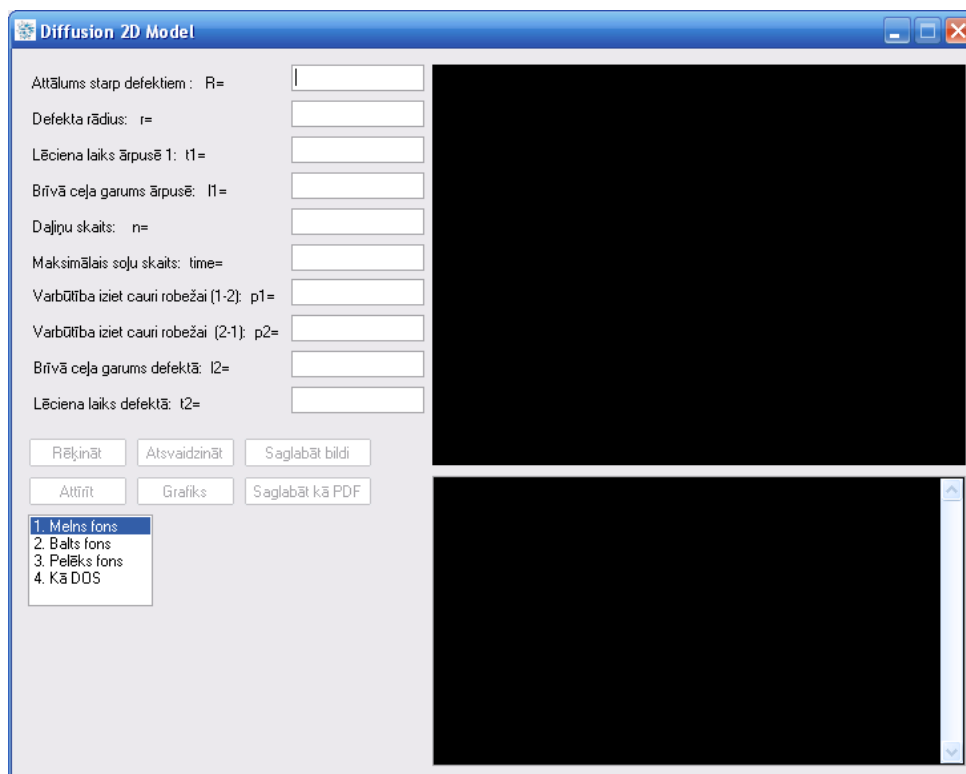
Un visbeidzot ir iespējams saglabāt visus modelēšanas rezultātus *PDF* failā. Programma izveido *PDF* failu, kurš satur visus rezultātus ieskaitot difūzijas grafisku attēlu, vidējās kvadrātiskās novirzes grafikus.

Papildus ir izveidots programmas instalators, kas atvieglo programmas uzstādīšanu, jo programma iekļauj sevī papildus bibliotēkas.

3.1. Grafiskās vides izstrāde

Grafisko vidi ar MS Visual Studio ir viegli izveidot, jo ir pieejami daudzi grafiski rīki, ar kuriem vienkārši var izveidot visus ievades laukus izmantojot *Toolbox* pieliekam *TextBox* objektus, kā nosaukumus ievades laukiem pieliekam *Label* objektus. Vēl ir nepieciešams laukums, kurā attēlosim difūzijas procesu grafiski. Priekš tā izmantosim *PictureBox* objektu, un uz tā zīmēsim attēlu.

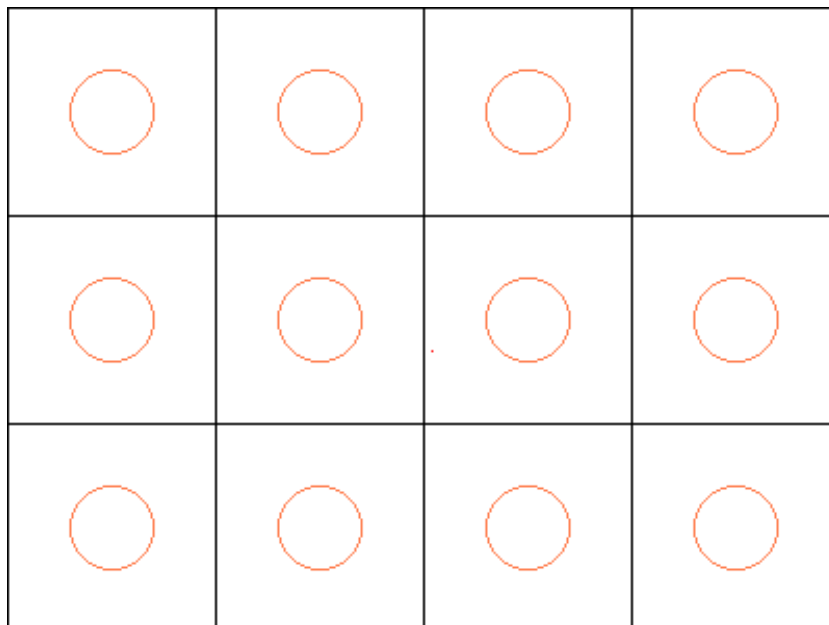
Tālāk pieliekam vajadzīgās pogas izmantojot *Button* objektus. Un vēl pieliekam sarakstu *ListBox* objektu, no kura varēs izvēlēties krāsas. Rezultātā tiek iegūts šāds vizuālais izskats:



3.1. att. Grafiskā vide

3.2. Galvenais algoritms

Sākuma nepieciešams izveidot lakumu, uz kura notiks difūzijas procesa modelēšana. Programmā izveidojam kvadrātisku režģi ar apaļiem ieslēgumiem.



3.2. att. Kvadrātisks režģis ar apaļiem ieslēgumiem

Lai uzzīmētu šādu laukumu nepieciešams *PictureBox* objekts, uz kura notiks zīmēšana. Kā arī izveidoju funkciju, kura maina krāsas, ar kurām tiek zīmēts process.

```
private void setColor(int id)
{
    if(colorID == 0)
    {
        cl_circle = Color.Blue;
        cl_square = Color.Crimson;
        cl_dot = Color.Lime;
        cl_bg = Color.Black;
    }
    else if (colorID == 1)
    {
        cl_circle = Color.Coral;
        cl_square = Color.Black;
        cl_dot = Color.Red;
        cl_bg = Color.White;
    }

    else if (colorID == 2)
    {
        cl_circle = Color.PaleGreen;
        cl_square = Color.Silver;
        cl_dot = Color.SpringGreen;
        cl_bg = Color.SlateGray;
    }
}
```

```

        else if (colorID == 3)
        {
            cl_circle = Color.Yellow;
            cl_square = Color.Silver;
            cl_dot = Color.Moccasin;
            cl_bg = Color.Blue;
        }
    }

```

Šai funkcijai padod parametru un tā atkarībā no parametra nomaina krāsas.

Tātad viss sākās, kad lietotājs ir ievadījis visus eksperimenta datus un nospiedis pogu 'Rēķināt'.

```

private void buttonCalculate_Click(object sender, EventArgs e)
{
    setColor(colorID);
    Console.WriteLine("Calculating...");

    double radius = 100 * rrr / rr;
    radius = radius * 2;
    bmp = new Bitmap(sizeX, sizeY);

    g = drawArea.CreateGraphics();
    g2 = Graphics.FromImage(bmp);
    g.Clear(cl_bg);
    g2.Clear(cl_bg);
    Pen pen1 = new Pen(cl_square);
    Pen pen2 = new Pen(cl_circle);

    for (int i = 0; i < 3; i++)
    {
        for (int i2 = 0; i2 < 4; i2++)
        {
            g.DrawRectangle(pen1, i2 * 100, i * 100, 100, 100); //uzzime
cetrsturus

            g2.DrawRectangle(pen1, i2 * 100, i * 100, 100, 100);

            g.DrawEllipse(pen2, ((i2 * 100) + 50) - (int)radius / 2, ((i *
100) + 50) - (int)radius / 2, (int)radius, (int)radius); //uzziimee rinkjus

            g2.DrawEllipse(pen2, ((i2 * 100) + 50) - (int)radius / 2, ((i *
100) + 50) - (int)radius / 2, (int)radius, (int)radius);

        }
    }

    Dif();
}

```

Sākuma izsauca funkciju, kura uzstāda izvēlētās krāsas. Pēc tam aprēķinu rādiusu riņķim, kas ir ieslēgums kvadrātiskā režģī. Izmantoju divus *Graphics* objektus, kuri paralēli zīmē vienu un to pašu attēlu, jo tas attēls, kurš zīmē tieši uz *PictureBox drawArea*

var pazust minimizējot logu, tāpēc, kad aprēķini beidzas, iestādu bildi, kura ir uzzīmēta ar *Graphics g2* objektu, kā:

```
drawArea.Image = bmp;
```

Tā, lai kods ir īsāks uzzīmēju četrstūrus un riņķus četrstūru centrā izmantojot divus *for* ciklus. Un tiek uzzīmēts laukums līdzīgs kā attēlā 3.2.

Tālāk tiek izsaukta funkcija *Diff()*, kura aprēķinās nepieciešamos datus.

Funkcijā *Diff()* aprēķinām D_1 un D_2 pēc formulas (2.1.).

```
dif1 = 11 * 11 / (4 * t1);  
if (t2 > 0)  
{  
    dif2 = 12 * 12 / (4 * t2);  
}  
  
else  
{  
    dif2 = 0;  
}
```

Tālāk aprēķinām Φ pēc formulas (2.40.).

```
f = pi * (r * r) / (rr * rr);
```

Tad aprēķinām D_{eff} no ievadītajiem parametriem. Lai aprēķinātu D_{eff} vispirms nepieciešams aprēķināt k vērtību. Pēc formulas (2.30.) vai (2.39.) aprēķinām k vērtību, jo formula (2.39.) ir pārveidota formula (2.30.). Tagad var aprēķināt D_{eff} pēc formulas (2.28.) vai (2.37.).

```
if (p1 == 0)  
{  
    Dteor = dif1 / (1 + f);  
}  
  
else  
{  
    k = (p2 * (12 / t2)) / (p1 * (11 / t1));  
  
    Dteor = (dif1 / (1 - f + f / k)) * (1 + (2 * (dif2 - k * dif1) * f) / (dif2 + k * dif1 - (dif2 - k * dif1) * f));  
}
```

Kā redzams kodā, ja daļiņas varbūtība p_1 šķērsot vides no 1 – 2 ir nulle, tad D_{eff} tiek aprēķināts pēc formulas (2.38.).

Tad seko galvenais *for* cikls, kurš strādā tik ilgi, cik liels ir daļiņu skaits. Šajā ciklā tiks aprēķinātas konkrētas daļiņas pozīcija, un atlikts galapunkts uz *PictureBox*, kā arī citas nepieciešamās vērtības.

```
for (n1 = 1; n1 <= nn; n1++)  
{  
    .  
    .  
    .  
}
```

Mainīgais *nn* ir daļiņu skaits, ko ievadījis lietotājs.

Tālāk kodā ir daudzas pārbaudes, daļiņas gadījuma kustība, skaitās daļiņu koncentrācija katrā vidē un citas operācijas, bet tās sīkāk var apskatīties programmas kodā.

3.3. Konsoles izvads uz *TextBox*

Lai izvadītu konsoles tekstu uz *TextBox* ir nepieciešams izveidot klasi, kura novirzīt konsoles izvada plūsmu uz *TextBox* objekta. Klase izskatās šādi:

```
public class TextBoxStreamWriter : TextWriter
{
    TextBox _output = null;

    public TextBoxStreamWriter(TextBox output)
    {
        _output = output;
    }

    public override void Write(char value)
    {
        base.Write(value);
        _output.AppendText(value.ToString());
    }

    public override Encoding Encoding
    {
        get { return System.Text.Encoding.UTF8; }
    }
}
```

Tālāk ir jāizveido izvades plūsmas objekts no klases *TextWriter* un jānorāda, kurā *TextBox* objektā izvadīt datus no *TextWriter*. Visbeidzot jānorāda, ka izvades plūsma no konsoles ir jāpāradresē uz *TextWriter* objektu.

```
TextWriter _writer = null;
_writer = new TextBoxStreamWriter(textBoxConsole);
Console.SetOut(_writer);
```

Tādā veidā viss, kas būs rakstīts konsoles izvadā, nonāk uz *TextBox*. Piemēram, ierakstot:

Console.WriteLine(„Helo world!“);

plūsma pāradresēsies un nonāks uz *TextBox*.

3.4. Grafiku zīmēšanas bibliotēka

Programmā ir iespējams uzzīmēt grafiku, kā arī saglabāt to formātā *BMP* kā bildi. Lai zīmētu grafikus es izmantoju bibliotēku *ZedGraph*. Tā ir bezmaksas bibliotēka dažādu grafiku un diagrammu zīmēšanai, vairāk informācijas par šo bibliotēku ir pieejams šeit [30]. Šajā mājas lapā var atrast dažādus piemērus, kā veidot grafikus izmantojot C#.

Programmā izmantojot šo bibliotēku, es attēloju vidējo kvadrātisko novirzi. Vispirms bija nepieciešams pievienot *ZedGraph.dll* failu projektam, lai varētu izmantot bibliotēkas funkcijas.

```
double[] x;  
double[] y;  
  
x = new double[nn + 1];  
y = new double[nn + 1];  
  
for (n1 = 1; n1 <= nn; n1++)  
{  
    .  
    .  
    .  
    rkvad = (ntr1 * rr + nxt1 - nx0) * (ntr1 * rr + nxt1 - nx0) + (ntr2 * rr + nyt1 -  
    ny0) * (ntr2 * rr + nyt1 - ny0);  
  
    sum = sum + (int)(rkvad);  
  
    r2vd = sum / n1;  
  
    x[n1] = n1 - 1;  
    y[n1] = r2vd;  
  
}
```

Tad klasē *Diffusion()* bija nepieciešams iegūt datus, no kuriem uzzīmēt grafiku. Lielajā ciklā notiek vidējās kvadrātiskās novirzes aprēķināšana. Cikla laikā iegūtos datus sarakstām masīvos *x* un *y*.

```
class DrawGraph  
{  
    public Image returnGraphImage(double[] x, double[] y, int xAxis)  
    {  
        ZedGraphControl zdk = new ZedGraphControl();  
        zdk.Width = 350;  
        zdk.Height = 250;  
        zdk.GraphPane.CurveList.Clear();  
        GraphPane myPane = zdk.GraphPane;  
        PointPairList spl = new PointPairList(x, y);  
        LineItem myCurve = myPane.AddCurve("<math>r^2>", spl, Color.Red,  
        SymbolType.None);  
        myPane.Title.Text = "<math>r^2>";  
        myPane.XAxis.Scale.Max = xAxis;  
        zdk.AxisChange();  
    }  
}
```

```

        zdk.Invalidate();
        zdk.Refresh();
        return zdk.MasterPane.GetImage();
    }
}

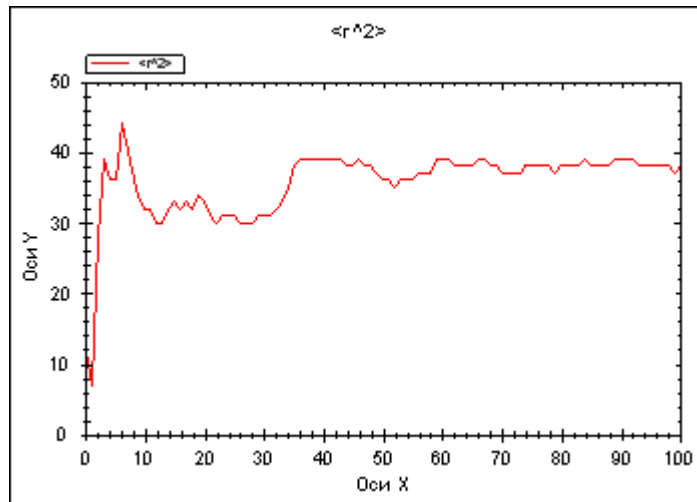
```

Pēc tam izmantojot klasi *DrawGraph()* uzzīmēju grafiku, kā redzams attēlā (3.2.). Funkcija *returnGraphImage()* sākumā uzzīmē grafiku no padotajiem parametriem un pēc tam atgriež grafika attēlu kā bildi, kuru vēlāk uzstādu uz kāda no *PictureBox* objektiem.

```

gra.returnGraphImage(x, y, x.Length)

```



3.3. att. Vidējās kvadrātiskās novirzes grafiks līdz $n = 100$

3.5. Eksperimenta rezultātu saglabāšana PDF formātā

Lai saglabātu rezultātus *PDF* formātā es izmantoju bezmaksas bibliotēku *PDFsharp*. Informācija par šo bibliotēku ir pieejama pa šo adresi [31]. Šajā mājas lapā var atrast dažādus piemērus, kā noformēt un saglabāt *PDF* dokumentus C# vidē.

Pašā sākumā bija nepieciešams iegūt *PDFsharp.dll* failu, nokompilējot bibliotēku no izejas koda. Tad pievienoju bibliotēku, kā referenci *Visual Studio* projektam.

```

using PdfSharp;
using PdfSharp.Drawing;
using PdfSharp.Pdf;
using PdfSharp.Pdf.IO;

```

Sākumā definēju izmantotās bibliotēkas.

```

PdfDocument s_document = new PdfDocument();

```

Nodefinēju mainīgo *PDF* dokumenta saglabāšanai.

```

s_document = new PdfDocument();
s_document.Info.Title = "Diffusion 2D Model";
s_document.Info.Author = "USER";
s_document.Info.Subject = "Diffusion 2D Model simulation results";
s_document.Info.Keywords = "Diffusion";
s_document.Options.NoCompression = true;
s_document.Options.CompressContentStreams = false;

```

```

DrawPage(s_document.AddPage());

SaveFileDialog dialog = new SaveFileDialog();
dialog.Filter = "PDF file (*.pdf)|*.pdf";
try
{
    if (dialog.ShowDialog() == DialogResult.OK)
    {
        s_document.Save(dialog.FileName);

    }
}
catch (Exception)
{
}

```

Tad piespiežot pogu *Saglabāt kā PDF* nostrādā notikums un tiek izveidots *PDF* dokuments, kuram uzstādām autoru, nosaukumu un citus parametrus, kā redzams kodā. Tad tiek izsaukta funkcija, kura izveido lapu, uz kuras tiks attēloti modelēšanas rezultāti. Kad *PDF* dokuments ir izveidots to saglabājam uz cietā diska, vai citiem datu nesējiem.

```

XStringFormat format = new XStringFormat();
XPdfFontOptions options = new XPdfFontOptions(PdfFontEncoding.Unicode,
PdfFontEmbedding.Always);
XGraphics gfx = XGraphics.FromPdfPage(page);
XRect rect = new XRect(0, 15, page.Width, page.Height);
XFont font = new XFont("Verdana", 14, XFontStyle.Bold, options);
gfx.DrawString("Rezultāts", font, XBrushes.MidnightBlue, rect,
XStringFormats.TopCenter);
DrawImageScaled(gfx, 1);

```

Funkcijas *DrawPage()* daļējs kods ir redzams augstāk. Kodā ir redzams kā izveidot virsrakstu uz *PDF* dokumenta lapas. *XRect* objektā norādām atkāpi no augšējās malas 15 pikseļu lielumā un atkāpi no kreisās malas 0 pikseļu, jo virsraksts „Rezultāts” tiks centrēts atkarībā no lapas izmēra. Kā redzams objekta *gfx* funkcijas *DrawString()* viens no parametriem ir teksta orientācija, tas var būt centrēts, nolīdzināts pēc labās vai kreisās malas.

```

private void DrawImageScaled(XGraphics gfx, int number)
{
    XImage image = XImage.FromGdiPlusImage(drawArea.Image);
    gfx.DrawImage(image, 150, 150);
}

```

Šī bibliotēka paver daudz plašākas iespējas, ir iespējams zīmēt grafikus un citas interesantas lietas.

3.6. Mersenne Twister algoritms

Mersenne twister ir pseidogadījumskaitļu ģenerators, kuru 1997. gadā izgudroja Makoto Matsumoto un Takudži Nišimura. Šis algoritms ir bāzēts uz matricas lineāru atkārtoto pāri galīgam binārajam laukam F_2 . Tas nodrošina ļoti augstas kvalitātes pseidogadījumskaitļus.

Algoritma nosaukums cēlies no tā, ka perioda garums ir izvēlēts Mersenne prime skaitlis. Jaunākais un visvairāk izmantotais Mersenne Twister algoritms ir *MT19937* ar 32-bitu vārdu garumu. Ir arī 64-bitu *MT19937-64* algoritms, kurš ģenerē citu virkni.

Šī algoritma ģenerēto skaitļu atkārtotāšanās periods ir $[0, 2^{32} - 1]$ 32-bitu versijai un $[0, 2^{64} - 1]$ 64-bitu versijai. [23]

Tātad izmantojot šo algoritmu var iegūt labākus gadījuma skaitļus veicot kāda eksperimenta veikšanu, kā manā gadījumā daļiņas gadījuma klejošanu.

Mersenne Twister algoritma izejas kodi C, C++, C# valodās ir pieejami šeit [27].

Programmā es izmantoju *MT19937* algoritmu. Sākumā izveidoju *PseudoRandom.dll* failu no izejas koda šeit [28]. Pievienoju iegūto *dll* bibliotēku projektam.

```
using PseudoRandom;
```

Tad izveidoju *MersenneTwister* klases objektu un izsauca *genrand_real32()* funkciju, kura atgriež Mersenne Twister algoritmā ģenerētu gadījumskaitli intervālā $[0, 1]$.

```
Random rnd = new Random();
ulong h = (ulong)rnd.Next();

MersenneTwister mersenneTwister = new MersenneTwister(h);
nxt1 = mersenneTwister.genrand_real2() * rr;
nyt1 = mersenneTwister.genrand_real2() * rr;
```

Piezīme: *rr* – attālums starp defektiem.

```
for (int t = 0; t < 10; t++)
{
    ulong h = (ulong)rnd.Next(100);
    mersenneTwister = new MersenneTwister(h);
    Console.Write(mersenneTwister.genrand_real2() + ", ");
}
```

Piemēram, šāda koda rezultāts būs:

```
0,214686890132725, 0,548813502304256, 0,307934952201322, 0,885880498681217,
0,944496605312452, 0,989011513767764, 0,154162841616198, 0,675731420051306,
0,374540114309639, 0,51729788328521,
```

Gadījuma skaitļi ir iegūti no vairākām neatkarīgām rindām, kodu atkārtoti palaižot tie neatkārtosies, bet ja sēklas skaitlis h būs viens un tas pats, tad rezultāts būs vienāds un skaitļi nemainīsies, palaižot kodu atkārtoti.

Tādā veidā, visur, kur nepieciešams ģenerēt gadījuma skaitļus tiek izmatots šis algoritms.

3.7. Programmas instalatora izveide

Lai izveidotu programmas instalatoru izmantoju *Nullsoft Scriptable Install System* (NSIS). Pati programma ir pieejama šeit [29]. Lai izveidotu instalatoru ir nepieciešams uzrakstīt skriptu, kurā tiek norādīts, kuri faili ir jāinstalē un kurā vietā tos instalēt. Programmas instalatoru

Skripta kods:

```
;iekļauj Modern UI

!include "MUI2.nsh"

;-----
;pamatdala

;programmas nosaukums un faila nosaukums
Name "Diffusion 2D Model"
OutFile "dif2Dmodel-install.exe"

;noklseta instalesanas direktorija

InstallDir "$PROGRAMFILES\Diffusion 2D Model"

;nepieciešamas privilēģijas, lai instalētu šo programmu
RequestExecutionLevel admin

;-----
;interfeisa iestatījumi

!define MUI_ABORTWARNING

;-----

;lapas

!insertmacro MUI_PAGE_LICENSE "C:\Documents and Settings\Рабочий
стол\installer\License.rtf"
!insertmacro MUI_PAGE_DIRECTORY
!insertmacro MUI_PAGE_INSTFILES
```

```
!insertmacro MUI_UNPAGE_CONFIRM
!insertmacro MUI_UNPAGE_INSTFILES
```

```
;-----
```

Šeit pievienojam licences tekstu, kas parādīsies noteiktajā lapā. Tālāk izveidojam lapu, kurā tiks prasīts izvēlēties instalēšanas direktoriju.

Section

```
SetOutPath $INSTDIR
```

```
File dif.exe
File dif2Dmodel.exe
File PdfSharp.dll
File PseudoRandom.dll
File ZedGraph.dll
File ZedGraph.xml
```

```
SetOutPath "$INSTDIR\de"
File "de\ZedGraph.resources.dll"
```

```
...
```

```
SetOutPath $INSTDIR
```

```
WriteUninstaller "$INSTDIR\uninstall.exe"
```

```
CreateDirectory "$SMPROGRAMS\Diffusion 2D Model"
CreateShortCut "$SMPROGRAMS\Diffusion 2D Model\uninstall.lnk"
"$INSTDIR\uninstall.exe"
```

```
...
```

```
SectionEnd
```

Šajā daļā tiek norādīti visi faili, kurus nepieciešams ieinstalēt, kā arī palaidēju izveidošana uz darba virsmas un galvenās izvēlnes.

; atinstalesanas sekcija

```
Section "uninstall"
```

```
; sakuma izdzesam instalatoru un programmas failus, direktorijas
```

```
Delete "$INSTDIR\uninstall.exe"
Delete "$INSTDIR\dif2Dmodel.exe"
Delete "$INSTDIR\PdfSharp.dll"
Delete "$INSTDIR\dif.exe"
```

```
RMDir "$INSTDIR\de"
```

```
...
```

```
...
```

```
RMDir "$INSTDIR"
```

```
Delete "$DESKTOP\Diffusion 2D Model.lnk"
Delete "$DESKTOP\Diffusion 2D Model Console.lnk"
RMDir "$SMPROGRAMS\Diffusion 2D Model"
```

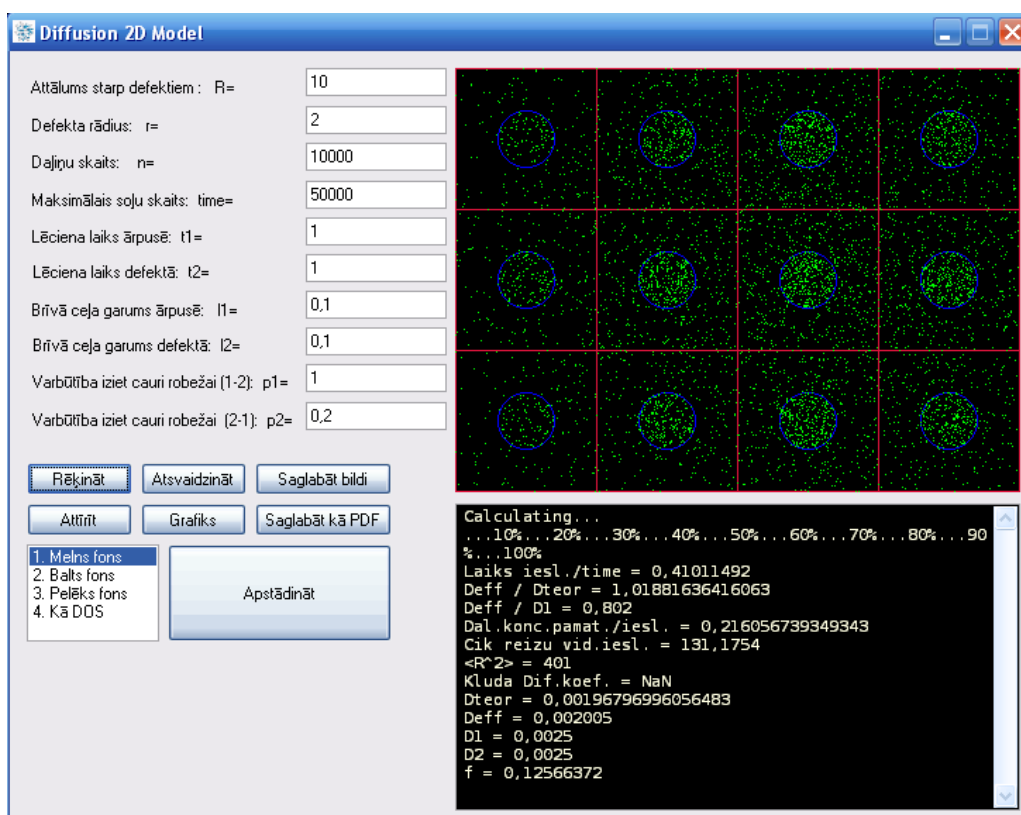
SectionEnd

Šajā sekcijā tiek norādīti visi faili, kuri ir jāizdzēš.

Kad skripts ir uzrakstīts saglabāju to kā .nsi failu un izmantojot NSIS palaižu un instalators tiek izveidots. Instalators ir notestēts uz Windows XP un Windows 7 versijām.

3.8. Modelēšana

Tagad, kad programma ir izveidota var veikt modelēšanu un rezultātu analīzi. Ir plānots salīdzināt D_{eff} teorētiskās formulas rezultātus ar iegūtiem difūzijas eksperimenta datiem un salīdzināt to atbilstību.



3.4. att. Modelēšanas rezultāts

Var redzēt, ka pie dotajiem parametriem tiek aprēķinātas vērtības. Šajā gadījumā D_{eff} ir modeļa difūzijas efektīvais koeficients, bet D_{teor} ir pēc formulas (2.37.) aprēķinātais difūzijas efektīvais koeficients. Tagad var salīdzināt iegūtos rezultātus.

Salīdzināšu teorētisko formulu ar praktiski aprēķināto difūzijas efektīvo koeficientu.

- Rezultāts pie parametriem:
- Attālums starp defektiem $R = 10$,
- Defekta rādiuss $r = 2$,

- Daļiņu skaits $n = 10000$,
- Maksimālais soļu skaits $time = 50000$,
- Lēciena laiks ārpusē $t_1 = 1$,
- Lēciena laiks defektā $t_2 = 1$,
- Brīvā ceļa garums ārpusē $l_1 = 0.1$,
- Brīvā ceļa garums defektā $l_2 = 0.1$,
- Varbūtība iziet cauri robežai (1 – 2): $p_1 = 1$,
- Varbūtība iziet cauri robežai (2 – 1): $p_2 = 0.2$.

Var redzēt, ka pie šiem parametriem iegūtas vērtības, kur:

$$D_{eff} / D_{teor} = 1,01881636416063$$

$$f = 0,12566372$$

Tālāk izmainām defekta rādiusu r , bet pārējās vērtības atstājam nemainīgas.

Rezultāts, kad $r = 1.5$:

$$D_{eff} / D_{teor} = 1,00153338994438$$

$$f = 0,0706858425$$

Rezultāts, kad $r = 3$:

$$D_{eff} / D_{teor} = 1,00978946186608$$

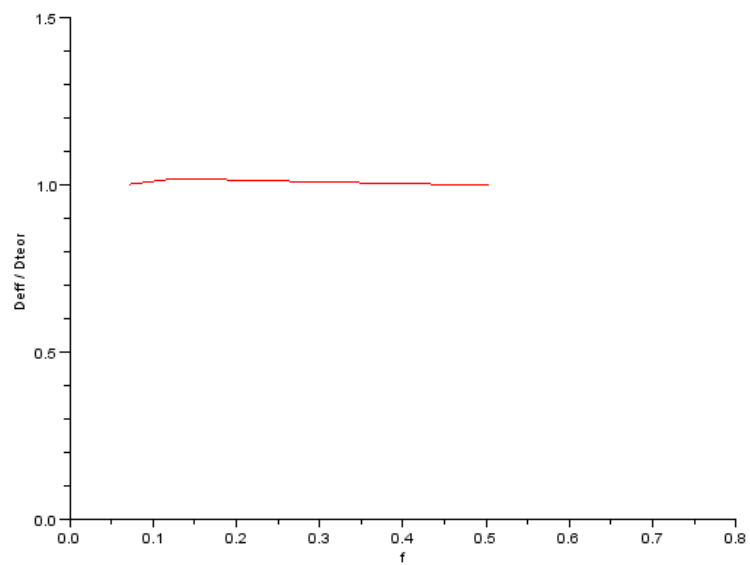
$$f = 0,28274337$$

Rezultāts, kad $r = 4$:

$$D_{eff} / D_{teor} = 0,998549172473922$$

$$f = 0,50265488$$

Tagad no iegūtajiem rezultātiem uzzīmēšu grafiku, kurā tiks attēlota D_{eff} / D_{teor} atkarībā no f , kur f ir tas pats Φ no teorijas.



3.5. att. D_{eff} / D_{teor} grafiks atkarībā no f vai Φ

Pēc attēlā 3.5. redzamā grafika var secināt, ka teorētiskā formula (2.37.) dod pietiekami precīzu rezultātu ar kļūdu aptuveni 2%.

SECINĀJUMI UN PRIEKŠLIKUMI

Secinājumi

Ir izdevies sasniegt darba mērķi – ir izstrādātas datorprogrammas C# un C++ programmēšanas valodās. Ir izveidots simulācijas modelis efektīvā difūzijas koeficienta noteikšanai. Ir izveidots interfeiss programmai, kas ļauj viegli mainīt modeļa parametrus un sekot aprēķina gaitai. Testa simulācijas rezultāti salīdzināti ar teoriju. Gūts apstiprinājums teorijas lietošanai gadījumos, kad ieslēgumi ir ķērāj centri ar atgriezenisku reakciju.

Programmai ir praktisks pielietojums, ir iespējams veikt datora eksperimentus pētot difūziju, kā arī saglabāt iegūtos datus. Teorētiskās formulas un modeļa rezultāti ir pietiekami precīzi ar kļūdu līdz 2%.

Priekšlikumi

Ir iespējams publicēt kodu kādā no kodu izplatīšanai paredzētām tīmekļa vietnēm (piem., SourceForge), lai to varētu izmantot citi cilvēki ar iespēju labot, papildināt un pielāgot programmu savām vajadzībām.

Iespējams būtu labi uztaisīt līdzīgu programmu, programmēšanas valodā JAVA, jo valoda JAVA ir vairākplatformu valoda, un dos iespēju lietot programmu arī uz Linux operētājsistēmām.

IZMANTOTĀS LITERATŪRAS UN AVOTU SARAKSTS

1. J.R. Kalnin, E.A. Kotomin, J. Maier, *Calculations of the effective diffusion coefficient for inhomogenous media*. Institute of Solid State Physics, 8 Kengaraga str., Riga LV-1063, Latvia. Planck-Institut fur Festkörperforschung, Heisenbergstr. 1, D-70569 Stuttgart, Germany, 2001. 8 lpp.
2. S. Traquoto, In Chan Kim, *Efficient simulation technique to compute effective properties of heterogeneous media*. Department of Mechanical and Aerospace Engineering and Department of Chemical Engineering, North Carolina State University, Raleigh, North Carolina 27695-7910, 1989, 3 lpp.
3. J. R. Kalniņš, G. Hilkeviča, E. Vītola, *Matemātiskā modelēšana*. Ventspils Augstskola Inženieru iela 101, Ventspils, LV-3601, 2008, 192 lpp., ISBN 9984-648-86-9
4. Andreas Ochner, Graeme E.Murch, *Cellular and Porous Materials: Thermal Properties Simulation and Prediction*. Willey VCH, 2009.
5. Dmitry S Novikov, Valerij G. Kiselev. *Effective medium theory of a diffusion-weighted signal*. NMR in Biomedicine 2010, Volume: 23, Issue: 7, Pages: 682-697
6. Isabelle Déléris, Imen Zouid, Isabelle Souchon, Ioan Cristian Tréléa, *Calculation of apparent diffusion coefficients of aroma compounds in dairy emulsions based on fat content and physicochemical properties in each phase*.
7. Els Fieremans, Yves De Deene, Steven Delputte, Mahir S.O. zdemir, Yves D'Asseler, Jelle Vlassenbroeck, Karel Deblaere, Eric Achten, Ignace Lemahieu, *Simulation and experimental verification of the diffusion in an anisotropic fiber phantom*.
8. Wolfgang Preis, *Surface exchange reactions and diffusion in composite materials: A finite element approach*. Chair of Physical Chemistry, University of Leoben, Franz-Josef-Strasse 18, A-8700 Leoben, Austria
9. Ying Shi (a), Jinsheng Xiaob (c), Shuhai Quana, Mu Panb, Liyan Zhanga, *Fractal model for prediction of effective hydrogen diffusivity of gas diffusion layer in proton exchange membrane fuel cell*.
10. Hsien-Ming Chang, Chungpin Liao, *A Parallel Derivation to the Maxwell-Garnett Formula for the Magnetic Permeability of Mixed Materials*.
11. Felderhof B.U., Deutch J.M., *J.Chem. Phys.* 64, 4551, 1976
12. Torquato S., *J. Stat. Phys.* 65, 1173, (1991)
13. Kalnin Yu.R, *Proc Latvian Acad. Sci. physics No 6, 13*, 1985
14. Kalnin Yu.R. and Zapol P., *Radiation Effects and Defects in Solids*. 137, 295, 1995

15. Manning J.R., *Diffusion Kinetics for Atoms in Crystals*. Princeton, Toronto, 1968
16. Jönsson B., Wennerström H., Nilsson P.G., and Line P., *Colloid and Polymer Science*. 264, 77, 1986
17. Ronis G.M., *Phys Rev. A* 36, 1908, 1987
18. Cukier R.I. *J.Phys.Chem. Solids* 87, 582, 1983
19. Bell G.M., *Trans.Farad.Soc.* 60, 1752, 1964
20. Diffusion, [cites April 10, 2012]. Available: <http://en.wikipedia.org/wiki/Diffusion>
21. Effective diffusion coefficient. [cites April 12, 2012 February]. Available: http://en.wikipedia.org/wiki/Effective_diffusion_coefficient
22. Point defects, [cites April 15, 2012]. Available: http://www.ndt-ed.org/EducationResources/CommunityCollege/Materials/Structure/point_defects.htm
23. Mersenne twister, [cites, April 14, 2012]. Available: http://en.wikipedia.org/wiki/Mersenne_twister
24. Диффузия, [просмотрен 10 апреля 2012 года]. Доступен: <http://ru.wikipedia.org/wiki/Диффузия>
25. Brauna kustība, [skatīts 2012.g. 10. aprīlī]. Pieejams: http://lv.wikipedia.org/wiki/Brauna_kustība
26. Диффузия. Статья из Химической энциклопедии, [просмотрен 10 апреля 2012 года]. Доступен: http://www.chemport.ru/chemical_encyclopedia_article_1184.html
27. Mersenne Twister in C, C++, C#, [cites April 15, 2012]. Available: <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/VERSIONS/C-LANG/c-lang.html>
28. Mersenne Twister C# source, [cites April 15, 2012]. Available: <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/VERSIONS/C-LANG/stlalv-MersenneTwister.zip>
29. Nullsoft Scriptable Install System (NSIS), [cites April 20, 2012]. Available: <http://nsis.sourceforge.net>
30. ZedGraph C# library, [cites April 21, 2012]. Available: <http://zedgraph.sourceforge.net/samples.html>
31. PDFsharp C# library, [cites April 21, 2012]. Available: <http://www.pdfsharp.net/>
32. Collision theory, [cites April 14]. Available: http://en.wikipedia.org/wiki/Collision_theory

GALVOJUMS

Ar šo es, Vitālijs Daņilevičs, galvoju, ka bakalaura darbs ir izpildīts patstāvīgi, konsultējoties ar darba vadītāju. No svešiem pirmavotiem ņemtā informācija ir norādīta ar atsaucēm, dati un definējumi ir uzrādīti darbā. Šis darbs tādā vai citādā veidā nav nekad iesniegts nevienai citai pārbaudījumu komisijai.

2012.gada _____

(paraksts)

Bakalaura darbs aizstāvēts Gala pārbaudījumu komisijas sēdē

2012. gada _____

un novērtēts ar atzīmi _____ .

Protokols Nr. _____

Gala pārbaudījumu komisijas

priekšsēdētājs _____
(paraksts)