

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Черкаський національний університет
імені Богдана Хмельницького
MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
Bohdan Khmelnytsky
National University of Cherkasy

DOI: 10.31651/2076-5886-2019-2

ISSN 2076-5886 (Print)

ВІСНИК
ЧЕРКАСЬКОГО НАЦІОНАЛЬНОГО
УНІВЕРСИТЕТУ
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО

Серія
ПРИКЛАДНА МАТЕМАТИКА. ІНФОРМАТИКА

BULLETIN
OF THE CHERKASY
BOHDAN KHMELNYTSKY
NATIONAL UNIVERSITY

APPLIED MATHEMATICS. INFORMATICS

ВИПУСК 2
ISSUE 2

Черкаси, 2019
Cherkasy, 2019

Засновник, редакція, видавець і виготовлювач –
Черкаський національний університет імені Богдана Хмельницького.
Свідоцтво про державну реєстрацію КВ № 16161-4633 ПР від 11.12.2009.
Свідоцтво про державну реєстрацію КВ № 23973-13813 Р від 21.05.2019.

Журнал розрахований на математиків, спеціалістів у галузі ІТ, викладачів, науковців,
аспірантів, студентів.

Випуск № 2 наукового журналу «Вісник Черкаського національного університету імені Богдана Хмельницького. Серія «Прикладна математика. Інформатика» рекомендовано до друку та до поширення через мережу Інтернет Вченою радою Черкаського національного університету імені Богдана Хмельницького (протокол № 2 від 31.10.2019 року).

Журнал реферується Українським реферативним журналом «Джерело» (засновники: Інститут проблем реєстрації інформації НАН України та Національна бібліотека України імені В.І. Вернадського), індексується Google Scholar.

Редакційна колегія серії:

Пасічний М.О., к.ф.-м.н., доц., ЧНУ ім. Б. Хмельницького (головний редактор);
Головня Б.П., д.т.н., доц., ЧНУ ім. Б. Хмельницького (заступник головного редактора);
Сердюк О.А., к.е.н., ЧНУ ім. Б. Хмельницького (відповідальний секретар);
Соловійов В.М., д.ф.-м.н., проф., КДПУ; Запорожець Т.В., д.ф.-м.н., проф., ЧНУ ім. Б. Хмельницького; Шквар Є.О., д.т.н., проф., Zhejiang Normal University (Zhejiang, China);
Ляшенко Ю.О., д.ф.-м.н., проф., ЧНУ ім. Б. Хмельницького; Дідковський Р.М., д.т.н., доц., провідний інженер, програміст, ТОВ «ФАЄРБЛІНК» (м. Черкаси, Україна);
Гаєв Є.О., д.т.н., проф., НАУ; Сторожук Н.В., к.ф.-м.н., ЧНУ ім. Б. Хмельницького;
Лиля Д.М., к.ф.-м.н., ЧНУ ім. Б. Хмельницького; Бабенко С.В., к.ф.-м.н., ЧНУ ім. Б. Хмельницького.

За дотримання права інтелектуальної власності, достовірність матеріалів та обґрунтування висновків відповідають автори.

Адреса редакційної колегії:

18031, Черкаси, бул. Шевченка, 79
Черкаський національний університет імені Богдана Хмельницького, корпус № 3, к. 307
e-mail: ami.cdu@ukr.net

З електронною версією журналу можна ознайомитися за адресою: <http://ami-ejournal.cdu.edu.ua/>

Founder, editorial, publisher and manufacturer –
Bohdan Khmelnytsky National University of Cherkasy.
State registration certificate: KV No. 16161-4633 PR dated 11.12.2009.
State registration certificate: KV No. 23973-13813 R dated 21.05.2019.

This journal is meant for mathematicians, IT specialists, teachers, researchers, postgraduates and students.

Issue № 2 of the scientific journal «Bulletin of the Cherkasy Bohdan Khmelnytsky national university. Applied mathematics. Informatics» is recommended for publication and dissemination through the Internet by the Academic Council of Bohdan Khmelnytsky National University of Cherkasy (protocol number 2 dated 31.10.2019).

The journal is referenced by the Ukrainian abstract journal "Dzherelo" (founders: the Institute of Information Recording Problems of the National Academy of Sciences of Ukraine and Vernadsky National Library), is indexed in Google Scholar.

Editorial board of the series:

Pasichnyy M.O., Candidate of Physical and Mathematical Sciences, Associate Professor (Editor in Chief); Golovnya B.P., Doctor of Technical Sciences, Associate Professor (Deputy Chief Editor); Serdiuk O.A., Candidate of Economic Sciences (executive secretar); Soloviev V. M., Doctor of Physical and Mathematical Sciences, Professor; Zaporozhets T.V., Doctor of Physical and Mathematical Sciences, Professor; Shkvar Ye.O., Doctor of Technical Sciences, Professor; Lyashenko Y.O., Doctor of Physical and Mathematical Sciences, Professor; Didkovsky R.M., Doctor of Technical Sciences, Associate Professor; Gayev Ye.A., Doctor of Technical Sciences, Professor; Storozhuk N.V., Candidate of Physical and Mathematical Sciences; Lila D.M., Candidate of Physical and Mathematical Sciences; Babenko S.V., Candidate of Physical and Mathematical Sciences.

The authors are responsible for the observance of the intellectual property right, for the reliability of the materials and for the substantiation of the conclusions.

Editorial office address:
18031, Cherkasy, Shevchenko Blvd., 79
Bohdan Khmelnytsky National University of Cherkasy, building 3, ap. 307
e-mail: ami.cdu@ukr.net

All electronic versions of articles are available on the website edition <http://ami-ejournal.cdu.edu.ua/>

© Bohdan Khmelnytsky National University of Cherkasy, 2019
© Copyright by the contributors

СЕКЦІЯ «ПРИКЛАДНА МАТЕМАТИКА»

УДК 532.135+612.11

DOI 10.31651/2076-5886-2019-2-4-10

PACS 47.50

БАТЮК Лілія Василівна

кандидат біологічних наук, доцент, доцент кафедри медичної і біологічної фізики та медичної інформатики, Харківський національний медичний університет
e-mail: liliyabatyuk24@gmail.com
ORCID 0000-0003-1863-0265

КІЗІЛОВА Наталія Миколаївна

доктор фізико-математичних наук, професор, професор кафедри прикладної математики, Харківський національний університет ім.В.Н. Каразіна
e-mail: n.kizilova@gmail.com
ORCID: 0000-0001-9981-7616

ДОСЛІДЖЕННЯ ТЕЧІЙ БІНГАМІВСЬКИХ НАНОРІДИН ПО МІКРОКАНАЛАХ

Розглядаються стаціонарні течії нанорідин, які являють собою концентровані суспензії наночастинок та широко використовуються в системах охолодження мікроприладів, в біомедичних системах та в багатьох інших. Рух нанорідини описується рівняннями Нав'є-Стокса з умовами прослизання другого порядку на шорстких поверхнях каналів, а їх реологія відповідає бінгамівським рідинам. Отримані розв'язки задачі у випадку течій по круговим трубкам, між паралельним пластинами та співвісними циліндрами. Обчислені в'язке тертя на стінках, об'ємна витрата рідини, обговорені відміни від випадку класичних бінгамівських рідин та можливість практичного застосування нових ефектів, зокрема для обробки мікрооб'ємів суспензії крові, яка є бінгамівською рідиною.

Ключові слова: *нанофлюїдика, бінгамівська рідина, стаціонарні течії, шорсткі стінки.*

Вступ

Останнім часом велику увагу привертають новітні мікро- та нанотехнології, які дозволяють розробляти різні мікроскопічні пристрої для численних технічних, біомедичних та інформаційних технологій [1-3]. Велика роль в таких пристроях належить системам циркуляції рідин, які використовуються для постачання речовин, виносу продуктів хімічних реакцій, нагрівання або охолодження компонент пристроїв та ін. Особливо перспективними є мікро/нанорідини, які являють собою суспензії мікро або наночастинок з розмірами $a \sim 10-500 \mu\text{m}$ та $a \sim 10-500 \text{nm}$ [4]. Завдяки унікальним фізичним властивостям наночастинок отримувані з ними мікро/нанорідини є більш ефективними, ніж звичайні рідини, тому що мають значно вищі теплопровідності, теплоємності, електропровідності та інші властивості. Для розробки таких мікрорідинних систем потрібні детальні інженерні розрахунки параметрів течій та оптимізація дизайну пристроїв, для чого потрібна розробка відповідних математичних моделей та розв'язки задач оптимізації.

Численні експериментальні дослідження течій рідин в мікро/наноканалах показали, що навіть при ламінарних течіях по циліндричних трубках об'ємна витрата рідини є набагато більшою, ніж передбачувана за формулою Пуазейля [2]. Це означає,

що гідравлічний опір трубки є за яких-то причин значно меншим, ніж опір макроскопічних трубок. Так, при фільтрації води в фільтрах, утворених паралельними пучками вуглеводних нанотрубок, загальний опір був в 200 разів менший, ніж розрахований за формулою Пуазейля для сукупності трубок [5]. Оскільки на довжинах >10 нм квантові ефекти ще не проявляються і рух рідини визначається рівняннями Нав'є-Стокса [2], було показано, що цей ефект пов'язаний з дифузійним відбиттям частинок мікро/нанорідин на шорстких стінках [1,2,6].

Характерний розмір мікро-, а особливо наночастинок, відповідає розміру неоднорідностей твердої стінки трубки або каналу, тому характерні для в'язких рідин умови прилипання до стінки не виконуються, а мають місце умови прослизання [6]. На шорстких поверхнях відбувається дифузійне відбиття частинок в напрямку руху рідини, що може розглядатися як тангенціальний переніс частинок вздовж стінки [1-3]. Якісною мірою цього процесу є коефіцієнт переносу імпульсу (tangential momentum accommodation factor σ); для мікро/нанорідин $\sigma = 1$, а для звичайних рідин $\sigma = 2$ [1-3].

Особливості руху неньютонівських рідин по мікро/нанотрубках та каналах з умовами прослизання на стінках раніше досліджувались, зокрема для в'язкопружних рідин Максвелла [7], Джефрі [8], Бонджерно [9], та ряду інших. В даній роботі досліджується вплив дифузійного віддзеркалення та просковзання на стінках мікроканалів на гідродинамічні параметри течії бінгамівської рідини.

Постановка проблеми

Розглядається Пуазейлівська течія нестисливої бінгамівської рідини по трубці з круговим поперечним перерізом і радіусом R . Рух описується стаціонарними рівняннями Нав'є-Стокса

$$\operatorname{div}(\vec{v}) = 0, \quad \Delta \vec{v} = \frac{1}{\mu} \nabla p, \quad (1)$$

де $\vec{v} = (v_x, 0, 0)$ - швидкість руху рідини, p - гідростатичний тиск, μ - в'язкість рідини.

Реологічні властивості бінгамівської рідини залежать від граничного напруження зсуву τ_0 [10]

$$\dot{\gamma} = \begin{cases} \frac{\tau - \tau_0}{\mu}, & \tau > \tau_0 \\ 0, & \tau \leq \tau_0 \end{cases} \quad (2)$$

де $\dot{\gamma} = \partial v_x / \partial r$ - швидкість зсуву.

Граничні умови для (1)-(2) є умовами симетрії на осі трубки та прослизання на шорстких стінках трубки [1-4]

$$\left(v - C_1 Kn \frac{\partial v_x}{\partial r} - C_2 Kn^2 \frac{\partial^2 v_x}{\partial r^2} \right) \Big|_{\partial\Omega} = 0 \quad (3)$$

де $\partial\Omega$ - поверхня трубки, $C_{1,2} = \text{const}$, $Kn = \lambda / R$ - число Кнудсена, λ - довжина вільного пробігу частинок рідини.

Для класичних макрорідин $Kn < 0.01$ і (3) відповідає умовам не прослизання на стінках. Для мікрорідин $0.01 < Kn \leq 0.1$ і останнім доданком в (3) можна знехтувати у порівнянні з першими, що відповідає умовам прослизання першого порядку ($C_2 = 0$) [1-4]. Для нанорідин $Kn > 0.1$ і $C_{1,2} \neq 0$.

Умови (3) отримані шляхом розкладання швидкості рідини біля стінки в ряд Тейлора зі зберіганням двох перших членів ряду. Згідно з експериментальними даними, для більшості нанорідин $C_1 \in [1; 1.1466]$, $C_2 \in [0.5; 1.309]$ [2].

Як відомо, для випадку течії звичайної бінгамівської рідини з умовами непротікання на стінках трубки замість (3), розв'язок (1)-(2) має вигляд [10]

$$v(r) = \begin{cases} \frac{\Delta p R^2}{4\mu L} \left(1 - \frac{r^2}{R^2}\right) - \left(1 - \frac{r}{R}\right) \frac{R\tau_0}{\mu}, & r > R^* \\ \frac{\Delta p R^2}{4\mu L} \left(1 - \frac{2L\tau_0}{R\Delta p}\right)^2, & 0 < r < R^* \end{cases} \quad (4)$$

де Δp - перепад тисків на кінцях трубки, L - довжина трубки, $R^* = 2L\tau_0 / \Delta p$ - критичний радіус, який розділяє рідку та квазітверду зони течії (Рис.1).

Загальна об'ємна витрата течії (4) визначається формулою Букінгема [10]

$$Q = Q^p \left(1 - \frac{4}{3} \left(\frac{R^*}{R}\right)^2 + \frac{1}{3} \left(\frac{R^*}{R}\right)^4\right) = Q^p \left(1 - \frac{4}{3} \left(\frac{\tau_0}{\tau_w^p}\right)^2 + \frac{1}{3} \left(\frac{\tau_0}{\tau_w^p}\right)^4\right), \quad (5)$$

де $Q^p = \Delta p \pi R^4 / (8\mu L)$, $\tau_w^p = \Delta p R / 2L$ - об'ємна витрата і напруження зсуву на стінках в течії Пуазейля ньютонівської рідини.

Напруження зсуву мають лінійний розподіл вздовж радіусу трубки

$$\tau(r) = \frac{(\tau_w + \tau_0)r}{R} - \tau_0, \quad r > R^*, \quad (6)$$

де $\tau_w = \tau_w^p - \tau_0$ - напруження зсуву на стінці трубки

Треба отримати розв'язок задачі (1)-(2) у вигляді (4)-(6) за наявності умов непротікання (3).

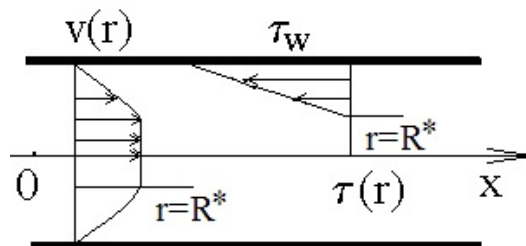


Рис.1. Розподіл швидкостей та напружень зсуву в течії бінгамівської рідини по циліндричній трубці за умовою непротікання на стінці.

Метод розв'язання

Ґрунтуючись на (2), розглядаємо $\dot{\gamma} = \dot{\gamma}(\tau)$ як складну функцію, де $\tau = \tau(r)$. Тоді

$$\dot{\gamma}(\tau(r)) \equiv \frac{dv}{dr} = \frac{dv}{d\tau} \frac{d\tau}{dr} = \frac{\tau_w}{R} \frac{dv}{d\tau} = \begin{cases} \frac{\tau - \tau_0}{\mu}, & \tau > \tau_0 \\ 0, & \tau < \tau_0 \end{cases} \quad (7)$$

Згідно з (7), при $\tau > \tau_0$

$$v(\tau) = \frac{R}{\tau_w} \int_{\tau}^{\tau_w} \frac{\tau - \tau_0}{\mu} d\tau = \frac{R\tau_w}{2\mu} \left(1 - \frac{\tau}{\tau_w}\right) \left(1 + \frac{\tau}{\tau_w} - 2\frac{\tau_0}{\tau_w}\right), \quad (8)$$

а при $\tau < \tau_0$

$$v = \text{const} = v(\tau_0) = \frac{R\tau_w}{2\mu} \left(1 - \frac{\tau_0}{\tau_w}\right)^2. \quad (9)$$

Переходячи в (8),(9) від τ до r , отримуємо замість (4)

$$v_{\text{slip}}(r) = \begin{cases} \frac{\Delta p R^2}{4\mu L} \left(1 - \frac{r^2 + 2\text{Kn}(C_1 R + \text{Kn} C_2)}{R^2} \right) - \frac{R\tau_0}{\mu} \left(1 - \frac{r}{R} + \frac{\text{Kn} C_1}{R} \right), & r > R^*, \\ \frac{\Delta p R^2}{4\mu L} \left(1 - \left(\frac{\tau_0}{\tau_w^p} \right)^2 - \frac{2\text{Kn}(C_1 R + \text{Kn} C_2)}{R^2} \right) - \frac{R\tau_0}{\mu} \left(1 - \frac{\tau_0}{\tau_w^p} + \frac{\text{Kn} C_1}{R} \right), & 0 < r < R^*, \end{cases} \quad (10)$$

де $r^* = 2L\tau_0 / \Delta p$.

Тоді з (10) для об'ємної витрати замість (5) отримуємо

$$Q_{\text{slip}} = Q^p \left(1 - 4\text{Kn} \left(\frac{C_1}{R} + \text{Kn} \frac{C_2}{R^2} \right) + 4 \frac{\text{Kn} C_1}{R} \frac{\tau_0}{\tau_w^p} + \left(2\text{Kn} \left(\frac{C_1}{R} + \text{Kn} \frac{C_2}{R^2} \right) - \frac{4}{3} \right) \left(\frac{\tau_0}{\tau_w^p} \right)^2 - 4 \frac{\text{Kn} C_1}{R} \left(\frac{\tau_0}{\tau_w^p} \right)^3 + \frac{1}{3} \left(\frac{\tau_0}{\tau_w^p} \right)^4 \right), \quad (11)$$

а для напружень зсуву – ту ж саму залежність (6), тому що (10) і (4) відрізняються лише константами, які зникають після диференціювання.

В класичних макрорідинах $\text{Kn}=0$ і (10),(11) переходять в (4),(5). При тих самих значеннях Δp різниця ΔQ між об'ємними витратами (11) і (5) складає

$$\Delta Q = Q^p 2\text{Kn} \left(\frac{2C_1}{R} \frac{\tau_0}{\tau_w^p} \left(1 - \left(\frac{\tau_0}{\tau_w^p} \right)^2 \right) - \left(\frac{C_1}{R} + \text{Kn} \frac{C_2}{R^2} \right) \left(2 - \left(\frac{\tau_0}{\tau_w^p} \right)^2 \right) \right). \quad (12)$$

Як було показано в [4], при ламінарних течіях ньютонівських мікро/нанорідин за рахунок тангенціального переносу частинок на шорстких стінках $\Delta Q > 0$, тобто мікро/нанотрубки та канали мають менші гідравлічні опори, що пояснює результати чисельних експериментів з течіями різних рідин по мікро- та нанотрубках [1-3,5]. З (12) виходить, що для бінгамівської рідин значення ΔQ залежать від співвідношень між параметрами стінки $C_{1,2}$, Kn , реології рідини τ_0 , відносного розміру трубки R/L та режиму руху рідини.

Результати чисельних розрахунків та їх обговорення

Профілі швидкості у безрозмірному вигляді можна переписати як

$$v^o(r) = \begin{cases} 1 - (r^o)^2 + 2\text{Kn}(C_1^o + \text{Kn}C_2^o) - \zeta(1 - r^o + \text{Kn}C_1^o), & r^o > r^*, \\ 1 - (\tau_{\text{rel}})^2 - 2\text{Kn}(C_1^o + \text{Kn}C_2^o) - \zeta(1 - \tau_{\text{rel}} + \text{Kn}C_1^o), & 0 < r^o < r^*, \end{cases} \quad (13)$$

де $v^o = \frac{v_{\text{slip}}}{v_{\text{max}}}$, $v_{\text{max}} = \frac{\Delta p R^2}{4\mu L}$, $\zeta = \frac{4L\tau_0}{\Delta p R}$, $\tau_{\text{rel}} = \frac{\tau_0}{\tau_w^p}$, $r^o = r/R$, $r^* = R^*/R$, а вираз (12) для об'ємної витрати – у вигляді

$$Q^o = 1 - 4\text{Kn}(C_1^o + \text{Kn}C_2^o) + 4\text{Kn}C_1^o\tau_{\text{rel}} + \left(2\text{Kn}(C_1^o + \text{Kn}C_2^o) - \frac{4}{3} \right) (\tau_{\text{rel}})^2 - 4C_1^o (\tau_{\text{rel}})^3 + \frac{1}{3} (\tau_{\text{rel}})^4, \quad (14)$$

де $Q^o = \frac{Q_{\text{slip}}}{Q^p}$.

Чисельні розрахунки за (13), (14) проводилися для характерних параметрів ряду мікро- та нанорідин (Табл.1) [1-3,11]. Залежності $v^o(r)$ для різних наборів параметрів та залежності Q^o від параметрів C_1^o , C_2^o , τ_{rel} , Kn наведені на Рис.2-3. Для порівняння зображені також відповідні криві для класичних ($C_1^o=0$, $C_2^o=0$) бінгамівських рідин. Для випадків мікрорідин $C_2^o=0$ (умови прослизання першого порядку [1-3]).

Таким чином, у порівнянні з класичними рідинами, які задовольняють умовам непрослизання на твердих стінках, мікро- та нанорідини мають як вищі швидкості, так і менші розміри «твердого ядра» потоку бінгамівської рідини (Рис.2а). Поступове збільшення тангенціального переносу моменту на стінках (значення C_1) приводить до

поступового збільшення швидкості і зменшення розміру «ядра» (Рис.3а). Кривизна шару Кнудсена на стінках (величина C_2) грає незначну роль (Рис.3а) у порівнянні зі впливом C_1 (Рис.2б), а також різних комбінацій C_1 і C_2 (Рис.2а).

Таблиця 1. Параметри деяких мікро- та нанорідин, використаних для чисельних розрахунків.

| № | назва | K_n | C_1 (м) | C_2 (м ²) | τ_0 (мПа) | R(нм) | L(мкм) |
|---|---|-------|-----------|-------------------------|----------------|-------|--------|
| 1 | Наночастинки ZnO+H ₂ O | 0.6 | 1.15 | 0.647 | 2.6 | 50 | 0.5 |
| 2 | Наночастинки Al ₂ O ₃ +H ₂ O | 0.5 | 1.1 | 0.934 | 2.2 | 50 | 0.5 |
| 3 | Наночастинки CuO+H ₂ O | 0.4 | 1.3 | 0.5 | 2.8 | 120 | 1 |
| 4 | Наночастинки SiO ₂ +H ₂ O | 0.2 | 1 | 1.125 | 2.5 | 120 | 1 |
| 5 | Мікрочастинки резини +H ₂ O з гліцерином | 0.06 | 1.5 | 0 | 3.2 | 500 | 500 |
| 6 | Суспензія «тіней» еритроцитів | 0.04 | 1.3 | 0 | 3.5 | 500 | 500 |
| 7 | кров | 0.03 | 1.4 | 0 | 5.1 | 500 | 500 |

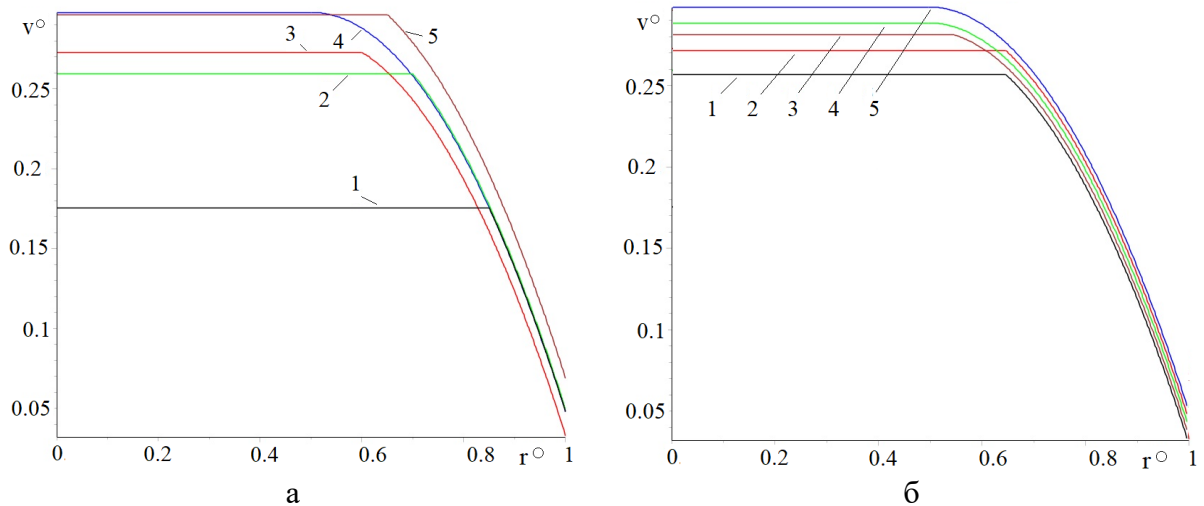


Рис.2. Залежності $v^o(r^o)$: для різних наборів параметрів з Табл.1 (а) (крива 1 відповідає класичній, а криві 2-5 - першим 4 рядкам таблиці); для значень $C_1=0;0.1;0.2;0.3;0.4$ (б).

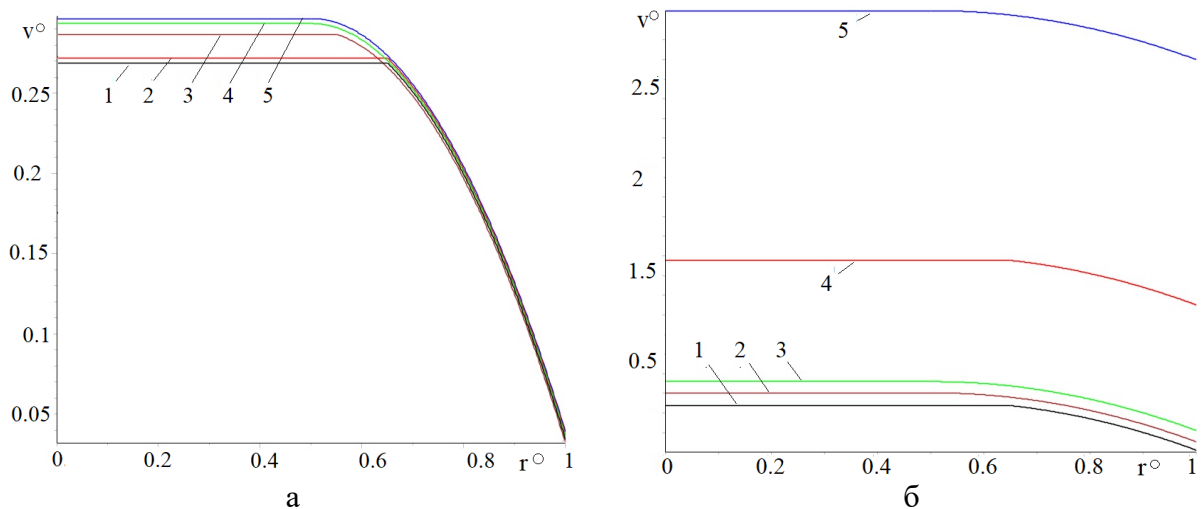


Рис.3. Залежності $v^o(r^o)$: для різних наборів параметрів з Табл.1 (а) (крива 1 відповідає класичній, а криві 2-5 - першим 4 рядкам таблиці) для значень $C_2=0;-0.5;0.1;0.2;0.3$ і (б) для $K_n=0;0.01;0.05;0.1;0.5;1$.

Найзначнішу роль грає число K_n , яке дозволяє при переході від нано- і мікротечій – до класичних рідин з $K_n=1$ (Рис.3б). Слід зауважити, що на Рис.2-3 наведені відносні (безрозмірні) параметри, тому ті ж самі значення τ° відповідають радіусам 1мм для течій з $K_n=1$ і радіусам 50нм – для течій з $K_n=0.01$. За рахунок підбору параметрів $K_n, C_1^\circ, C_2^\circ, \tau_{rel}, K_n$ можна отримати течію з необхідними швидкостями в ядрі, шарі Кнудсена та їх відносних розмірів, що дозволяє відокремлювати фізико-хімічні та біологічні процеси, які протікають в зонах течії з високими та низькими значеннями швидкостей зсуву.

Висновки

В роботі вперше отримані розв'язки системи рівнянь Нав'є-Стокса для течій мікро/нанорідин з умовами прослизання пуршого та другого порядку відповідно та досліджені відмінності профілей швидкості, об'ємної витрати та напружень тертя на стінках трубки в залежності від усіх параметрів задачі. Показано, що за рахунок вибору матеріалів мікро/наночастинок та стінки, параметрів шорсткості стінки, концентрації частинок та режиму руху можна отримати будь-які потрібні для конкретних технологій величин швидкості та розміру ядра бінгамівської рідини.

Список використаної літератури:

1. Gad-el-Hak, M. The MEMS Handbook / M. Gad-el-Hak. – New York: CRC Press. – 2006.
2. Karniadakis, G.E. Heat transfer and fluid flow in minichannels and microchannels / G.E. Karniadakis, S. Garimella, D. Li, S. Kolin, M.N. King. – Elsevier. – 2006.
3. MEMS Microfluidics for Lab-on-a-Chip Applications, Microelectromechanical Systems and Devices / editor Islam N. – InTech, 2012.
4. Cherevko, V. Complex flows of immiscible microfluids and nanofluids with velocity slip boundary conditions / V. Cherevko, N. Kizilova // Nanophysics, Nanomaterials, Interface Studies, and Applications. – 2017. – P. 207-230.
5. Noya, A. Nanofluidics in carbon nanotubes / A. Noya, H.G. Parka, F. Fornasiero, J.K. Holt, S.P. Grigoropoulos, O. Bakajin // Nanotoday. – 2007. – V. 2. – P. 22-29.
6. Michael, J.M. Slip over rough and coated surface / J.M. Michael, H.D. Stephen // J. Fluid Mech. – 1994. – Vol. 273. – P. 801-817.
7. Hayat, T. Three-dimensional boundary layer flow of Maxwell nanofluid: mathematical model / T. Hayat, T. Muhammad, S.A. Shehzad, et al. // Appl. Math. Mech. Ed. – 2015. – Vol. 36. – P. 747-762.
8. Santhosh, N. Flow of a jeffrey fluid through a porous medium in narrow tubes / N. Santhosh, G. Radhakrishnamacharya, A.J. Chamkha // J. Porous Media. – 2015. – Vol. 18 (1). – P. 71-78.
9. Zhu, J. Effects of second order velocity slip and nanoparticles migration on flow of Buongiorno nanofluid / J. Zhu, D. Yang, L. Zheng, X. Zhang // Appl. Math. Letters. – 2016. – Vol. 52. – P. 183-191.
10. Лойцянский, Л.Г. Механика жидкости и газа / Л.Г. Лойцянский. – М.: Наука. – 1978. – 736 с.
11. Левтов, В.А. Реология крови / В.А. Левтов, С.А. Регирер, Н.Х. Шадрина. – М.: Медицина. – 1982. – 270 с.

References:

1. Gad-el-Hak, M. (2006). The MEMS Handbook. New York: CRC Press.
2. Karniadakis, G.E., Garimella, S., Li, D., Kolin, S., & King, M.N. (2006). Heat transfer and fluid flow in minichannels and microchannels. Elsevier.
3. Islam, N. (Eds.). (2012). MEMS Microfluidics for Lab-on-a-Chip Applications, Microelectromechanical Systems and Devices. InTech.
4. Cherevko, V., & Kizilova, N. (2017). Complex flows of immiscible microfluids and nanofluids with velocity slip boundary conditions. *Nanophysics, Nanomaterials, Interface Studies, and Applications*. (pp. 207-230).
5. Noya, A., Parka, H.G., Fornasiero, F., Holt, J.K., Grigoropoulos, S.P., & Bakajin, O. (2007). Nanofluidics in carbon nanotubes. *Nanotoday*, 2, 22-29.
6. Michael, J.M., & Stephen H.D. (1994). Slip over rough and coated surface. *J. Fluid Mech*, 273, 801-817.
7. Hayat, T., Muhammad, T., Shehzad, S.A. et al. (2015) Three-dimensional boundary layer flow of Maxwell nanofluid: mathematical model. *Appl. Math. Mech. Ed.*, 36, 747-762.

8. Santhosh, N., Radhakrishnamacharya, G., & Chamkha, A.J. (2015). Flow of a jeffrey fluid through a porous medium in narrow tubes. *J. Porous Media*, 18 (1), 71-78.
9. Zhu, J., Yang, D., Zheng, L., & Zhang, X. (2016). Effects of second order velocity slip and nanoparticles migration on flow of Buongiorno nanofluid. *Appl. Math. Letters*, 52, 183-191.
10. Lojtsiansky, L.G. (1978). *Mekhanika zhidkosti i gaza* [Mechanics of fluids and gases]. M.: Nauka [in Russian].
11. Levitov, V.A., Regirer, S.A., Shadrina, N.X. (1982). *Reologiya krovi* [Blood rheology]. M.: Medicine [in Russian].

BATYUK Liliya,

Candidate of Biological Sciences, Associate Professor of Department of Medical and Biological Physics and Medical Informatics, Kharkiv National Medical University

KIZILOVA Natalya,

Doctor of Physical and Mathematical Sciences, Professor of Department of Applied Mathematics, V.N. Karazin Kharkiv National University

INVESTIGATION OF FLOWS OF BINGHAM FLUIDS THROUGH THE MICROCHANNELS

Summary. Introduction. *A brief concise literature review on the experimental data of flows of micro- and nanofluids through the micro tubes and ducts and the corresponding theoretical models is given. It is shown, the theoretical estimations based on the Poiseuille formulas for the fluid flows through the ducts of different shaped can be explained by the theory of complex diffusion scatter on the micro- or nanoparticles at the roughness of the tubes' walls which has the same order of magnitude as the particles have.*

Purpose. *The aim of the article is to find out the differences in the hydromechanical parameters of the flows of classical Bingham fluids with no-slip boundary conditions at the walls and the micro/nanofluids with the first/second order velocity slip boundary conditions at the same flow regimes.*

Results. *The analytical solution for the Bingham fluid flow through a circular tube with velocity slip boundary conditions has been obtained with the same technique that had been used for the classical Bingham fluid flows. The expressions for the velocity profiles, volumetric flow rates, shear stress distributions, wall shear stress and radius of the "solid core" in the Bingham fluid flow has been obtained and compared to the ones for the classical Bingham fluid flows with velocity no-slip boundary conditions. The main parameters influenced the flow differences are the Knudsen number, first and second order velocity slip coefficients,*

Conclusion. *The influence of each flow parameter, materials parameters of the particles and basic fluid on the velocity profile of the moving fluid and the "solid core" of the Bingham fluid, its volumetric flow rate through the tube, shear stress distribution and wall shear stress is studied. It is shown, the different variations of the parameters measured in the experimental studies presented in the literature published produce multilateral influence of the velocity and size of the "solid core" of different Bingham fluids. The monotonous increase in the first order slip parameter produced the monotonous growth of the fluid velocity on the slip wall and small variations in the core velocity and size. The second order slip coefficient evoked the smallest influence of the flow parameters. The most significant influence was found for the Knudsen number that corresponds to the transfer from the flows of classical fluids to micro- and then to nanofluids. The presented data is related to the relative (i.e. non-dimensional) data, but the significant acceleration of the nanotube/microtube flows in comparison to the classical flows in the $>1\text{mm}$ channel width is proven by the obtained formulae and confirmed by the experiments in the nanofluid flows through the bunches of nanotubes (nanofilters) that was estimated as 200 higher than those estimated by the Poiseuille law for circular tubes.*

Keywords: *nanofluidics, Bingham fluid, steady flows, rough walls.*

*Одержано редакцією 05.07.2019 р.
Прийнято до публікації 09.10.2019 р.*

УДК 536.23

DOI 10.31651/2076-5886-2019-2-11-20

PACS 02.60.-x, 02.60.Pn, 02.70.Wz

ГОЛОВНЯ Борис Петрович

доктор технических наук, доцент кафедры
прикладной математики и информатики
Черкаского национального университета
имени Богдана Хмельницкого
e-mail: bpgolovnya@gmail.com
ORCID 0000-0002-9242-3937

РАСЧЕТ ЛАМИНАРНО-ТУРБУЛЕНТНОГО ПЕРЕХОДА В ПОГРАНИЧНОМ СЛОЕ НА ПЛОСКОЙ ПЛАСТИНЕ, ОБДУВАЕМОЙ ТУРБУЛИЗИРОВАННЫМ ПОТОКОМ

Проблема ламинарно-турбулентного перехода в пограничном слое на плоской пластине, обдуваемой турбулизированным потоком, или байпасного перехода, играет очень важную роль в вопросах проектирования турбинных установок. Известны многие попытки моделирования байпасного перехода, но все они ориентированы на расчет коэффициента трения и плохо воспроизводят сам процесс возникновения турбулентности. В работе делается попытка частично исправить эти недостатки.

Ключевые слова: ламинарно-турбулентный переход в пограничном слое, байпасный переход, переход при повышенной турбулентности внешнего потока.

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

| | | | |
|---------------------------------------|--|---|--|
| $C_f = 2\tau_w / \rho U_e^2$ | коэффициент трения; | u, v | пульсации скорости, м/сек; |
| $F = U/U_e$ | безразмерная скорость; | $u^* = \sqrt{\tau_w / \rho}$ | динамическая скорость, м/сек; |
| F_v, f_0 | функции модели; | $u_+ = \bar{U}/u^*$ | безразмерная скорость; |
| $H = \delta^*/\delta^{**}$ | формпараметр; | Греческие символы | |
| $k = 0.5\overline{u_i u_i}$ | энергия турбулентности, м ² /с ² ; | δ | толщина пограничного слоя, м; |
| k_0 | энергия больших вихрей, м ² /с ² ; | $\delta^* = \int_0^\infty (1-F) dy$ | толщина вытеснения, м; |
| $L_\varepsilon = k^{3/2}/\varepsilon$ | диссипативный масштаб, м; | $\delta^{**} = \int_0^\infty F(1-F) dy$ | толщина потери импульса, м; |
| $Re_x = U_e x / \nu$ | число Рейнольдса; | $\varepsilon = \overline{\nu(\partial u_i / \partial x_i)^2}$ | скорость диссипации k , м ² /с ³ ; |
| $Re_L = U_e L_\varepsilon / \nu$ | число Рейнольдса; | ν_t | турбулентная вязкость, м ² /с; |
| $Re_y = \sqrt{k} y / \nu$ | число Рейнольдса; | Индексы | |
| $Re^{**} = U_e \delta^{**} / \nu$ | число Рейнольдса; | e | в свободном течении; |
| $Tu_e = \sqrt{2/3} k_e / U_e$ | уровень внешней турбулентности; | w | на стенке; |

Постановка задачи

Инженерная постановка задачи и краткий обзор существующих расчетов

Проблема ламинарно-турбулентного перехода в пограничном слое на плоской пластине, обдуваемой турбулизированным потоком, или байпасного перехода, играет очень важную роль в вопросах проектирования турбинных установок. Следует

отметить, что эту проблему нельзя признать решенной. В настоящее время известно, что на процесс перехода оказывают влияние, как интенсивность пульсаций внешнего течения, так и масштабы его вихрей. При нулевом или очень малом уровне пульсаций переход возникает в результате развития волн Толлмина-Шлихтинга и их разрушения в результате вторичной неустойчивости. При возрастании уровня турбулентности внешнего потока уже до уровня 0.5-1% переход к турбулентности возникает значительно раньше и имеет место байпасный переход. Другими словами, турбулентность внешнего потока оказывает очень заметное дестабилизирующее воздействие на пограничный слой.

Естественный переход в плоском пограничном слое с нулевым градиентом давления изучался многими авторами. В подавляющем большинстве работы сводятся к расчету развития волн Толлмина-Шлихтинга и их последующему разрушению через вторичную и третичную неустойчивости. Численные результаты в основном находятся в хорошем соответствии с экспериментами.

Но, как известно, механизм байпасного перехода существенно отличается от механизма естественного перехода. Волны Толлмина-Шлихтинга и форма возмущений перестают играть в этом случае столь важную роль, как при естественном переходе. Известны многочисленные попытки расчета этого процесса на базе RANS моделей. К сожалению, практически все модели адекватно воспроизводят только финальную стадию перехода. Начальные этапы развития процесса этими моделями игнорируются. В качестве примера подобных моделей можно привести работы [1-5]. К наиболее успешным и популярным подобным моделям можно отнести работы [6-7].

Известны попытки расчета байпасного перехода методом типа LES.

К наиболее успешным следует отнести работу Янга и Воука [8]. Авторы [8] проводили расчеты LES методом. Авторам удалось добиться хорошего соответствия с экспериментами по всем важнейшим параметрам.

Более сложный расчет байпасного перехода проведен в работах [9] и [10] DNS методом. Точность полученных результатов очень хорошая. Но, в то же время, следует отметить, что в [10] расчет проводился на параллельном компьютере, содержащем 128 процессоров, и потребовал порядка 500 часов времени на процессор. Используемая сетка содержала $2048 \times 180 \times 192$ узлов. Очевидно, что при всех его достоинствах, подобный подход не может быть рекомендован к внедрению в инженерную практику.

Физическая постановка задачи

В однородном турбулентном потоке параллельно основному направлению течения установлена плоская пластина. Известны распределения всех пульсационных и осредненных параметров течения во внешнем течении и перед пластиной. На поверхности пластины развивается пограничный слой. На некотором расстоянии от начала пластины, благодаря воздействию внешней турбулентности, характеристики пограничного слоя на пластине начинают отличаться от характеристик ламинарного пограничного слоя, т.е. начинается переход. Со временем слой на пластине становится турбулентным. Требуется рассчитать процесс развития турбулентности в пограничном слое. Из соображений экономичности расчета уравнения Навье-Стокса требуется решать в приближении пограничного слоя. Начальные условия должны задаваться в виде прямоугольных профилей всех параметров непосредственно в начальной точке пластины, т.е. в единственной физически обоснованной форме.

Цель статьи

На основе модели турбулентности [1] предложить модель для расчета байпасного перехода. Тем самым подтвердить большие возможности данной модели.

Методы решения

Модель, использованная в расчетах

В расчетах использовалась модель автора [11]

$$\frac{Dk_0}{Dt} = f_0 \frac{\partial}{\partial x_k} \left(v + \frac{v_t}{C_k} \right) \frac{\partial k_0}{\partial x_k} + f_0 P - \varepsilon_0, \quad (1)$$

$$\frac{D\varepsilon_0}{Dt} = f_0 \frac{\partial}{\partial x_k} \left(v + \frac{v_t}{C_\varepsilon} \right) \frac{\partial \varepsilon_0}{\partial x_k} + \frac{\varepsilon_0}{k_0} (C_1 f_0 P - C_2 \varepsilon_0). \quad (2)$$

$$f_0 = f_{0-wall} f_{0-base}, \quad (3)$$

$$f_{0-wall} = 1 - \exp(-\text{Re}_{y_0}/5.5), \quad f_{0-base} = 1 - \exp(-2.4 y/L_{\varepsilon_0}),$$

$C_v=0.09, C_\varepsilon=1.3, C_k=1, C_2=1.45, C_1=0.9C_2.$

$y=0 - k_0=\varepsilon_0=0, y \rightarrow \infty - k_0=k_e, \varepsilon_0=\varepsilon_e.$

Модель потребовала небольшой модификации. Как известно, генерация корреляции \overline{uv} описывается выражением $-\overline{v^2} \frac{\partial U}{\partial y}$. Поэтому корреляция \overline{uv} должна

быть пропорциональна $\overline{v^2} \frac{\partial U}{\partial y}$, откуда следует, что v_t пропорциональна $\overline{v^2}$. При

расчете турбулентной вязкости по выражению $v_t = C_v F_v k^2/\varepsilon$ соотношение между $\overline{v^2}$ и k учитывается функцией F_v . Из экспериментов известно, что в случае переходного слоя это соотношение резко отличается от соотношения в случае развитой турбулентности. По этой причине в функцию F_v вводился дополнительный множитель φ_v , учитывающий эту разницу в ламинарном режиме и равный единице в турбулентном режиме течения

$$\varphi_v = 1 - C_0 \exp(-2.5\sqrt{v_t}) / \exp\left(0.25\left(\frac{C_f}{C_{f_lam}} - 1\right)^2\right).$$

Здесь $C_{f_lam} = 0.664/\sqrt{\text{Re}}$ – коэффициент трения при ламинарном режиме течения, $C_0=1-1.0e-5$ – используется только для того, чтобы не обнулять F_v в самом начале расчета (там имеем $v_t=0, C_f=C_{f_lam}$ откуда следует, что $\varphi_v=0$, если $C_0=1$) так как тогда турбулентная вязкость обнулится навсегда. Как видно из выражения, при ламинарном режиме трения знаменатель равен единице. При возрастании трения, т.е. начале перехода, знаменатель также резко возрастает и, в итоге, практически превращает второе слагаемое поправки в нуль, а саму поправку в единицу

Краевые условия

Традиционной формой постановки краевых условий при расчете байпасного перехода служит аппроксимация экспериментальных данных вида $Tu_e = C(X + X_b)^n$. Здесь X - расстояние от начала пластины, C, X_b и n - параметры расчета. Типичное значение $n = -5/7$. Краевые условия для диссипации как правило в опубликованных работах не указываются. В данной работе аналогичные условия получены аналитически.

Известно, что уравнения k - ε модели в свободном течении выглядят следующим образом

$$U_e \frac{dk_e}{dx} = -\varepsilon_e, \quad (4)$$

$$U_e \frac{d\varepsilon_e}{dx} = -C_2 \frac{\varepsilon_e^2}{k_e}, \quad (5)$$

$$k_e(x_{start}) = k_{start}, \quad \varepsilon_e(x_{start}) = \varepsilon_{start}.$$

Эта система имеет точное решение

$$k_e = k_{start} \left(1 + \frac{x - x_{start}}{x_a} \right)^{\frac{1}{1-C_2}}, \quad \varepsilon_e = \varepsilon_{start} \left(1 + \frac{x - x_{start}}{x_a} \right)^{\frac{C_2}{1-C_2}} \quad (6)$$

где $x_a = \frac{k_{start} U_e}{\varepsilon_{start} (C_2 - 1)}$, $\tilde{x} = x_a + x - x_{start}$.

Выражения (6) переписываем в безразмерном виде

$$k_e = k_{start} \left(1 + \frac{Re - Re_{start}}{Re_a} \right)^{\frac{1}{1-C_2}}, \quad (7)$$

$$\varepsilon_e = \varepsilon_{start} \left(1 + \frac{Re - Re_{start}}{Re_a} \right)^{\frac{C_2}{1-C_2}} \quad (8)$$

Значение C_2 подбиралось из требований соответствия (7) с экспериментальными данными.

На рис.1 приведено сравнение краевых условий (7) с экспериментальными данными [12] и LES расчетами [8].

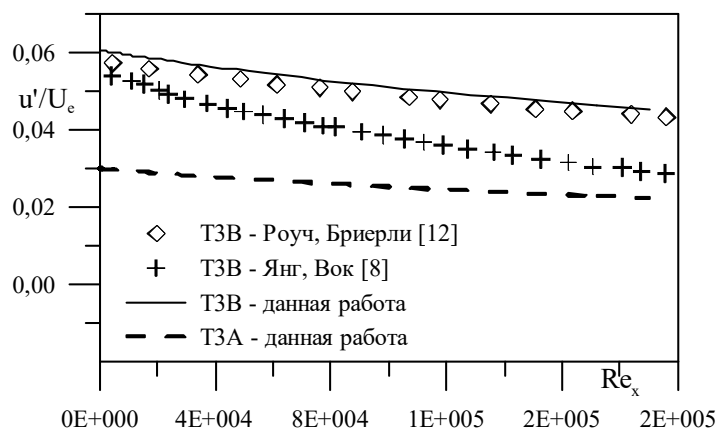


Рис. 1. Расчетные значения турбулентности внешнего потока

Начальные условия

Во всех расчетах начальные условия задавались в точке, соответствующей $Re=1$. В качестве начальных условий использовались прямоугольные профили всех параметров, т.е. единственно возможные физически осмысленные значения. Модель проверялась расчетами пограничного слоя с двумя уровнями турбулентности внешнего потока - 3% и 6%. Результаты расчетов сравнивались с экспериментальными данными, известными в литературе, как Т3А и Т3В [12], а также с результатами расчетов на основе LES подхода. LES данные взяты из работы [8]. На рис. 1 приведены использовавшиеся в расчетах значения Tu_e . Как видно из рис. 1, в данной работе уровень турбулентности внешнего потока, а значит и условия эксперимента, воспроизведен точнее, чем в [8].

Результаты расчетов

На рис. 2 и 3 приведены результаты расчетов коэффициента трения и

формпараметра H для двух случаев турбулентности набегающего потока. Для сравнения здесь же приведены экспериментальные данные Роуча и Бриерли [12]. Высокая точность расчетов, в особенности параметра H , очевидна.

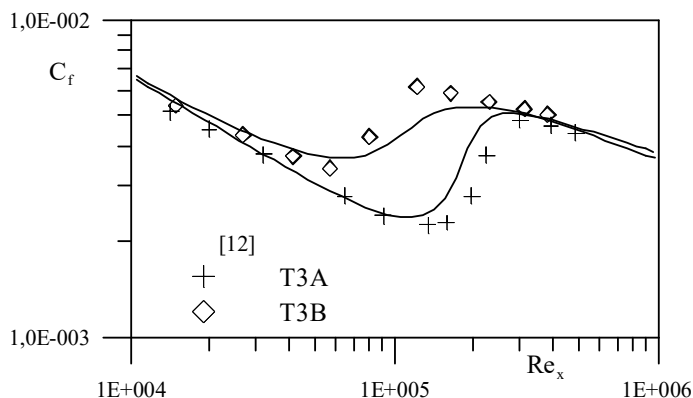


Рис. 2. Байпасный переход. Расчет коэффициента трения для двух случаев турбулентности внешнего течения.

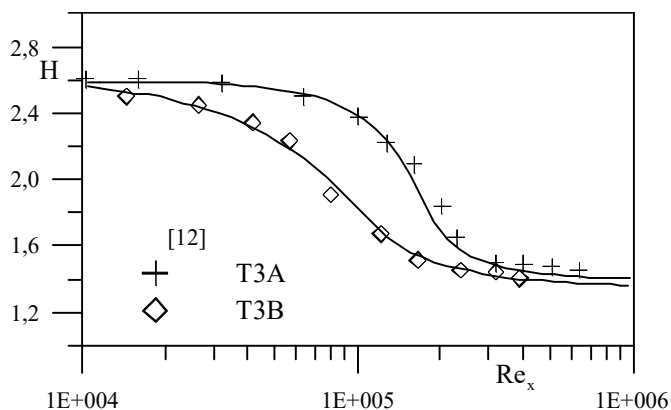


Рис. 3. Байпасный переход. Расчет формпараметра H для двух случаев турбулентности внешнего течения.

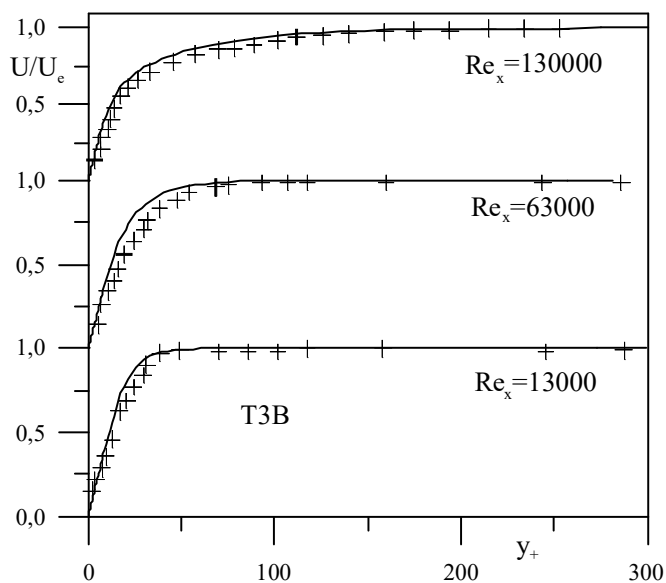


Рис. 4. Байпасный переход. Расчет средней скорости при трех числах Re .

На рис. 4 сплошной линией приведены результаты расчетов профиля средней

скорости – данные ТЗВ. Для сравнения на графике символами приведены экспериментальные данные Роуча и Бриерли [12]. Соответствие вполне удовлетворительное.

На рис. 5 приводится сопоставление расчетов корреляции $\overline{u_0 v_0}$ с экспериментальными данными Роуча и Бриерли [12] и LES данными Янга и Воука [8]. Нетрудно видеть, что данная работа обеспечивает лучшее соответствие с экспериментом, чем LES.

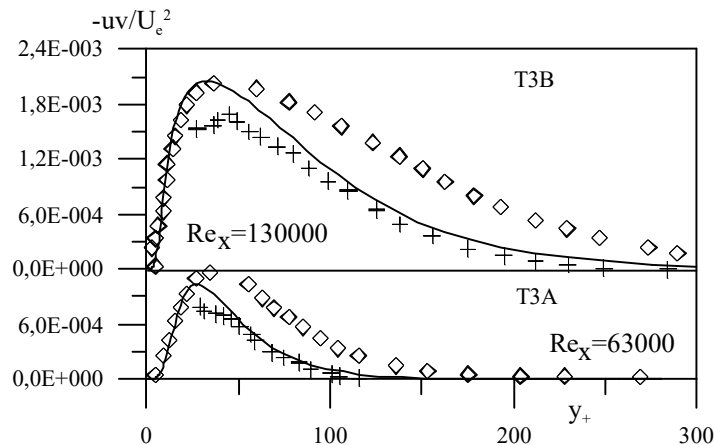


Рис. 5. Байпасный переход. Расчет турбулентного трения при двух числах R

На рис. 6 показаны результаты сравнения расчетов энергии турбулентности с экспериментальными данными Роуча и Бриерли [12]. Видно, что данный расчет дает несколько заниженные значения энергии, что вполне объясняется расчетом только первичных вихрей. В то же время общая структура распределения, положение максимума и т.д. воспроизводятся достаточно точно.

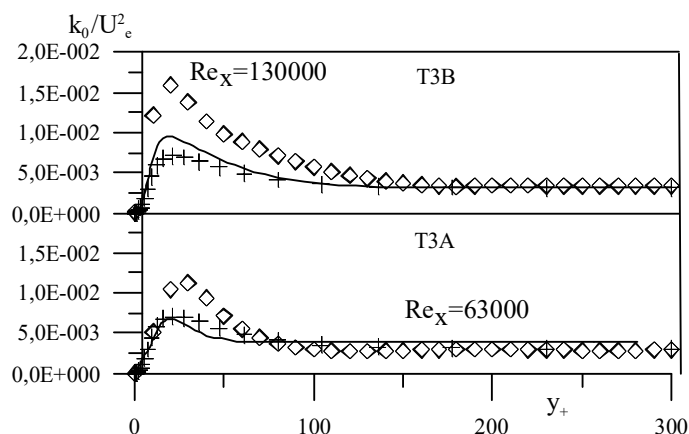


Рис. 6. Байпасный переход. Расчет энергии турбулентности при двух числах R

Таким образом, данный подход позволяет с достаточной точностью воспроизводить все основные как интегральные, так и локальные параметры байпасного переходного пограничного слоя.

Расчет надслоя при байпасном ламинарно-турбулентном переходе. Гипотетические причины его возникновения

Тщательный анализ проведенных расчетов показывает очень важную особенность развития байпасного перехода, не отмечавшуюся ранее в расчетных работах. При

больших энергиях и диссипативных масштабах внешнего течения корреляция \overline{uv} распространяется гораздо дальше пограничного слоя. Это отчетливо видно на рис. 7. Причины этого можно объяснить следующим образом. Автором была выдвинута следующая гипотеза – корреляция \overline{uv} порождается в результате деформации вихрей, возникающей из-за наличия сдвига в течении.

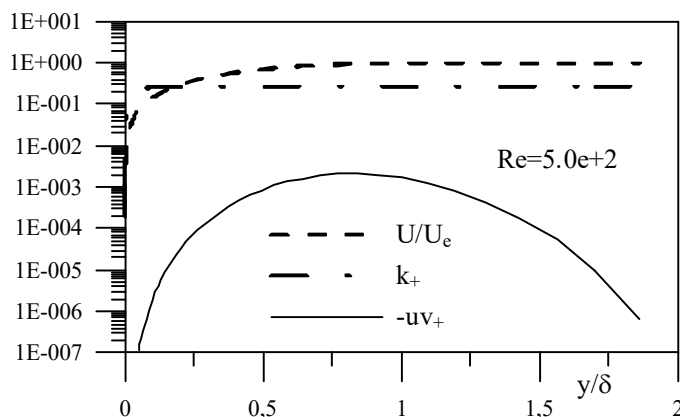


Рис. 7. Расчет существования надслоя.

Представим себе эластичную сферу, подвергаемую периодическим воздействиям в двух диаметрально противоположных точках. Если эти воздействия равны между собой по амплитуде и противоположны по фазе, то центр сферы останется неподвижным. Если воздействия не противофазны, то центр можно считать неподвижным в среднем. В любом случае в центре произведение $u \times v = 0$. Но в случае неравенства амплитуд центр начнет двигаться, а произведение скоростей в точке, совпадающей с центром неподвижной сферы, станет неравным нулю. Очевидно, что этот процесс максимизируется, если сфера касается какой-либо твердой поверхности, т.е. одна из точек остается неподвижной.

Согласно современным представлениям основная кинетическая энергия турбулентности сосредоточена в вихрях, имеющих размеры, соизмеримые с толщиной пограничного слоя. Если считать диссипативный масштаб адекватной характеристикой размера турбулентного вихря, то расчеты, проведенные по модели $k-\varepsilon$ типа, показывают, что, размеры этих вихрей во всем слое больше расстояния до стенки, т.е. эти вихри все время прижаты к стенке. В то же время эти вихри постоянно подвергаются воздействию турбулентных пульсаций, что в сочетании с воздействием стенки и порождает ненулевое значение корреляции \overline{uv} .

Но, если в развитом турбулентном течении размеры основных энергосодержащих вихрей соизмеримы с толщиной пограничного слоя, то при байпасном переходе со слоем взаимодействуют вихри очень значительных размеров, превосходящих толщину слоя в тысячи раз, особенно на начальных этапах развития слоя. Естественно, что деформация вихрей таких размеров практически не связана с толщиной пограничного слоя и, следовательно, корреляция \overline{uv} может распространяться от стенки очень далеко. Это явление наблюдалось экспериментально (см. например, Дыбан, Эпик [13]). Авторами [13] подобная турбулизирующая надстройка над пограничным слоем была названа надслоем.

Но тогда можно сказать, что на самом деле в байпасном переходе участвуют два различных по природе пограничных слоя – динамический пограничный слой, порождаемый осредненным течением, и слой турбулентности, порождаемый взаимодействием вихрей внешнего течения с поверхностью. Расчеты показывают, что

на начальной стадии перехода их толщины очень слабо связаны друг с другом. Разница в этих толщинах обеспечивает существование надслоя. В то же время, очевидно, что толщина надслоя, а значит и необходимая область расчета заранее неизвестны. Поэтому расчетная область должна быть, по крайней мере на начальной стадии расчета, достаточно большой для того, чтобы корректно учитывать всю область взаимодействия пограничного слоя с внешней турбулентностью. Но тогда отсюда следует, что пограничный слой при байпасном переходе нельзя рассчитывать в отрыве от внешнего течения, т.е. использовать внешнее течение только в качестве краевых условий для расчета пограничного слоя. Более того. Развитие байпасного перехода необходимо рассматривать как взаимодействие двух процессов – развития собственно пограничного слоя и затухания внешней турбулентности.

Результаты исследования и выводы

1. Автором были проведены расчеты байпасного перехода на плоской пластине. Расчеты проводились, начиная непосредственно от носика пластины, т.е. с единственными физически обоснованными начальными условиями. Аналогичных расчетов на основе k-ε модели обнаружить в литературе не удалось. Результаты расчетов находятся в очень хорошем соответствии с экспериментальными данными (Роуч, Бриерли [12]). Сопоставление с LES расчетами Янга и Воука [8], показывает, что данная модель воспроизводит основные закономерности байпасного перехода не хуже, а местами и лучше, чем LES метод.

2. Модель позволяет обнаружить в расчетах байпасного перехода существование турбулизированной надстройки над пограничным слоем, т.н. надслоя. Это явление экспериментально исследовалось в работе Дыбан, Эпик [13].

Список использованной литературы:

1. ERCOFTAC. A Summary of ERCOFTAC Transition SIG Progress for the 1st WORKSHOP // UMIST. – December 14-15th, 1993.
2. Savill, A.M. A synthesis of T3 test case predictions / A.M. Savill; O. Pironneau (Eds) // Numerical Simulation of Unsteady Flows and Transition to Turbulence. – 1992.
3. Liou, W.W. Bypass transitional flow calculations using a Navier-Stokes solver and two-equations models / W.W. Liou, S. Tsan-Hsing. – NASA CR-2000-209923. – 2000.
4. Yang, Z. Calculations of inlet/nozzle flows using a new k-ε model / Z. Yang, N. Georgiadis, J. Zhy, T.-H. Shih // AIAA paper. – 1995. – V. 95-2761.
5. Pèneau, F. Bypass Transition of a Boundary Layer subjected to Free-Stream Turbulence / F. Pèneau, H.C. Boisson, A. Kondjoyan, A. Uranga // An international conference on boundary and interior layers – computational and asymptotic methods ONERA. – Centre de Toulouse, France. – 2004.
6. Langtry, R. B. Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes / R. B. Langtry, F. R. Menter // AIAA Journal. – 2009. – V. 47 (12). – P. 2894-2906.
7. Durbin, P. A. An intermittency model for bypass transition / P. A. Durbin // International Journal of Heat and Fluid Flow. – 2006. – V.36. – P. 1-6.
8. Yang, Z. Large-Eddy Simulation Studies of Bypass Transition / Z. Yang, P.R. Voke // Engineering Turbulence Modelling and Experiments 2, W. Rodi & F. Martelli (Eds). – Amsterdam. – 1993.
9. Jacobs, R.G. Simulation of bypass transition / R.G. Jacobs, P.A. Durbin // J. of Fluid Mech. – 2001. – V. 428. – P. 185-212.
10. Ovchinnikov, V. Numerical simulations of boundary-layer bypass transition due to high-amplitude free-stream turbulence / V. Ovchinnikov, M. Choudhari, U. Piomelli // J. Fluid Mech. – 2008. – V. 613. – P. 135-169
11. Golovnya, B.P. Important properties of turbulent near-wall flows which are not accounted by modern RANS models / B.P. Golovnya // Int. J. Heat Mass Transfer. – 2020. – V. 146. – 118813.
12. Roach, P.E. The Influence of a Turbulent Free-Stream on Zero Pressure Gradient Transitional Boundary Layer Development / P.E. Roach, D.H. Brierley // Numerical Simulation of Unsteady Flows and Transition to Turbulence. O. Pironneau (ed). – W. – 1992.
13. Дыбан, Е.П. Тепломассобмен и гидродинамика турбулизированных потоков / Е.П. Дыбан, Э.Я. Эпик. – Киев: Наукова думка. – 1985. – 295 с.

Bibliography:

1. (1993). ERCOFTAC. A Summary of ERCOFTAC Transition SIG Progress for the 1st WORKSHOP. UMIST. December 14-15th.
2. Savill, A.M. (1992). A synthesis of T3 test case predictions. *Numerical Simulation of Unsteady Flows and Transition to Turbulence*.
3. Liou, W.W., & Tsan-Hsing, S. (2000). Bypass transitional flow calculations using a Navier-Stokes solver and two-equations models. NASA CR-2000-209923.
4. Yang, Z., Georgiadis, N., Zhy, J., Shih, T.-H. (1995). Calculations of inlet/nozzle flows using a new k- ϵ model. *AIAA paper*, 95-2761.
5. Pénéau, F., Boisson, H.C., Kondjoyan, A., Uranga, A. (2004). Bypass Transition of a Boundary Layer subjected to Free-Stream Turbulence. *An international conference on boundary and interior layers – computational and asymptotic methods ONERA*.
6. Langtry, R. B., Menter, F. R. (2009). Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes. *AIAA Journal*, 47 (12), 2894-2906.
7. Durbin, P. A. (2006). An intermittency model for bypass transition. *International Journal of Heat and Fluid Flow*, 36, 1-6.
8. Yang, Z., & Voke, P.R. (1993). Large-Eddy Simulation Studies of Bypass Transition. *Engineering Turbulence Modelling and Experiments*. W. Rodi & F. Martelli (Ed). Amsterdam.
9. Jacobs, R.G., Durbin, P.A. (2001). Simulation of bypass transition. *J. of Fluid Mech*, 428, 185-212.
10. Ovchinnikov, V, Choudhari, M., & Piomelli, U. (2008). Numerical simulations of boundary-layer bypass transition due to high-amplitude free-stream turbulence. *J. Fluid Mech*, 613, 135-169.
11. Golovnya, B.P. (2020). Important properties of turbulent near-wall flows which are not accounted by modern RANS models. *Int. J. Heat Mass Transfer*, 146, 118813.
12. Roach, P.E., & Brierley, D.H. (1992). The Influence of a Turbulent Free-Stream on Zero Pressure Gradient Transitional Boundary Layer Development. *Numerical Simulation of Unsteady Flows and Transition to Turbulence*. O. Pironneau (Ed.), W.
13. Dyban, E.P., & Epik, E.Y. (1985). Teplomasssobmen i gidrodinamika turbulizirovannykh potokov [Heat and mass transfer and hydrodynamics of turbulent flows]. Kyiv, Naukova Dumka [in Russian].

GOLOVNYA Boris,

Doctor of Technical Sciences, Associated Professor, Department of Applied Mathematics and Informatics, The Bohdan Khmelnytsky National University of Cherkasy

CALCULATION OF A LAMINAR-TURBULENT TRANSITION IN A BOUNDARY LAYER ON A FLAT PLATE BLOWED BY A TURBULIZED FLOW

Summary. Introduction. *The task of modeling the bypass transition can be formulated as follows. In a uniform turbulent flow, a flat plate is installed parallel to the main flow direction. The distributions of all pulsating and averaged flow parameters in the external flow and in front of the plate are known. A boundary layer develops on the surface of the plate. At a certain distance from the beginning of the plate, due to external turbulence, the characteristics of the boundary layer on the plate begin to differ from the characteristics of the laminar boundary layer, i.e. the transition begins. Over time, the layer on the plate becomes turbulent. It is required to calculate the process of development of turbulence in the boundary layer. For reasons of economy, the calculation of the Navier-Stokes equation needs to be solved in the approximation of the boundary layer. The initial conditions should be specified in the form of rectangular profiles of all parameters directly at the starting point of the plate, i.e. in a single physically sound form.*

The Purpose. *To solve this problem, on the basis of the turbulence model developed by the author [11], a model is proposed for modeling the bypass transition.*

Results and conclusion. *1. Simulations of the bypass transition on a flat plate were provided by the author. The calculations were carried out starting directly from the leading edge of the plate, i.e. with the only physically significant initial conditions. The simulation results are in very good agreement with experimental data (Roach, Brierly [12]). Comparison with the LES calculations by Young and Wook [8] shows that this model reproduces the basic laws of the bypass transition not worse, but in some places better, than the LES method.*

2. The model allows us to detect the existence of a turbulized superstructure over the boundary layer in the bypass transition calculations, the so-called superlayer. This phenomenon was experimentally investigated in the work of Dyban, Epik [13].

Keywords: laminar-turbulent transition in the boundary layer, bypass transition, transition with high level of the external flow turbulence.

Одержано редакцією 06.08.2019 р.
Прийнято до публікації 09.10.2019 р.

UDK 519.24, 530.19, 330.46

DOI 10.31651/2076-5886-2019-2-20-34

PACS 02.50.Cw, 02.70.Rr, 03.65.-w,
89.65.Gh

СОЛОВІЙОВ Володимир Миколайович,
доктор фізико-математичних наук,
професор, завідувач кафедри інформатики
та прикладної математики Криворізького
державного педагогічного університету
e-mail: vnsoloviev2016@gmail.com
ORCID 0000-0002-4945-202X

СЕРДЮК Олександр Анатолійович,
кандидат економічних наук, старший
викладач кафедри прикладної математики
та інформатики Черкаського національного
університету імені Богдана Хмельницького
e-mail: serdyuk@ukr.net
ORCID 0000-0002-3919-4661

МОДЕЛІ ЗАСТОСУВАННЯ ЕНТРОПІЇ ПЕРІОДУ РЕКУРЕНТНОСТІ ТА ЕНТРОПІЇ РЕКУРЕННОСТІ ДО ДОСЛІДЖЕННЯ ДИНАМІКИ СКЛАДНИХ СИСТЕМ

У статті розглянуто методику розрахунку ентропії періоду рекурентності та ентропії рекурентності на основі методу рекурентних діаграм, запропонованого Н. Марваном. Проаналізовано поведінку ентропій на основі синтетичних даних. Продемонстровано стабільність отримуваних показників з використанням процедури рухомого вікна на синтетичних даних. Проведено оцінку поведінки динаміки показників на основі економічних часових рядів: індексу Доу Джонса та цін на нафту. Зроблено висновки про можливість використання досліджуваних показників для аналізу часових рядів економічної природи та визначення критичних періодів у економічних системах.

Ключові слова: складна система, ентропія, рекурентна діаграма, кризове явище

Вступ

Складні системи – це системи з великою кількістю агентів, які взаємодіють між собою. Під час такої взаємодії на макроскопічному рівні колективної поведінки генеруються нові характеристики систем, що проявляються шляхом самовільної організації значних часових, функціональних чи просторових структур. Застосування кількісних методів у якості процесів моделювання включає у себе процедури вимірювання, де досить важливими є показники складності системи, для кількісного опису яких існує велике різноманіття підходів.

Одним з таких підходів є застосування поняття ентропії [1], на якому ґрунтується другий закон термодинаміки. Узагальнено - другий закон полягає у властивості закритих систем прагнути набути стану термодинамічної рівноваги. У відповідності з цим у замкнутій системі, яка досягла рівноважного стану, ніяка частина внутрішньої енергії системи не може бути перетворена у механічну енергію, і внутрішній хаос системи поступово зростає [1].

Як показано у багатьох попередніх працях, економічні системи, що є яскравими зразками складних систем, можуть досліджуватись за допомогою методів, призначених для використання у інших науках, зокрема – фізиці. Досить успішно для дослідження економічних систем використовується й поняття ентропії [2, 3, 4, 5, 6]. Однак, з появою нових моделей та методів дослідження складних систем, з'являються й нові підходи до оцінки ентропії.

На поточний момент серед різноманіття способів визначення ентропії складної системи з'явилося кілька нових, що опираються на аналіз рекурентних діаграм – методуку, розроблену під керівництвом Н. Марвана [7]. Розгляд двох таких способів і було взято для дослідження, проведеного у даній роботі.

Метою статті є опис та оцінка можливості застосування до дослідження економічних рядів двох понять ентропії, що опираються на аналіз рекурентних діаграм: ентропії періоду рекурентності та ентропії рекурентності.

Виклад основного матеріалу

Метод рекурентного відображення (recurrence plot) використовується для відображення поведінки динамічної системи у часі шляхом візуалізації траєкторії $\vec{x}_i \in R^n$ ($i = 1, \dots, N$) у n -вимірному фазовому просторі [8]. Фактично, він відображує подібність траєкторії у фазовому просторі до певного стану, що є фундаментальною властивістю детерміністичної динамічної системи [9, 10]. Головним кроком такої візуалізації є обчислення матриці розміру $N \times N$:

$$\mathbf{R}_{ij} = \Theta(\varepsilon_i - \|\vec{x}_i - \vec{x}_j\|), \quad i, j = 1, \dots, N, \quad (1)$$

де ε_i – окіл, у якому точки вважаються сусідами;

$\|\cdot\|$ – норма (наприклад, Евклідова норма);

$\Theta(x)$ – функція Хевісайда.

Вектори фазового простору для одновимірному часового ряду $\{t_i\}$ реконструюються за допомогою методу часової затримки Такенса: $\vec{x}_i = (t_i, t_{i+\tau}, \dots, t_{i+(m-1)\tau})$ [11]. Розмірність m може бути отриманою за допомогою методу фальшивих найближчих сусідів (теоретично, $m = 2n + 1$ [10, 11]).

Радіус околу сусідніх точок ε_i визначає сферу, центр якої знаходиться у точці \vec{x}_i . Якщо \vec{x}_j попадає у сферу точки \vec{x}_i , то стани відповідних точок вважаються близькими, і $\mathbf{R}_{ij} = 1$. Значення радіуса ε_i може бути як фіксованим, так і варіюватись таким чином, щоб у кожен сферу попадало не більше, ніж задана кількість сусідніх точок. Однак переважно використовується стале значення радіуса для усього ряду.

В якості норми часто використовується також максимум модулів різниць між відповідними координатами:

$$\|\vec{a} - \vec{b}\| = \max \{a_i - b_i\}, \quad i = 1, \dots, m. \quad (2)$$

З врахуванням сталого радіуса ε_i та норми (2) матриця \mathbf{R} отримується симетричною, а бінарні значення у \mathbf{R} досить просто візуалізувати у вигляді відображення на площині точок з координатами, що відповідають рядкам і стовпчикам

у \mathbf{R} , та зображеними чорним кольором при $\mathbf{R}_{ij} = 1$ і білим кольором при $\mathbf{R}_{ij} = 0$. На рис. 1 подано графіки двох числових рядів та відповідні їм рекурентні діаграми.

На основі матриці крос-рекурентності \mathbf{R} може бути розраховано ряд характеристик вихідної динамічної системи, зокрема, коефіцієнт рекурентності (recurrence rate) [7], коефіцієнт визначеності (determinism) [12], ламінарність (laminarity) [13], час рекурентності першого типу (recurrence times of first type) [14], коефіцієнт кластеризації (clustering coefficient) [15] тощо.

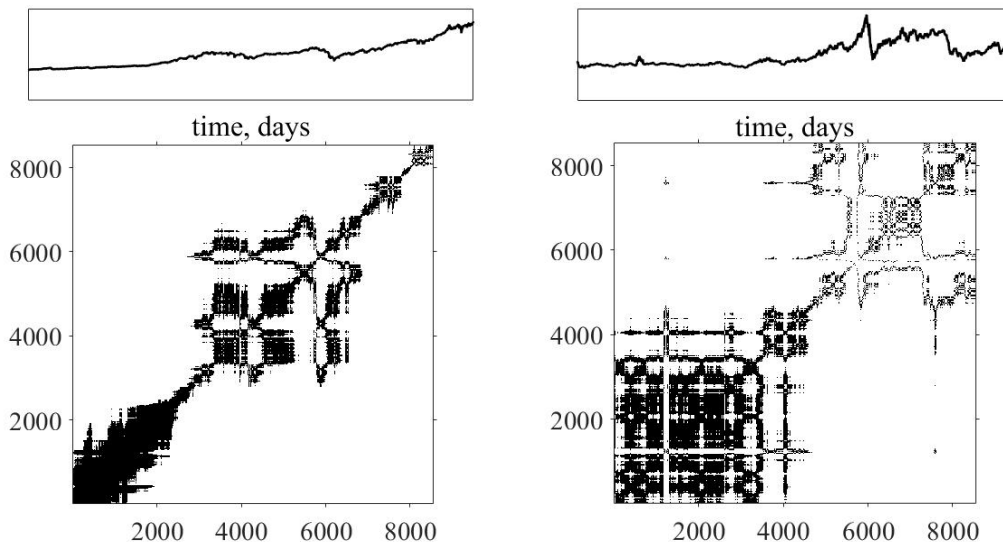


Рис. 1. Угорі: синтетичний часовий ряд, унизу: відповідна крос-рекурентна діаграма.

Шляхом дослідження конфігурацій, утворених з одиниць матриці \mathbf{R} , можна отримувати інформацію про природу фазового простору. Зокрема: 1) паралельний рух точок описується діагональними рядами; 2) ламінарність можна досліджувати шляхом аналізу вертикальних та/чи горизонтальних рядів; 3) періодична чи квазіперіодична природа фазового простору визначається на основі наявності лише діагональних рядів з відповідно однаковою чи різною часовою затримкою; 4) прямокутні структури складаються з діагональних ліній, утворених ізольованими точками; 5) вертикальні ряди відповідають хаотичній природі системи.

За допомогою рекурентних діаграм можуть бути досліджені й більш складні структури, що виникають у стохастично хаотичних чи суто хаотичних системах. Наприклад, наявність рівномірно розподілених ізольованих точок свідчить про процес, що є білим шумом, а окремі ізольовані точки свідчать про сильні флуктуації, наявні у процесі, що відповідає некорельованому чи антикорельованому процесу у системі. Вказані характеристики можуть також бути оцінені за допомогою щільності періоду рекурентності.

Одним зі способів оцінки хаотичності системи є аналіз показника ентропії, що базується на концепції ентропії Шеннона:

$$H = -\sum_{i=1}^N p(x_i) \log_2(p(x_i)), \quad (3)$$

де $p(x_i)$ – ймовірність події x_i (за домовленістю, $0 \log_2 0 = 0$).

Далі у роботі розглядається два види ентропії на основі ентропії Шеннона, які можна розрахувати шляхом аналізу характеристик матриці крос-рекурентності \mathbf{R} : ентропію періоду рекурентності (recurrence period density entropy, RPDE) та ентропію рекурентності (recurrence entropy, RecEn).

Ентропія періоду рекурентності

Міра RPDE підходить з ідеї про рекурентність часу між рекурентними точками [16]. Час рекурентності обчислюється шляхом підрахунку кількості нерекурентних точок, які йдуть підряд у матриці крос-рекурентності, що відповідає довжині нулів у матриці крос-рекурентності \mathbf{R} чи довжині білих ліній на рекурентній діаграмі. Так, для двох рекурентних точок \vec{x}_i та \vec{x}_j час рекурентності визначається як $T_k = i - j$. Тоді для усіх точок, відображених у матриці \mathbf{R} , можна отримати послідовність інтервалів часів рекурентності $R(T_k)$, підрахувавши кількість кожного з інтервалів T_k . На основі послідовності $R(T_k)$ отримується щільність періодів рекурентності $P(T_k)$, що визначається як ймовірність кожного з періодів рекурентності T_k у просторі $\{R(T_k)\}$:

$$P(T_k) = \frac{R(T_k)}{\sum_{k=1}^{T_{\max}} R(T_k)}, \quad (4)$$

де $T_{\max} = \max\{T_k\}$.

Отримані ймовірності періодів рекурентності є основою для обчислення ентропії періоду рекурентності на основі ентропії Шеннона:

$$RPDE = -\sum_{k=1}^{T_{\max}} P(T_k) \ln P(T_k). \quad (5)$$

Оскільки T_{\max} може змінюватись з часом, для показника RPDE може бути необхідним застосування нормалізації, внаслідок чого також використовується нормалізована ентропія періоду рекурентності:

$$NRPDE = -\frac{1}{\ln T_{\max}} \sum_{k=1}^{T_{\max}} P(T_k) \log P(T_k), \quad \ln T_{\max} = -\sum_{i=1}^{T_{\max}} P(i) \ln P(i). \quad (6)$$

Фактично, показник ентропії періоду рекурентності визначає міру упорядкованості системи, опираючись на довжини послідовностей сусідніх точок фазового простору: чим більше точок будуть сусідами у фазовому просторі – тим меншим буде значення показника ентропії, розрахованого за формулами (5) чи (6).

Ентропія рекурентності

Однак, у поданому вище визначення ентропії оцінка «сусідства» точок проводиться на основі застосування норми $\|\cdot\|$ до пар відповідних координат. Наприклад, для двох точок m -вимірного фазового простору $\mathbf{a}(a_1, a_2, \dots, a_m)$ та $\mathbf{b}(b_1, b_2, \dots, b_m)$ порівнюються лише пари координат (a_1, b_1) , (a_2, b_2) , ..., (a_m, b_m) , при

цьому відкидається велика кількість пар, утворених різними координатами точок \mathbf{a} та \mathbf{b} .

Врахування й інших пар пропонується у роботі [17], де авторами вводиться поняття *мікростанів*, що визначає взаємне розташування двох точок. Для визначення мікростану використовуються точки у 2-, 3- чи 4-вимірному просторі. Розмірність простору визначається параметром N , який назовемо *порядком мікростану*.

Розглянемо визначення мікростану 2-го порядку для двох точок $\mathbf{a}(a_1, a_2)$ та $\mathbf{b}(b_1, b_2)$. Маємо для оцінки 4 пари координат: (a_1, b_1) , (a_1, b_2) , (a_2, b_1) , (a_2, b_2) . Вибравши норму $\|\cdot\|$ та застосувавши до кожної з пар функцію Хевісайда $\Theta(\varepsilon - \|a_i - b_j\|)$, отримаємо послідовність нулів та одиниць, довжина якої рівна кількості оцінюваних пар координат. Така послідовність і є мікростаном.

Оскільки кожний з мікростанів можна подати числом у двійковій системі числення, розрядність якого відповідає кількості оцінюваних пар, для мікростанів порядку 2, існує 16 різних значень: від 0000 до 1111.

На основі мікростанів автори роботи [17] вводять поняття *ентропії рекурентності* $S(A)$, яка визначається наступним чином:

$$RecEn = S(A) = -\sum_A p_A \ln p_A, \quad (7)$$

де $A = A(\varepsilon)$ – розподіл по мікростанам множини точок, поданих у фазовому просторі розмірності N , з використанням радіусу околу сусідніх точок ε .

Зупинимось на кількох особливостях визначення ентропії рекурентності $S(A)$.

1. Кількість мікростанів для порядку N визначається за формулою 2^{N^2} , де основа степеня є основою системи числення для представлення мікростанів, а показник степеня N^2 відповідає кількості порівнюваних пар координат. Зважаючи на використання двійкової системи числення для представлення мікростанів, маємо, наприклад, $2^{2^2} = 16$ мікростанів порядку 2: від $0000_2 = 0_{10}$ до $1111_2 = 15_{10}$, $2^{3^2} = 512$ мікростанів порядку 3, або ж $2^{4^2} = 65536$ мікростанів порядку 4. Очевидно, зважаючи на кількість мікростанів порядку 4, більші порядки авторами роботи [17] не використовуються. З іншого боку, порядок мікростанів висуває вимоги й до кількості точок часового ряду, що оцінюються: бажано, щоб ця кількість була більшою за максимальну кількість мікростанів, інакше буде багато мікростанів, кількість яких у досліджуваній системі рівна 0. Останнє, однак, не є необхідною умовою.

2. Розподіл точок по мікростанам прямо залежить від околу сусідства ε . Так, якщо при використанні норми (2) для числового ряду $\{t_i\}$ довжиною K вибрати $\varepsilon > \max\{|t_i - t_j|\}$, де $i, j = 1, \dots, K$, то усі можливі мікростани будуть рівні $\underbrace{1\dots 1}_{N^2}$, і тоді значення ентропії рекурентності становитиме $S(A) = 1 \cdot \ln 1 = 0$. Таке ж значення буде отримане й у випадку, коли вибрати $\varepsilon < \min\{|t_i - t_j|\}$, оскільки усі мікростани будуть рівні $\underbrace{0\dots 0}_{N^2}$. Таким чином, окіл сусідства повинен вибиратись з інтервалу $\varepsilon \in (\min\{|t_i - t_j|\}, \max\{|t_i - t_j|\})$. Наприклад, у роботі [15] для часових рядів, що містять білий шум, експериментально отримано значення $\varepsilon \in [0.14, 0.45]$ і для таких рядів

пропонується $\varepsilon = 0.293$, для якого отримується максимальне значення ентропії рекурентності $S(A)$.

Використовуване програмне забезпечення

Для дослідження часових рядів нами використано наступне програмне забезпечення.

Розрахунок ентропії за Н. Марваном, яку автор називає *ентропія щільності періоду рекурентності* (recurrence period density entropy, RPDE) [12], проводилась за допомогою функцій пакету інструментів Cross Recurrence Plot [18].

Для розрахунку ентропії рекурентності було реалізовано окремий пакет у інструментальному середовищі MatLab.

Обидва методи об'єднано в інструменті з графічним інтерфейсом, що дозволяє обирати аналізований часовий ряд та параметри для розрахунку вказаного показника ентропії. Результати обчислень як візуалізуються, так і зберігаються у файл.

Вибір радіусу сусідства можна робити, задавши його числове значення, або ж провівши попередню оцінку. У останньому випадку для числового ряду $\{t_i\}$ проводиться розрахунок значень ентропії для послідовності значень $\varepsilon_j \in (\min\{|t_i - t_j|\}, \max\{|t_i - t_j|\})$, взятих з певним кроком $\Delta\varepsilon$, після чого в якості початкового значення параметра ε автоматично обирається те, для якого отримано максимальне значення ентропії, або ж користувачем вибирається бажане значення ε на основі розрахованого профіля ентропій $E(\varepsilon)$.

Додатково, оскільки, у першу чергу, нас цікавить саме динаміка зміни показника ентропії у часовому ряді, в інструменті реалізовано процедуру зміщеного вікна (moving window procedure), де показник ентропії розраховується для поточного підряду довжиною K , після чого вибір підряду зміщується на деякий крок h .

Аналіз показників ентропії на синтетичних часових рядах

Оцінка значень ентропії періоду рекурентності RPDE та ентропії рекурентності RecEn спочатку проводилась на синтетичних часових рядах, що містять $1/f$ -шум з різними показниками α від 0.0 (білий шум) до 2.0 (броунівський рух) з кроком $\Delta\alpha = 0.2$. При дослідженні використовувався набір показників ε з відрізка $[0,5]$, взятих з кроком $\Delta\varepsilon = 0.1$. Для кожного набору показників ε та α генерувалось по 10 рядів, для яких розраховані значення ентропій потім усереднювались для візуалізації результатів (рис. 2).

Аналіз результатів дозволив отримати наступні висновки.

1. У кожному випадку для значення ентропії рекурентності при збільшенні околу сусідніх точок ε спостерігається спочатку швидке зростання, потім – повільне спадання. При цьому для профіля значень ентропії рекурентності існує максимум у точці, близькій до $\varepsilon = 1.0$, який зміщується до $\varepsilon = 0.7$ зі зростанням значення α при розрахунку ентропії порядку 2, та до $\varepsilon = 0.3$ при розрахунку ентропії порядку 3.

2. Отримані усереднені значення ентропії рекурентності дозволяють використовувати у більшості випадків значення $\varepsilon = 1.0$ при використанні порядку 2 та $\varepsilon = 0.5$ при використанні порядку 3.

3. Максимальне значення ентропії рекурентності зменшується зі зростанням α : при використанні порядку 2 від $RecEn \approx 2.8$ для $\alpha = 0$ до $RecEn \approx 1.1$ для $\alpha = 2$ та при використанні порядку 3 від $RecEn \approx 6$ для $\alpha = 0$ до $RecEn \approx 1.9$ для $\alpha = 2$ (рис. 3).

Таким чином, оцінку отриманих значень ентропії рекурентності для довільного ряду можна проводити і шляхом порівняння аналізованого ряду з рядом $1/f$ -шуму.

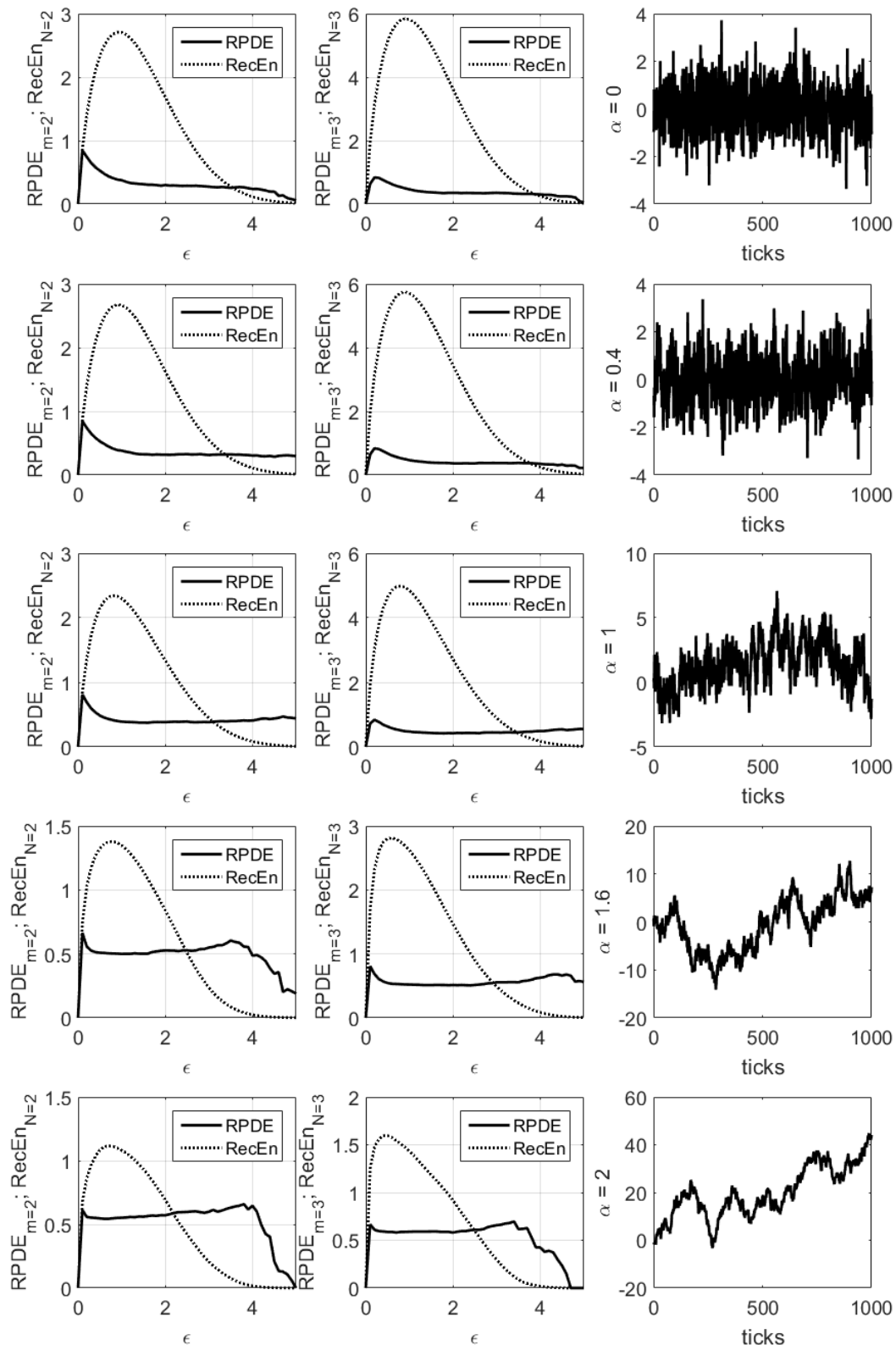


Рис. 2. У кожному ряду зображень: зліва – профіль значень ентропії періодів (RPDE) та ентропії рекурентності (RecEn), взятих для розмірності фазового простору $m = 2$ і порядку $N = 2$ відповідно; по центру – профіль значень ентропії періодів та ентропії рекурентності, взятих для розмірності фазового простору $m = 3$ і порядку $N = 3$ відповідно; справа – зразок часового ряду $1/f$ -шуму з заданим значенням α .

4. Значення ентропії періодів різко зростає при збільшенні ε від 0 до 0.1, після

чого у більшості випадків стабілізується, і лише при наближенні ε до 0.5 починає спадати до 0. Отриманий профіль ентропії періодів підтверджує вибір Н. Марваном значення радіуса $\varepsilon = 0.1$ за замовчуванням у пакеті CRPTool.

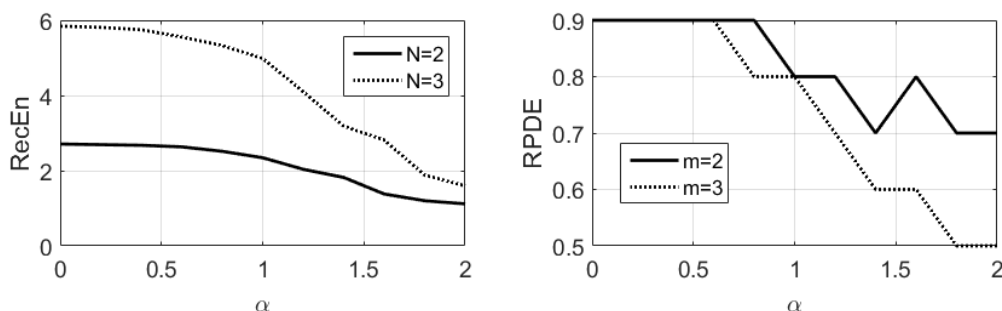


Рис. 3. Профілі максимальних значень ентропії для рядів $1/f$ -шуму з різними значеннями α : зліва – профіль значень ентропії рекурентності (RecEn), справа – профіль значень ентропії періодів рекурентності (RPDE).

Для оцінки значень більш упорядкованих систем використано періодичну функцію $\sin(x)$.

Оцінка повністю упорядкованої системи проводилась на основі часових рядів, згенерованих за допомогою функції $\sin(x + \omega)$ з різними фазами ω . Профілі отриманих ентропій подано на рис. 4, угорі.

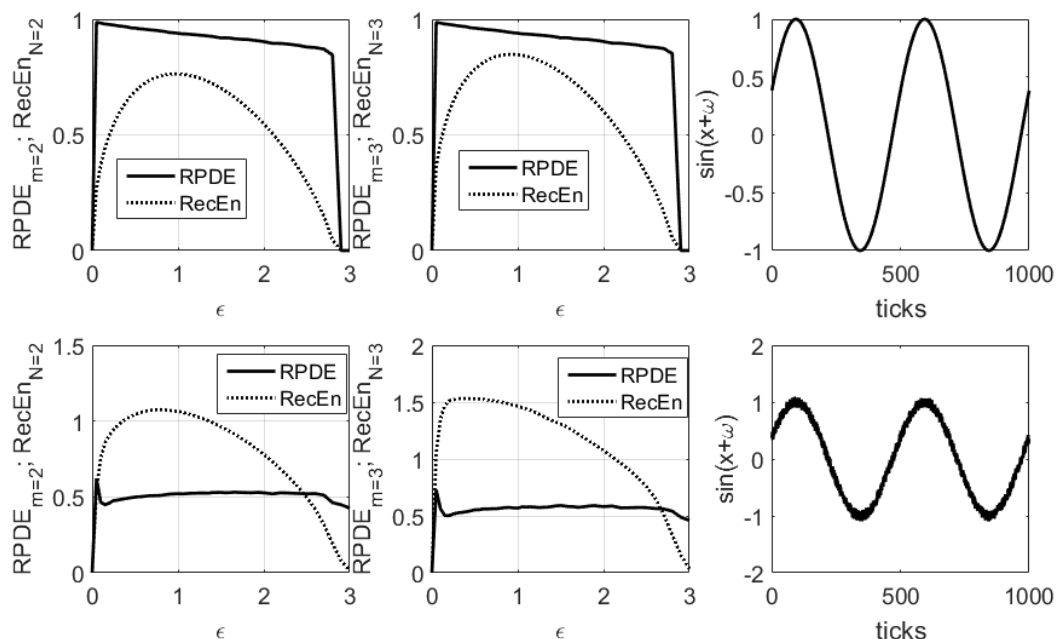


Рис. 4. У кожному ряду зображень: зліва – профіль значень ентропії періодів (RPDE) та ентропії рекурентності (RecEn), взятих для розмірності фазового простору $m = 2$ і порядку $N = 2$ відповідно; по центру – профіль значень ентропії періодів та ентропії рекурентності, взятих для розмірності фазового простору $m = 3$ і порядку $N = 3$ відповідно; справа – зразок часового ряду $\sin(x + \omega)$ (угорі) та $\sin(x + \omega) + rnd$ (унизу).

Як видно з профілів, загальні характеристики ентропій, визначені вище, повторюються. Відмінністю є менше максимальне значення ентропії рекурентності,

рівне $RecEn \approx 0.75$ для порядку 2 та $RecEn \approx 0.85$ для порядку 3, та вище максимальне значення ентропії періодів. Знайдені значення ε , при яких ентропії мають максимальні значення, узгоджуються зі знайденими раніше.

Додатково було досліджено поведінку профілів ентропій для часових рядів, згенерованих за допомогою функції $\sin(x+\omega)+rnd$, де rnd – випадкове число з рівномірним розподілом на відрізку $[-0.1,0.1]$. Генерувались такі ряди з метою оцінки поведінки ентропій у випадку різних механізмів роботи систем на низькочастотному та високочастотному рівнях. Для такого випадку поведінка обох ентропій схожа на їх поведінку при броунівському русі, тобто, при $1/f$ -шумі при $\alpha = 2.0$ (див. рис. 4, унизу).

Таким чином, на основі проведених вище досліджень було зроблено додатковий висновок про оданкову поведінку досліджуваних ентропій, незалежно від природи формування часових рядів, що дозволяє зробити припущення про універсальність ентропії рекурентності та ентропії періодів.

Оскільки дослідження динаміки зміни показників у часових рядах проводиться з використанням процедури рухомого вікна, таку ж процедуру було застосовано й для тестових рядів. Метою було визначення стійкості отримуваних значень при суттєво меншій ширині використовуваного вікна порівняно з довжинами аналізованих вище синтетичних рядів та загалом визначення можливості застосування досліджуваних ентропій у процедурі рухомого вікна.

Процедура проводилась для синтетичних рядів $1/f$ -шуму з різними показниками α при ширині вікна $win = 250$ та кроці вікна $step = 5$ (самі такі параметри вікон переважно застосовується нами при аналізі часових рядів економічної природи з денними показниками).

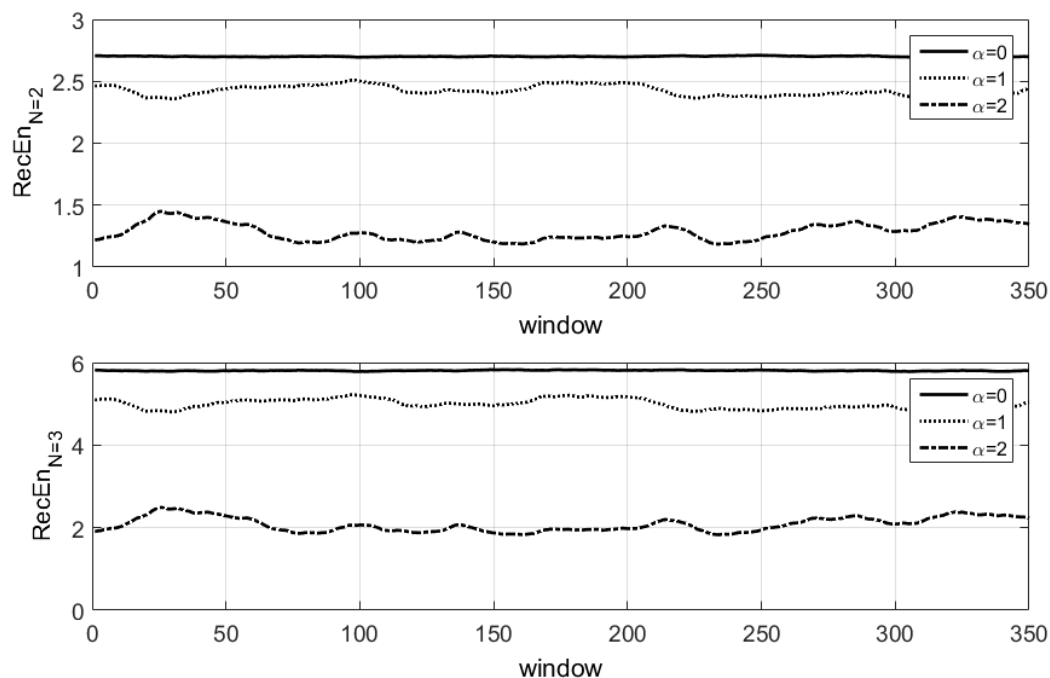


Рис. 5. Динаміка значення ентропії рекурентності ($RecEn$) при використанні процедури рухомого вікна шириною $win = 250$ та кроком вікна $step = 5$, розрахованих для рядів $1/f$ -шуму з показниками $\alpha = 0,1,2$ для порядку $N = 2$ (угорі) та $N = 3$ (унизу).

Для ентропії рекурентності отримано показники ResEn, близькі до наведених вище. Звичайно, невелика ширина вікна вносить свої збурення, що стають більш чутливими зі зростанням показника α (див. рис. 5). При цьому порядок ентропії практично не є суттєвим.

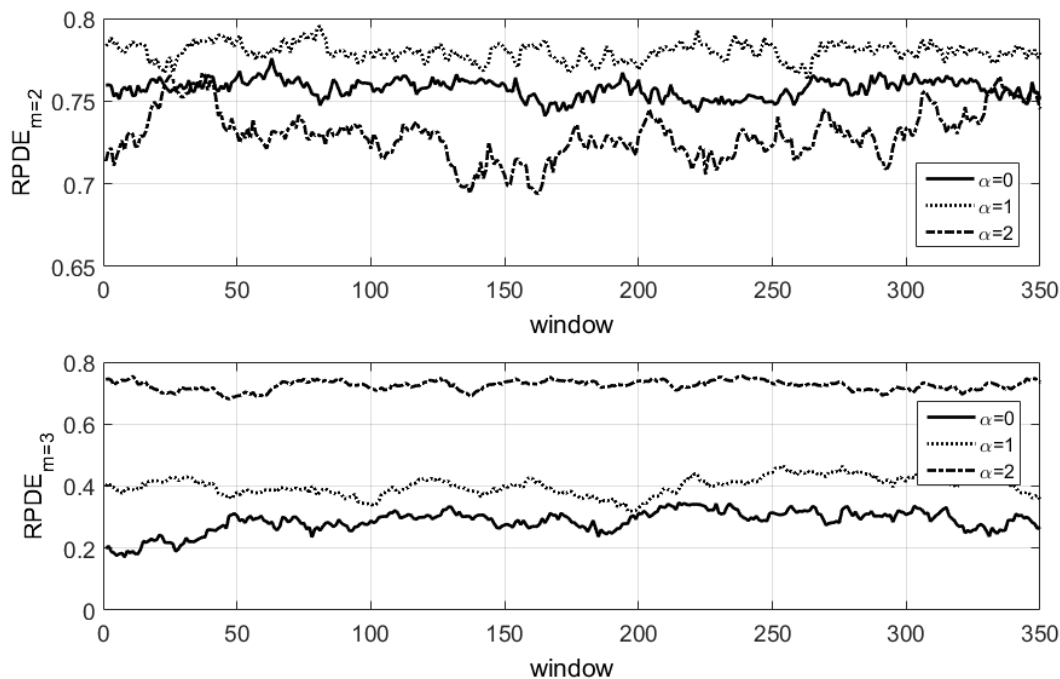


Рис. 6. Динаміка значення ентропії періодів (RPDE) при використанні процедури рухомого вікна шириною $win = 250$ та кроком вікна $step = 5$, розрахованих для рядів $1/f$ -шуму з показниками $\alpha = 0,1,2$ для розмірності фазового простору $m = 2$ (угорі) та $m = 3$ (унизу).

Подібна ситуація отримана й для ентропії періодів (рис. 6).

Отже, ентропія рекурентності та ентропія періодів рекурентності можуть бути використані у віконних процедурах, при цьому ентропія рекурентності показує сталі показники навіть при порівняно малій ширині рухомого вікна.

Аналіз курсу Dow Jones Industrial Average та цін на нафту

Дослідження можливості застосування ентропії рекурентності та ентропії періодів у економічних системах проводилось на основі денних даних індекса Dow Jones Industrial Average (DJIA), взятих за період з 1990 р. по 2019 р., та денних значень ціни на спотовому ринку нафти, взятих за період з 1987 р. по 2019 р. З метою оцінки динаміки зміни значень показників ентропій у окремі критичні періоди економічних систем було складено таблиці критичних та кризових явищ на відповідних ринках.

У таблиці 1 подано список критичних та кризових явищ на фондовому ринку Доу Джонса за досліджуваний період. Відповідні номери явищ відмічено й на рис. 7 разом з графіками динаміки значень ентропії рекурентності та ентропії періоду рекурентності, отриманими за допомогою процедури рухомого вікна при використанні ширини вікна $win = 500$ та кроку вікна $step = 1$.

Аналіз отриманих результатів дозволив зробити наступні висновки.

1. Значення RPDE різко падає, щонайменше, на 10%, при попаданні у вікно періоду критичного явища, яке проявляється у часовому ряді у вигляді поступового

зменшення чи різкого падіння індексу. Поступове зменшення показника RPDE спостерігається іноді з деякий час до входження у рухоме вікно періоду відповідно критичного явища.

2. Поведінка значення RecEn подібна до поведінки RPDE, за деякими відмінностями, а саме: графік динаміки зміни RecEn є більш гладким і поступове зменшення значення RecEn починається за більший час порівняно з RPDE.

Таблиця 1

Критичні та кризові явища на фондовому ринку Доу-Джонса за період з 01.01.1990 по 01.06.2019 рр.

| N | Часовий період | Тривалість, днів | Падіння показника, % |
|---|-----------------------|------------------|----------------------|
| 1 | 17.07.1990-23.08.1990 | 28 | 17.21 |
| 2 | 01.10.1997-21.10.1997 | 15 | 12.43 |
| 3 | 17.08.1998-31.08.1998 | 11 | 18.44 |
| 4 | 14.08.2002-01.10.2002 | 34 | 19.52 |
| 5 | 16.10.2008-15.12.2008 | 42 | 30.21 |
| 6 | 09.08.2011-22.09.2011 | 32 | 11.94 |
| 7 | 18.08.2015-25.08.2015 | 6 | 10.53 |
| 8 | 29.12.2015-20.01.2016 | 16 | 11.02 |
| 9 | 03.12.2018-24.12.2018 | 15 | 15.62 |

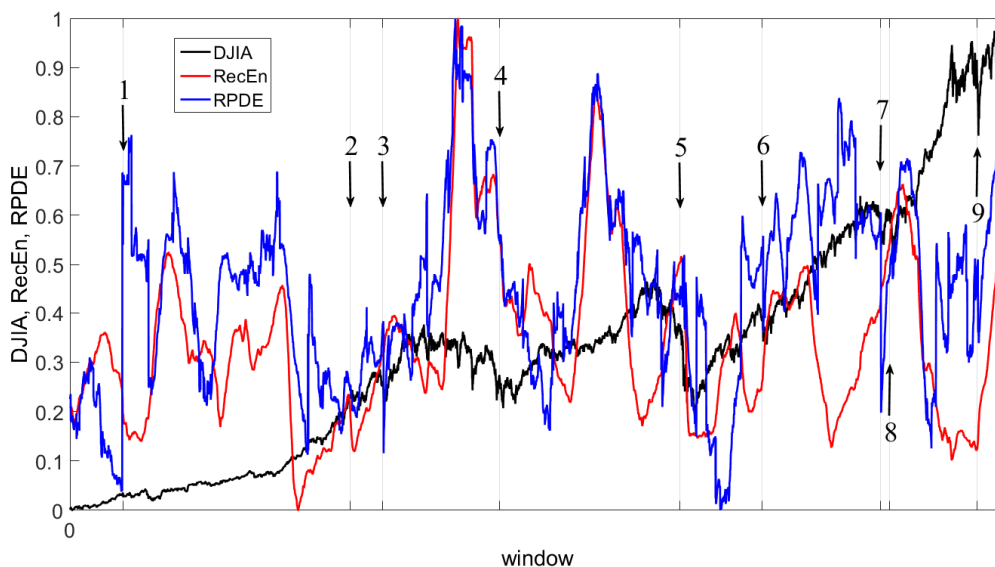


Рис. 7. Динаміка значень ентропії рекурентності (RecEn) та ентропії періоду рекурентності (RPDE) при використанні процедури рухомого вікна шириною $win = 500$ та кроком вікна $step = 1$, розрахованих для Dow Jones Index Average. Стрілками відображено кризові явища, подані у табл. 1.

У таблиці 2 подано список критичних та кризових явищ на спотовому ринку нафти за досліджуваний період. Відповідні номери явищ відмічено на рис. 8 разом з графіками динаміки значень ентропії рекурентності та ентропії періоду рекурентності, отриманими за допомогою процедури рухомого вікна при використанні ширини вікна $win = 500$ та кроку вікна $step = 1$.

Таблиця 2

Критичні та кризові явища на спотовому ринку нафти
за період з 01.01.1987 по 01.06.2019 рр.

| N | Часовий період | Тривалість, днів | Падіння показника, % |
|----|-----------------------|------------------|----------------------|
| 1 | 09.12.1987-21.12.1987 | 9 | 18 |
| 2 | 11.10.1990-23.08.1990 | 8 | 31 |
| 3 | 17.11.1993-17.12.1993 | 22 | 18 |
| 4 | 11.04.1996-05.06.1996 | 38 | 22 |
| 5 | 30.09.1998-25.11.1998 | 40 | 33 |
| 6 | 07.03.2000-10.04.2000 | 24 | 29 |
| 7 | 27.11.2000-20.12.2000 | 17 | 29 |
| 8 | 14.09.2001-24.09.2011 | 6 | 27 |
| 9 | 12.03.2003-21.03.2003 | 7 | 28 |
| 10 | 26.10.2004-10.12.2004 | 31 | 28 |
| 11 | 07.08.2006-17.11.2006 | 73 | 27 |
| 12 | 03.07.2008-23.12.2008 | 120 | 80 |
| 13 | 03.05.2010-25.05.2010 | 16 | 25 |
| 14 | 29.04.2011-17.05.2011 | 12 | 15 |
| 15 | 24.02.2012-28.06.2012 | 87 | 27 |
| 16 | 06.09.2013-27.11.2013 | 58 | 17 |
| 17 | 20.06.2014-29.01.2015 | 152 | 59 |
| 18 | 03.11.2015-20.01.2016 | 52 | 44 |
| 19 | 03.10.2018-27.12.2018 | 56 | 41 |

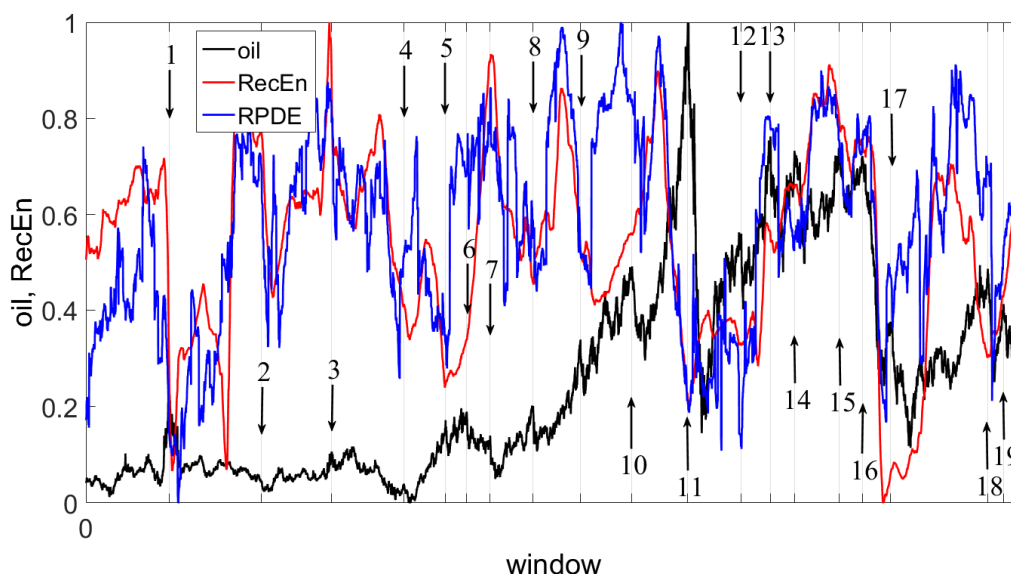


Рис. 8. Динаміка значень ентропії рекурентності (RecEn) та ентропії періоду рекурентності (RPDE) при використанні процедури рухомого вікна шириною $win = 500$ та кроком вікна $step = 1$, розрахованих для цін на спотовому ринку нафти. Стрілками відображено кризові явища, подані у табл. 2.

Висновки, отримані у результаті аналізу отриманих графіків, підтвердили висновки, зроблені для індекса ДІА, й дозволили їх узагальнити наступним чином.

1. У поведінці динаміки значення RPDE спостерігається яскраво виражена волатильність, обумовлена, очевидно, способом розрахунку показника ентропії. У той же час графік значення RecEn є більш згладженим.

2. На обидва показники суттєво впливає значення околу сусідства, а тому перед проведенням дослідження на заданому часовому ряді необхідне проведення процедури оцінки використовуваного значення.

3. Значення RecEn є більш чутливим до критичних явищ і дозволяє виявляти їх за деякий час до початку безпосередньо самого явища, у період його підготовки. Це узгоджується з деякими теоріями виникнення та розвитку критичних явищ у економічних системах, що наголошують про внутрішні зміни системи під час наближення критичних та кризових явищ.

Висновки

У результаті проведеної роботи розглянуто застосування ентропійного підходу до дослідження економічних систем, результати діяльності яких подано у вигляді часових рядів. Розглянуто два методи розрахунку показників ентропії на основі часових рядів – ентропію рекурентності та ентропію періоду рекурентності – що опираються на методіку рекурентних відображень Н. Марвана. Досліджено поведінку розглянутих ентропій на основі часових рядів, що містять економічні показники: ряду денних значень індекса DJIA та ряду денних значень цін нафти на спотовому ринку. Зроблено висновки про чутливість розглянутих показників ентропії до критичних та кризових явищ, що відбуваються у економічних системах. Відмічено більшу чутливість ентропії періоду рекурентності до змін, що відбуваються у системі, та більший час, за який ентропія рекурентності починає реагувати на підготовку у економічній системі критичного явища.

Отже, ентропії періоду рекурентності та рекурентності можуть успішно застосовуватись до дослідження стану економічних систем та, потенційно, слугувати індикаторами-передвісниками надходження особливих станів економічної системи у майбутньому.

Список використаної літератури:

1. Кубо, Р. Термодинамика / Р. Кубо; пер с англ. – М.: Мир, 1970. – 307 с.
2. Sole, R.V. Information theory of complex networks: on Evolution and Architectural constraints / R.V. Sole, S. Valverde // *Lecture Notes in Physics*. – 2004. – Vol. 650. – p. 189-207.
3. Tsallis, C. Introduction to Nonextensive Statistical Mechanics, Approaching a Complex World / C. Tsallis. – New-York: Springer, 2009. – 382 p.
4. Joshua, S. Physiological time-series analysis using approximate entropy and sample entropy / S. Joshua, J. Richman, R. Moorman // *Am. J Physiol. Heart Circ. Physiol.* – 2000. – I. 278. – H2039-H2049.
5. Синергетичні та еконофізичні методи дослідження динамічних та структурних характеристик економічних систем: монографія / В. Д. Дербенцев, О. А. Сердюк, О. Д. Шарапов, В. М. Соловйов. – Черкаси: Брама-Україна, 2010. – 300 с.
6. Моделирование сложных систем: Навчально-методичний посібник для самостійного вивчення дисципліни / В.М. Соловйов, О.А. Сердюк, Г.Б. Данильчук. – Черкаси : Брама-Черкаси, 2016. – 194 с.
7. Marwan, N. Recurrence plots for the analysis of complex systems / N. Marwan, M.C. Romano, M. Thiel, J. Kurths // *Physics Reports*. – 2007. – V. 438. – P. 237-329. – Режим доступу: <https://doi.org/10.1016/j.physrep.2006.11.001>
8. Eckmann, J.-P. Recurrence Plots of Dynamical Systems / J.-P. Eckmann, S. Kamphorst, D. Ruelle // *Europhysics Letters (epL)*. – 1987. – V. 4. – P. 973-977. – Режим доступу: <https://doi.org/10.1209/0295-5075/4/9/004>
9. Ott E. *Chaos in Dynamical Systems* / E. Ott. – Cambridge University Press, 2012. – 478 p. – Режим доступу: <https://doi.org/10.1017/CBO9780511803260>
10. Argyris, J. H. *An Exploration of Chaos: An Introduction for Natural Scientists and Engineers (Texts on Computational Mechanics)* / J. H. Argyris, G. Faust, M. Haase. – North Holland, 1994. – 751 pp.

11. Kantz H. *Nonlinear Time Series Analysis* / H. Kantz, T. Schreiber. – Cambridge University Press, 2003. – 388 pp.
12. Cross Recurrence Plot Toolbox for Matlab. Reference Manual [Електронний ресурс]. – Режим доступу: <http://tocsy.pik-potsdam.de/CRPtoolbox/>
13. Marwan, N. Recurrence-plot-based measures of complexity and their application to heart-rate-variability data / N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan, J. Kurths // *Phys. Rev.* – 2002. – Vol. 66, I. 2. – 026702. – Режим доступу: <https://doi.org/10.1103/PhysRevE.66.026702>
14. Gao, J. On the structures and quantification of recurrence plots / J. Gao, H. Cai // *Physics Letters A.* – 2000. – V. 270, I. 1-2. – P. 75-87. – Режим доступу: [https://doi.org/10.1016/S0375-9601\(00\)00304-2](https://doi.org/10.1016/S0375-9601(00)00304-2)
15. Marwan, N. Complex network approach for recurrence analysis of time series / N. Marwan, J. Donges, Y. Zou, R. Donner, J. Kurths // *Physics Letters A.* – 2009. – V. 373. – P. 4246-4254. – Режим доступу: <https://doi.org/10.1016/j.physleta.2009.09.042>
16. Little, M.A. Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection / M.A. Little, P.E. McSharry, S.J. Roberts, D.AE. Costello, I.M. Moroz // *BioMedical Engineering OnLine.* – 2007. – V. 6, Article number: 23. – Режим доступу: <https://doi.org/10.1186/1475-925X-6-23>
17. Corso, G. A novel entropy recurrence quantification analysis / G. Corso, T. L. Prado, G. Z. dos S. Lima, S. R. Lopes [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/1707.00944>
18. CROSS RECURRENCE PLOT TOOLBOX [Електронний ресурс]. – Режим доступу: <http://tocsy.pik-potsdam.de/CRPtoolbox/index.html>

Bibliography:

1. Kubo, R. (1970). *Termodinamika* [Thermodynamics]. M.: Mir [in Russian].
2. Sole, R.V., & Valverde S. (2004). Information theory of complex networks: on Evolution and Architectural constraints. *Lecture Notes in Physics*, 650, 189-207.
3. Tsallis, C. (2009). *Introduction to Nonextensive Statistical Mechanics, Approaching a Complex World*. New-York: Springer.
4. Joshua, S., Richman, J., & Moorman R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *Am. J Physiol. Heart Circ. Physiol.* 278, H2039-H2049.
5. Derbentsev, V.D., Serdiuk, O.A., Sharapov, O.D., & Soloviev, V.M. (2010). Synerhetychni ta ekonofizychni metody doslidzhennia dynamichnykh ta strukturnykh kharakterystyk ekonomichnykh system: monohrafiia [Synergetic and econophysical methods for studying the dynamic and structural characteristics of economic systems: a monograph]. Cherkasy: Brama-Ukraine [in Ukrainian].
6. Soloviev, V.M., Serdiuk, O.A., Danylchuk, G.B. (2016). Modeliuvannia skladnykh system [Modeling of Complex Systems]. *Navchalno-metodychnyi posibnyk dlia samostiinoho vyvchennia dysypliny* [A Guide for Independent Study of a Discipline]. Cherkasy: Brama-Ukraine [in Ukrainian].
7. Marwan, N., Romano, M.C., Thiel, M., & Kurths J. (2007). Recurrence plots for the analysis of complex systems. *Physics Reports*, 438, 237-329. Retrieved from: <https://doi.org/10.1016/j.physrep.2006.11.001>
8. Eckmann, J.-P., Kamphorst, S., & Ruelle D. (1987). Recurrence Plots of Dynamical Systems. *Europhysics Letters (epL)*, 4, 973-977. – Retrieved from: <https://doi.org/10.1209/0295-5075/4/9/004>
9. Ott, E. (2012). *Chaos in Dynamical Systems*. Cambridge University Press. – Retrieved from: <https://doi.org/10.1017/CBO9780511803260>
10. Argyris, J. H., Faust, G., & Haase, M. (1994). *An Exploration of Chaos: An Introduction for Natural Scientists and Engineers* (Texts on Computational Mechanics). North Holland.
11. Kantz, H., & Schreiber, T. (2003). *Nonlinear Time Series Analysis*. Cambridge University Press.
12. Cross Recurrence Plot Toolbox for Matlab. Reference Manual. Retrieved from: <http://tocsy.pik-potsdam.de/CRPtoolbox/>
13. Marwan, N., Wessel, N., Meyerfeldt, U., Schirdewan, A., & Kurths J. (2002). Recurrence-plot-based measures of complexity and their application to heart-rate-variability data. *Phys. Rev.* 66(2), 026702. Retrieved from: <https://doi.org/10.1103/PhysRevE.66.026702>
14. Gao, J., & Cai, H. (2000). On the structures and quantification of recurrence plots. *Physics Letters A*, 270(1-2), 75-87. – Retrieved from: [https://doi.org/10.1016/S0375-9601\(00\)00304-2](https://doi.org/10.1016/S0375-9601(00)00304-2)
15. Marwan, N., Donges, J., Zou, Y., Donner, R., & Kurths, J. (2009). Complex network approach for recurrence analysis of time series. *Physics Letters A*, 373, 4246-4254. Retrieved from: <https://doi.org/10.1016/j.physleta.2009.09.042>
16. Little, M.A., McSharry, P.E., Roberts, S.J., Costello, D.AE., & Moroz, I.M. (2007). Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection. *BioMedical Engineering OnLine*, 6, article number: 23. Retrieved from: <https://doi.org/10.1186/1475-925X-6-23>
17. Corso, G., Prado, T. L., Lima, G. Z. dos S., & Lopes, S. R. (2017). A novel entropy recurrence quantification analysis. Retrieved from: <https://arxiv.org/abs/1707.00944>

18. CROSS RECURRENCE PLOT TOOLBOX. Retrieved from: <http://tocsy.pik-potsdam.de/CRPtoolbox/index.html>

SOLOVIEV Vladimir,

Doctor of Physical and Mathematical Sciences, Professor, Chair of the Department of Informatics and Applied Mathematics, Kryvyi Rih State Pedagogical University, Ukraine

SERDIUK Oleksandr,

Candidate of Economic Sciences, Department of Informatics and Applied Mathematics, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

THE MODELS OF APPLICATION THE RECURRENCE ENTROPY AND RECURRENCE PERIOD DENSITY ENTROPY TO THE ANALYSIS OF COMPLEX SYSTEMS DYNAMICS

Summary. Introduction. *Complex systems are the systems with a large number of agents that interact with each other, and during such interactions new characteristics of the systems are generated. The using of quantitative methods in the modeling processes involves measurement procedures where the system complexity indicators are quite important. One such approach is to use the concept of entropy, which has been successfully used to study economic systems, which are also complex systems. Currently, among many ways of determining the entropy of a complex system, there are several new ones that rely on the analysis of recurrence diagrams, a technique developed by Norbert Marwan. Consideration of two such methods was taken for the study conducted in this paper.*

The purpose of the article is to describe and evaluate the possibility of applying to the economic series two concepts of entropy based on the analysis of recurrence diagrams: recurrence entropy and recurrence period density entropy.

Results. *The paper deals with the method of calculation of two entropy types on the basis of Norbert Marwan recurrence diagrams: entropy of the recurrence period density and recurrence entropy. The behavior of entropies on synthetic time series with and without noise is demonstrated. The general neighborhood error that can be used in practical research are identified. Based on the application of the moving window procedure, the stability of the calculated values on synthetic time series is demonstrated. The results of application the entropies to time series DJIA and oil prices are demonstrated to determine the possibility of applying the entropy values considered.*

Conclusions. *The analysed indicators have shown a high sensitivity to critical and crisis phenomena in economic systems. There is a greater sensitivity of the recurrence period density entropy to changes in the system, and more time during which the recurrence entropy begins to respond to the preparation in the economic system of a critical phenomenon. Thus, the recurrence period density entropy and recurrence entropy can be successfully applied to the study of the state of economic systems and serve as pre-cursor of the arrival of unique states in the economic system in the future.*

Keywords: *complex system, entropy, recurrence plot, critical event.*

Одержано редакцією 17.06.2019 р.
Прийнято до публікації 04.09.2019 р.

УДК 519.6: 004.023

DOI 10.31651/2076-5886-2019-2-34-43

PACS PACS 02.60.-x, 02.60.Pn, 02.70.-c

ЧЕРКАС Дар'я Валеріївна

магістрантка спеціальності «Прикладна математика» Черкаського національного університету імені Богдана Хмельницького
e-mail: cherkas.dasha1@gmail.com

КРАСНОШЛИК Наталія

Олександрівна

кандидат технічних наук, доцент,
доцент кафедри прикладної математики та інформатики Черкаського національного університету імені Богдана Хмельницького

e-mail: wlik007@ukr.net

ДОСЛІДЖЕННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ РОЗВ'ЯЗУВАННЯ ЗАДАЧ ОПТИМІЗАЦІЇ

Генетичні алгоритми є одним з важливіших напрямів дослідження у еволюційному моделюванні. Їх часто використовують для знаходження розв'язку задач в різних областях, коли традиційні методи виявляються недостатньо ефективними.

У роботі розглянуто генетичні алгоритми Continuous genetic algorithm та Binary genetic algorithm для розв'язування задач глобальної оптимізації. Описано особливості розв'язування задачі комівояжера за допомогою генетичного алгоритму. Всі методи були реалізовані у середовищі GNU Octave. Генетичний алгоритм використано для розв'язання задачі комівояжера. Досліджено ефективність генетичних алгоритмів в залежності від значень його параметрів при знаходженні глобального мінімуму деяких тестових функцій. Проведено порівняння роботи генетичних алгоритмів з методом кажанів, методом імітації відпалу та класичними методами оптимізації.

Ключові слова: генетичний алгоритм, задача оптимізації, задача комівояжера.

Вступ

Дана робота присвячена одному з оптимізаційних методів. Як відомо, оптимізаційні задачі полягають у знаходженні мінімуму (максимуму) заданої функції. Таку функцію називають цільовою. Як правило, цільова функція – складна функція, що залежить від деяких вхідних параметрів. У оптимізаційних задачах потрібно знайти значення вхідних параметрів, при яких цільова функція досягає мінімального (максимального) значення. Існує цілий клас оптимізаційних методів. Умовно всі оптимізаційні методи можна розділити на методи, що використовують поняття похідної (градієнтні методи) і стохастичні методи (наприклад, методи групи Монте-Карло). За їх допомогою можна знайти екстремальне значення цільової функції, але не завжди можна бути впевненим, що отримано значення глобального екстремуму. Знаходження локального екстремуму замість глобального називається передчасною збіжністю. Крім проблеми передчасної збіжності існує інша проблема – час процесу обчислень. Найчастіше більш точні оптимізаційні методи працюють дуже довго. Для вирішення поставлених проблем і проводиться пошук нових оптимізаційних алгоритмів. Запропоновані у 1975 році Джоном Холландом генетичні алгоритми (ГА) засновані на принципах природного відбору Ч. Дарвіна. ГА відносяться до стохастичних методів. Ці алгоритми успішно застосовуються у різних областях діяльності (економіка, фізика, технічні науки і т.п.) [1].

Постановка проблеми

Генетичний алгоритм (genetic algorithm) – це еволюційний алгоритм пошуку, що використовується для розв'язання задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію. Задача кодується таким чином, щоб її розв'язання могло бути представлено у вигляді масиву подібного до інформації складу хромосоми. Цей масив часто називають саме так «хромосома». Випадковим чином в масиві створюється деяка кількість початкових елементів «осіб», або початкова популяція. Особи оцінюються з використанням функції пристосування, в результаті якої кожній особі присвоюється певне значення пристосованості, яке визначає можливість виживання особи. Після цього з використанням отриманих значень пристосованості вибираються особи допущені до схрещення (селекція). До осіб

застосовується «генетичні оператори» схрещення (crossover) і мутації (mutation), створюючи таким чином наступне покоління осіб. Особи наступного покоління також оцінюються застосуванням генетичних операторів і виконується селекція і мутація. Так моделюється еволюційний процес, що продовжується декілька життєвих циклів, поки не буде виконано критерій зупинки алгоритму:

Метою оптимізаційної задачі є вибір допустимого або оптимального розв'язку з множини альтернатив для досягнення поставленої мети. Оптимізаційна задача повинна відповідати двом основним вимогам: повинні існувати як мінімум два розв'язки, і треба знати, в якому сенсі шукане рішення має бути найкращим.

Математична модель оптимізаційної задачі складається з трьох складових: цільової функції, обмежень, граничних умов [2].

Пошук глобального мінімуму функції $f: R^n$ при наявності явних обмежень здійснюється на деякій власній підмножині Ω метричного простору R^n :

$$f(x) = f(x_1, \dots, x_n) \rightarrow \min, \quad x \in \Omega, \quad \Omega \subset R^n, \quad (1)$$

де підмножина Ω визначається обмеженнями типу рівностей:

$$q(x) = 0, \quad \text{де } q: R^n. \quad (2)$$

Метою даної статті є реалізація та дослідження генетичних алгоритмів розв'язування задач оптимізації.

Методи розв'язання

Класичний генетичний алгоритм оптимізації

В теорії ГА використовується біологічна термінологія в спрощеному вигляді [3]:

- хромосома – вектор чисел;
- ген – біт хромосоми;
- популяція – сукупність особин.

Як критерій життєздатності особини виступає функція пристосованості F – відображення сукупності хромосом особини на множину дійсних чисел. Точний вираз для функції пристосованості складається індивідуально для кожної задачі. В цьому випадку елементи x підмножини Ω в (1) можна розглядати як особини деякої популяції точок n -мірного простору, а значення функції $f(x)$ в цих точках можна розглядати як значення функції пристосованості $F = -f(x)$. Основні кроки роботи класичного ГА можна подати у наступному вигляді [3]:

- 1) генерація початкової популяції з k особин;
- 2) обчислення функції пристосованості кожної особини і відкидання найменш пристосованих особин;
- 3) вибір серед решти особин двох батьків;
- 4) застосування до батьківських хромосом оператора схрещування з певною ймовірністю (при відсутності схрещування дочірні хромосоми рівні батьківським);
- 5) застосування до отриманих дочірніх хромосом оператора мутації з певною ймовірністю (за відсутності мутації хромосоми не змінюються);
- 6) повторення кроків 3-5, поки популяція знов не буде містити k особин;
- 7) повторення кроків 1-6 до тих пір, поки не буде досягнуто критерію закінчення процесу.

Схрещування (кросовер) – це операція, при якій дві хромосоми обмінюються своїми частинами. У моделі, описуваної ГА, кросовер виконує оператор схрещування. Існують різні алгоритми оператора схрещування, найпростіші з них – це одно- і багатоточковий. В одноточковому варіанті відбувається розрив двох батьківських хромосом у випадковій позиції, після чого вони обмінюються отриманими ділянками. При багатоточковому кросовері хромосоми обмінюються ділянками, отриманими в результаті декількох розривів. Мутацією називається випадкова зміна одного або декількох генів хромосоми. Як критерій закінчення алгоритму можуть виступати дві умови:

- досягнення заданої максимальної кількості ітерацій (покоління);
- збіжність популяції.

Збіжністю популяції називається такий її стан, коли всі особини популяції практично однакові і знаходяться в області деякого екстремуму. ГА контролюється наступними параметрами [3]:

- обсягом популяції k ;
- ймовірністю мутації;
- ймовірністю схрещування;
- числом нових особин на кожному етапі розвитку популяції m ;
- максимальним числом ітерацій алгоритму N ;
- при використанні критерію збіжності популяції – точністю збіжності Q .

Особливості розв'язування задачі комівояжера за допомогою генетичного алгоритму

Постановка задачі комівояжера полягає у наступному. Нехай відстані між містами відомі та задані в матриці відстаней:

$$C = \begin{pmatrix} \infty & c_{12} & \dots & c_{1n} \\ c_{21} & \infty & \dots & c_{11} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & \infty \end{pmatrix}. \quad (3)$$

Комівояжер повинен побувати у кожному місті по одному разу. Це означає, що він повинен увійти в кожне місто і вийти з нього по одному разу. Маршрут обходу міст може починатися з будь-якого міста. Але закінчитися він повинен у тому ж місті, з якого почався. Необхідно побудувати маршрут, що проходить через усі міста найменшої довжини. Знаки « ∞ » у матриці C , які розташовані по діагоналі, використовуються для заборони переходу з кожного міста в це ж місто, тобто для заборони повернення [4].

Задача полягає у тому, щоб мінімізувати функцію мети

$$F = \sum_{j=1}^N \sum_{i=1}^N C_{ij} \rightarrow \min. \quad (4)$$

Для застосування генетичного алгоритму у якості особини приймають запис, а у кожній особині є набір хромосом. Здають його одновимірним масивом, який є послідовністю обходу заданих пунктів із поверненням у пункт відправлення. Так само є покоління із k особин, у яких є власний маршрут обходу. Наступним етапом є їх розмноження та мутація. Для кожної особини рахують функцію придатності, в даній задачі це довжина маршруту. Чим вона менша, тим краще. Найбільш пристосованих особин схрещують, перевіряючи, щоб не було повторень. Це необхідно для того, що в кожному пункті потрібно побувати тільки один раз. Після схрещення отримують нове покоління, у якому знов шукають найбільш пристосовані особини і т.д., поки не буде знайдено мінімальне значення функції пристосованості. Перевага генетичного

алгоритму в тому, що при розмноженні не потрібно обирати від батьків найкраще, тому що краще рішення все одно отримаємо через декілька поколінь, а іноді обираючи гірше можна отримати кращий результат [2].

Один з операторів, що відіграє вирішальну роль у розв'язанні задачі комівояжера, здійснюючи обмін інформацією між особинами під час генерації та допомагає отримати оптимальний розв'язок – це кросовер. В останні 20-30 років були запропоновані кілька кросоверів для задачі комівояжера. Один із них – PMX (partially mapped crossover) [5], принципи його роботи полягає у наступному:

1) вибір підрядків: вирізати два підрядки однакового розміру для кожного з батьків на одних і тих же позиціях;

2) обмін підрядками: обміняти два обрані підрядки, щоб створити прототип нащадка;

3) визначення списку відображень: визначити співвідношення відображення на основі вибраних підрядків;

4) прийняття потомства: прийняти прототипи нащадків за допомогою співвідношення відображення.

Результати дослідження

Continuous genetic algorithm та Binary genetic algorithm були реалізовані у середовищі розробки GNU Octave. Проведено дослідження Continuous genetic algorithm та Binary genetic algorithm для розв'язання задачі (1) з використанням різних значень параметрів $nparg$ – кількість змінних оптимізаційної функції (параметрів), $maxit$ – максимальне число ітерацій, $popsize$ – розмір популяції, $mutrate$ – швидкість мутації, $selection$ – частка популяції, що зберігається. Отримані результати наведено у табл. 1.

Було проведено порівняння Continuous genetic algorithm та Binary genetic algorithm. Отримані результати свідчать, що при збільшенні частоти мутацій погіршується ефективність роботи алгоритмів, отже, оптимальним значенням цього параметру є значення від 0.1 до 0.5. Якщо популяцію ($popsize$) зробити зовеликою або замалою, то алгоритми теж показують гірші результати. При дуже великому значенні $selection$, наприклад, якщо ми залишимо 70% «населення», також отримаємо гірші результати, ніж якщо б залишили 50%. Порівняльний аналіз результатів роботи Continuous genetic algorithm та Binary genetic algorithm показав, що Continuous genetic algorithm демонструє кращі результати.

Таблиця 1

Результати обчислювальних експериментів

| Вхідні параметри | Назва алгоритму | |
|---|---|---|
| | Continuous genetic algorithm | Binary genetic algorithm |
| | Тестова функція: $f(x_1, x_2) = x_1 \sin(4x_1) + 1.1x_2 \sin(2x_2)$ Точний розв'язок: $f(9.039, 8.668) = -18.5547$, $0 \leq x_1, x_2 \leq 10$ | |
| $nparg = 2$, $maxit = 300$ $popsize = 12$ $mutrate = 0.15$ $selection = 0.5$ | $f(x_1, x_2) = -18.5547$ $x_1 = 9.0384$; $x_2 = 8.6693$ | $f(x_1, x_2) = -18.5275$ $x_1 = 9.0196$; $x_2 = 8.6667$ |

Продовження таблиці 1

| | | |
|--|--|--|
| npar = 2 maxit = 200 popsize = 20 mutrate = 0.2 selection = 0.5 | $f(x_1, x_2) = -18.5351$ $x_1 = 9.039; x_2 = 8.6361$ | $f(x_1, x_2) = -18.5275$ $x_1 = 9.0196; x_2 = 8.6667$ |
| npar = 2 maxit = 200 popsize = 12 mutrate = 0.8 selection = 0.5 | $f(x_1, x_2) = -18.552$ $x_1 = 9.0389; x_2 = 8.6564$ | $f(x_1, x_2) = -18.4945$ $x_1 = 9.0588; x_2 = 8.6275$ |
| npar = 2 maxit = 200 popsize = 12 mutrate = 0.15 selection = 0.7 | $f(x_1, x_2) = -18.5518$ $x_1 = 9.0337; x_2 = 8.6751$ | $f(x_1, x_2) = -18.5249$ $x_1 = 9.0187; x_2 = 8.6663$ |
| npar = 2 maxit = 200 popsize = 8 mutrate = 0.15 selection = 0.7 | $f(x_1, x_2) = -18.551$ $x_1 = 9.046; x_2 = 8.6709$ | $f(x_1, x_2) = -18.268$ $x_1 = 9.0196; x_2 = 8.7843$ |

Порівняємо роботу ГА з іншими евристичними методами оптимізації. Для порівняння обрали метод кажанів та метод імітації відпалу, які також були реалізовані у середовищі Octave. Отримані результати наведено у табл. 2.

Таблиця 2

Результати обчислювальних експериментів

| Назва алгоритму | | | |
|--|---|--|---|
| Continuous genetic algorithm | Binary genetic algorithm | Метод кажанів | Метод імітації відпалу |
| Тестова функція: $f(x_1, x_2) = 0.5 + \frac{\sin^2(x_1^2 + x_2^2) - 0.5}{1 + 0.001(x_1^2 + x_2^2)}$ Точний розв'язок: $f(0,0) = 0, -100 \leq x_1, x_2 \leq 100$ | | | |
| $f(x_1, x_2) = 1.26e-09$ $x_1 = 0.037726$ $x_2 = 0.012659$ | $f(x_1, x_2) = 0$ $x_1 = 0$ $x_2 = 0$ | $f(x_1, x_2) = 1.1218e-07$ $x_1 = -0.080498$ $x_2 = 0.08097$ | $f(x_1, x_2) = 3.967854$ $x_1 = 0.015863$ $x_2 = 0.005481$ |
| Тестова функція: $f(x_1, x_2) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{1 + 0.01(x_1^2 + x_2^2)}$ Точний розв'язок: $f(1.897, 1.006) = -0.5231, -\infty \leq x_1, x_2 \leq \infty$ | | | |
| $f(x_1, x_2) = -0.52311$ $x_1 = 0.30007$ $x_2 = 2.126$ | $f(x_1, x_2) = -0.52311$ $x_1 = 2.1176$ $x_2 = 0.35294$ | $f(x_1, x_2) = -0.52311$ $x_1 = 1.4073$ $x_2 = 1.6216$ | $f(x_1, x_2) = -0.48758$ $x_1 = 0.058699$ $x_2 = -0.201340$ |

Продовження таблиці 2

| | | | |
|---|---|---|--|
| Тестова функція: $f(x_1, x_2) = J_0(x_1^2 + x_2^2) + 0.1 1 - x_1 + 0.1 1 - x_2 $ | | | |
| Точний розв'язок: $f(1, 1.6606) = -0.3356, -\infty \leq x_1, x_2 \leq \infty$ | | | |
| $f(x_1, x_2) = -0.32108$ $x_1 = 0.9074$ $x_2 = 1.7111$ | $f(x_1, x_2) = -0.33482$ $x_1 = 1.6471$ $x_2 = 1.0196$ | $f(x_1, x_2) = -0.33211$ $x_1 = 1.5932$ $x_2 = 1.0964$ | $f(x_1, x_2) = 0.30187$ $x_1 = -0.41483$ $x_2 = 0.40925$ |
| Тестова функція: $f(x_1, x_2) = x_1 \sin(4x_1) + 1.1x_2 \sin(2x_2)$ | | | |
| Точний розв'язок: $f(9.039, 8.668) = -18.5547, 0 \leq x_1, x_2 \leq 10$ | | | |
| $f(x_1, x_2) = -18.5547$ $x_1 = 9.0384;$ $x_2 = 8.6693$ | $f(x_1, x_2) = -18.5275$ $x_1 = 9.0196;$ $x_2 = 8.6667$ | $f(x_1, x_2) = -18.5275$ $x_1 = 9.0196;$ $x_2 = 8.6667$ | $f(x_1, x_2) = -18.220$ $x_1 = 8.5295;$ $x_2 = 8.5057$ |

Було проведено порівняння Continuous genetic algorithm, Binary genetic algorithm з методом кажанів та методом імітації відпалу. Порівняльний аналіз результатів роботи показав, що алгоритми Continuous genetic algorithm, Binary genetic algorithm показали кращі результати для більшості функцій.

Порівнюємо роботу ГА з класичними чисельними методами безумовної багатовимірної мінімізації, які були реалізовані у середовищі Octave. Отримані результати наведено у табл. 3.

Таблиця 3

Результати обчислювальних експериментів

| Назва алгоритму | | | | | | | |
|---|------------------------------------|-------------------------------------|--------------------------------------|---------------------------------------|---|-------------------------------------|---|
| Continuous genetic algorithm | Binary genetic algorithm | Метод координатного спуску | Метод конфігурацій (Хука-Дживса) | Метод найшвидшого градієнтного спуску | Метод спряжених градієнтів (Флетчера-Рівса) | Метод Ньютона | Метод Марквардта |
| Тестова функція: $f(x) = 45x_1^2 - 88x_1x_2 + 45x_2^2 + 102x_1 + 268x_2 - 21$ | | | | | | | |
| Точний розв'язок: $f(-92.9613, -93.9173) = -1.7172e + 04$ | | | | | | | |
| -17170.4132 -92.9613 -93.9173 | 17170.4132 -92.9613 -93.9173 | -1.7172e+04 -92.0335 -92.9661 | -1.7172e+04 -92.0337 -92.9663 | -1.7172e+04 -92.0337 -92.9663 | -1.7172e+04 -92.0334 -92.9660 | -1.7172e+04 -92.0337 -92.9663 | -1.7172e+04 -92.0337 -92.9663 |
| Тестова функція: $f(x) = \sin^2 3\pi x_1 + (x_1 - 1)^2(1 + \sin^2 3\pi x_2) + (x_2 - 1)^2(1 + \sin^2 3\pi x_2)$ | | | | | | | |
| Точний розв'язок: $f(1, 1) = 0, -10 \leq x_1, x_2 \leq 10$ | | | | | | | |
| 1.8415e-05 1.0001 1.0041 | 0.034559 1.0196 1.0196 | 3.5299 2.3183 2.3291 | 0.0051012 1.0075 1.0075 | 0.10987 1.3296 1.0000 | 0.10987 1.3296 1.0000 | 7.9491 -0.97702 1.02298 | 7.9491 -0.97702 1.02298 |
| Тестова функція: $f(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$ | | | | | | | |
| Точний розв'язок: $f(0, 0) = 0, -5 \leq x_1, x_2 \leq 5$ | | | | | | | |
| 8.52e-07 0.00060753 -0.00075773 | 0 0 0 | 0.29864 -1.74755 0.87378 | 0.00062497 -0.012500 -0.012500 | 0.29864 -1.74755 0.87373 | 0.50234 -1.53909 0.58679 | 0.87736 -1.07054 0.92946 | 0.000000001 4035 0.0000049927 2.0000049927 |

Було проведено порівняння Continuous genetic algorithm, Binary genetic algorithm з методом координатного спуску, методом найшвидшого градієнтного спуску, метод спряжених градієнтів (Флетчера-Рівса), методом Ньютона і методом Марквардта. Порівняльний аналіз результатів роботи показав, що алгоритми Continuous genetic algorithm, Binary genetic algorithm показали кращі результати для більшості функцій.

Застосуємо генетичний алгоритм до розв'язання задачі комівояжера на тестовому прикладі з 20 міст. Отримані результати наведено на рис. 1.

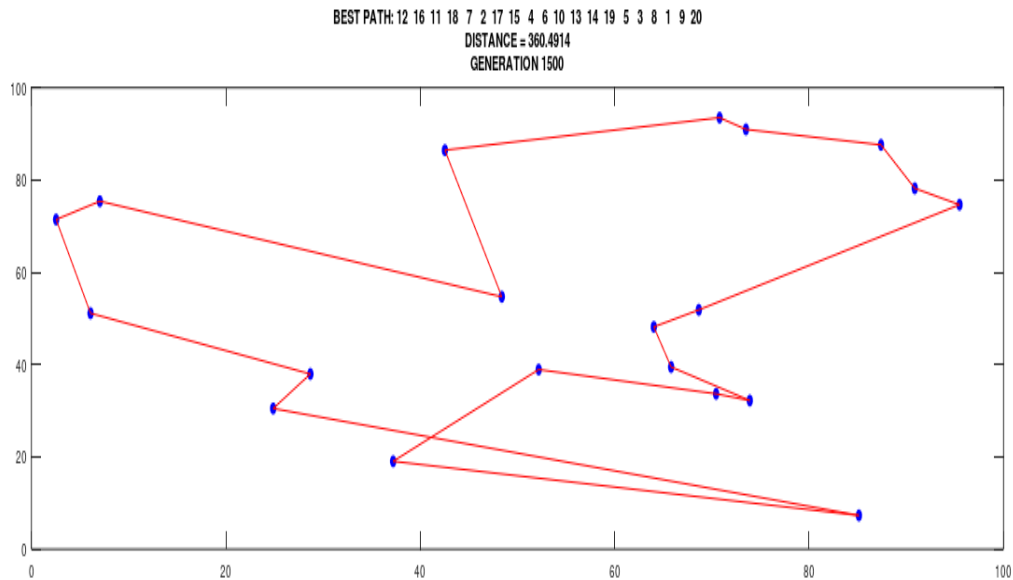


Рис. 1 Маршрут комівояжера на ітерації 1500

Мінімальна відстань, якої вдалося досягти в за допомогою генетичного алгоритму 360.4914. Це доводить ефективність ГА в таких важких проблемах, як задача комівояжера. На прикладі цієї задачі можна спостерігати, як ГА створює розв'язок, не маючи попередніх знань про маршрути подорожі. На відміну від інших евристичних методів, ГА використовує природні правила відбору, кросовера та мутації, щоб зробити обчислення легшими та швидшими. Ці речі роблять його більш цінним, більш ефективним ніж інші евристичні алгоритми.

Висновки

У роботі розглянуто основні поняття і принцип роботи генетичних алгоритмів. Розроблено програми у середовищі GNU Octave, які реалізують Continuous genetic algorithm, Binary genetic algorithm для знаходження мінімуму функцій та розв'язання задачі комівояжера.

Досліджено залежність ефективності даних алгоритмів від розміру популяції, швидкості мутації, частки популяції, що зберігається та максимального числа ітерацій. В результаті обчислювальних експериментів визначено оптимальні значення для наведених параметрів.

Проведено порівняння роботи генетичних алгоритмів з евристичними та класичними методами оптимізації. Серед класичних методів розглянуто метод координатного спуску, метод найшвидшого градієнтного спуску, метод спряжених градієнтів (Флетчера-Рівса), метод Ньютона і метод Марквардта, а серед евристичних – метод кажанів, метод імітації відпалу. Встановлено, що алгоритми Continuous genetic

algorithm та Binary genetic algorithm показали кращі результати для більшості тестових функцій.

Таким чином, доцільно застосовувати розглянуті генетичні алгоритми для задач мінімізації складних функцій.

Список використаної літератури:

1. Панченко, Т. В. Генетические алгоритмы: учеб.-метод. пособие / Т. В. Панченко. – Астрахань: Астрахан. ун-т., 2007. – 87 с.
2. Гладков, Л.А. Генетические алгоритмы / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. – М.: ФИЗМАТЛИТ, 2010. – 368 с.
3. Савин, А. Н. Применение генетических алгоритмов для решения задач оптимизации на параллельных и распределенных вычислительных системах / А. Н. Савин, И. В. Дружинин, А. А. Ерофтиев // Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. – 2013. – Т. 13, вып. 1. – С. 99-109.
4. Івченко, І.Ю. Математичне програмування: Навчальний посібник / І.Ю. Івченко. – К.: Центр учбової літератури, 2007. – 232 с.
5. Goldberg, D.E. Alleles, Loci and the Traveling Salesman Problem / D.E. Goldberg, & R. Lingle // Proceedings of the First International Conference on Genetic Algorithms. – 1985. – P. 154-159.

Bibliography:

1. Panchenko, T.V. (2007). Geneticheskie algoritmy: ucheb.-metod. posobie [Genetic algorithms: the methodical manual]. Astrakhan: Univer. Astrakhan [in Russian].
2. Gladkov, L.A., Kurejchik, V.V., & Kurejchik, V.M. (2010). Geneticheskie algoritmy [Genetic algorithms]. Moskva: FIZMATLIT [in Russian].
3. Savin, A. N., Druzhinin, I. V., Eroftiev, A. A. (2013). Primenenie geneticheskikh algoritmov dlya resheniya zadach optimizacii na parallelnykh i raspredelennykh vychislitelnykh sistemakh [The use of genetic algorithms to solve optimization problems on parallel and distributed computing systems]. *Izv. Sarat. un-ta. Nov. ser. Ser. Matematika. Mekhanika. Informatika*, 13, 1, 99-109 [in Russian].
4. Ivchenko, I. Yu. (2007), Matematichne programuvannya: Navchalnij posibnik [Mathematical Programming: A Tutorial]. Kyiv: Czentr uchbovoyi literaturi [in Ukraine].
5. Goldberg, D.E., & Lingle, R. (1985). Alleles, Loci and the Traveling Salesman Problem. *Proceedings of the First International Conference on Genetic Algorithms*, 154-159.

CHERKAS Dariya,

student, The Bohdan Khmelnytsky National University of Cherkasy

KRASNOSHLYK Natalia

Candidate of Technical Sciences, Associate Professor, The Bohdan Khmelnytsky National University of Cherkasy

RESEARCH OF GENETIC ALGORITHMS OF SOLUTION OF OPTIMIZATION PROBLEMS

Summary. Introduction. Genetic algorithms are one of the most important areas of research in evolutionary modeling. They are often used to solve problems in different areas when traditional methods are not effective enough.

A genetic algorithm is an evolutionary search algorithm used to solve optimization and modeling problems by sequentially selecting, combining, and varying the parameters sought using mechanisms that resemble biological evolution.

The purpose of this article is to implement and investigate genetic algorithms for solving optimization problems.

Results. The basic concepts and principle of operation of genetic algorithms are considered in the work. The formulation of the problem of unconditional optimization and the travelling salesman problem is given. The peculiarities of solving the travelling salesman problem using a genetic algorithm are described.

GNU Octave applications have been developed that implement Continuous genetic algorithm and Binary genetic algorithm for minimization of functions and solve the travelling salesman problem. The dependence of the efficiency of these algorithms on the size of the population, the mutation rate, the proportion of the stored population and the maximum number of iterations are investigated. As a result of computational experiments, the optimal values for the given parameters are determined.

Comparison of genetic algorithms with heuristic and classical optimization methods is compared. The classic methods include coordinate descent method, gradient descent method, conjugate gradient method (Fletcher-Reeves method), Newton method and Marquardt method, and among the heuristic methods, bat algorithm and simulated annealing.

Conclusions. *Genetic algorithms have been used for minimization of functions and solve the traveling salesman problem. The efficiency of genetic algorithms is studied depending on the values of its parameters when finding the global minimum of some test functions. Comparison of genetic algorithms with bat algorithm, simulated annealing and classical optimization methods are compared. Continuous genetic algorithm and Binary genetic algorithm have been found to perform better for most test functions.*

Thus, it is advisable to apply the genetic algorithms under consideration to solve optimization problems.

Keywords: *genetic algorithm, optimization problem, traveling salesman problem.*

*Одержано редакцією 18.07.2019 р.
Прийнято до публікації 09.10.2019 р.*

УДК 519.6

DOI 10.31651/2076-5886-2019-2-43-58

ГОЛОВНЯ Борис Петрович

доктор технических наук, доцент кафедры
прикладной математики и информатики
Черкасского национального университета
имени Богдана Хмельницкого
e-mail: bpgolovnya@gmail.com
ORCID 0000-0002-9242-3937

К ВОПРОСУ О ПРЕПОДАВАНИИ МЕТОДА СОПРЯЖЕННЫХ ГРАДИЕНТОВ

Метод сопряженных градиентов является лучшим из известных итерационных методов решения систем линейных уравнения с симметричной матрицей. На его основе разработано много высокоскоростных методов решения произвольных систем алгебраических уравнений. В то же время, традиционное объяснение принципов работы этого метода очень сложно. Как показывает практика, студенты плохо понимают его. В работе предложено интуитивно понятное объяснение принципов работы метода сопряженных градиентов.

Ключевые слова: *системы линейных алгебраических уравнений, итерационные методы решения СЛАУ, метод сопряженных градиентов.*

Постановка задачи

Метод сопряженных градиентов является одним из самых быстросходящихся итерационных методов для решения систем линейных алгебраических уравнений с симметричной матрицей. Но, при всей простоте программной реализации метода, его математические основы достаточно сложны (см. например [1],[2],[3]). Очень часто при его пояснении используются такие понятия, как проекционные методы, ортогонализация по Арнольди и Ланцошу, скорейший спуск, пространства Крылова, сопряженные направления и многое другое. Все это делает метод непонятным на интуитивном уровне. По этой причине метод сопряженных градиентов очень непросто объяснять студентам. В работе приведено несколько отличающееся от традиционных обоснование метода. Здесь используются только понятие ортогональности, метод Грамма-Шмидта и понятие А-скалярного произведения. Понятие проекции используется только для улучшения уже построенного и понятного метода. Такой подход оказывается интуитивно понятным и гораздо более простым методически чем

традиционный подход. В итоге в данном описании метод усваивается студентами гораздо легче.

Цель статьи

Задача работы – предложить простое и понятное обоснование метода сопряженных градиентов

Методы решения

В основе метода сопряженных градиентов лежит понятие ортогонализации. Рассмотрим его подробнее

Мы имеем векторное пространство со скалярным произведением.

Согласно определению в n -мерном линейном пространстве задано скалярное умножение векторов, если любой паре векторов a и b из этого пространства поставлено в соответствие действительное число, обозначаемое (a, b) и называемое скалярным произведением, причем выполняются следующие условия:

1. $(a, b) = (b, a)$.
2. $(a + b, c) = (a, c) + (b, c)$.
3. $(\alpha a, b) = \alpha(a, b)$.
4. $a \neq 0 \rightarrow (a, a) > 0$.

Методика нахождения этого числа, т.е. скалярного произведения, в определении не оговаривается. Общеизвестный вариант скалярного произведения - $(a, b) = |a||b|\cos\alpha$. Здесь α - угол между векторами a и b . Нетрудно убедиться, что оно удовлетворяет всем вышеперечисленным требованиям. По определению вектора считаются ортогональными, если их скалярное произведение равно 0.

Имеем задачу $Ax = b$. Разложим матрицу A на множители $A = C \times D$ так, чтобы одна из матриц легко обращалась, а система с другой матрицей легко решалась. Пусть матрица C легко обращается, а система с матрицей D легко решается. Перепишем систему как $(C \times D)x = C(Dx) = b$ и введем переменную $y = Dx$. Тогда исходная система распадается на две легко решаемых системы $Cy = b$ и $Dx = y$. Какая именно из матриц легко обращается роли не играет.

Классическим примером легко обращающихся матриц служат унитарные матрицы, т.е. матрицы вида $A \times A^* = A^* \times A = E$. Если элементами матрицы являются действительные числа, то выражение упрощается $A \times A^T = A^T \times A = E$, а матрица называется ортогональной. Рассмотрим, что значит требование ортогональности. Пусть матрица имеет порядок n и состоит из n строк. Мы трактуем каждую строку как n -мерный вектор. Тогда требование ортогональности выглядит как

$$A \times A^T = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix}^T = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix} \begin{pmatrix} a_1^T & a_2^T & \dots & a_n^T \end{pmatrix} = (a_i a_j^T) = E.$$

Отсюда мы сразу имеем $(a_i a_j^T) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$. Но тогда можно сказать, что строки

матрицы представляют собой ортонормированную систему векторов. Транспонировав это выражение, получаем, что столбцы матрицы также представляют собой ортонормированную систему.

Таким образом, одна из стоящих задач – преобразовать систему векторов, образующих столбцы (строки) матрицы в ортонормированную систему. Наиболее известный метод решения этой задачи это процесс Грама-Шмидта.

Процесс Грама-Шмидта

Имеем векторное пространство, натянутое на векторы a_1, a_2, \dots, a_n . Построим в нем ортогональный базис u_1, u_2, \dots, u_n . Процедура выглядит следующим образом. В качестве первого вектора нового базиса выбираем первый вектор исходного базиса.

$$u_1 = a_1.$$

Основная идея построения всех последующих векторов состоит в следующем. Векторы $a_1 = u_1$ и a_2 – элементы базиса и не могут быть параллельны. Поэтому на них можно натянуть плоскость. Любой вектор этой плоскости можно выразить как

$$u_2 = \alpha u_1 + \beta a_2. \tag{1}$$

Будем искать в этой плоскости вектор ортогональный u_1 . Очевидно, что у этого вектора множитель β не может равняться нулю, т.к. в этом случае мы получим параллельный u_1 вектор. В то же время нас не интересует длина ортогонального вектора, а только его направление. Поэтому общее выражение для этого вектора делим на неравное нулю число β и ищем его в виде (2).

$$u_2 = a_2 - \lambda_{2,1} u_1. \tag{2}$$

Минус перед $\lambda_{2,1}$ взят для упрощения дальнейших формул. Потребуем ортогональности этого вектора вектору u_1 . Для этого умножим (2) скалярно на u_1 и приравняем результат нулю. Имеем $(u_2 u_1) = (a_2 u_1) - \lambda_{2,1} (u_1 u_1) = 0$. Отсюда $\lambda_{2,1} = (a_2 u_1) / (u_1 u_1)$. Понятно, что плоскость, натянутая на векторы u_1 и u_2 , совпадает с плоскостью, натянутой на a_1 и a_2

Построим третий вектор. Векторы a_1, a_2 и a_3 принадлежат к базису, поэтому вектор a_3 не лежит в плоскости, натянутой на векторы u_1 и u_2 . В связи с этим вектор u_3 строим в виде $u_3 = a_3 - \lambda_{3,2} u_2 - \lambda_{3,1} u_1$. Из требования ортогональности вектора u_3 векторам u_1 и u_2 получаем $\lambda_{3,1} = (a_3 u_1) / (u_1 u_1)$, $\lambda_{3,2} = (a_3 u_2) / (u_2 u_2)$

Окончательное описание ортогонализации по Граму-Шмидту выглядит следующим образом

$$\begin{aligned} u_1 &= a_1, \\ u_2 &= a_2 - \lambda_{2,1} u_1, \\ &\dots\dots\dots \\ u_k &= a_k - \sum_{j=1}^{k-1} \lambda_{k,j} u_j. \end{aligned} \tag{3}$$

Проводя ортогонализацию по методике Грама-Шмидта умножим (3) скалярно на u_i ($i < k$). Получаем

$$(u_k u_i) = (a_k u_i) - \sum_{j=1}^{k-1} \lambda_{k,j} (u_j u_i). \tag{4}$$

Из требования ортогональности $(u_k u_i) = 0$ при $i < k$ получаем

$$\lambda_{k,i} = \frac{(a_k u_i)}{(u_i u_i)}. \quad (5)$$

Элементарная модификация алгоритма позволяет получить ортонормированный базис. Для этого мы будем считать, что выражение (3) рассчитывает не окончательный, а промежуточный вариант вектора u_k , т.е. операторы (3) и (4) перепишем как

$$u_{k-temp} = a_k - \sum_{j=1}^{k-1} \lambda_{k,j} u_j.$$

$$(u_{k-temp} u_i) = (a_k u_i) - \sum_{j=1}^{k-1} \lambda_{k,j} (u_j u_i).$$

При расчете $\lambda_{k,i}$ считаем, что все полученные ранее векторы уже нормированы, поэтому в знаменателе стоит единица

$$\lambda_{k,i} = (a_k u_i).$$

В заключение находим новый базисный вектор как

$$u_k = u_{k-temp} / \|u_{k-temp}\|$$

Задача построения A -ортogonalного базиса

Очень удобные методы решения систем линейных алгебраических уравнений можно построить используя понятие A -ортogonalности.

Пусть мы имеем скалярное произведение и симметричную, положительно определенную матрицу A . Напомним, что согласно одному из эквивалентных определений матрица называется положительно определенной если для любого ненулевого вектора a $(Aa, a) > 0$. Введем операцию над векторами вида $(a, b)_A = (Aa, b)$. Очевидно, что эта операция удовлетворяет всем требованиям, предъявляемым к скалярному произведению и может быть названа как A -скалярное произведение. Теперь можно ввести понятие A -нормы $\|x\|_A = \sqrt{(x, x)_A}$. Как известно ортogonalными векторами называются векторы с нулевым скалярным произведением. Отсюда мы можем ввести понятие A -ортogonalности. Часто A -ортogonalные векторы называют сопряженными.

Рассмотрим, что это нам дает. Решается система уравнений $Ax = b$ с симметричной, положительно определенной матрицей A . В пространстве \mathbb{R}^n , т.е. в пространстве, которому принадлежит решение системы, построен A -ортogonalный базис $R^n = span\{u_1, u_2, \dots, u_n\}$. Разложим решение по этому базису $x = \sum c_i u_i$ и подставим разложение в систему $\sum c_i A u_i = b$. Умножим результат скалярно на u_k - $\sum c_i (A u_i, u_k) = (b, u_k)$. Из требования A -ортogonalности имеем $c_k (A u_k, u_k) = (b, u_k)$, где

$$c_k = \frac{(b, u_k)}{(A u_k, u_k)}. \quad (6)$$

Ситуацию можно интерпретировать так. После всех этих преобразований система уравнений приобрела диагональный вид

$$\begin{pmatrix} (A u_1, u_1) & & & \\ & (A u_2, u_2) & & \\ & & \dots & \\ & & & (A u_n, u_n) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{pmatrix} = \begin{pmatrix} (b, u_1) \\ (b, u_2) \\ \dots \\ (b, u_n) \end{pmatrix}.$$

Мы можем находить векторы базиса последовательно, компонента решения, связанная с очередным найденным вектором не влияет на последующие выкладки. Само решение исходной системы мы также находим в процессе расчетов как

$$x_{k+1} = x_k + c_k u_k, \quad (7)$$

где x_{k+1} – очередное приближение к решению. Замечательно, что хотя процесс очень похож на итерационный, он приводит к точному решению ровно за n шагов.

A-ортогонализация на основе процесса Грама-Шмидта

Построим алгоритм перехода от базиса (r_1, r_2, \dots, r_n) к A-ортогональному базису (u_1, u_2, \dots, u_n) . Процедура ортогонализации по Граму-Шмидту выглядит следующим образом

$$\begin{aligned} u_1 &= r_1, \\ u_2 &= r_2 - \lambda_{1,2} u_1, \\ u_3 &= r_3 - \lambda_{2,3} u_2 - \lambda_{1,3} u_1, \\ u_4 &= r_4 - \lambda_{3,4} u_3 - \lambda_{2,4} u_2 - \lambda_{1,4} u_1, \\ &\dots\dots\dots \\ u_j &= r_j - \sum_{i=1}^{j-1} \lambda_{i,j} u_i. \end{aligned} \quad (8)$$

Умножим (8) A-скалярно на u_k $k < j$. Получаем $(Au_j, u_k) = (Ar_j, u_k) - \sum_{i=1}^{j-1} \lambda_{i,j} (Au_i, u_k)$. Требование A-ортогональности значит, что $(Au_i, u_j) = (Au_j, u_i) = 0$ при $j \neq i$, откуда

$$(Ar_j, u_k) - \sum_{i=1}^{j-1} \lambda_{i,j} (Au_i, u_k) = 0$$

Но $(Au_i, u_k) = 0$ при $i \neq k$. В итоге в сумме остается единственное слагаемое

$$(Ar_j, u_i) - \lambda_{i,j} (Au_i, u_i) = 0. \quad (9)$$

Коэффициент $\lambda_{i,j}$ находим из выражения (9)

$$\lambda_{i,j} = \frac{(Ar_j, u_i)}{(Au_i, u_i)} = \frac{(Au_i, r_j)}{(Au_i, u_i)}.$$

Последний переход возможен, т.к. матрица A по условию симметрична.

Мы построили процедуру A-ортогонализации на основе классического процесса Грама-Шмидта

Оформим эти действия в виде алгоритма на псевдокоде. Представим его в двух вариантах. Первый вариант полностью соответствует вышеописанным действиям

1. $u_1 = r_1$
2. for $i=2:n$
3. $u_i = r_i$
4. for $j=1:i-1$
5. $\lambda_{i,j} = (Au_i, r_j) / (Au_i, u_i)$
6. $u_i = u_i - \lambda_{i,j} u_j$.

7. end
8. end

Второй вариант строит A-ортономмированную систему векторов

1. $u_1 = r_1 / \|r_1\|$
2. for $i=2:n$
3. $u_{temp} = r_i$
4. for $j=1:i-1$
5. $\lambda_{i,j} = (Au_i, r_j)$
6. $u_{temp} = u_{temp} - \lambda_{i,j} u_j$
7. end
8. $u_i = u_{temp} / \sqrt{(Au_{temp}, u_{temp})}$
9. end

Простой метод решения систем уравнений основанный на A-ортогонализации.

Теперь мы можем построить простой метод решения систем уравнений основанный на A-ортогонализации. Собственно метод был описан немного выше. Нам только нужно дополнить его процедурой построения A-ортогонального базиса. Здесь, как и ранее, (r_1, r_2, \dots, r_n) - уже имеющийся базис, (u_1, u_2, \dots, u_n) - A-ортогональный базис. Алгоритм используется в виде без нормировки. Его можно оформить следующим образом

1. $x_1 = 0$
2. $u_1 = r_1$
3. for $i=1:n-1$
4. $c_i = (b, u_i) / (Au_i, u_i)$
5. $x_{i+1} = x_i + c_i u_i$
6. $u_i = r_i$
7. for $j=1:i-1$
8. $\lambda_{i,j} = (Au_i, r_j) / (Au_i, u_i)$
9. $u_i = u_i - \lambda_{i,j} u_j$
10. end
11. end

Алгоритм вполне работоспособен. Недостатки его хорошо заметны. Он требует очень большой оперативной памяти. В частности, мы должны помнить исходный базис. Для этого необходима память равная полной матрице исходной задачи. Кроме того, память необходима для запоминания A-ортогонального базиса. А это еще один такой же объем памяти. Эти два факта сводят на нет все достоинства метода.

Метод сопряженных градиентов по идеологии полностью совпадает с описанным методом. Но в нем, путем удачного выбора исходного базиса, удалось резко сократить потребности в оперативной памяти. Чтобы разобраться в способе построения этого базиса нам понадобится понятие проектирования (проекции).

Проектирование

Проектирование очень широко применяется в численных методах. Например. Приближенные соотношения для производных получаются отбрасыванием хвоста ряда Тейлора. Но использование только части базиса в разложении произвольного вектора, функции и т.д. и есть проектирование. Отсюда следует, что все конечно-разностные методы, а заодно и методы решения задачи Коши по смыслу являются проекционными. Конечно-элементные методы сразу называют себя проекционными. Очевидно, что все методы численного интегрирования попадают сюда же. Т.е. прием упрощения задачи путем отбрасывания части ее компонент очень популярен в вычислительной математике. Другое дело, что при решении систем уравнений мы будем использовать его несколько иначе.

Что такое проекция

Рассмотрим проекцию вектора в двумерном случае (рис. 1). В этом случае вектор может проектироваться только на прямую, т.е. на направление. Саму проекцию можно рассматривать как тень от вектора, отбрасываемую на это направление. Естественно, что в этом случае нам надо знать направление на источник света.

Например. Мы проектируем вектор \overline{ab} на направление \overline{acd} . В случае если источник света находится на линии \overline{cb} , то результатом будет вектор \overline{ac} , если же на направлении \overline{db} - то вектор \overline{ad} . Для нахождения этой проекции достаточно разложить исходный вектор по базису, состоящему из вектора – направления проекции и вектора – направления источника света. Отбросив компоненту, связанную с источником света, получаем решение. В дальнейшем вместо выражения «вектор направления на источник» будем использовать выражение «параллельно вектору». Естественно, что этот вектор и задает направление на источник. Для простоты выкладок мы будем описывать направление, параллельно которому идет проектирование, ортогональной этому направлению прямой. Тогда отбрасывание компоненты сводится к скалярному умножению разложения на вектор, параллельный этой прямой.

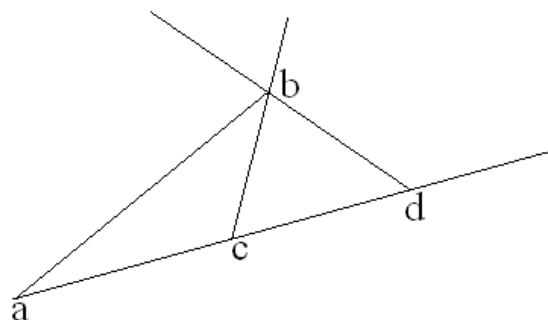


Рис. 1.

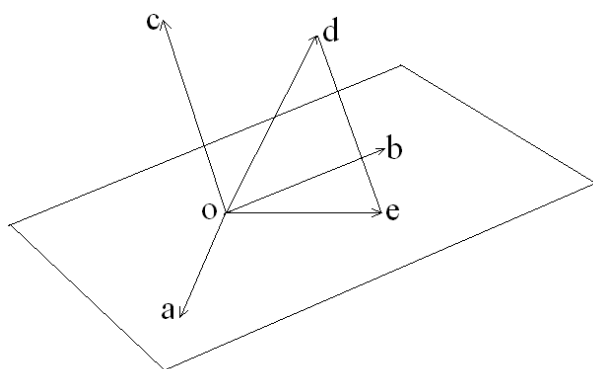


Рис.2.

Расчет выглядит следующим образом. Построим проекцию вектора u на направление заданное вектором v_1 , параллельно вектору w , причем вектор v_2 ортогонален вектору w . Разложим вектор u векторам v_1 и w - $u = c_1 v_1 + c_2 w$. Вектор $c_2 w = u - c_1 v_1$ ортогонален v_2 . Запишем это условие - $c_2 w = (u v_2) - c_1 (v_1 v_2)$. Отсюда $c_1 = \frac{(u v_2)}{(v_1 v_2)}$. Искомая проекция имеет вид

$P(u) = \frac{(u v_2)}{(v_1 v_2)} v_1$. Наибольшее распространение получили ортогональные проекции, т.е. проекции, когда вектор, параллельно которому строится проекция, ортогонален

направлению проекции. В этом случае можно выбрать $v_2=v_1$ и формула упрощается $P(u) = \frac{(uv_1)}{(v_1v_1)}v_1$. Если вектор v_1 нормирован, то имеем $P(u) = (uv_1)v_1$.

И еще. Отброшенную компоненту разложения, т.е. вектор c_2w , вполне можно рассматривать как погрешность замены исходного вектора его проекцией. Будем измерять эту погрешность длиной отброшенного вектора. Очевидно, что расстояние от точки до прямой будет кратчайшим, если измерять его по перпендикуляру, опущенному из точки на прямую. Отсюда имеем, что в случае ортогональной проекции эта погрешность минимизируется.

Теперь рассмотрим проекцию вектора на плоскость параллельно заданному вектору. Мы будем проектировать вектор $v = \overline{od}$ параллельно направлению $w = \overline{oc}$. Плоскость на которую строится проекция задается двумя векторами - $v_1 = \overline{oa}$ и $v_2 = \overline{ob}$. Разложим вектор v по базису из векторов v_1, v_2 и w - $v = c_1v_1 + c_2v_2 + c_3w$. Как и в прошлый раз, будем задавать направление проекции ортогональным этому направлению подмножеством трехмерного пространства. Ортогональность мы будем понимать следующим образом – любой вектор, принадлежащий этому подмножеству ортогонален вектору w . Очевидно, что в нашем случае это будет плоскость. Будем считать, что эта плоскость натянута на векторы w_1 и w_2 , причем $(ww_1) = (ww_2) = 0$. Тогда, умножая соотношение $c_3w = v - c_1v_1 - c_2v_2$ на векторы w_1 и w_2 , получим систему уравнений для расчета коэффициентов c_1 и c_2

$$\begin{cases} c_1(v_1, w_1) + c_2(v_2, w_1) = (v, w_1), \\ c_1(v_1, w_2) + c_2(v_2, w_2) = (v, w_2). \end{cases}$$

Итоговая проекция выглядит как $P(v) = c_1v_1 + c_2v_2$.

Несложными преобразованиями получаем решение

$$\begin{aligned} & \begin{pmatrix} (v_1, w_1) & (v_2, w_1) \\ (v_1, w_2) & (v_2, w_2) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} (v, w_1) \\ (v, w_2) \end{pmatrix} \\ & \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} (v_1, w_1) & (v_2, w_1) \\ (v_1, w_2) & (v_2, w_2) \end{pmatrix}^{-1} \begin{pmatrix} (v, w_1) \\ (v, w_2) \end{pmatrix} = \begin{pmatrix} (v_1, w_1) & (v_2, w_1) \\ (v_1, w_2) & (v_2, w_2) \end{pmatrix}^{-1} \begin{pmatrix} (w_1, v) \\ (w_2, v) \end{pmatrix} \\ & P(v) = c_1v_1 + c_2v_2 = \begin{pmatrix} v_1 & v_2 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} v_1 & v_2 \end{pmatrix} \begin{pmatrix} (v_1, w_1) & (v_2, w_1) \\ (v_1, w_2) & (v_2, w_2) \end{pmatrix}^{-1} \begin{pmatrix} (w_1, v) \\ (w_2, v) \end{pmatrix} \end{aligned}$$

Решение заметно упрощается, если проектирование производится ортогонально плоскости и векторы v_1 и v_2 ортогональны. В этом случае можно выбрать $w_1 = v_1$ и $w_2 = v_2$. Тогда $c_1 = \frac{(v, v_1)}{(v_1, v_1)}$ и $c_2 = \frac{(v, v_2)}{(v_2, v_2)}$. Если же векторы ортонормированны, то соотношения получаются следующими $c_1 = (v, v_1)$ и $c_2 = (v, v_2)$, или $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} (v, v_1) \\ (v, v_2) \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, v$, а проекция вычисляется как

$$P(v) = c_1 v_1 + c_2 v_2 = (v_1 \quad v_2) \begin{pmatrix} (v, v_1) \\ (v, v_2) \end{pmatrix} = (v_1 \quad v_2) \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, v.$$

Решение может трактоваться следующим образом. Сначала мы находим коэффициенты разложения исходного вектора по базисным векторам v_1 и v_2 , а затем умножаем эти коэффициенты на базисные вектора и находим проекцию.

При переходе к произвольному n -мерному пространству технология решения задачи не меняется. Нужно только окончательно определить, что значит одно подпространство ортогонально другому. Пусть (u_1, u_2, \dots, u_n) ортогональный базис в пространстве R^n . Выделим в базисе (u_1, u_2, \dots, u_n) две непересекающихся части (u_1, u_2, \dots, u_k) и $(u_{k+1}, u_{k+2}, \dots, u_m) = (v_1, v_2, \dots, v_l)$, $m = r + l \leq n$. Можно сказать, что мы выделили в исходном пространстве R^n два подпространства $R^k = \text{span}\{u_1, u_2, \dots, u_k\}$ и $R^l = \text{span}\{v_1, v_2, \dots, v_l\}$. Здесь $\text{span}\{u_1, u_2, \dots, u_k\}$ - линейная оболочка векторов u_1, u_2, \dots, u_k . Так как каждый вектор базиса (u_1, u_2, \dots, u_k) ортогонален каждому вектору базиса (v_1, v_2, \dots, v_l) , то любой вектор подпространства $R^k = \text{span}\{u_1, u_2, \dots, u_k\}$ ортогонален любому вектору подпространства $R^l = \text{span}\{v_1, v_2, \dots, v_l\}$. В этом случае мы считаем подпространства ортогональными. Определение несколько отличается от бытового понимания ортогональности. В частности, в трехмерном пространстве не может быть двух ортогональных плоскостей. В самом деле, если плоскости пересекаются, то на обеих плоскостях можно найти совпадающие векторы, лежащие на их пересечении.

Проектирование систем уравнений

Рассмотрим проектирование системы уравнений в пространстве R^n на некое подпространство $K = R^k \subset R^n$. Под проекцией будет пониматься решение следующей задачи. Решается система уравнений $A\bar{x} = b$, где A - невырожденная матрица $n \times n$. Пусть x проекция точного решения \bar{x} этой системы на подпространство K . Требуется построить систему, решением которой будет x . Подпространство, на которое производится проекция, и подпространство, ортогонально которому производится проекция, заданы.

Пусть $(v_1 \quad v_2 \quad \dots \quad v_k)$ - базис в K . Если x искомая проекция, то она принадлежит K и может быть разложена по базису $(v_1 \quad v_2 \quad \dots \quad v_k)$ -

$$x = \sum_{i=1}^k x_i v_i = (v_1 \quad v_2 \quad \dots \quad v_k) \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{pmatrix}.$$

Здесь x_i - коэффициенты разложения, $(v_1 \quad v_2 \quad \dots \quad v_k)$ - матрица, состоящая из векторов-столбцов базиса. Подставим это выражение в исходную систему. Обозначая произведение матрицы A на $(v_1 \quad v_2 \quad \dots \quad v_k)$ как $A_1 = (A(v_1 \quad v_2 \quad \dots \quad v_k))$, получаем

$$A \begin{pmatrix} v_1 & v_2 & \dots & v_k \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{pmatrix} = \begin{pmatrix} A(v_1 & v_2 & \dots & v_k) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{pmatrix} = A_1 \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{pmatrix} = b.$$

В итоге мы перебрались в подпространство, но получили матрицу размерности $k \times n$. Полученную задачу решать нельзя. Причины очевидны: проектирование не только переход к новому базису, а и отбрасывание «лишних» деталей. Способов отбрасывания существует много. Для более детального понимания, что надо делать с системой дальше, рассмотрим простой пример.

Решается система с двумя неизвестными

$$\{ Ax = b \} = \begin{cases} a_1 x = b_1 \\ a_2 x = b_2 \end{cases}.$$

Очевидно, что задача имеет точное решение если и только если вектор b коллинеарен вектору a , или вектор b находится в подпространстве, натянутом на вектор a . В общем случае задача не имеет решения. Будем искать его по методу наименьших квадратов. Т.е. будем искать такое значение x чтобы выражение

$$(b_1 - a_1 x)^2 + (b_2 - a_2 x)^2 = (b - ax)^T (b - ax) = b^T b - 2a^T b x + a^T a x^2$$

приобретало минимальное значение. Дифференцируя по x и приравнявая нулю производную, получаем

$$a^T b - a^T a x = a^T (b - ax) = 0, \quad (10)$$

откуда следует $x = \frac{a^T b}{a^T a} = \frac{(a, b)}{(a, a)}$. Решение совпадает со случаем проектирования вектора

b на вектор a , т.е. мы нашли ближайшую к точке b точку на направлении, заданном вектором a . Можно сказать, что мы нашли ближайшую к точке b точку в подпространстве, заданном вектором a . Отметим некоторые особенности полученного решения. Во-первых, мы минимизировали норму $\|b - ax\|$. Во-вторых, вектор невязки $b - ax$ в этом случае получается ортогональным подпространству, заданному вектором a коэффициентов исходной системы

Вернемся к исходной задаче. Мы проектировали систему на подпространство L с базисом $(v_1 \ v_2 \ \dots \ v_k)$. Согласно только что сказанному, для получения ортогонально проекции нам достаточно умножить слева соотношение (10) на транспонированную матрицу, составленную из базисных векторов.

$$\begin{pmatrix} v_1 \\ v_2 \\ \dots \\ v_k \end{pmatrix} \begin{pmatrix} b - A(v_1 \ v_2 \ \dots \ v_k) \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{pmatrix} \end{pmatrix} = 0. \quad (11)$$

Чтобы избежать громоздкости записи обозначим матрицу из базисных векторов как $V = (v_1 \ v_2 \ \dots \ v_k)$ и вектор коэффициентов разложения проекции x по этому базису как $x_V = (x_1 \ x_2 \ \dots \ x_k)^T$ и перепишем выражение (11) как

$$\{V^T(b - AVx_V) = 0\} = \{V^T AVx_V = V^T b\}. \quad (12)$$

Для повышения общности можно потребовать ортогональности невязки не подпространству K , а какому-либо другому подпространству L такой же размерности. Тогда соотношение (12) перепишется как

$$\{W^T(b - AVx_V) = 0\} = \{W^T AVx_V = W^T b\}, \quad (13)$$

где W – матрица, составленная из векторов-столбцов подпространства L . Сокращать W^T нельзя, т.к. сокращение эквивалентно умножению на обратную матрицу. Но матрица W^T не квадратная и обратной не имеет.

Напомним, что вектор проекции x здесь разложен по базису V , т.е. $x = Vx_V$, откуда $b - AVx_V = b - Ax = r$ – невязка исходной системы уравнений при подстановке в нее вектора проекции. Отсюда выражение (13) можно переписать как $W^T r = 0$.

С точки зрения итерационного расчета задачу можно сформулировать следующим образом. Решается система уравнений $A\bar{x} = b$. Имеем некоторое приближение x^k к точному решению \bar{x} . Требуется дополнить это приближение до проекции, т.е. найти такую поправку $\delta^k \in K$ к приближению, чтобы вектор невязки $b - A(x^k + \delta^k)$ стал ортогонален подпространству L . Тогда можно записать

$$W^T(b - A(x^k + \delta^k)) = W^T(b - AV(x_V^k + \delta_V^k)) = W^T(b - AVx_V^k) - W^T AV\delta_V^k = 0.$$

и

$$\{W^T AV\delta_V^k = W^T(b - AVx_V^k)\} = \{\delta_V^k = (W^T AV)^{-1} W^T(b - AVx_V^k)\} = \{\delta_V^k = (W^T AV)^{-1} W^T r^k\}.$$

Окончательно получаем

$$\delta^k = V\delta_V^k = V(W^T AV)^{-1} W^T r^k$$

и

$$x^{k+1} = x^k + \delta^k = x^k + V(W^T AV)^{-1} W^T r^k.$$

Здесь r^k – вектор невязки системы при подстановке в нее x^k . Отметим, что так как матрицы W^T и V не квадратные и обратных не имеют, то выражение $(W^T AV)^{-1} = V^{-1} A^{-1} (W^T)^{-1}$ неверно. В то же время, если удачно выбрать матрицы W и V , то вычисление $(W^T AV)^{-1}$ может оказаться очень несложным.

В завершение отметим несколько полезных соотношений. Во первых

$$r^{k+1} = b - A(x^k + \delta^k) = b - Ax^k - A\delta^k = r^k - A\delta^k.$$

Отсюда получаем

$$r^{k+1} - r^k = -A\delta^k = -A(x^{k+1} - x^k)$$

и

$$W^T(r^k - A\delta^k) = 0.$$

Метод сопряженных градиентов

Метод сопряженных градиентов вполне соответствует предложенному методу. Но если в этом методе в качестве исходного использовался некий уже имеющийся базис, то в методе сопряженных градиентов используется базис из векторов-невязок системы, рассчитываемых на каждом шаге. Рассмотрим, что это нам дает. Для этого вспомним некоторые свойства ортогонального проектирования систем уравнений.

В пространстве решается система уравнений $A\bar{x} = b$. Задано некоторое подпространство $K^k = R^k \subset R^n$. Имеем некоторое приближение $x_k \in K^k$ к точному решению \bar{x} . Дополнить это приближение до проекции, значит найти такую поправку $\delta_k \in K^k$ к приближению, чтобы вектор невязки $r^{k+1} = b - A(x^k + \delta^k) = b - Ax^{k+1}$ стал ортогонален подпространству K^k . Эта ортогональность эквивалентна минимальности нормы невязки.

В то же время $r_k \in K^k$ и из ортогональности $r_{k+1} \in K^{k+1}$ подпространству K^k следует, что $r_{k+1} \perp r_k$. Рассуждая по индукции имеем $r_1 \perp r_2 \perp r_3 \perp \dots \perp r_n$, т.е. невязки образуют ортогональный базис в R^n , причем этот базис находится по ходу вычислений. Но тогда этот базис мы можем преобразовывать в А-ортогональный базис, который позволяет нам находить новые поправки к вектору неизвестных. Алгоритм замкнулся.

Учитывая, что

$$r_{j+1} = b - A(x_j + \delta_j) = b - Ax_j - A\delta_j = r_j - A\delta_j.$$

его можно записать следующим образом.

1. x_1 - выбрать
2. $r_1 = b - Ax_1$
3. $u_1 = r_1$
4. For $j=1:n$ do
4. $c_j = (b, u_j) / (Au_j, u_j)$
6. $x_{j+1} = x_j + c_j u_j$
7. $r_{j+1} = r_j - c_j Au_j$
8. For $i=1:j$ do
9. $\lambda_{i,j+1} = (Au_i, r_{j+1}) / (Au_i, u_i)$
10. $u_{j+1} = r_{j+1} - \sum_{i=1}^j \lambda_{i,j+1} u_i$.
11. End

Ортогональность базиса r_1, r_2, \dots, r_n и методика его построения вносит в процедуру расчета заметные упрощения. В самом деле. Рассмотрим сумму в 9-ой строке алгоритма. Для этого оператор в 7-ой строке запишем в виде $r_{i+1} = r_i - c_i Au_i$ и умножим скалярно на r_{j+1} . Получаем

$$c_i(Au_i, r_{j+1}) = (r_i, r_{j+1}) - (r_{i+1}, r_{j+1}) \quad (13)$$

В силу ортогональности r_k правая часть (13) не равна нулю только если $(r_i, r_{j+1}) \neq 0$ или $(r_{i+1}, r_{j+1}) \neq 0$, т.е. $i = j+1$ или $i+1 = j+1$. Но из оператора цикла в 8-ой строке следует, что $i \leq j$. Но тогда получаем, что в сумме в 9-ой строке осталось единственное слагаемое. В результате оператор 10-ой строки приобретает вид

$$u_{j+1} = r_{j+1} - \lambda_{j,j+1} u_j = u_{j+1} = r_{j+1} - \frac{(Au_j, r_{j+1})}{(Au_j, u_j)} u_j.$$

Теперь алгоритм метода можно переписать как

1. x_1 - выбрать
2. $r_1 = b - Ax_1$
3. $u_1 = r_1$
4. For $j=1:n$ do
4. $c_j = (b, u_j)/(Au_j, u_j)$
6. $x_{j+1} = x_j + c_j u_j$
7. $r_{j+1} = r_j - c_j Au_j$
8. $\beta_j = (Au_j, r_{j+1})/(Au_j, u_j)$
9. $u_{j+1} = r_{j+1} - \beta_j u_j$
10. End

Чтобы получить традиционную форму алгоритма сделаем небольшие преобразования. Во первых. Умножив равенство $r_{j+1} = r_j - c_j Au_j$ на u_j получаем $c_j (Au_j, u_j) = (r_j, u_j) - (r_{j+1}, u_j)$. Но из смысла процесса Грамма-Шмидта следует, что $\forall j \text{ span}\{r_1, r_2, \dots, r_k\} = \text{span}\{u_1, u_2, \dots, u_k\}$, а из ортогональности базиса r_1, r_2, \dots, r_{j+1} следует, что $r_{j+1} \perp \text{span}\{r_1, r_2, \dots, r_j\}$ откуда $r_{j+1} \perp \text{span}\{u_1, u_2, \dots, u_j\}$. Следовательно $r_{j+1} \perp u_j$ и $c_j = (r_j, u_j)/(Au_j, u_j)$. В то же время $u_j = r_j - \beta_{j-1} u_{j-1}$, откуда $(r_j, u_j) = (r_j, r_j) - \beta_{j-1} (r_j, u_{j-1})$. Так как r_j ортогонально u_{j-1} , то $(r_j, u_{j-1}) = 0$ и $(r_j, u_j) = (r_j, r_j)$. В результате получаем стандартное выражение $c_j = (r_j, r_j)/(Au_j, u_j)$.

Во вторых. Из $r_{j+1} = r_j - c_j Au_j$ имеем $Au_j = (r_j - r_{j+1})/c_j$ откуда $(r_{j+1}, Au_j) = (r_{j+1}, r_j - r_{j+1})/c_j = -(r_{j+1}, r_{j+1})/c_j$. Сопоставляя это выражение с выражением для c_j и формулой $\beta_j = (r_{j+1}, Au_j)/(Au_j, u_j)$ получаем $\beta_j = -(r_{j+1}, r_{j+1})/(r_j, r_j)$.

Окончательно

1. x_1 - выбрать
2. $r_1 = b - Ax_1$
3. $u_1 = r_1$
4. For $j=1:n$ do
4. $c_j = (r_j, r_j)/(Au_j, u_j)$
6. $x_{j+1} = x_j + c_j u_j$

7. $r_{j+1} = r_j - c_j Au_j$
8. $\beta_j = (r_{j+1}, r_{j+1}) / (r_j, r_j)$
9. $u_{j+1} = r_{j+1} + \beta_j u_j$
10. *End*

Отметим, что знак у β заменен на плюс.

Достоинства и недостатки метода сопряженных градиентов.

Метод очень экономичен. Он требует хранения в памяти только трех векторов. При грамотном написании программы на один шаг цикла используется всего одна операция умножения матрицы на вектор, два скалярных умножения векторов и три операции сложения векторов. При этом программу можно организовать так, что матрица явно в вычислениях участвовать не будет, а будет доступна лишь через программу, вычисляющую произведение матрицы на вектор. Этот факт делает метод сопряженных градиентов и родственные ему методы чрезвычайно привлекательным при хранении матриц в любом разреженном формате.

Основной недостаток метода – возможная потеря ортогональности. Что это значит. Как уже говорилось, теоретически метод дает точное решение за n шагов. Но этот результат будет достигаться при вычислениях с бесконечным количеством цифр. Мы же работаем в машинной арифметике, т.е. с ограниченным числом значащих цифр. В связи с этим точное решение достигается за n шагов разве что случайно. Причины состоят в следующем. Мы считаем, что невязки образуют ортогональный базис. Этот факт гарантирует теория метода. В то же время, поиск очередной невязки требует умножения матрицы на вектор и сложения двух векторов. Ошибки плавающей арифметики здесь практически гарантированы. Поэтому мы получим $|(r_{k+1}, r_k)| = \varepsilon > 0$. Эта погрешность остается в расчете навсегда. Более того. Эта погрешность накапливается, т.е. $|(r_{k+2}, r_k)| = \varepsilon_1 > \varepsilon > 0$ и т.д. В результате базис из невязок получается не ортогональным. При больших системах уравнений отклонения от ортогональности могут быть весьма значительными. Можно сказать, что ортогональность нам не нужна, достаточно иметь базис. Это верно, но при выводе соотношений для c_k и β_k теоретическая ортогональность невязок использовалась очень активно. Это значит что используемые выражения для c_k и β_k только приближенны. В результате система уравнений для c_k

$$\begin{pmatrix} (Au_1, u_1) & & & \\ & (Au_2, u_2) & & \\ & & \dots & \\ & & & (Au_n, u_n) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{pmatrix} = \begin{pmatrix} (b, u_1) \\ (b, u_2) \\ \dots \\ (b, u_n) \end{pmatrix}.$$

получается не диагональной, а сами векторы u_k оказываются не A-ортогональными.

Эти эффекты привели к тому, что метод сопряженных градиентов сначала был объявлен неустойчивым и заброшен. И лишь потом пришло понимание, что это очень хороший итерационный метод.

Для преобразования метода в итерационный достаточно заменить оператор цикла со счетчиком на оператор цикла «пока не будет достигнута требуемая точность» или «пока невязка не станет достаточно мала». Окончательная форма метода

1. x_1 - *выбрать*
2. $r_1 = b - Ax_1$
3. $u_1 = r_1$
4. *For* $j=1,2,3\dots$ *до досягнення сходимости do*
4. $c_j = (r_j, r_j) / (Au_j, u_j)$
6. $x_{j+1} = x_j + c_j u_j$
7. $r_{j+1} = r_j - c_j Au_j$
8. $\beta_j = (r_{j+1}, r_{j+1}) / (r_j, r_j)$
9. $u_{j+1} = r_{j+1} + \beta_j u_j$
10. *End*

И наконец, для борьбы с накапливающимися ошибками рекомендуется время от времени перезапускать расчет. Это значит, что мы объявляем x_1 равным уже найденному приближению и начинаем расчет с начала.

Метод сопряженных градиентов и по нынешнее время остается самым эффективным итерационным методом решения систем линейных алгебраических уравнений с симметричной матрицей. В основе метода лежит построение А-ортогонального базиса пространства, в котором решается система – ортогонального базиса, состоящего из невязок. Это можно делать на основе любого подхода, в частности на основе рассмотренного процесса Грама-Шмидта, и алгоритм будет работать. Но на основе базиса составленного из невязок получается меньше выкладок, т.к. невязки получаются автоматически и не требуют усилий для нахождения. Ортогональность этого базиса нужна единственно в качестве гарантии того, что на очередном шаге мы повысим размерность подпространства, в котором решении уже найдено на единицу. Отсюда следует, что за n шагов мы исчерпаем пространство, и система будет решена. Естественно, что ортогональность упрощает вычисления, но она не изменяет конечный результат. Если мы сможем на каждом шаге гарантировать увеличение размерности подпространства каким-либо другим способом, то мы также придем к решению.

Результаты исследования. Предложена достаточно простая методика преподавания метода сопряженных градиентов в курсе вычислительной алгебры. Практика показывает, что в данном изложении метод не вызывает у студентов каких-либо затруднений.

Список использованной литературы:

1. Shewchuk, J.R. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain / J.R. Shewchuk // Tech. rep. School of Computer ScienceCarnegie Mellon University. – 1994.
2. Gower, R.M. Conjugate Gradients: The short and painful explanation with oblique projections. [Електронний ресурс] – 2015. – Режим доступу: https://gowerrobert.github.io/pdf/reports/GowerR_Painful_PCG_projections.pdf
3. Saad, Y. Iterative methods for sparse linear systems / Y. Saad. – Philadelphia, Pa.: Society for Industrial and Applied Mathematics. – 2003. – p. 195.

Bibliography:

1. Shewchuk, J.R. (1994). An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. *Tech. rep. School of Computer ScienceCarnegie Mellon University.*
2. Gower, R.M. (2015). Conjugate Gradients: The short and painful explanation with oblique projections. Retrieved from: https://gowerrobert.github.io/pdf/reports/GowerR_Painful_PCG_projections.pdf

3. Saad, Y. (2003). Iterative methods for sparse linear systems. Philadelphia, Pa.: Society for Industrial and Applied Mathematics.

GOLOVNYA Boris,

Doctor of Science, Professor, Department of Applied Mathematics and Informatics, The Bohdan Khmelnytsky National University of Cherkasy

ON THE TEACHING OF THE CONJUGATE GRADIENTS METHOD

Summary. Introduction. The conjugate gradient method is one of the fastest converging iterative methods for solving systems of linear algebraic equations with a symmetric positive definite matrix. But, if the software implementation of the method is very simple, then its mathematical foundations are quite complicated (see, for example, [1], [2], [3]). Very often, when explaining it, such concepts as projection methods, orthogonalization according to Arnoldi and Lanczos, the steepest descent, Krylov spaces, conjugate directions, and much more are used. All this makes the method incomprehensible on an intuitive level. For this reason, the conjugate gradient method is very difficult to explain to students. The paper presents a slightly different substantiation of the method. Here we use only the concept of orthogonality, the Gram-Schmidt method, and the concept of an A-scalar product. The concept of projection is used only to improve an already constructed and understandable method. This approach is intuitive and much simpler methodically than the traditional approach. As a result, in this description, the method is assimilated by students much easier.

The Purpose is to offer a simple and understandable explanation of the conjugate gradient method.

Result and conclusion. The simple and understandable explanation of the conjugate gradient method is offered.

Keywords: systems of linear algebraic equations, iterative methods for solving systems of linear algebraic equations, conjugate gradient method.

*Одержано редакцією 16.04.2019 р.
Прийнято до публікації 04.09.2019 р.*

СЕКЦІЯ «ІНФОРМАТИКА»

УДК 004.032.26

DOI 10.31651/2076-5886-2019-2-59-72

PACS 07.05.Mh, 07.05.Kf, 07.05.Pj

ГОНТАРЕНКО Яна Дмитрівна
магістрантка спеціальності «Прикладна
математика»
Черкаського національного університету
імені Богдана Хмельницького
e-mail: yana.hontarenko@gmail.com

**КРАСНОШЛИК Наталія
Олександрівна**
кандидат технічних наук, доцент,
доцент кафедри прикладної математики та
інформатики
Черкаського національного університету
імені Богдана Хмельницького
e-mail: wlik007@ukr.net
ORCID 0000-0003-4661-6997

ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ДІЙ ЛЮДИНИ ПО ВІДЕО

Розпізнавання дій людини по відео є важливою задачею в області комп'ютерного зору, яка знаходить широке застосування у різних сферах діяльності людини.

У даній роботі розглянуто методи розв'язання даної задачі з використанням штучних нейронних мереж. Описано два підходи застосування нейронні мереж: Transfer Learning та метод зміни простору. Метод Transfer Learning дозволяє використовувати досвід, отриманий під час розв'язання однієї задачі, для розв'язання іншої. Метод зміни простору полягає у використанні прогнозу попередньо-навченої моделі, як вхідних ознак для ще однієї нейронної мережі. У якості такої нейронної мережі обирали мережі різних архітектур з повнозв'язними шарами або шаром LSTM. Також були використані попередньо-треновані мережі MobileNet, ResNet та DenseNet.

Реалізацію розглянутих нейронних мереж здійснено за допомогою бібліотеки Keras. Для навчання моделей використано два типи вхідних даних: відео фрагменти та координати суглобів у просторі.

Для класифікації дій людини за координати суглобів у просторі також застосовували класичні алгоритми машинного навчання: метод найближчих сусідів, логістична регресія, випадковий ліс та метод опорних векторів.

Досліджено ефективність використання запропонованих моделей для розпізнавання дій людини по відео за долею правильних відповідей на тестовій вибірці і часом навчання.

Ключові слова: нейронна мережа, розпізнавання дій людини по відео, Transfer Learning, метод зміни простору, задача класифікації.

Постановка проблеми

Розпізнавання дій людини на відео є важливою частиною систем штучного інтелекту. Незважаючи на великий прогрес у цьому напрямку, розпізнавання рухів все

ще залишається нерозв'язаною задачею у сфері комп'ютерного зору та є перспективною областю для подальших досліджень [1].

Однією з кінцевих цілей розвитку штучного інтелекту в цьому напрямку є створення машин, які б могли досконало розуміти людей, їхні емоції, стан, рухи, дії та наміри, для того, щоб краще аналізувати ситуації і виконувати відповідні запрограмовані дії.

Іншими важливими областями використання цієї задачі є, наприклад, відеоспостереження, сфера розваг та інші. Тут ключовими аспектами є обчислювальні алгоритми, які можуть розпізнавати людські рухи. Вони повинні позначати дію відповідним ярликом після спостереження цілої послідовності людських рухів, або її частини, аналогічно до людської системи зору. Побудову таких алгоритмів зазвичай досліджують у сфері комп'ютерного зору, яка вивчає як запрограмувати комп'ютери на розуміння зображень або послідовності зображень – відео.

Термін розпізнавання дій людини в сфері досліджень комп'ютерного зору коливається від простого руху кінцівки до комплексного руху декількох кінцівок та суглобів. Цей процес динамічний і як правило відображається у відео, що триває декілька секунд. З цих причин, важко описати задачу розпізнавання рухів людини в сфері комп'ютерного зору, проте є багато проектів вартих уваги [2].

Метою статті є використання нейронних мереж для розпізнавання дій людини по відео.

Методи розв'язання

1. Методи розв'язання задачі розпізнавання дій людини за допомогою штучних нейронних мереж

Штучні нейронні мережі (ШНМ) – це математичні моделі, за основу в яких взято, біологічні нейронні мережі, а саме модель людського мозку. Мозок людини складається з нейронів, з'єднаних між собою синапсами. В свою чергу, штучна нейронна мережа складається з штучних нейронів - обчислювальних елементів. Аналогічно до людського мозку в штучних нейронних мережах є зв'язки, що визначають їх характеристики.

У нейронних мереж є такі переваги: паралелізм, можливість навчання та здатність до узагальнення. Оскільки кожен нейрон людського мозку одночасно зв'язаний із сотнями, тисячами інших нейронів, можна сказати, що він володіє високим ступенем паралелізму. Саме це є причиною того, що ШНМ підходять для вирішення задач, з якими раніше могла впоратись тільки людина [3].

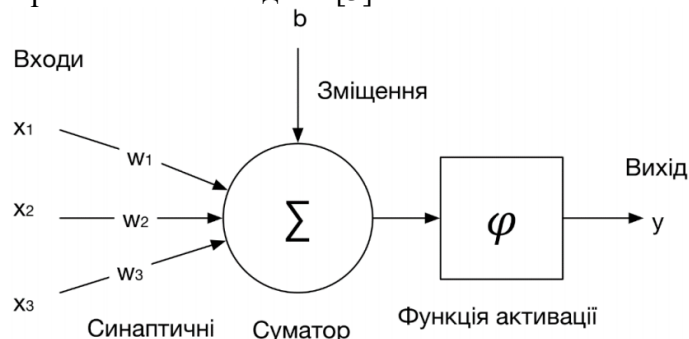


Рис. 1. Зображення математичної моделі ШНМ з одним прихованим шаром (суматор та функція активації)

Для реалізації штучних нейронних мереж використали бібліотеку Keras, яка є надбудовою бібліотеки TensorFlow. Вона в свою чергу використовує графічні

процесори від компанії Nvidia, в яких є вбудована архітектура CUDA – програмно-апаратна архітектура паралельних обчислень, яка дозволяє істотно збільшити обчислювальну продуктивність. Ці графічні процесори працюють набагато швидше звичайних центральних процесорів. На жаль, в мене немає можливості використання такого обладнання, тож було прийнято рішення використовувати Google Colaboratory.

Що б працювати з відеоданими, враховуючи обмеженість обчислювальних ресурсів, було прийнято рішення не працювати з відео, як цілісною структурою, а з кожним кадром цього відео, як з картинкою. Використовуючи цей метод, було обрано декілька підходів: transfer learning та метод зміни простору – використання попередньо тренованої моделі для прогнозу того, на що вона вчилася і потім використання цього результату як вхідних ознак для ще одної нейронної мережі.

Transfer Learning

Transfer Learning дозволяє використовувати досвід, що накопичувався під час вирішення одної задачі, для вирішення іншої. Часто цей метод використовується для класифікації зображень.

Бібліотека Keras надає доступ до багатьох уже реалізованих архітектур: ResNet, VGG та інших. Вони є попередньо-тренованими на базі даних ImageNet. ImageNet – проект по створенню великої бази даних розмічених зображень призначених для тестування методів розпізнавання образів та машинного зору. Із запропонованих архітектур було обрано MobileNet, ResNet та DenseNet.

MobileNet (рис. 2) вважається однією з кращих нейронних мереж. Вона має небагато параметрів, швидка та може прогнозувати з точністю, яку можна прирівняти до моделей такого рівня як ResNet або VGG. Це дає можливість використовувати мінімум ресурсів для гарного результату.

| Type / Stride | Filter Shape | Input Size | |
|---------------|--------------------------------------|------------------------------------|---------------------------|
| Conv / s2 | $3 \times 3 \times 3 \times 32$ | $224 \times 224 \times 3$ | |
| Conv dw / s1 | $3 \times 3 \times 32$ dw | $112 \times 112 \times 32$ | |
| Conv / s1 | $1 \times 1 \times 32 \times 64$ | $112 \times 112 \times 32$ | |
| Conv dw / s2 | $3 \times 3 \times 64$ dw | $112 \times 112 \times 64$ | |
| Conv / s1 | $1 \times 1 \times 64 \times 128$ | $56 \times 56 \times 64$ | |
| Conv dw / s1 | $3 \times 3 \times 128$ dw | $56 \times 56 \times 128$ | |
| Conv / s1 | $1 \times 1 \times 128 \times 128$ | $56 \times 56 \times 128$ | |
| Conv dw / s2 | $3 \times 3 \times 128$ dw | $56 \times 56 \times 128$ | |
| Conv / s1 | $1 \times 1 \times 128 \times 256$ | $28 \times 28 \times 128$ | |
| Conv dw / s1 | $3 \times 3 \times 256$ dw | $28 \times 28 \times 256$ | |
| Conv / s1 | $1 \times 1 \times 256 \times 256$ | $28 \times 28 \times 256$ | |
| Conv dw / s2 | $3 \times 3 \times 256$ dw | $28 \times 28 \times 256$ | |
| Conv / s1 | $1 \times 1 \times 256 \times 512$ | $14 \times 14 \times 256$ | |
| 5× | Conv dw / s1 | $3 \times 3 \times 512$ dw | $14 \times 14 \times 512$ |
| | Conv / s1 | $1 \times 1 \times 512 \times 512$ | $14 \times 14 \times 512$ |
| Conv dw / s2 | $3 \times 3 \times 512$ dw | $14 \times 14 \times 512$ | |
| Conv / s1 | $1 \times 1 \times 512 \times 1024$ | $7 \times 7 \times 512$ | |
| Conv dw / s2 | $3 \times 3 \times 1024$ dw | $7 \times 7 \times 1024$ | |
| Conv / s1 | $1 \times 1 \times 1024 \times 1024$ | $7 \times 7 \times 1024$ | |
| Avg Pool / s1 | Pool 7×7 | $7 \times 7 \times 1024$ | |
| FC / s1 | 1024×1000 | $1 \times 1 \times 1024$ | |
| Softmax / s1 | Classifier | $1 \times 1 \times 1000$ | |

Рис. 2 Архітектури нейронної мережі MobileNetV1 [4]

MobileNetV1 складається з одного звичайного згорткового шару 3×3 , який використовує шар «Batch Normalization» та функцію активації «Relu», та тринадцяти блоків з шести елементів: поглиблений згортковий шар 3×3 з шаром «Batch Normalization» та «Relu» та звичайний згортковий шар 1×1 , шар «Batch Normalization» та «Relu». Особливістю її архітектури є те, що в ній немає pooling-шарів. Саме цю нейронну мережу буде використано для експериментів.

ResNet (residual neural network) – залишкова нейронна мережа. Це означає, що вона має глибоку залишкову структуру навчання (рис. 3), додаються з'єднання швидкого доступу. Вони пропускають один або декілька шарів і виконуються співставлення ідентифікаторів цього прошарку та попереднього залишкового.

Архітектура ResNet50 має наступну структуру: згортковий шар 7×7 , шар max-pool, блоки згорткових шарів з розміром 1×1 , 3×3 , 1×1 , кожен з яких відрізняється від попереднього їх кількістю в мережі та кількістю нейронів у кожному згортковому шарі.

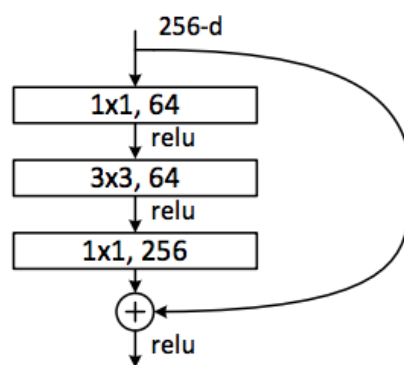


Рис. 3 Перший блок згорткових шарів в ResNet50 [5]

DenseNet – це нейронна мережа, яка не має дуже значних досягнень в сфері глибокого навчання, але її архітектура є досить цікавою (рис. 4). Як видно, кожен з виходів блоку з'єднаний з входом в наступні. Така архітектура дозволяє зменшити ризик того, що не зійдеться градієнт, та зменшує кількість параметрів тощо.

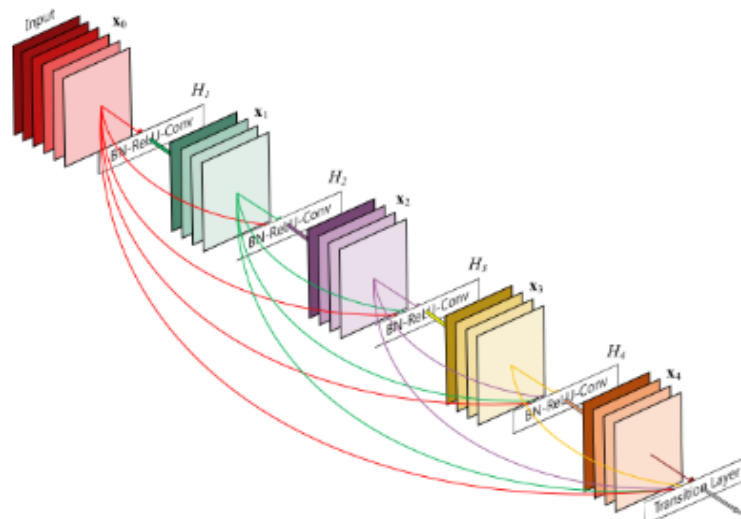


Рис. 4 Архітектура DenseNet [6]

DenseNet121 має послідовність з 7×7 згорткового шару, шару max-pool та Dense блоку, 1×1 згорткового шару, average-pool, що повторюються 3 рази. Після цього йде

ще один Dense блок. Цей блок в собі містить два згорткових шари розмірами 1x1 та 3x3 відповідно. Але в кожному блоці різна кількість такого набору шарів.

Метод зміни простору

Метод зміни простору передбачає використання прогнозу попередньо-навченої моделі, як вхідних ознак для ще одної нейронної мережі.

У цьому випадку було побудовано декілька варіантів нейронної мережі, які поділялись на два типи: нейронні мережі з повнозв'язними шарами та шаром LSTM. Для кожного з цих типів було побудовано декілька варіантів. Для моделі з повнозв'язними шарами архітектура будувалась таким чином:

- визначався шар, який перетворює тензор вхідних ознак;
- обиралась кількість повнозв'язних шарів;
- підбирались значення в шарі dropout.

Для моделі з шарами LSTM підбирались тільки значення в шарі dropout.

2. Методи розв'язання задачі розпізнавання дій людини за допомогою класичних алгоритмів класифікації

Ще одним підходом до розпізнавання дій людини по відео є застосування алгоритмів класифікації даних. Ці алгоритми можна застосувати для класифікації рухів за розміщенням суглобів у просторі.

Задача класифікації – це задача розбиття множини об'єктів на задані групи, всередині кожної з яких вони вважаються схожими один на одного. Основні проблеми, з якими зустрічаються при розв'язку задач класифікації, – це незадовільна якість вхідних даних. Також алгоритми стандартного машинного навчання, які працюють з задачею класифікації дуже чутливі і легко піддаються проблемам перенавчання і недонавчання. В першому випадку модель дуже підлаштовується під всі дані, на яких тренується, під всі аномалії які там є, і не зможе правильно описати тестові дані, тому що не матиме узагальнюючої здатності. Терміном недонавчання позначають ситуацію, коли модель, що використовувалась, не змогла навчитись на тренувальній вибірці, виділити важливі ознаки, класифікувати її об'єкти [7].

Було використано наступні алгоритми класифікації.

Метод k-найближчих сусідів – це один з найпростіших методів класичного машинного навчання. Він працює за наступним алгоритмом для кожного об'єкту з набору даних:

- 1) обчислити відстань до всіх інших об'єктів;
- 2) обрати k елементів відстань до яких мінімальна. Це й будуть сусіди об'єкта;
- 3) класом об'єкту буде той клас, який найчастіше зустрічається в сусідів [8].

Логістична регресія дозволяє оцінити ймовірність належності об'єкту до певного класу. Основна задача логістичної регресії, побудувати пряму (гіперплощину) так, щоб вона максимально розділяла класи, що знаходяться на цій площині (в цьому просторі).

Випадковий ліс – це ансамблевий метод навчання, за основу якого взято побудову визначеної кількості дерев рішень.

Нехай у нас є вибірка з N елементів, M ознак та параметр m . Для всіх дерев рішень в ансамблі будуть, незалежно один від одного, виконуватись такі дії:

- 1) генерується підвибірка з повторенням розміром n з навчальної вибірки;
- 2) будується дерево рішень, яке класифікує елементи даної підвибірки, за m ознаками;
- 3) дерево будується до повного вичерпання підвибірки та не піддається процедурі відсікання.

Після цього проводиться класифікація об'єктів: кожне дерево відносить елемент, для якого проводилась класифікація до одного з класів та визначається той клас, за який проголосувало найбільше дерев [9].

Метод опорних векторів – метод класифікації, що використовує розділяючу пряму або гіперплощину. Він відрізняється від інших гіперплощинних методів класифікації (таких, як логістична регресія) тим, що дозволяє обирати оптимальне розташування гіперплощини. Вона обирається таким чином, щоб бути на максимальній відстані від елементів кожного з класів.

Результати дослідження

1. Пошук та підготовка даних

Перша задача, яку потрібно було вирішити – це пошук даних, які будуть використовуватись для розпізнавання дій людини по відео. Необхідно було знайти два набори даних. Перший набір мав містити відео з різними діями людей. Другий набір повинен був бути розмічений, тобто містити координати, за якими суглоби розміщувались у просторі.

Для першого випадку було обрано набір даних Kinetics-600 (рис. 5). Він містить 500 000 відео та 600 рухів. Це один з найкращих та найбільших наборів даних для задач такого типу.

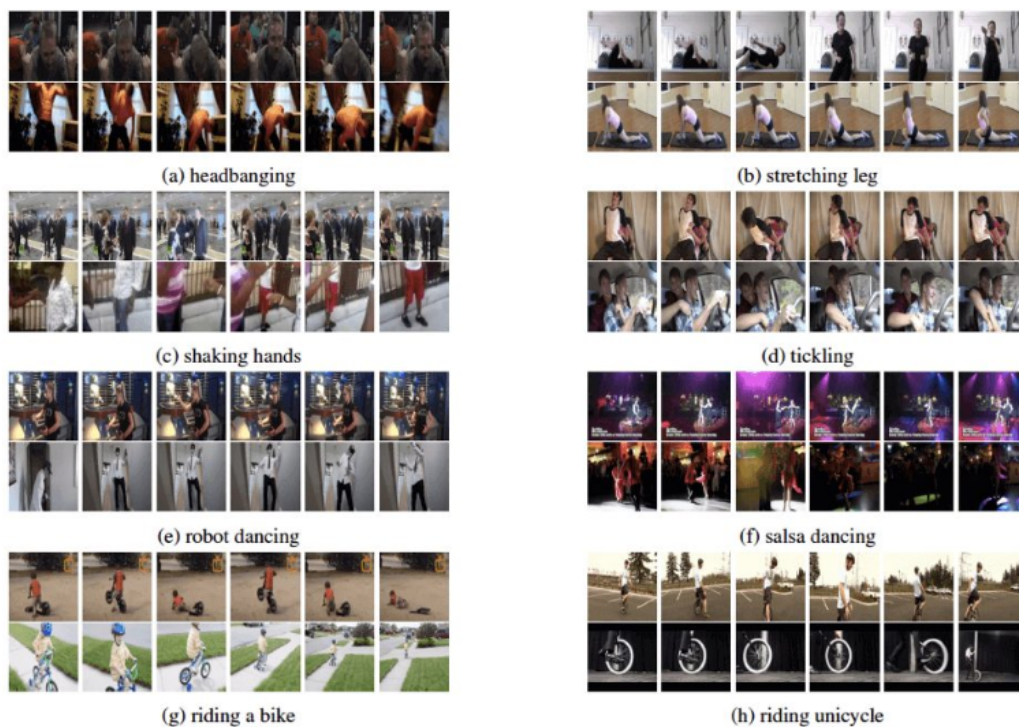


Рис. 5 Ілюстрація деяких відео з набору даних Kinect-600

Даний набір даних для обробки потребує великого об'єму пам'яті та значних обчислювальних ресурсів. Для подальших досліджень було зменшено кількість використаних відео. Випадковим чином обрали 50 дій із 600, і на кожен дію було виділено певну кількість відео. В результаті було збережено близько 22800 відео для тренування, 2500 відео для валідації та 4650 відео для тестування.

Для другого набору даних ключовим було питання про те, як його використовувати. Такі дані отримуються завдяки спеціальним RGB-D камерам або сенсорам. Вони мають спеціальний сенсор глибини, який може визначити, наскільки

далеко об'єкт знаходиться від нього. Завдяки цьому, вони можуть відслідковувати людину, її рухи, її образ із постійного потоку відео. Таким чином, можна це інтерпретувати так, із потоку RGB відео виділяються кадри. З потоку «глибинного» розділеного на кадри відео прибирається шум. Потім ці кадри об'єднуються у картинки. Далі відбувається процес витягування ознак з цих даних та сама класифікація.

Для застосування алгоритму класифікації рухів за допомогою RGB-D послідовності було обрано набір даних UTKinect-Action3D. Він містить дані 10 людей, які виконують 10 різних дій по два рази. Все це зберігається в трьох архівах:

- набір RGB-картинок;
- набір «глибинних» картинок;
- набір координат 20 точок (рис. 6) суглобів в просторі.

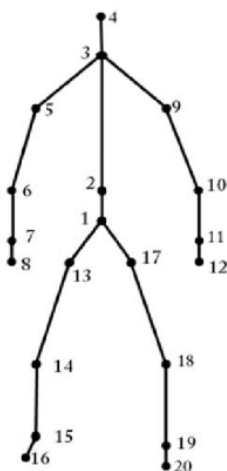


Рис. 6 Нумерація суглобів на тілі людини

Вважається, що набір координат це і є ознаки, які потрібно виявити, тож з даної задачі буде вирішуватись тільки підзадача класифікації.

2. Попередня обробка даних

Попередня обробка даних передбачала попередню обробку відео фрагментів та попередню обробку координат суглобів.

При попередній обробці відео фрагментів необхідно було представити відео у вигляді тензору – багатовимірної матриці. Для цього потрібно було зробити такі кроки:

1) завантажити відео з сервісу YouTube;

У наборі даних, що було використано була така інформація: посилання на розміщення відео на ресурсі YouTube, секунда ролика на якому починалась дія, секунда на якій вона закінчувалась, та назва цієї дії.

2) обрізати розмір відео (до 10 відведених секунд) та зберегти цей відрізок відео у новому розширенні з іншим FPS;

Змінювати розширення та кількість кадрів в секунду потрібно було для того, щоб потім отримувати матриці однакового розміру.

3) завантажити отримане нове відео, нормалізувати його і зберегти у вигляді матриці, для подальшого використання.

Отримані файли містять тензори розміром (50, 20, 200, 200, 3). Тобто це – (кількість відео, секунди * кількість кадрів, ширина, висота, канали кольору).

Метою попередньої обробки координат суглобів також було отримання матриць. З самого початку одне відео містить всі 10 дій. Назва цих дій та їх час (номера кадрів з

ними) вже містяться у наборі даних. Головним завданням було нормалізувати довжину дії, та, відповідно, і розмір майбутньої матриці. Найвні у наборі дії займають різну кількість кадрів. Тобто є дії, які відмічені менш ніж на 10 кадрах, а є ті, що займають більше 100. Для стабілізації було обрано 80 кадрів. Якщо в послідовності кадрів, що зображає дія їх не вистачало, то вони дублювались. Якщо ж їх було забагато, то послідовність обрізалась з двох кінців. Після цього відповідно до кожного елементу нової послідовності будувалась матриці з координат суглобів у просторі, які теж були попередньо отримані з текстового файлу у змінну. У результаті на кожне відео одного користувача було збережено матрицю розміром (10, 80, 20, 3), де 10 – кількість дій, 80 – послідовність кадрів, 20 – кількість суглобів та три їх координати.

3. Реалізації методів роботи з послідовністю картинок

Transfer Learning

Для методу transfer learning було використано моделі MobileNetV1, DenseNet121 і ResNet50. Важливою частиною є завантаження даних в саму модель. Враховуючи обмеженість обчислювальних ресурсів використовували генератори для формування даних. Головна ідея використання генераторів у тому, щоб не завантажувати весь набір даних за раз. Можна завантажити його частинами за певну кількість кроків. Отримати частину тренувальних даних, частину відповідей на них та віддавати це на вхід моделі.

Процес розпізнавання дій по відео передбачає роботу з картинками і класифікацію рухів, опираючись на прогноз для всіх картинок, які цей рух зображали. Тобто, для того, щоб спрогнозувати дію на одному відео потрібно було зробити прогноз для 20 картинок. В процесі «розкладання» відео на картинки кожна матриця змінює форму з матриці форми (кількість відео, кадри в секунду * секунди, ширина, висота, канали кольорів) на матрицю форми (кількість відео * кадри в секунду * секунди, ширина, висота, відео).

Також деякі перетворення проводяться і для відповідей. Кожна відповідь замінюється на вектор з нулів, в якому одиничка стоїть на тому місці, номер якої має відповідь (3 це [0, 0, 0, 1, 0, 0]). Оскільки для використаного методу нам потрібно мати одну відповідь на 20 елементів, то він дублюється 20 разів. Всі ці перетворення виконувались для кожного файлу завдяки генератору і не займали всю пам'ять.

Було використано три попередньо-навчені моделі. Фрагмент коду для роботи з цими моделями наступний:

```
def model(input_shape):
    inp = Input(input_shape)
    base_model = applications.MobileNet(include_top=False, weights='imagenet', input_shape=input_shape,
pooling='avg')
    base_model.trainable = False
    resnet = base_model(inp)
    drop = Dropout(0.7)(resnet)
    fc = Dense(50)(drop)
    out = Activation('softmax')(fc)
    return Model(input=inp, output=out)
model = model((WIDTH, HEIGHT, 3))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Спочатку створюється функція, яка буде повертати об'єкт типу Model. Це спеціальний об'єкт бібліотеки Keras, який дозволяє створити об'єкт нейронної мережі з усіма зв'язками, які пов'язують вхідний та вихідний шар. У самій функції записуються шари нейронної мережі від вхідного шару до вихідного. У даному випадку ми отримуємо на вхід об'єкт з певним розміром. Запам'ятовуємо цей об'єкт у першому вхідному шарі. Далі завантажуюмо попередньо-треновану модель: параметр include_top вказує на те, чи ми будемо використовувати її класифікатор (include_top=True) чи

будемо додавати свій (`include_top=False`), `weights` – які ваги будемо використовувати та задається розмір даних, які подаються на вхід. Після цього вказуємо, що тренувати модель не будемо, додаємо свій класифікатор, тобто вказуємо нашу кількість класів у вихідному шарі та функцію активації, яку хочемо використовувати.

Після чого ми створюємо об'єкт описаної моделі – це робиться викликом функції, та вона компілюється. Параметри компіляції були обрані з рекомендацій, найчастіше використовують саме такі параметри для задач мульти-класифікації. Останнім етапом є тренування.

Метод зміни простору

За допомогою створеного раніше генератору та попередньо-тренованих моделей, до яких надає доступ бібліотека Keras було зроблено прогнози, але не конкретних дій, а того, що вміє дана модель (MobileNetV1) прогнозувати. Ідея полягає в тому, що нейронна мережа змінює простір і вже на іншому просторі буде навчатись для розпізнавання дій. Можна інтерпретувати це так: перша нейронна мережа розпізнає певні предмети на відео, а за допомогою її прогнозів інша вчиться відповідати на питання що за дія на відео.

Для цього спочатку завантажується файл, який містить декілька відео. Для кожного з цих відео робиться прогноз на завантаженій моделі та зберігається в окремому файлі. Таким чином, для одного відео буде вже не матриця розміром (кількість кадрів в секунду * секунди, ширина, висота, канал кольору), а (кількість кадрів в секунду * секунди, 1000). Також паралельно з цим знову зберігались і позначки на ці відео, але вже по одному елементу. Після того, як уже було створено такі вхідні дані, вони були завантажені у матриці і за допомогою генератора, який поділяв ці матриці на частини, подавались на вхід моделям двох типів: що вміщали в собі тільки повноз'язні шари та модель з шаром LSTM.

4. Реалізація методів роботи з координатами суглобів у просторі

Нейронні мережі

Для використання штучних нейронних мереж при класифікації рухів за розміщенням суглобів у просторі, потрібно було представити дані у вигляді картинки за наступним алгоритмом:

- кожна координату було перетворено за формулою

$$(x'_i, y'_i, z'_i) = F(x_i, y_i, z_i) ,$$

$$x'_i = 255 \cdot \frac{x_i - \min\{C\}}{\max\{C\} - \min\{C\}} , y'_i = 255 \cdot \frac{y_i - \min\{C\}}{\max\{C\} - \min\{C\}} , z'_i = 255 \cdot \frac{z_i - \min\{C\}}{\max\{C\} - \min\{C\}} ,$$

де $\min\{C\}$ та $\max\{C\}$ – мінімум та максимум серед усіх координат набору даних. Після перетворення дані зберігаються у матрицях розміру (80, 20, 3), але в форматі RGB;

- змінюємо чергування груп суглобів – нова послідовність $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5$.

В результаті цієї операції змінюється розмір матриці, тоді отримуємо матрицю (40, 40, 3) [1].

Класичні методи машинного навчання

Для використання класичних методів машинного навчання при класифікації рухів за розміщенням суглобів у просторі, потрібно було представити дані у вигляді матриці, рядками якої були б самі дії. Тобто в результаті обробки даних потрібно отримати матрицю розміру (200, 80 * 20 * 3).

Для всіх зазначених раніше методів машинного навчання виконували наступні кроки:

- поділити матрицю на два фрагменти: для тренування та для тестування;
- застосувати класичні методи машинного навчання для тренування;
- перевірити результати;
- підібрати параметри для використаних методів;
- перевірити результати для кращих параметрів.

5. Результати проведених обчислювальних експериментів

Результати навчання попередньо-тренованих моделей продемонстрували, що мережа ResNet відразу почала перенавчатись, DenseNet з самого початку дала непоганий результат, але найкраще спрацювала мережа MobileNet. Тому її обрали для методу зміни простору, щоб отримати прогнози, які будуть вхідними ознаками для наступної нейронної мережі. У якості цієї мережі розглянули повнозв'язні нейронні мережі та нейронні мережі з шаром LSTM різної архітектури. Процес навчання найкращих з них представлено на рис. 7., з якого видно, що хоч модель з шаром LSTM навчалась значно довше, але все одно продемонструвала гірші результати, ніж модель з повнозв'язними шарами, хоча і ця модель почала перенавчатись і не дала бажаного результату.

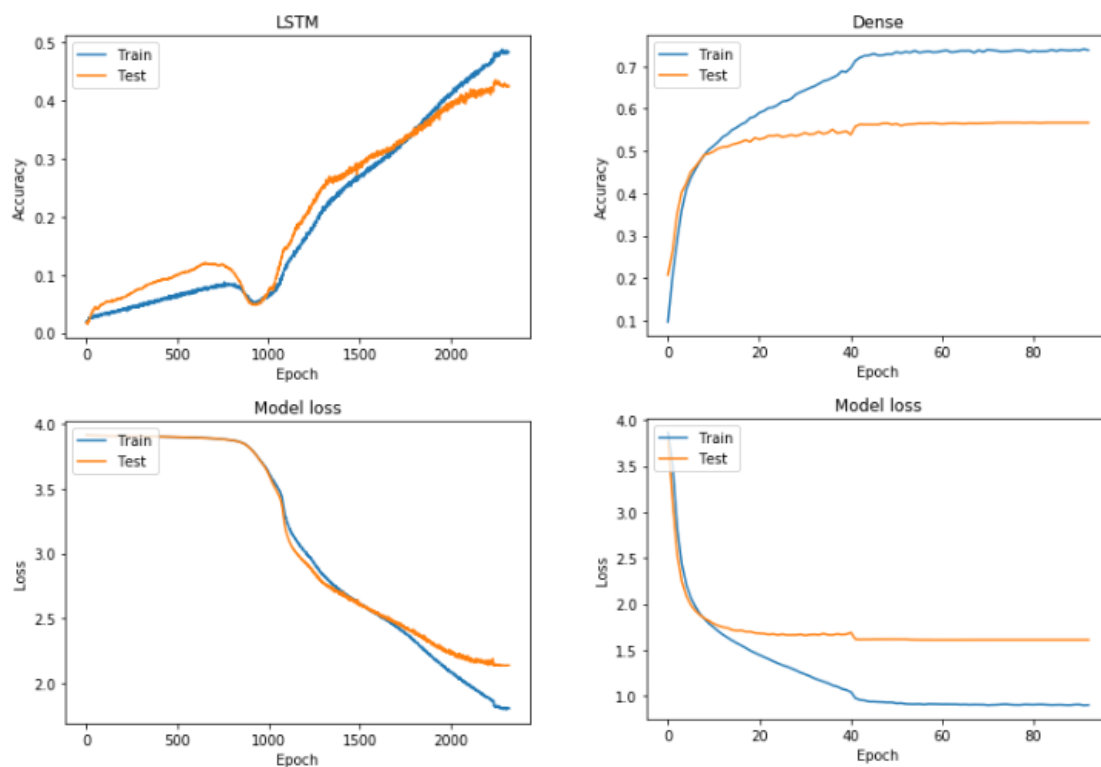


Рис. 7 Процес навчання моделей у зміненому просторі

Таблиця 1

| Метод та модель | Час навчання для однієї епохи | Кількість епох | Доля правильних відповідей на тестовій вибірці |
|---|-------------------------------|----------------|--|
| Попередньо-тренована модель ResNet50 | 540 с | 2 | 0.018750 |
| Попередньо-тренована модель DenseNet121 | 520 с | 14 | 0.505625 |

Продовження таблиці 1

| | | | |
|---|-------|------|----------|
| Попередньо-тренована модель MobileNetV1 | 430 с | 7 | 0.555625 |
| Зміна простору, повнозв'язна модель | 1 с | 63 | 0.546076 |
| Зміна простору, модель з шаром LSTM | 9 с | 2271 | 0.422025 |

Результати проведених обчислювальних експериментів наведено у табл. 1.

Далі розглянемо метод роботи з координатами суглобів у просторі. Було проведено експерименти з різними типами нейронних мереж, але в результаті було обрано згорткову нейронну мережу з декількома блоками. Процес навчання нейронної мережі для цих даних представлено на рис. 8.

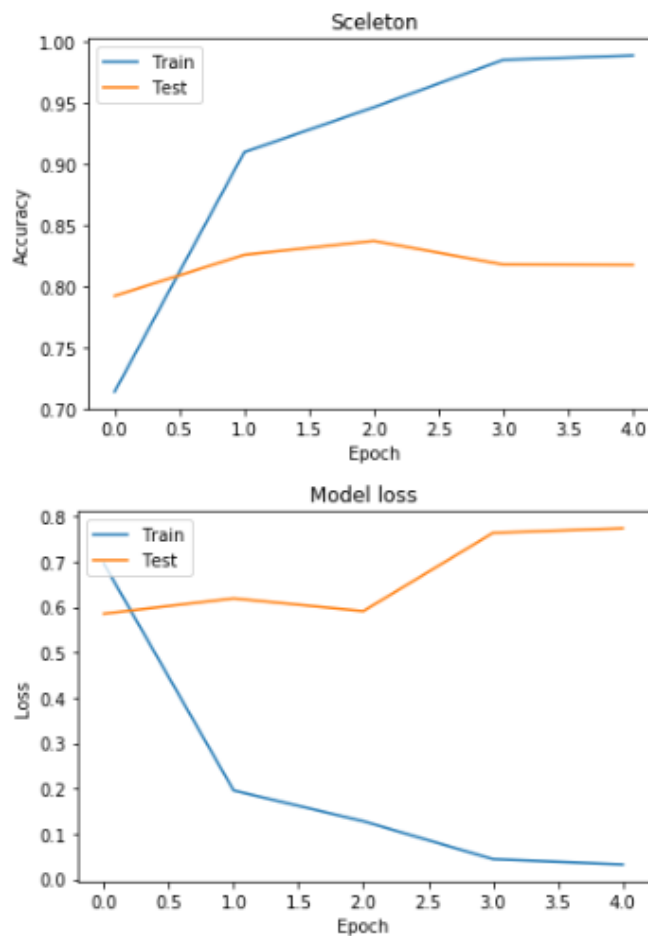


Рис. 8 Процес навчання згорткової нейронної мережі для форматуваних координат суглобів

З наведеного графіку видно, що дані добре оптимізуються і з перших кроків мережа добре навчається.

Також для цього типу даних було використано і класичні методи машинного навчання. Спочатку було проведено обчислювальні експерименти для загальних моделей, а потім підібрано параметри для кожної них. Остаточні результати представлено у табл. 2.

Таблиця 2

| Метод та модель | Час навчання | Доля правильних відповідей на тестовій виборці |
|---------------------------|--------------|--|
| Згорткова нейронна мережа | 50 с | 0.80 |
| Метод найближчих сусідів | 7.15 мс | 0.55 |
| Логістична регресія | 7.15 мс | 0.85 |
| Випадковий ліс | 5.25 мс | 0.58 |
| Метод опорних векторів | 6.68 мс | 0.80 |

В результаті проведеного дослідження дійшли висновку, що реалізувати розпізнавання дій людини без значних обчислювальних ресурсів досить важко. Також потрібно багато пам'яті, щоб була можливість завантажувати всі дані в ході виконання програми, а не порціями, як це було реалізовано.

Висновки

У роботі розглянуто методи розв'язання задачі розпізнавання дій людини по відео. Для цього було використано два типи даних: відео та координати суглобів у просторі.

Для кожного типу даних було використано декілька типів нейронних мереж.

Для набору даних з відео було реалізовано два підходи: у першому було використано три попередньо-треновані моделі; у другому, після зміни простору було використано 2 типи моделей. Найкращі результати за долею правильних відповідей на тестовій виборці було отримано для попередньо-тренованої моделі MobileNetV1. Також хороші результати отримано за допомогою повнозв'язної нейронної мережі, використовуючи метод зміни простору.

Для набору даних з координатами суглобів у просторі проводили експерименти з різними архітектурами нейронної мережі: з повнозв'язними шарами та згортковими. Також для цих даних використали класичні методи машинного навчання: метод найближчих сусідів, логістична регресія, випадковий ліс та метод опорних векторів. Серед класичних моделей найкращі результати на тестовій виборці було отримано для логістичної регресії.

Отже, розглянуті моделі мали непоганий результат враховуючи, що у розглянутій вибірці було 50 різних типів рухів.

Також варто зазначити, що для реалізації моделей розпізнавання дій людини, потрібна значна попередня обробка даних. Мається на увазі, що багато часу займає завантаження та обробка відео і вибір моделей для роботи з ними.

Враховуючи те, що кращі результати отримали для моделей які працювали з координатами суглобів у просторі, можна сказати, що було б доцільно реалізувати модель, яка працюватиме з потоком відео та автоматично виділятиме точки суглобів на тілі людини і аналізуючи їх могла визначити, що за дію виконує людина.

Список використаної літератури:

1. Pham, H.-H. Learning and Recognizing Human Action from Skeleton Movement with Deep Residual Neural Networks / H.-H. Pham, L. Khoudour, A. Crouzil, P. Zegers, S.A. Velastin. [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/1803.07780.pdf>
2. Yu, K. Human Action Recognition and Prediction: A Survey / K. Yu, F. Yun // Journal of latex class files. – 2018. – Vol. 13, No. 9. – 1-20 p.

3. Мошенченко, М. С. Штучні нейронні мережі / М. С. Мошенченко // Topical problems of modern science. – 2017. – Vol.4. – С. 47-49.
4. Akogo, D.A. CellLineNet: End-to-End Learning and Transfer Learning For Multiclass Epithelial Breast cell Line Classification via a Convolutional Neural Network / D.A. Akogo, V. Appiah, X.-L. Palmer. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/327134257_CellLineNet_End-to-End_Learning_and_Transfer_Learning_For_Multiclass_Epithelial_Breast_cell_Line_Classification_via_a_Convolutional_Neural_Network
5. He, K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun. [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/1512.03385>
6. Сайт PyTorch [Електронний ресурс]. – Режим доступу: https://pytorch.org/hub/pytorch_vision_densenet
7. Біла, Н.І. Інформаційні системи та технології в управлінні: методичні вказівки / Н.І. Біла. – Запоріжжя: ЗНТУ, 2014. – 50 с.
8. Классификатор kNN / Хабр [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/149693>
9. Hastie T. The Elements of Statistical Learning / T. Hastie, R. Tibshirani, J. Friedman // Chapter 15. Random Forests. – 2009. – 587-623 p.

Bibliography:

1. Pham, H.-H., Khoudour, L., Crouzil, A., Zegers, P., & Velastin, S.A. (2018). Learning and Recognizing Human Action from Skeleton Movement with Deep Residual Neural Networks. Retrieved from <https://arxiv.org/pdf/1803.07780.pdf>
2. Yu, K., Yun, F. (2018). Human Action Recognition and Prediction: A Survey. *Journal of latex class files*, 13(9), 1-20.
3. Moshchenko, M. S. (2017). Shtuchni neironni merezhi [Artificial Neural Nets]. *Topical problems of modern science*, 4, 47-49 [in Ukrainian].
4. Akogo, D.A., Appiah, V., & Palmer, X.-L. (2018). CellLineNet: End-to-End Learning and Transfer Learning For Multiclass Epithelial Breast cell Line Classification via a Convolutional Neural Network. Retrieved from https://www.researchgate.net/publication/327134257_CellLineNet_End-to-End_Learning_and_Transfer_Learning_For_Multiclass_Epithelial_Breast_cell_Line_Classification_via_a_Convolutional_Neural_Network
5. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. Retrieved from <https://arxiv.org/abs/1512.03385>
6. PyTorch *pytorch.org*. Retrieved from https://pytorch.org/hub/pytorch_vision_densenet
7. Bila, N.I. (2014), Informatsiini systemy ta tekhnolohii v upravlinni: metodychni vказivky [Information Systems and Technologies in Management: Guidelines]. Zaporizhzhia: ZNTU [in Ukrainian].
8. (2012). Klassifikator kNN *habr.com*. Retrieved from <https://habr.com/ru/post/149693>
9. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Chapter 15. Random Forests, 587-623.

HONTARENKO Yana,

Student, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

KRASNOSHLYK Nataliia,

Candidate of Technical Sciences, Associate Professor, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

HUMAN ACTION RECOGNITION FROM VIDEOS USING NEURAL NETWORKS

Summary. Introduction. Human action recognition is an important problem in the field of computer vision, which is widely used in various areas of human activity.

This paper describes methods for solving this problem using neural networks. Two approaches to using neural networks are described: Transfer Learning and Space Modification. The Transfer Learning method allows you to use the experience gained when solving one task to solve another. The method of changing space is to use the prediction model learned as inputs for another neural network. As such a neural network was chosen networks of different architectures with fully connected layers or LSTM layer. MobileNet, ResNet and DenseNet pre-trained networks were also used. Two types of input are used to train models: video snippets and joint coordinates in space.

Classical machine learning algorithms were also used to classify human actions by the coordinates of joints in space: the nearest neighbor method, logistic regression, random forest, and support vector machine.

The Purpose of this paper is to use the neural networks to human action recognition from videos.

Results. The neural networks considered were implemented using the Keras library.

Pre-processing of data is performed to represent them in the form of multidimensional matrices (tensors). Several types of neural networks were used for each data type.

The effectiveness of the use of the proposed models for the recognition of human actions on video by the share of the correct answers on the test sample and the time of training is investigated.

The best results in the proportion of correct answers were obtained for the MobileNetV1 pre-trained model. Also, good results were obtained using a fully connected neural network using the space-changing method. Among the classical machine learning algorithms, the best results on the test sample were obtained for logistic regression.

The models presented show a good result considering that there were 50 different types of human movements in the sample used.

Conclusion. *This paper describes the problem of human action recognition. Neural networks were used to solve this problem. Various approaches have been implemented to use neural networks to recognize human actions through video, and computational experiments have been conducted. Classical machine learning algorithms have been applied to classify joint motion in space.*

The study found that human action recognition from videos requires significant computing resources. Significant pre-processing is also required.

Considering that the best results were obtained for models that worked with joint coordinates in space, it can be concluded that it would be advisable to implement a model that will work with the video stream and automatically highlight the joint points on the body.

Keywords: *neural network, human action recognition from videos, transfer learning, space change method, classification.*

*Одержано редакцією 19.07.2019 р.
Прийнято до публікації 09.10.2019 р.*

УДК 538.9

DOI 10.31651/2076-5886-2019-2-72-86

PACS 07.05.Tr, 61.43.Bn, 64.60.De, 66.30.-h,
82.60.Lf

ПАСІЧНИЙ Микола Олександрович
кандидат фізико-математичних наук,
доцент, завідувач кафедри фізики,
Черкаський національний університет
імені Богдана Хмельницького
e-mail: pasichnyu@ukr.net
ORCID: 0000-0002-8434-1544

ДОСЛІДЖЕННЯ КРИВИХ ФАЗОВОЇ РІВНОВАГИ БІНАРНИХ ГЦК-СПЛАВІВ З ОБМЕЖЕНОЮ РОЗЧИННІСТЮ КОМПОНЕНТІВ РЕШІТКОВИМИ МЕТОДАМИ МОНТЕ-КАРЛО

У роботі проведено дослідження кривих фазової рівноваги бінарних ГЦК-сплавів з обмеженою розчинністю компонентів на основі решіткових методів Монте-Карло. Підтверджено існування єдиної фазової діаграми модельної системи, що не залежить від конкретного дифузійного механізму та ймовірного алгоритму. Виявлено лінійні залежності різниці приведених температур кривих фазової рівноваги від концентрації для моделі регулярного твердого розчину та модельної бінарної системи на основі решіткових методів Монте-Карло.

Ключові слова: *Монте-Карло, бінарний сплав, регулярний твердий розчин, метод дифузійної пари.*

Вступ

Поява і стрибкоподібний розвиток ЕОМ став стимулом для удосконалення широкого класу чисельних методів для моделювання фізичних процесів. Обсяги оперативної пам'яті та швидкодія сучасної обчислювальних систем дають змогу

досліджувати на атомістичному рівні фізичні системи з кількістю атомів порядку 10^6 - 10^9 , чого цілком достатньо для моделювання наносистем та наноструктур.

Одним із популярних сьогодні є метод Монте-Карло, котрий знайшов своє застосування у багатьох галузях сучасної науки, як для розв'язування задач прикладного так і фундаментального характеру. Якщо не вдаватися до деталей, то будь який спосіб розв'язання конкретної задачі, що використовує випадкові величини, відноситься до класу методів Монте-Карло. Вперше метод Монте-Карло використали Нейман, Улам та Метрополіс при розгляді задачі про розповсюдження нейтронів. Термін „Монте-Карло” був запропонований Метрополісом у 1947 році для характеристики класу методів з використанням випадкових величин. Вперше термін „Монте-Карло” був використаний у статті Метрополіса і Улама 1949 року.

Загальною характеристикою даного класу методів є багаторазове застосування певної послідовності математичних та логічних операцій відповідно до імовірнісного алгоритму. Метод Монте-Карло є ефективним при дослідженні дифузійних процесів у твердих тілах. Його основу складає випадковий вибір дозволених траєкторій системи у фазовому просторі конфігурацій. Положення системи у фазовому просторі змінюється шляхом розігрування випадкового числа. У дифузійних задачах це досягається зміщенням атомів у реальному просторі при виконанні умов відповідно до конкретного ймовірнісного алгоритму. Метод Монте-Карло може бути використаний для широкого класу задач від знаходження інтегралів до моделювання кінетики фізичних процесів на атомістичному рівні [1-5]. На даний час залишаються нез'ясованими до кінця процеси нуклеації та конкуренції фаз на початкових етапах твердофазних реакцій. Дослідження даних процесів є актуальним для нанорозмірних систем, оскільки в таких системах, початкові етапи реакційної дифузії визначають характеристики всієї системи.

Мета статті. Метою роботи є дослідження кривих фазової рівноваги бінарних ГЦК-сплавів з обмеженою розчинністю компонентів решітковими методами Монте-Карло та аналіз результатів для різних дифузійних механізмів та ймовірнісних алгоритмів. Дослідження даного питання є актуальним не лише з наукової, а й з методологічної точки зору, оскільки Монте-Карло методи є досить розповсюдженим інструментом для моделювання кінетики дифузійних фазових перетворень на атомістичному рівні. Однак на сьогодні залишаються невисвітленими деякі ключові питання. Зокрема, при моделюванні кінетики дифузійних процесів вирішальним фактором є термодинамічний стимул перетворення, який визначається початковим станом системи відносно кривих фазової рівноваги на діаграмі стану модельної системи. Однак нез'ясованим залишається питання щодо ідентичності діаграм стану модельних систем при використанні різних дифузійних механізмів та ймовірнісних алгоритмів. Для реалізації мети було проведено серію комп'ютерних експериментів по дослідженню кривих фазової рівноваги бінарних ГЦК-сплавів з використанням обмінного та вакансійного механізмів для алгоритмів Метрополіса та Глаубера і вакансійного механізму для «Residence Time» алгоритму з подальшим їх порівнянням між собою та теоретичною моделлю регулярного твердого розчину.

Виклад основного матеріалу

1. Основи решіткових методів Монте-Карло

Клас Монте-Карло методів що базується на розгляді твердого тіла як системи атомів у вузлах певної жорсткої решітки і зміні положення атомів відповідно до певного імовірнісного алгоритму дістав назву решіткових методів Монте-Карло.

Суттєве обмеження даних методів полягає у використанні жорсткої решітки вузлів, в яких можуть знаходитися атоми. Дане обмеження робить можливим

визначити гамільтоніан системи у якому відсутній член з кінетичною енергією. У наближенні гамільтоніана Ізінга, енергію системи можна представити через суму парних енергій взаємодій атомів. Відповідно, можна знайти енергію системи і її зміну в результаті зміщення точки у фазовому просторі.

В залежності від умов задачі, обмежуються розглядом парних енергій взаємодії в межах певної координаційної сфери n , тоді, для енергетичного опису системи, необхідно задати набір параметрів: $E_{i,j}^1, E_{i,j}^2, \dots, E_{i,j}^n$. Верхній індекс вказує номер координаційної сфери в якій знаходиться один атом відносно іншого. Індекси i та j мають зміст при розгляді багатокомпонентної системи. Наприклад, для двокомпонентної системи і наближення другої координаційної сфери отримуємо набір із шести парних енергій взаємодії: $E_{A,A}^1, E_{A,B}^1, E_{B,B}^1, E_{A,A}^2, E_{A,B}^2, E_{B,B}^2$. Підбір параметрів парної енергії взаємодії для конкретної досліджуваної системи становить окрему термодинамічну задачу.

1.1. Механізми решіткових МК-методів

Певний час при моделюванні дифузійного руху досить широко був розповсюджений метод динаміки Кавасаки, у якому допускався безпосередній обмін двох сусідніх атомів з певною ймовірністю, що залежить від зміни повної енергії системи при такому обміні [6, 7]. В даній моделі немає необхідності розглядати сам механізм обміну. Спрощений алгоритм даного методу за обмінним механізмом можна представити наступною послідовністю подій: випадково обирається вузол модельної ґратки; випадково обирається один з його сусідів; обраховується зміна енергії системи в результаті їх обміну; обмін приймається або ні у відповідності до імовірнісного алгоритму (Рис. 1). Однак це не зовсім задовільна репрезентація кінетики дифузійного процесу, адже атоми можуть мігрувати по вузлам решітки лише за певним механізмом, наприклад вакансійним.

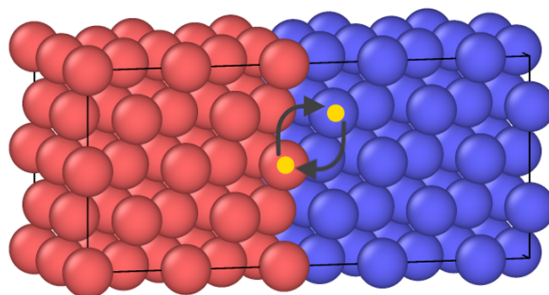


Рис. 1. Схема обмінного механізму для ГЦК структури.

Також для МК-моделювання досить широко використовується метод вакансійної динаміки [8-10]. Для реалізації цього механізму в модельну решітку вводиться вакансія, котра представляє собою відсутність атома у певному вузлі. Даний механізм відмінний від механізму Кавасаки тим, що лише атоми котрі знаходяться біля вакантного вузла можуть здійснити стрибок (помінятися місцями з вакансією). За даним механізмом можуть бути реалізовані стрибки вакансії як в першу так і в наступні координаційні сфери, однак відповідь на питання щодо розподілу ймовірностей таких стрибків є неоднозначною.

Зрозуміло, що наведені вище механізми доцільно використовувати для модельних систем у котрих вузли повністю заповнені атомами (за виключенням вакансій). Однак, їх використання не доцільне при моделюванні систем у яких значна частина вузлів решітки не зайнята атомами. До них можна віднести моделювання процесів напилення, поверхневої дифузії, еволюції пор на атомістичному рівні та ін. Для моделювання даного класу явищ доцільно використовувати модифікацію вакансійного механізму,

коли атом може змінити своє положення лише у випадку наявності вільного вузла. В залежності від конкретної задачі можуть бути різні реалізації з можливістю стрибків атомів як у першу так і в наступні координаційні сфери. Необхідно наголосити, що відмінність даного методу від вакансійного дещо умовна. В обох випадках зміщуються лише атоми і лише у вакантні вузли. Для вакансійного механізму ми можемо вважати рухомим об'єктом вакансію, тоді як для даного методу рухомими об'єктами є самі ж атоми. Перераховані механізми не вичерпують всіх можливих варіантів динаміки системи, адже кінетика системи (траєкторія точки у фазовому просторі) визначається імовірнісним алгоритмом.

1.2. Алгоритм Метрополіса

Згідно алгоритму Метрополіса, котрий був запропонований у 1953 році, ймовірність зміни стану системи, а у випадку решіткового методу Монте-Карло – ймовірність стрибка атома з положення i в положення j (або двох обміну атомів місцями), визначається наступною залежністю [11, 12]:

$$p_{i \rightarrow j} = \begin{cases} 1, & (\Delta E \leq 0); \\ \exp\left(-\frac{\Delta E}{kT}\right), & (\Delta E > 0). \end{cases} \quad (1)$$

де $\Delta E = E_j - E_i$, E_i та E_j – енергії системи до та після зміни стану відповідно, k – стала Больцмана, T – температура системи.

Очевидно, що зміни стану котрі приводять до зменшення енергії системи однозначно приймаються, якщо ж зміна енергії додатна, то ймовірність визначається експоненційною залежністю. Даний алгоритм є коректним при знаходженні рівноважного стану системи, однак не враховуються енергетичні бар'єри котрі долає система у процесі еволюції. Тому його застосування при дослідженні кінетики процесів не є достатньо обґрунтованим.

1.3. Алгоритм Глаубера

Згідно алгоритму Глаубера, ймовірність для атома здійснити стрибок з положення i в положення j визначається залежністю [13]:

$$p_{i \rightarrow j} = \frac{\exp\left(-\frac{\Delta E}{kT}\right)}{1 + \exp\left(-\frac{\Delta E}{kT}\right)}. \quad (2)$$

Як і в алгоритмі Метрополіса, ймовірність залежить лише від зміни енергії системи. Однак даний метод, у комбінації з вакансійним механізмом, широко використовується для моделювання кінетики процесів. Так, наприклад, авторами [14, 15], на основі алгоритму Глаубера, досліджено кінетику релаксації параметру порядку високовпорядкованих бінарних систем. Паралельно проведені резистометричні дослідження підтверджують результати отримані в МК-експериментах.

1.4. Residence-Time Algorithm

Відмінність даного алгоритму від двох попередніх полягає у тому, що розглядаються ймовірності для кожної можливої зміни стану системи і виключається можливість системи залишитися в попередньому стані [16-20]. Застосування Residence-Time алгоритму до решіткового методу Монте-Карло є доцільним для вакансійного механізму. Адже у цьому випадку ми маємо обмежену кількість атомів, що мають змогу здійснити стрибок. Згідно даного алгоритму, ймовірність атома у вузлі i обмінятися місцями з вакансією у вузлі j визначається наступним виразом:

$$P_{i \rightarrow j} = \frac{v_{0i} \exp\left(-\frac{Q_{i \rightarrow j}}{kT}\right)}{\sum_{k=1}^{nn} v_{0k} \exp\left(-\frac{Q_{k \rightarrow j}}{kT}\right)}, \quad (3)$$

де v_{0i} – частота спроб здійснити стрибок (може бути різною для різних типів атомів), $Q_{i \rightarrow j}$ – енергія міграції атома з вузла i у вузол j , nn – кількість атомів, що мають змогу стрибнути. Сумування у знаменнику (3) проводиться по атомам, сусіднім до вакансії, у межах визначеної конкретною моделлю координаційної сфери.

Відповідно до алгоритму, вакансія здійснює стрибок на кожному МК-кроці. Очевидно, що це приводить до нерівномірності часу, тому проміжок часу за який відбувається зміна стану системи на даному МК-кроці визначається наступним виразом:

$$\Delta t = \frac{1}{\sum_{k=1}^{nn} v_{0k} \exp\left(-\frac{Q_{k \rightarrow j}}{kT}\right)}. \quad (4)$$

Визначення енергій міграції атомів $Q_{i \rightarrow j}$ становить окрему задачу, адже зрозуміло, що дана величина характеризує енергетичний бар'єр, котрий потрібно подолати системі. Існує широкий клас першопринципних методів що дозволяють оцінити енергії міграції атомів [21-23]. В такій реалізації даного алгоритму необхідно мати повний набір величин $Q_{i \rightarrow j}$ для можливих стрибків вакансії при всіх конфігураціях оточення. Для розв'язання цієї проблеми користуються певними наближеннями, згідно яких $Q_{i \rightarrow j}$ визначається зміною енергії системи у результаті зміни стану та певними енергетичними параметрами. Як приклад, для активаційної енергії міграції можна використовувати наближення (Рис. 2):

$$Q = \Delta E / 2 + E_0,$$

де ΔE – зміна енергії системи; $E_0 = \text{const}$ – параметр моделі, що характеризує середню висоту бар'єра модельної системи.

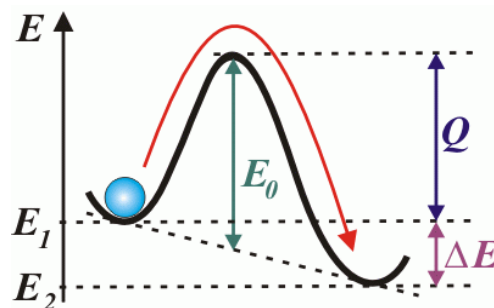


Рис. 2. Енергетичний профіль зміни стану системи, E_1 і E_2 – енергії системи до і після зміни стану.

За даної модифікації енергії міграції, ймовірність стрибка вакансії в i -му напрямі визначається залежністю:

$$p_i = f_i \exp\left(-\frac{\Delta E_i}{2kT}\right) / \sum_{j=1}^6 f_j \exp\left(-\frac{\Delta E_j}{2kT}\right), \quad (5)$$

де $f_i = v_{0i}/v_{0B}$ – характеризує відношення між рухливостями компонентів.

1.5. Граничні умови та поверхні розділу

Для моделювання решітковими методами Монте-Карло використовується жорстка решітка у вузлах якої знаходяться атоми, тобто ідеального решітка. Оскільки модельна система має скінченні розміри, необхідно певним чином обрати граничні умови адекватні до поставленої задачі.

При дослідженні об'ємних властивостей твердих тіл, коли необхідно виключити вплив поверхневих ефектів, використовують періодичні умови Борна-Кармана. Застосування періодичних умов можна інтерпретувати як доповнення системи самою ж собою методом паралельного перенесення. Дані граничні умови забезпечують замкнутість і перетворюють модельну систему у нескінченну.

Однак існує цілий ряд задач в яких необхідно враховувати скінченність системи. У таких випадках граничні умови можуть задаватися різними способами. Наприклад, для моделювання утворення і руху нанопор при електроміграції [24], граничні атоми діелектрика фіксуються нерухомо у вузлах решітки. Для моделювання процесу наплення [25, 26], по двох осях, паралельних підложці, задаються періодичні граничні умови, граничні атоми підложки фіксуються як нерухомі, а поверхня на яку проводиться наплення вважається вільною, тобто атоми межують з частиною незайнятих вузлів, котрі заповнюються в процесі моделювання за рахунок наплення та поверхневої дифузії.

Більшість реальних систем це полікристали – мають у своїй структурі границі (границі зерен, міжфазні границі, границі двійників). Моделювання таких систем ускладнене використанням єдиної решітки вузлів для всього зразка. Дана проблема може бути розв'язана декількома способами: 1) Поняття границі вводиться шляхом приписування атомів до певного зерна. В такому наближенні границя представляє собою уявну поверхню розділу між атомами що належать різним зернам. Додатково вводиться енергія парної взаємодії для атомів різних зерен [24-26]; 2) Границі вводяться шляхом визначення послідовності атомів, котрі представляють границю. При цьому для атомів границі вводиться додаткова енергія і підвищується їх рухливість для забезпечення переваги дифузії по границям у порівнянні з об'ємною [27]; 3) Модельний зразок представляє собою полікристал, зерна якого мають різну орієнтацію [28], тобто для кожного зерна вводиться своя жорстка решітка з певною орієнтацією.

1.6. Застосування решіткових МК-методів до моделювання дифузійних процесів на атомістичному рівні

Моделювання процесів реакційної дифузії атомістичними МК-методами дає змогу проаналізувати початкові стадії даного процесу – нуклеацію нової фази та кінетику морфології дифузійної зони. Складність застосування МК-методів для моделювання даного класу явищ полягає у появі міжфазних границь, котрі, фізично, мають певну поверхневу енергію. Наступна проблема це опис термодинаміки нової фази, адже зрозуміло, поява нової фази викликана енергетичною вигідністю. Одними із перших робіт по моделювання початкових етапів реакційної дифузії МК-методами можна вважати [29-31]. Авторами даних робіт був розроблений новий підхід для моделювання двофазних систем. Основа запропонованого методу полягає у введенні різних енергій парної взаємодії для впорядкованої фази та розчину. На основі „Residence-Time” алгоритму для вакансійного механізму було встановлено існування двостадійного росту фази в мультишарах. Підтверджено гіпотезу про відповідність першої стадії реакції латеральному розростанню острівків нової фази, другої стадії –

нормальному росту. У роботах [32, 33] розроблено Монте-Карло модель в основу якої покладено можливість існування в дифузійній зоні двох проміжних фаз. Особливість запропонованої моделі полягає у різкій залежності парних потенціалів взаємодії від оточення. Для випадку двох проміжних фаз – введено чотири типи потенціалів: 1) для материнської фази (чистий А і чистий В, включно); 2) для проміжної фази типу А1В2; 3) для проміжної фази типу А2В1; 4) для атомів котрі належать різним фазам (аналог поверхневої енергії).

2. Методика експерименту

Для визначення рівноважних концентрацій бінарної системи з обмеженою розчинністю компонентів використовувався метод дифузійної пари. Суть даного методу полягає в тому, що модельна система задається у вигляді дифузійної пари, одна частина якої заповнена атомами сорту А, а інша половина атомами сорту В. У результаті моделювання кінетики дифузійних процесів за Монте-Карло алгоритмом відбувається релаксація системи до рівноваги, що відповідає контакту двох бінарних твердих розчинів на основі компонентів А (Alfa) та В (Beta) відповідно (Рис. 3). Даний метод був використаний авторами [34] для визначення точок кривої рівноваги діаграми стану модельної системи і отримано значну відмінність їх значень від моделі регулярного твердого розчину.

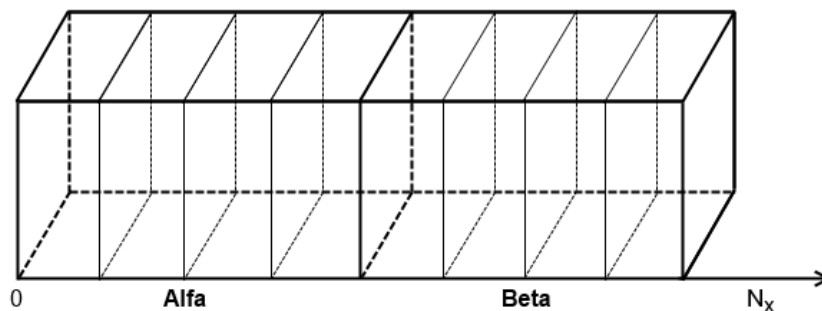


Рис. 3. Схема методу дифузійної пари.

У якості модельної системи для дослідження використано тривимірну ґратку з гранецентрованою кубічною решіткою (ГЦК) розміром 50x20x20 вузлів заповнених атомами двох сортів. Початково частина системи заповнювалася атомами одного сорту, інша – атомами другого сорту, створюючи таким чином дифузійну пару А-В. Для ГЦК решітки кожен вузол має $z = 12$ найближчих сусідів. При моделюванні використано наближення першої координаційної сфери для визначення енергії системи та можливості обміну. У якості граничних умов використано періодичні граничні умови Борна-Кармана по всім трьом осям.

Основним термодинамічним параметром модельної системи є приведена енергія змішування:

$$V = \frac{E_{mix}}{kT} = \frac{E_{AB} - \frac{E_{AA} + E_{BB}}{2}}{kT}. \quad (6)$$

3. Результати комп'ютерного експерименту

За результатами серії комп'ютерних експериментів встановлено рівноважний склад фаз C_{alfa} і C_{beta} при їх контакті для різних значень приведеної енергії змішування бінарної системи. Дослідження проводилися із використанням різних дифузійних механізмів та ймовірнісних алгоритмів. Результати представлені у таблицях 1-5:

- Таблиця 1 – Обмінний механізм за алгоритмом Метрополіса;
- Таблиця 2 – Обмінний механізм за алгоритмом Глаубера;

- Таблиця 3 – Вакансійний механізм за алгоритмом Метрополіса;
- Таблиця 4 – Вакансійний механізм за алгоритмом Глаубера;
- Таблиця 5 - Вакансійний механізм за Residence Time Algorithm.

Таблиця 1.
Обмінний механізм за алгоритмом Метрополіса.

| E_{mix}/kT | C_{alfa} | C_{beta} | kT/E_{mix} |
|--------------|-------------|-------------|--------------|
| 0,208 | 0,680011475 | 0,313967558 | 4,807692308 |
| 0,21 | 0,724345042 | 0,27321015 | 4,761904762 |
| 0,215 | 0,785609733 | 0,213038417 | 4,651162791 |
| 0,22 | 0,819313742 | 0,1817348 | 4,545454545 |
| 0,23 | 0,861066642 | 0,137791333 | 4,347826087 |
| 0,25 | 0,909325492 | 0,089480017 | 4 |
| 0,3 | 0,962094142 | 0,038059896 | 3,333333333 |
| 0,4 | 0,990804442 | 0,009428488 | 2,5 |

Таблиця 2.
Обмінний механізм за алгоритмом Глаубера.

| E_{mix}/kT | C_{alfa} | C_{beta} | kT/E_{mix} |
|--------------|-------------|-------------|--------------|
| 0,208 | 0,697516692 | 0,30579395 | 4,807692308 |
| 0,21 | 0,733112633 | 0,266149117 | 4,761904762 |
| 0,215 | 0,78476975 | 0,215027133 | 4,651162791 |
| 0,22 | 0,816802533 | 0,183115008 | 4,545454545 |
| 0,23 | 0,862494508 | 0,13799945 | 4,347826087 |
| 0,25 | 0,9086616 | 0,090969631 | 4 |
| 0,3 | 0,962367375 | 0,038422723 | 3,333333333 |
| 0,4 | 0,99058115 | 0,009319213 | 2,5 |

Таблиця 3.
Вакансійний механізм за алгоритмом Метрополіса.

| E_{mix}/kT | C_{alfa} | C_{beta} | kT/E_{mix} |
|--------------|-------------|-------------|--------------|
| 0,208 | 0,705218325 | 0,290373825 | 4,807692308 |
| 0,21 | 0,734239592 | 0,260308467 | 4,761904762 |
| 0,215 | 0,787069317 | 0,213377967 | 4,651162791 |
| 0,22 | 0,819631683 | 0,1804902 | 4,545454545 |
| 0,23 | 0,859286925 | 0,139530517 | 4,347826087 |
| 0,25 | 0,908969442 | 0,089775928 | 4 |
| 0,3 | 0,961201325 | 0,038469373 | 3,333333333 |
| 0,4 | 0,990719267 | 0,009467731 | 2,5 |

Таблиця 4.
Вакансійний механізм за алгоритмом Глаубера.

| E_{mix}/kT | C_{alfa} | C_{beta} | kT/E_{mix} |
|--------------|-------------|-------------|--------------|
| 0,208 | 0,7124315 | 0,291251442 | 4,807692308 |
| 0,21 | 0,73837975 | 0,269286875 | 4,761904762 |
| 0,215 | 0,789483042 | 0,215157067 | 4,651162791 |
| 0,22 | 0,822855575 | 0,180125467 | 4,545454545 |
| 0,23 | 0,862682967 | 0,1372936 | 4,347826087 |
| 0,25 | 0,910053883 | 0,090623065 | 4 |

Продовження таблиці 4.

| E_{mix}/kT | C_{alfa} | C_{beta} | kT/E_{mix} |
|--------------|-------------|-------------|--------------|
| 0,3 | 0,961617242 | 0,038483348 | 3,333333333 |
| 0,4 | 0,990861425 | 0,009564916 | 2,5 |

Таблиця 5.

Вакансійний механізм за Residence Time Algorithm.

| E_{mix}/kT | C_{alfa} | C_{beta} | kT/E_{mix} |
|--------------|-------------|-------------|--------------|
| 0,208 | 0,73012425 | 0,285057008 | 4,807692308 |
| 0,21 | 0,736232558 | 0,260327467 | 4,761904762 |
| 0,215 | 0,780804608 | 0,210252767 | 4,651162791 |
| 0,22 | 0,813826942 | 0,1812084 | 4,545454545 |
| 0,23 | 0,861944 | 0,138785517 | 4,347826087 |
| 0,25 | 0,909113417 | 0,091611846 | 4 |
| 0,3 | 0,961750533 | 0,037998778 | 3,333333333 |
| 0,4 | 0,990674092 | 0,009225173 | 2,5 |

Для аналізу, результати зручно представити у вигляді діаграми стану модельної системи. На рис. 4 побудована фазова діаграма стану модельного бінарного ГЦК сплаву з обмеженою розчинністю компонентів. Неперервно кривою представлено лінію фазової рівноваги отриману на основі теоретичної моделі регулярного твердого розчину [34]. Пунктирними кривими та маркерами представлено результати комп'ютерного моделювання решітковими методами Монте-Карло, за даними представленими у таблицях 1-5.

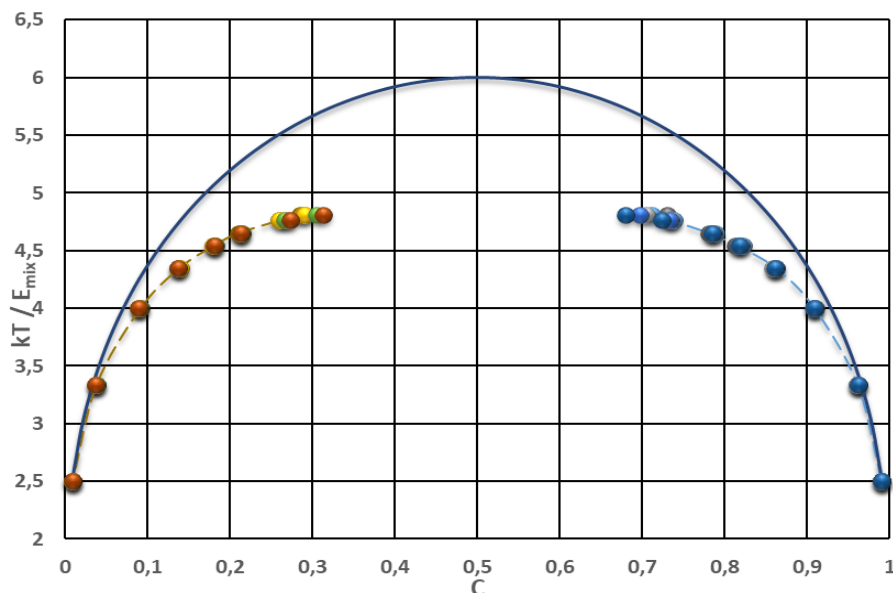


Рис. 4. Фазова діаграма стану бінарного ГЦК сплаву з обмеженою розчинністю компонентів. Неперервно кривою представлено лінію фазової рівноваги отриману на основі теоретичного методу регулярного твердого розчину. Пунктирною кривою та маркерами представлено результати комп'ютерного моделювання решітковими методами Монте-Карло за даними представленими у таблицях 1-5.

Можна бачити, що значення отримані за різними дифузійними механізмами та ймовірнісними алгоритмами відповідають певній єдиній кривій фазової рівноваги

модельної бінарної системи. Відсутність даних в центральній області концентрацій викликана складністю визначення рівноважних концентрацій методом дифузійної пари при відповідних параметрах системи в області близькій до однофазного рівноважного стану системи. Типовою є значна різниця між кривими фазової рівноваги бінарної системи отриманих на основі решіткових Монте-Карло і моделі регулярного твердого розчину.

На Рис. 5 представлено залежності приведеної різниці температур кривих фазової рівноваги для моделі регулярного твердого розчину і решіткових Монте-Карло методів від складу бінарного сплаву. Залежності є симетричними відносно концентрації 1/2. Крім того, вони є прямолінійними. Цей результат відкриває можливість, після подальших досліджень, встановити аналітичний вираз для кривої фазової рівноваги модельної системи для решіткових методів Монте-Карло шляхом удосконалення відповідних виразів для моделі регулярного твердого розчину.

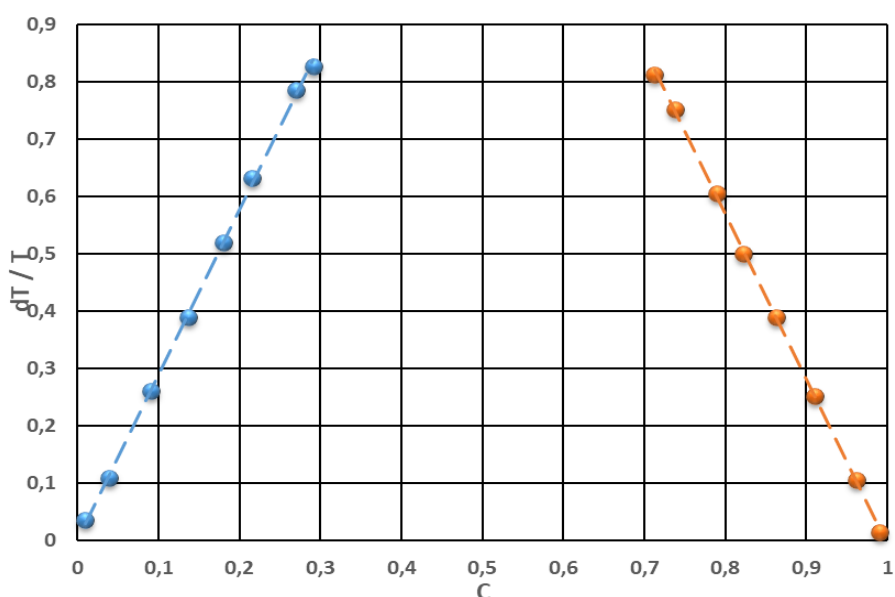


Рис. 5. Залежність приведеної різниці температур кривих фазової рівноваги для моделі регулярного твердого розчину і решіткових Монте-Карло моделей від складу.

Висновки

У роботі досліджено криві фазової рівноваги бінарних ГЦК-сплавів з обмеженою розчинністю компонентів на основі комп'ютерного моделювання решітковими методами Монте-Карло з використанням методу дифузійної пари. Для узагальнення результатів використано ймовірнісні алгоритми Глаубера та Метрополіса на основі обмінного та вакансійного дифузійних механізмів, а також Residence Time Algorithm для вакансійного механізму.

За результатами серії комп'ютерних експериментів побудовано фазову діаграму модельної бінарної системи з ГЦК ґраткою що базується на використанні решіткових Монте-Карло алгоритмів для моделювання кінетики дифузійних процесів на атомістичному рівні.

Підтверджено існування єдиної кривої фазової рівноваги модельної бінарної системи для решіткових методів Монте-Карло, що не залежить від конкретного дифузійного механізму та ймовірнісного алгоритму.

Виявлено лінійні залежності різниці приведених температур кривих фазової рівноваги від концентрації для моделі регулярного твердого розчину та модельної системи на основі решіткових методів Монте-Карло.

Подяки

Статтю написано згідно з держбюджетними темами «Сингулярні розв'язки рівнянь математичної фізики в анізотропних і неоднорідних середовищах, моделювання процесів дифузії та абсорбції» (№ державної реєстрації 0119U100421) і «Мультимасштабне моделювання конкурентної нуклеації, росту і коалесценції фаз в ізотермічних та СВС-реакціях» (№ державної реєстрації 0118U003861). Автор висловлює подяку Каленіченку Віталію Анатолійовичу за створення програмного забезпечення і проведення розрахунків.

Список використаної літератури:

1. Thijssen, J. M. Computational Physics / J. M. Thijssen. – Cambridge: Cambridge University Press, 1999. – 560 p. – Режим доступу: <http://www.cambridge.org/9781107677135>.
2. Гулд, Х. Компьютерное моделирование в физике / Х. Гулд, Я. Тобочник. – Москва: Мир, 1990. – 752 с.
3. Хеерман, Д. В. Методы компьютерного эксперимента в теоретической физике / Д. В. Хеерман. – Москва: Наука, 1990. – 176 с.
4. Поттер, Д. Вычислительные методы в физике / Д. Поттер. – Москва: Мир, 1975. – 392 с.
5. Гусак, А. М. Статистична фізика – основні положення та моделі / А. М. Гусак, В. Г. Гриценко, Т. В. Запорожець. – Черкаси: ЧДУ, 1998. – 288 с.
6. Bortz, A. V. Time evolution of a quenched binary alloy: Computer simulation of a two-dimensional model system / A. V. Bortz, M. H. Kalos, J. L. Lebowitz, M. A. Zendejas // Physical Review B. – 1974. – Vol. 10. – P. 535-541. – Режим доступу: <https://doi.org/10.1103/PhysRevB.10.535>.
7. Marro, J. Time evolution of a quenched binary alloy. II. Computer simulation of a three-dimensional model system / J. Marro, A. V. Bortz, M. H. Kalos, J. R. Lebowitz // Physical Review B. – 1975. – Vol. 12. – P. 2000-2011. – Режим доступу: <https://doi.org/10.1103/PhysRevB.13.4328>.
8. Flinn, P. A. Monte Carlo calculation of the order-disorder transformation in the body-centered cubic lattice / P. A. Flinn, G. M. McManus // Physical Review. – 1961. – Vol. 124. – Issue 1. – P. 54-59. – Режим доступу: <https://doi.org/10.1103/PhysRev.124.54>.
9. Beller, J. R. Order-Disorder Events Produced by Single Vacancy Migration / J. R. Beller, J. A. Delaney // Physical Review. – 1963. – Vol. 130. – P. 962-971. – Режим доступу: <https://doi.org/10.1103/PhysRev.130.962>.
10. Beller, J. R. Order-Disorder Events Produced by Single-Vacancy Migration. II / J. R. Beller // Physical Review. – 1965. – Vol. 138. – P. A1259-A1270. – Режим доступу: <https://doi.org/10.1103/PhysRev.138.A1259>.
11. Metropolis, N. Equation of state calculations by fast computing machines / N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller // Journal of Chemical Physics. – 1953. – Vol. 21. – P. 1087-1092. – Режим доступу: <https://doi.org/10.1063/1.1699114>.
12. Liu, Z.-J. Effects of amorphous matrix on the grain growth kinetics in two-phase nanostructured films: a Monte Carlo study / Z.-J. Liu, Y. G. Shen // Acta Materialia. – 2004. – Vol. 52. – P. 729-736. – Режим доступу: <https://doi.org/10.1016/j.actamat.2003.10.010>.
13. Glauber, R. J. Time-dependent statistics of the Ising model / R. J. Glauber // Journal of Mathematical Physics. – 1963. – Vol. 4. – P. 294-307. – Режим доступу: <https://doi.org/10.1063/1.1703954>.
14. Oramus, P. Monte Carlo computer simulation of order-order kinetics in the L12-ordered Ni3Al binary system / P. Oramus, R. Kozubski // Physical Review B. – 2001. – Vol. 63. – P. 174109. – Режим доступу: <https://doi.org/10.1103/PhysRevB.63.174109>.
15. Oramus, P. Computer simulation of long-range order relaxation in homogeneous systems / P. Oramus, R. Kozubski, V. Perron-Bohnes, M. C. Cadeville, C. Massobrio, W. Pfeiler // Materials Science and Engineering A. – 2002. – Vol. 324. – P. 11-15. – Режим доступу: [https://doi.org/10.1016/S0921-5093\(01\)01275-8](https://doi.org/10.1016/S0921-5093(01)01275-8).
16. Young, W. M. Monte Carlo studies of vacancy migration in binary ordered alloys / W. M. Young, E. Elcoc // Proceedings of the Physical Society. – 1966. – Vol. 89. – P. 735-746. – Режим доступу: <https://doi.org/10.1088/0370-1328/89/3/329>.
17. Bortz, A. V. A new algorithm for Monte Carlo simulation of Ising spin systems / A. V. Bortz, M. H. Kalos, J. L. Lebowitz // Journal of Computational Physics. – 1975. – Vol. 17. – P. 10-18. – Режим доступу: [https://doi.org/10.1016/0021-9991\(75\)90060-1](https://doi.org/10.1016/0021-9991(75)90060-1).
18. Athenes, M. A Monte-Carlo study of B2 ordering and precipitation via vacancy mechanism in b.c.c. lattices / M. Athenes, P. Bellon, G. Martin, F. Haider // Acta Materialia. – 1996. – Vol. 44. – P. 4739-4748. – Режим доступу: [https://doi.org/10.1016/S1359-6454\(96\)00135-8](https://doi.org/10.1016/S1359-6454(96)00135-8).

19. Athenes, M. Identification of novel diffusion cycles in B2 ordered phases by Monte Carlo simulation / M. Athenes, P. Bellon, G. Martin // *Philosophical Magazine A*. – 1997. – Vol. 76. – P. 565-585. – Режим доступу: <https://doi.org/10.1080/01418619708214023>.
20. Haider, F. Monte-Carlo study of the phase stability of ordered fcc phases under irradiation / F. Haider // *Ordering and Disordering in Alloys*; Edited by A. R. Yavari. – London: Elsevier Applied Science, 1992. – P. 215-222. – Режим доступу: https://doi.org/10.1007/978-94-011-2886-5_23.
21. De Fontane D. Cluster approach to order-disorder transformations in alloys / D. de Fontane // *Solid State Physics*. – 1994. – Vol. 47. – P. 33-176. – Режим доступу: [https://doi.org/10.1016/S0081-1947\(08\)60639-6](https://doi.org/10.1016/S0081-1947(08)60639-6).
22. Wolverton, C. Short-range-order types in binary alloys: a reflection of coherent phase stability / C. Wolverton, V. Ozolinis, A. Zunger // *Journal of Physics: Condensed Matter*. – 2000. – Vol. 12. – P. 2749-2768. – Режим доступу: <https://doi.org/10.1088/0953-8984/12/12/314>.
23. Staunton, J. B. Compositional short-range ordering in metallic alloys: band-filling, charge-transfer, and size effects from a first-principles all-electron Landau-type theory. / J. B. Staunton, D. D. Johnson, F. J. Pinski // *Physical Review B*. – 1994. – Vol. 50. – P. 1450-1472. – Режим доступу: <https://doi.org/10.1103/PhysRevB.50.1450>.
24. Zaporozhets, T. V. 3D-Simulation of Void Formation, Growth and Migration under Electromigration / T. V. Zaporozhets, I. V. Sobchenko, A. M. Gusak // *Defect and Diffusion Forum*. – 2005. – Vol. 237. – P. 1306-1311. – Режим доступу: <https://doi.org/10.4028/www.scientific.net/DDF.237-240.1306>.
25. Tu, K. N. Linear rate of grain growth in thin films during deposition / K. N. Tu, A. M. Gusak, I. Sobchenko // *Physical Review B*. – 2003. – Vol. 67. – P. 245408. – Режим доступу: <https://doi.org/10.1103/PhysRevB.67.245408>.
26. Sobchenko, I. V. 3D Monte-Carlo model of deposition and grain growth in thin films / I. V. Sobchenko, A. M. Gusak, K. N. Tu // *Defect and Diffusion Forum*. – 2005. – Vol. 237. – P. 1281-1286. – Режим доступу: <https://doi.org/10.4028/www.scientific.net/DDF.237-240.1281>.
27. Hin, C. Atomistic Monte Carlo simulations of homogeneous and heterogeneous precipitation on grain boundaries of NbC in steels / C. Hin, F. Soisson, Ph. Maugis // *Defect and Diffusion Forum*. – 2005. – Vol. 237. – P. 721-726. – Режим доступу: <https://doi.org/10.4028/www.scientific.net/DDF.237-240.721>.
28. Rubio, J. E. Atomistic Monte Carlo simulations of three-dimensional polycrystalline thin films / J. E. Rubio, M. Jaraiz, I. Martin-Bragado, J. M. Hernandez-Mangas, J. Barbolla // *Journal of Applied Physics*. – 2003. – Vol. 94. – P. 163-168. – Режим доступу: <https://doi.org/10.1063/1.1577814>.
29. Gusak, A. Initial Stage of Reactive Diffusion: Nucleation and Avrami Kinetics / A. Gusak, A. O. Bogatyrev, A. O. Kovalchuk // *Defect and Diffusion Forum*. – 2001. – Vol. 194. – P. 1625-1630. – Режим доступу: <https://doi.org/10.4028/www.scientific.net/DDF.194-199.1625>.
30. Gusak, A. M. Oscillatory regime of ordering during interdiffusion / A. M. Gusak, A. O. Kovalchuk // *Physical Review B*. – 1998. – Vol. 58. – P. 2551-2555. – Режим доступу: <https://doi.org/10.1103/PhysRevB.58.2551>.
31. Гусак, А. М. Модели твердофазных реакций / А. М. Гусак, А. О. Богатырев, Т. В. Запорожец, А. А. Ковальчук, С. В. Корниенко, Г. В. Луценко, Ю. А. Ляшенко, А. С. Ширинян. – Черкассы: ЧНУ, 2004. – 314 с.
32. Pasichnyy, M. O. Modeling of Phase Competition and Diffusion Zone Morphology Evolution at Initial Stages of Reaction Diffusion / M. O. Pasichnyy, A. M. Gusak // *Defect and Diffusion Forum*. – 2005. – Vol. 237. – P. 1193-1198. – Режим доступу: <https://doi.org/10.4028/www.scientific.net/DDF.237-240.1193>.
33. Пасічний, М. О. Монте-Карло дослідження початкових етапів реакційної дифузії у мультишарах / М. О. Пасічний // *Вісник Черкаського університету. Серія «Фізико-математичні науки»*. – 2007. – Вип. 114. – С. 65-72.
34. Gusak, A. M. Modeling of Concentration and Temperature Dependencies of Incubation Time at Decomposition of Solid Solution by Monte Carlo Method / A. M. Gusak, V. M. Pasichna // *Вісник Черкаського університету. Серія «Фізико-математичні науки»*. – 2018. – Вип. 1. – С. 3-11. – Режим доступу: <https://doi.org/10.31651/2076-5851-2018-1-3-11>.

Bibliography:

1. Thijssen, J. M. (1999). *Computational Physics*. Cambridge: Cambridge University Press. Retrieved from <http://www.cambridge.org/9781107677135>.
2. Guld, Kh., Tobochnik, Ya. (1990). *Kompiuternoe modelirovanie v fizike [Computer modeling in physics]*. Moskva: Mir [in Russian].
3. Kheerman, D. V. (1990). *Metody kompiuternogo eksperimenta v teoreticheskoi fizike [Methods of computer experiment in theoretical physics]*. Moskva: Nauka [in Russian].
4. Potter, D. (1975). *Vychislitelnye metody v fizike [Computational methods in physics]*. Moskva: Mir [in Russian].

5. Gusak, A. M., Grytsenko, V. G., Zaporozhets, T. V. (1998). *Statystychna fizyka – osnovni polozhennia ta modeli* [Statistical physics – basic principles and models]. Cherkasy: ChDU [in Ukrainian].
6. Bortz, A. B., Kalos, M. H., Lebowitz, J. L., Zendejas, M. A. (1974). Time evolution of a quenched binary alloy: Computer simulation of a two-dimensional model system. *Physical Review B*, 10, 535-541. Retrieved from <https://doi.org/10.1103/PhysRevB.10.535>
7. Marro, J., Bortz, A. B., Kalos, M. H., Lebowitz, J. R. (1975). Time evolution of a quenched binary alloy. II. Computer simulation of a three-dimensional model system. *Physical Review B*, 12, 2000-2011. Retrieved from <https://doi.org/10.1103/PhysRevB.13.4328>.
8. Flinn, P. A., McManus, G. M. (1961). Monte Carlo calculation of the order-disorder transformation in the body-centered cubic lattice. *Physical Review*, 124(1), 54-59. Retrieved from <https://doi.org/10.1103/PhysRev.124.54>.
9. Beller, J. R., Delaney, J. A. (1963). Order-Disorder Events Produced by Single Vacancy Migration. *Physical Review*, 130, 962-971. Retrieved from <https://doi.org/10.1103/PhysRev.130.962>.
10. Beller, J. R. (1965). Order-Disorder Events Produced by Single-Vacancy Migration. II. *Physical Review*, 138, A1259-A1270. Retrieved from <https://doi.org/10.1103/PhysRev.138.A1259>.
11. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21, 1087-1092. Retrieved from <https://doi.org/10.1063/1.1699114>.
12. Liu, Z.-J., Shen, Y. G. (2004). Effects of amorphous matrix on the grain growth kinetics in two-phase nanostructured films: a Monte Carlo study. *Acta Materialia*, 52, 729-736. Retrieved from <https://doi.org/10.1016/j.actamat.2003.10.010>.
13. Glauber, R. J. (1963). Time-dependent statistics of the Ising model. *Journal of Mathematical Physics*, 4, 294-307. Retrieved from <https://doi.org/10.1063/1.1703954>.
14. Oramus, P., Kozubski, R. (2001). Monte Carlo computer simulation of order-order kinetics in the L12-ordered Ni₃Al binary system. *Physical Review B*, 63, 174109. Retrieved from <https://doi.org/10.1103/PhysRevB.63.174109>.
15. Oramus, P., Kozubski, R., Perron-Bohnes, V., Cadeville, M. C., Massobrio, C., Pfeiler, W. (2002). Computer simulation of long-range order relaxation in homogeneous systems. *Materials Science and Engineering A*, 324, 11-15. Retrieved from [https://doi.org/10.1016/S0921-5093\(01\)01275-8](https://doi.org/10.1016/S0921-5093(01)01275-8).
16. Young, W. M., Elcoc, E. (1966). Monte Carlo studies of vacancy migration in binary ordered alloys. *Proceedings of the Physical Society*, 89, 735-746. Retrieved from <https://doi.org/10.1088/0370-1328/89/3/329>.
17. Bortz, A. B., Kalos, M. H., Lebowitz, J. L. (1975). A new algorithm for Monte Carlo simulation of Ising spin systems. *Journal of Computational Physics*, 17, 10-18. Retrieved from [https://doi.org/10.1016/0021-9991\(75\)90060-1](https://doi.org/10.1016/0021-9991(75)90060-1).
18. Athenes, M., Bellon, P., Martin, G., Haider, F. (1996). A Monte-Carlo study of B2 ordering and precipitation via vacancy mechanism in b.c.c. lattices. *Acta Materialia*, 44, 4739-4748. Retrieved from [https://doi.org/10.1016/S1359-6454\(96\)00135-8](https://doi.org/10.1016/S1359-6454(96)00135-8).
19. Athenes, M., Bellon, P., Martin, G. (1997). Identification of novel diffusion cycles in B2 ordered phases by Monte Carlo simulation. *Philosophical Magazine A*, 76, 565-585. Retrieved from <https://doi.org/10.1080/01418619708214023>.
20. Haider, F. (1992). Monte-Carlo study of the phase stability of ordered fcc phases under irradiation. *Ordering and Disorder in Alloys*; Edited by A. R. Yavari. London: Elsevier Applied Science, 215-222. Retrieved from https://doi.org/10.1007/978-94-011-2886-5_23.
21. De Fontane, D. (1994). Cluster approach to order-disorder transformations in alloys. *Solid State Physics*, 47, 33-176. Retrieved from [https://doi.org/10.1016/S0081-1947\(08\)60639-6](https://doi.org/10.1016/S0081-1947(08)60639-6).
22. Wolverson, C., Ozolinis, V., Zunger, A. (2000). Short-range-order types in binary alloys: a reflection of coherent phase stability. *Journal of Physics: Condensed Matter*, 12, 2749-2768. Retrieved from <https://doi.org/10.1088/0953-8984/12/12/314>.
23. Staunton, J. B., Johnson, D. D., Pinski, F. J. (1994). Compositional short-range ordering in metallic alloys: band-filling, charge-transfer, and size effects from a first-principles all-electron Landau-type theory. *Physical Review B*, 50, 1450-1472. Retrieved from <https://doi.org/10.1103/PhysRevB.50.1450>.
24. Zaporozhets, T. V., Sobchenko, I. V., Gusak A. M. (2005). 3D-Simulation of Void Formation, Growth and Migration under Electromigration. *Defect and Diffusion Forum*, 237, 1306-1311. Retrieved from <https://doi.org/10.4028/www.scientific.net/DDF.237-240.1306>.
25. Tu, K. N., Gusak, A. M., Sobchenko, I. (2003). Linear rate of grain growth in thin films during deposition. *Physical Review B*, 67, 245408. Retrieved from <https://doi.org/10.1103/PhysRevB.67.245408>.
26. Sobchenko, I. V., Gusak, A. M., Tu, K. N. (2005). 3D Monte-Carlo model of deposition and grain growth in thin films. *Defect and Diffusion Forum*, 237, 1281-1286. Retrieved from <https://doi.org/10.4028/www.scientific.net/DDF.237-240.1281>.

27. Hin, C., Soisson, F., Maugis, Ph. (2005). Atomistic Monte Carlo simulations of homogeneous and heterogeneous precipitation on grain boundaries of NbC in steels. *Defect and Diffusion Forum*, 237, 721-726. Retrieved from <https://doi.org/10.4028/www.scientific.net/DDF.237-240.721>.
28. Rubio, J. E., Jaraiz, M., Martin-Bragado, I., Hernandez-Mangas, J. M., Barbolla, J. (2003). Atomistic Monte Carlo simulations of three-dimensional polycrystalline thin films. *Journal of Applied Physics*, 94, 163-168. Retrieved from <https://doi.org/10.1063/1.1577814>.
29. Gusak, A., Bogatyrev, A. O., Kovalchuk, A. O. (2001). Initial Stage of Reactive Diffusion: Nucleation and Avrami Kinetics. *Defect and Diffusion Forum*, 194, 1625-1630. Retrieved from <https://doi.org/10.4028/www.scientific.net/DDF.194-199.1625>.
30. Gusak, A. M., Kovalchuk, A. O. (1998). Oscillatory regime of ordering during interdiffusion. *Physical Review B*, 58, 2551-2555. Retrieved from <https://doi.org/10.1103/PhysRevB.58.2551>.
31. Gusak, A. M., Bogatyrev, A. O., Zaporozhets, T. V., Kovalchuk, A. A., Kornienko, S. V., Lutsenko, G. V., Liashenko, Yu. A., Shirinian, A. S. (2004). *Modeli tverdogfaznykh reaktsii [Models of solid-phase reactions]*. Cherkassy: ChNU [in Russian].
32. Pasichnyy, M. O., Gusak, A. M. (2005). Modeling of Phase Competition and Diffusion Zone Morphology Evolution at Initial Stages of Reaction Diffusion. *Defect and Diffusion Forum*, 237, 1193-1198. Retrieved from <https://doi.org/10.4028/www.scientific.net/DDF.237-240.1193>.
33. Pasichnyy, M. O. (2007). Monte-Karlo doslidzhennia pochatkovykh etapiv reaktsiinoi dyfuzii u multysharakh [Monte Carlo study of initial stages of reaction diffusion in multilayers]. *Visnyk Cherkaskoho universytetu. Seriya "Fyzyko-matematychni nauky"* – Bulletin of Cherkasy University. Series of Physical and Mathematical Sciences, 114, 65-72 [in Ukrainian].
34. Gusak, A. M., Pasichna, V. M. (2018) Modeling of Concentration and Temperature Dependencies of Incubation Time at Decomposition of Solid Solution by Monte Carlo Method. *Visnyk Cherkaskoho universytetu. Seriya "Fyzyko-matematychni nauky"* – Bulletin of Cherkasy University. Series of Physical and Mathematical Sciences, 1, 3-11. Retrieved from <https://doi.org/10.31651/2076-5851-2018-1-3-11>.

PASICHNYY Mykola,

Candidate of Physical and Mathematical Sciences, Associate Professor, Head of the Chair of Physics, The Bohdan Khmelnytsky National University of Cherkasy, Cherkasy, Ukraine,

MONTE-CARLO INVESTIGATION OF PHASE EQUILIBRIUM CURVES OF BINARY FCC ALLOYS WITH LIMITED SOLUBILITY

Summary. Introduction. *The phase equilibrium curves of binary FCC alloys with limited solubility of components were investigated by Monte Carlo lattice methods. The existence of a single phase diagram of the modeled system is confirmed. It does not depend on the diffusion mechanism and probability algorithm of Monte-Carlo methods. The linear dependences of the difference of the reduced temperatures of the phase equilibrium curves for the regular solid solution model and the modeled system based on the lattice Monte Carlo methods are approved.*

Purpose. *The purpose of the work is investigation of the phase equilibrium curves of binary FCC alloy with limited solubility of components by lattice Monte Carlo methods and analysis the results for different diffusion mechanisms and probability algorithms: the exchange and vacancy mechanisms for the Metropolis and Glauber algorithms and the vacancy mechanism for the Residence Time algorithm.*

Results. *Three-dimensional face-centered cubic lattice with 50x20x20 sites occupied by atoms of two species was used as a model system. The approximation of the first coordination shell was used for the simulation. Born-Karman periodic boundary conditions were used as boundary conditions. The diffusion couple method was used for calculation of equilibrium concentrations of the binary system with limited solubility of the components. The main thermodynamic parameter of the modeled system is the energy of mixing of binary solid solution. The series of computer simulations were performed using: Metropolis algorithm with exchange mechanism; Glauber algorithm with exchange mechanism; Metropolis algorithm with vacancy mechanism; Glauber algorithm with vacancy mechanism; Residence Time Algorithm with vacancy mechanism. As a result of modeling the kinetics of diffusion processes, the system is relaxed to equilibrium corresponding to the contact of two binary solid solutions. Significant difference of equilibrium concentrations at phase diagram curves was obtained between values modeled by lattice Monte-Carlo and a regular solid solution model. The values obtained by all considered Monte Carlo algorithms correspond to a single curve of phase equilibrium of the modeled binary system.*

Conclusion. In the paper, the phase equilibrium curves of binary FCC alloys with limited solubility of components were investigated by computer simulations by lattice Monte Carlo methods using the diffusion couple method. The Glauber and Metropolis algorithms based on exchange and vacancy diffusion mechanisms, as well as Residence Time Algorithm for the vacancy mechanism were used for the numerical simulation. The phase diagram of FCC binary system with limited solubility of components was constructed by lattice Monte Carlo algorithms based on the simulation of kinetics of diffusion processes at an atomic level. The existence of a single curve of phase equilibrium of the modeled binary system is confirmed for different lattice Monte Carlo methods. The linear dependences of the difference of the reduced temperatures of the phase equilibrium curves on the concentration for a model of a regular solid solution and a model binary system based on the lattice Monte Carlo method were obtained.

Keywords: Monte Carlo, binary alloy, regular solid solution, diffusion couple method.

Одержано редакцією 08.08.2019 р.
Прийнято до публікації 09.10.2019 р.

УДК 004.85:519.6

DOI 10.31651/2076-5886-2019-2-86-95

PACS 02.70.Wz, 07.05.Kf, 07.05.Mh,
07.05.Tr

ПІСКУН Олександр Варфоломійович
кандидат технічних наук, доцент,
завідувач кафедри прикладної математики
та інформатики, Черкаський національний
університет імені Богдана Хмельницького
e-mail: piskun@ukr.net
ORCID 0000-0001-5334-6337

АНАЛІЗ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ ЗАДАЧІ БІНАРНОЇ КЛАСИФІКАЦІЇ

У роботі був проведений аналіз існуючих найбільш поширених методів класифікації на предмет їх використання в задачі діагностики серцевих захворювань. Розглянуто основні метрики якості моделей бінарної класифікації, які можуть бути використані при ухваленні рішення про оптимальність розробленої моделі. Дослідження моделей проводились без і з оптимізацією параметрів. Оптимізація параметрів моделей проведена, використовуючи криві валідації з подальшим пошуком по сітці з крос-валідацією кожної комбінації параметрів. Найкращі результати показали методи *DecisionTreeClassifier*, *GradientBoostingClassifier* та *GaussianNB*.

Ключові слова: машинне навчання, метрики якості, бінарна класифікація, алгоритми класифікації

Вступ

Бінарна класифікація - одна з найбільш поширених проблем прикладної статистики та машинного навчання, яка вирішується в багатьох прикладних областях - в медицині, біології, метеорології, аналізі поштових повідомлень, класифікації текстів, зображень і т.д.

Розглянемо задачу бінарної класифікації об'єктів, в якій кожен об'єкт K_i ($i = 1, \dots, N$) характеризується m -мірним вектором ознак $(X_1 \dots X_m)$. Ці характеристики (або ознаки) можуть приймати як числові, так і нечислові значення та утворюють вибірку для подальших досліджень. Потрібно на підставі значень ознак передбачити вихідну характеристику об'єктів y , яка може приймати одне з двох значень (0 або 1).

Існує багато методів класифікації, такі як дерева рішень, нейронні мережі, байесовський класифікатор, метод опорних векторів, логістична регресія і ін., які використовують різний математичний апарат і різні підходи при реалізації [1]. Однак, їх ефективність залежить від конкретної задачі і на сьогоднішній день не існує методів,

які могли б однозначно ефективно вирішити задачу класифікації. Тому, доцільно проводити апробацію декількох методів бінарної класифікації та знаходити оптимальні для вирішення поставленої задачі.

Метою статті є знаходження найкращого алгоритму машинного навчання для вирішення задачі класифікації щодо визначення наявності або відсутності захворювань серця у людини на прикладі реальних даних.

Виклад основного матеріалу

Оцінка якості моделей класифікації є важливим аспектом у всіх областях, для яких розробляються моделі машинного навчання. Дана оцінка якості відповідає на питання, наскільки добре отриманий класифікатор розділяє класи, що нас цікавлять, на деякій вибірці.

Метрики якості моделей бінарної класифікації [2-4]

Стосовно багатокласової класифікації, метрика відображає точність класифікації для усіх класів разом узятих або для кожного окремого класу.

Accuracy. Нехай класифікатор видає мітку класу. Позначимо: y_i – мітка i -го об'єкту, \hat{y}_i – відповідь на цьому об'єкті нашого алгоритму, m – число об'єктів у вибірці, тоді частка об'єктів, по яким класифікатор прийняв правильне рішення, визначається як:

$$Accuracy(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^m I[\hat{y}_i = y_i].$$

Частка вірних відповідей не враховує ціни помилок.

Confusion matrix. Бінарна класифікація використовується в задачах, де об'єкти вибірки діляться на два класи – позитивні (positive) та негативні (negative). У свою чергу, сама модель бінарної класифікації привласнює об'єктам також дві мітки – positive або negative (рис. 1а). А оскільки модель буде працювати з помилками щодо тестової вибірки, то в результаті бінарної класифікації всі об'єкти вибірки розбиваються на чотири типи, утворюючи матрицю неточностей / помилок (confusion matrix) (рис. 1б):

- 1) істинно-позитивні (true positive – TP);
- 2) істинно-негативні (true negative – TN);
- 3) помилково-позитивні (false positive – FP) – помилка 1-го роду (type I error);
- 4) помилково-негативні (false negative – FN) – помилка 2-го роду (type II error).

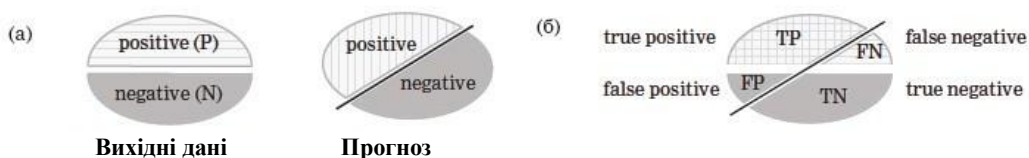


Рис. 1. Вихідні та прогнозовані дані бінарної класифікації (а) та матриця неточностей (б) [4]

Матриця неточностей бінарної класифікації - матриця розміру 2×2 , ij -я позиція якої дорівнює числу об'єктів i -го класу, яким алгоритм присвоїв мітку j -го класу (рис. 2).

| | | Predicted Labels | |
|-------------|---------|----------------------|----------------------|
| | | Class 0 | Class 1 |
| Real Labels | Class 0 | True Negatives (TN) | False Positives (FP) |
| | Class 1 | False Negatives (FN) | True Positives (TP) |

Рис. 2. Матриця неточностей для бінарної класифікації

Використовуючи наведену вище матрицю неточностей можна отримати кілька метрик якості моделі бінарної класифікації, які при цьому не є взаємовиключними, доповнюють одна одну та можуть бути використані в процесі прийняття рішення про оптимальну модель в кожному конкретному випадку. Наприклад, у медичній сфері помилка 1-го роду є найбільш критичною, тому що може бути краще поставити більш песимістичний діагноз, ніж більш оптимістичний [5].

Precision і Recall. Для оцінки якості роботи алгоритму на кожному з класів окремо вводяться метрики Precision (точність) і Recall (повнота):

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}.$$

Точність - це частка об'єктів, які дійсно належать позитивному класу щодо всіх об'єктів, віднесених класифікатором до цього класу.

Повнота - це частка знайдених класифікатором об'єктів, які належать позитивному класу щодо всіх об'єктів цього класу в тестовій вибірці.

F-міра. Зрозуміло, що чим вище точність і повнота, тим краще. Однак, максимальна точність та повнота недосяжні одночасно і доводиться шукати деякий баланс. F-міра об'єднує в собі інформацію про точність та повноту алгоритму.

Міра F_1 є гармонійне середнє між точністю та повнотою:

$$F_1 = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

При цьому, якщо ми хочемо віддати перевагу або точності, або повноті, можна скористатися розширеною версією F-міри, що має параметр β , який можна використовувати для балансування точності та повноти:

$$F_\beta = (1 + \beta^2) \frac{\text{Precision} \times \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}.$$

Опис набору даних

Для дослідження використовується набір даних (датасет) з серцевих захворювань Statlog (Heart) [6]. Дані містять 13 атрибутів (табл. 1) і 270 спостережень.

Таблиця 1

Атрибути даних з серцевих захворювань

| № | Назва | Опис |
|----|--------------------------------------|---|
| 1 | age | вік пацієнта в роках |
| 2 | sex | стать |
| 3 | chest_pain_type | тип болю в грудях |
| 4 | resting_blood_pressure | артеріальний тиск у спокої (мм рт. ст.) |
| 5 | serum_cholesterol_mg_per_dl | рівень холестерину в крові в мг/дл |
| 6 | fasting_blood_sugar_gt_120_mg_per_dl | рівень цукру в крові > 120 мг/дл |
| 7 | resting_ekg_results | ЕКГ результати в стані спокою |
| 8 | max_heart_rate_achieved | максимальна частота серцевих скорочень |
| 9 | exercise_induced_angina | стенокардія, викликана фізичними вправами |
| 10 | oldpeak_eq_st_depression | депресія сегменту ST на ЕКГ |
| 11 | slope_of_peak_exercise_st_segment | нахил пікового сегменту ST |
| 12 | num_major_vessels | кількість судин, забарвлених при флюороскопії |
| 13 | thal | тип дефекту |

Змінна для прогнозування y може приймати два значення: 0 – відсутність, 1 – наявність серцевих захворювань.

Дослідження проводимо з використанням бібліотеки машинного навчання Scikit-Learn, в середовищі Python 3.

Аналіз та підготовка даних

Завантажуємо дані та перевіряємо їх на наявність пропущених значень:

```
In [125]: X.isnull().sum()
Out[125]:
slope_of_peak_exercise_st_segment    0
thal                                   0
resting_blood_pressure                 0
chest_pain_type                       0
num_major_vessels                     0
fasting_blood_sugar_gt_120_mg_per_dl  0
resting_ekg_results                   0
serum_cholesterol_mg_per_dl           0
oldpeak_eq_st_depression              0
sex                                    0
age                                    0

max_heart_rate_achieved                0
exercise_induced_angina                0
dtype: int64
```

Пропусків немає.

Проводимо перевірку наявності дисбалансу цільових класів. Результати представлені на рис. 3.

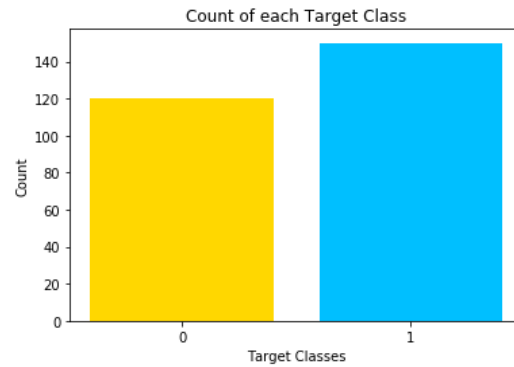


Рис. 3. Кількість об'єктів цільових класів

Класи можна вважати збалансованими (об'єктів класу «0» - 120, «1» - 150). Побудуємо гістограми атрибутів (рис. 4).

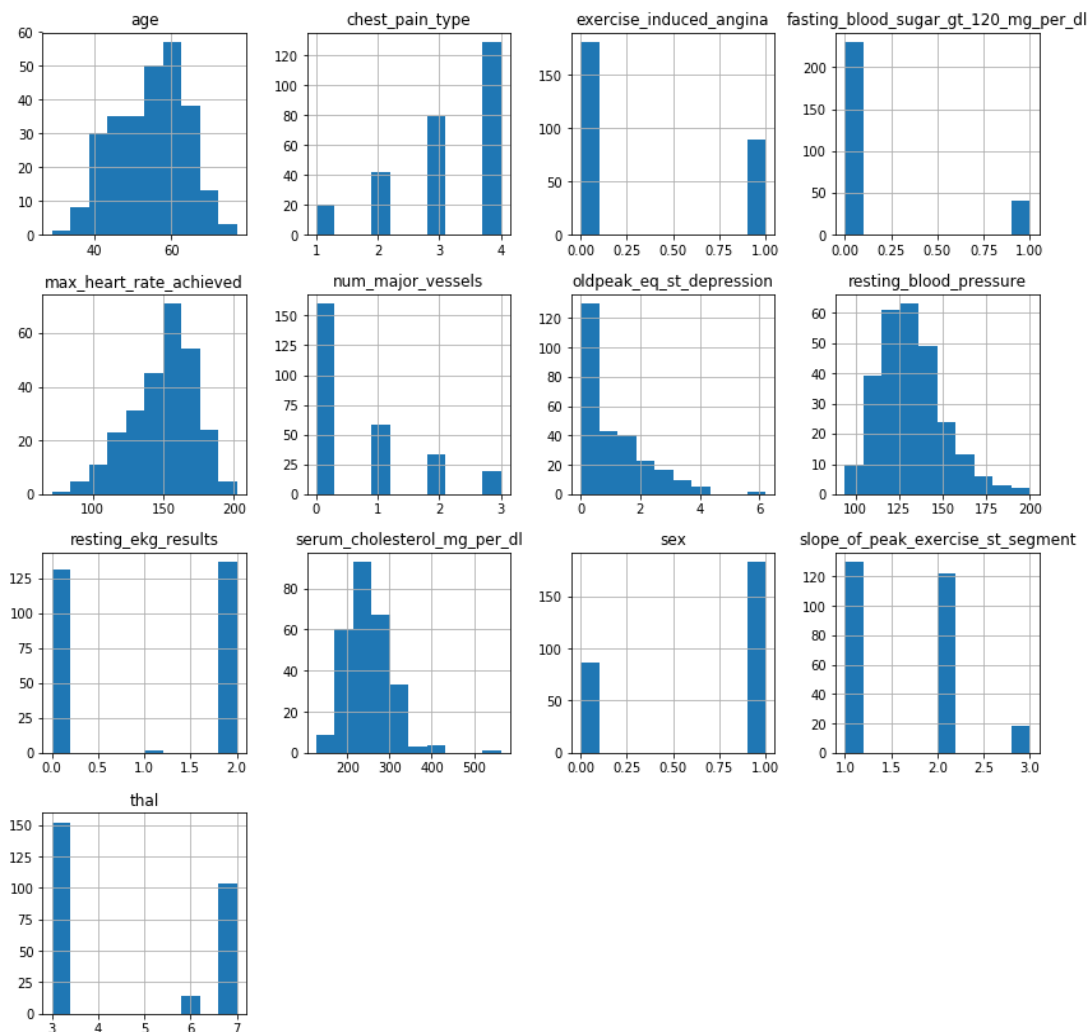


Рис. 4. Гістограми атрибутів

Як видно з гістограм, наш датасет включає в себе як кількісні, так і категоріальні дані, що вимагає їх попередньої обробки: категоріальні перевести в бінарні, а кількісні - нормалізувати.

Побудуємо матрицю кореляцій атрибутів (рис.5).

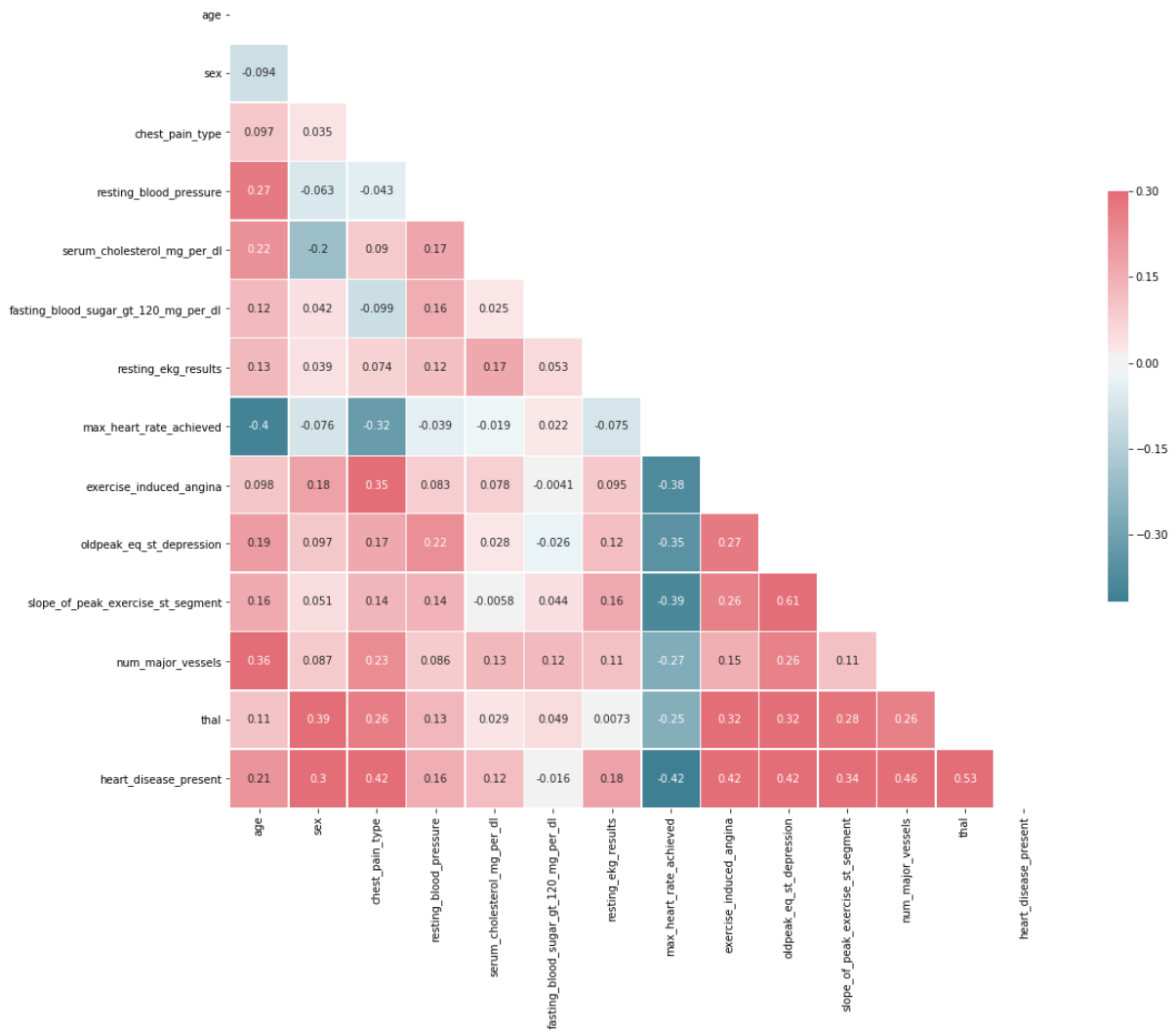


Рис. 5. Кореляційна матриця атрибутів

Як видно з матриці кореляцій, атрибути не мають між собою достатньо високих кореляцій, тому їх можна вважати незалежними.

Виконаємо попередню обробку даних: перетворимо категоріальні дані у бінарні, нормалізуємо усі числові до одного діапазону [0..1].

Для подальшого аналізу даних проведемо візуалізацію змінної у за допомогою методу t-SNE. T-Distributed Stochastic Neighbor Embedding (t-SNE) - це метод нелінійного зменшення розмірності, який добре підходить для візуалізації багатовимірних наборів даних. Метод моделює кожен об'єкт простору високої розмірності дво- або тривимірною точкою таким чином, що близькі за характеристиками елементи даних в багатовимірному просторі (наприклад, датасет з великим числом стовпців) проєктуються в сусідні точки, а різні об'єкти з великою ймовірністю моделюються точками, які стоять далеко одна від одної [7]. Результати представлені на рис. 6.

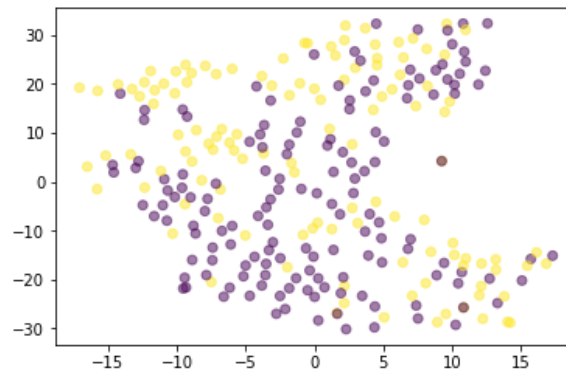


Рис. 6. T-SNE візуалізація цільових класів

Отримані результати показують, що ми маємо лінійно нероздільну вибірку.

Цільові класи не вдасться просто лінійно розділити, тому будемо застосовувати різні методи класифікації з подальшим порівнянням за допомогою відповідної метрики оцінки моделі.

Розділяємо вибірку на тренувальну та тестову (75% / 25%).

Застосуємо набір загально використовуваних класифікаторів [8] з параметрами по замовченню:

```
names = ['Logit', "Nearest Neighbors", "RBF SVM", "Gaussian Process", "Decision Tree", "Random Forest", "Multi-layer Perceptron", 'GradBoost', 'StochGradBoost', "AdaBoost", "Naive Bayes"]
```

```
classifiers = [
    LogisticRegression(),
    KNeighborsClassifier(),
    SVC(kernel="rbf", probability=True),
    GaussianProcessClassifier(1.0 * RBF(1.0)),
    DecisionTreeClassifier(),
    RandomForestClassifier(),
    MLPClassifier(),
    GradientBoostingClassifier(),
    SGDClassifier(),
    AdaBoostClassifier(),
    GaussianNB()]
```

Recall є для нас основною метрикою, так як невиявлення хвороби при її наявності може призвести до смерті пацієнта. Якщо ж діагноз буде поставлено здоровій людині, то додаткове обстеження дасть можливість виявити помилку без загрози життю людини. Тому далі ми будемо використовувати ассигасу, як метрику для оцінки ефективності моделі загалом, та recall – при вирішенні поставленої задачі.

Розраховуємо точність і повноту на тренувальному та тестовому наборах з крос-валідацією по 10 вікнах. Таким чином, ми отримаємо середнє значення точності і повноти моделі та їх стандартні відхилення, що дасть уявлення про стабільну ефективність моделі і знизить вплив вибірки на результат.

Отримані результати точності та повноти для моделей преставлені в табл. 2.

Значення точності і повноти недоцільно низькі для нашої задачі та до того ж мають великий розкид значень. Recall на тестовій вибірці є нижчим за 80%, тобто більше 200 з 1000 хворих пацієнтів віднесені до здорових, тільки DecisionTree Classifier і GaussianNB дали результати 84% і 93% відповідно, але при розкиді значень у 16% і

13%. Це говорить про істотну залежність результатів від вибірки та вимагає підвищення стабільності роботи моделей.

Таблиця 2

Значення метрик точності та повноти для моделей моделей з параметрами по замовченню на тренувальній та тестовій виборках

| Classifier (Default parameters) | Accuracy Train | Acc Tr Std | Accuracy Test | Acc Test Std | Recall Train | Recall Tr Std | Recall Test | Recall Test Std |
|---------------------------------|----------------|------------|---------------|--------------|--------------|---------------|-------------|-----------------|
| LogisticRegression | 0,85 | 0,05 | 0,78 | 0,09 | 0,80 | 0,10 | 0,74 | 0,20 |
| KNeighbors Classifier | 0,85 | 0,06 | 0,75 | 0,17 | 0,80 | 0,14 | 0,64 | 0,24 |
| SVC | 0,85 | 0,05 | 0,78 | 0,13 | 0,79 | 0,13 | 0,74 | 0,20 |
| GaussianProcess Classifier | 0,87 | 0,05 | 0,76 | 0,10 | 0,81 | 0,11 | 0,71 | 0,23 |
| DecisionTree Classifier | 0,76 | 0,05 | 0,81 | 0,13 | 0,78 | 0,14 | 0,84 | 0,16 |
| RandomForest Classifier | 0,82 | 0,04 | 0,75 | 0,11 | 0,74 | 0,12 | 0,64 | 0,18 |
| MLPClassifier | 0,86 | 0,07 | 0,75 | 0,11 | 0,83 | 0,09 | 0,74 | 0,20 |
| GradientBoosting Classifier | 0,82 | 0,07 | 0,79 | 0,10 | 0,80 | 0,12 | 0,71 | 0,23 |
| SGDClassifier | 0,78 | 0,05 | 0,69 | 0,13 | 0,70 | 0,17 | 0,67 | 0,26 |
| AdaBoostClassifier | 0,79 | 0,07 | 0,72 | 0,21 | 0,77 | 0,10 | 0,77 | 0,21 |
| GaussianNB | 0,80 | 0,08 | 0,63 | 0,15 | 0,61 | 0,20 | 0,93 | 0,13 |

Виконаємо оптимізацію параметрів для усіх застосованих моделей. Оптимізація проводиться, використовуючи криві валідації з подальшим пошуком по сітці з крос-валідацією кожної комбінації параметрів [9]. Результати представлені в табл. 3.

Таблиця 3

Значення метрик точності та повноти для моделей з оптимізованими параметрами на тренувальній та тестовій виборках

| Classifier (Best parameters) | Accuracy Train | Acc Tr Std | Accuracy Test | Acc Test Std | Recall Train | Recall Tr Std | Recall Test | Recall Test Std |
|------------------------------|----------------|------------|---------------|--------------|--------------|---------------|-------------|-----------------|
| LogisticRegression | 0,86 | 0,05 | 0,79 | 0,07 | 0,82 | 0,11 | 0,78 | 0,15 |
| KNeighbors Classifier | 0,86 | 0,06 | 0,76 | 0,16 | 0,81 | 0,11 | 0,67 | 0,26 |
| SVC | 0,87 | 0,05 | 0,78 | 0,09 | 0,83 | 0,11 | 0,74 | 0,20 |
| GaussianProcess Classifier | 0,85 | 0,07 | 0,79 | 0,10 | 0,78 | 0,16 | 0,71 | 0,18 |
| DecisionTree Classifier | 0,76 | 0,05 | 0,81 | 0,13 | 0,78 | 0,14 | 0,84 | 0,16 |
| RandomForest Classifier | 0,82 | 0,05 | 0,82 | 0,09 | 0,79 | 0,13 | 0,80 | 0,16 |
| MLPClassifier | 0,85 | 0,05 | 0,72 | 0,10 | 0,80 | 0,12 | 0,78 | 0,21 |
| GradientBoosting Classifier | 0,85 | 0,06 | 0,82 | 0,13 | 0,84 | 0,09 | 0,87 | 0,16 |
| SGDClassifier | 0,83 | 0,04 | 0,75 | 0,09 | 0,82 | 0,17 | 0,80 | 0,22 |
| AdaBoostClassifier | 0,82 | 0,06 | 0,82 | 0,15 | 0,82 | 0,09 | 0,83 | 0,22 |
| GaussianNB | 0,81 | 0,08 | 0,64 | 0,16 | 0,64 | 0,21 | 0,93 | 0,13 |

Як видно з табл. 3, оптимізація параметрів не призвела до суттєвого покращення якості моделей, за винятком GradientBoostingClassifier, яка посіла друге місце після GaussianNB. Так як ми маємо сбалансований набір даних без викидів, вибірка, швидше за все, недостатньо велика для забезпечення оптимального тренування моделей.

Таким чином, ми можемо виділити три моделі, які найбільш підходять для даної конкретної задачі - DecisionTree Classifier, GradientBoosting Classifier та GaussianNB з показниками recall 0,84; 0,87 та 0,93 відповідно. Стандартні відхилення 0,16; 0,16 та 0,13 вказують на чутливість моделей до вибірки даних.

Висновки

В роботі розглянуто та вибрано метрики якості моделей бінарної класифікації з урахуванням поставленого завдання. Для оцінки ефективності моделі загалом використовувалась метрика асигасу, при вирішенні поставленої задачі – recall.

Досліджено 11 найбільш поширених методів класифікації на предмет їх використання для вирішення задачі класифікації щодо визначення наявності або відсутності захворювань серця у людини на прикладі реальних даних. Отримано показники якості роботи моделей (з кросс-валидацією по 10 вікнам) з параметрами по замовченню та з оптимізацією параметрів. Найкращі результати показали методи DecisionTreeClassifier, GradientBoostingClassifier та GaussianNB, але вони не можуть бути визнані достатніми в медичній практиці.

Майбутніми дослідженнями будуть детальний розгляд моделей DecisionTreeClassifier, GradientBoostingClassifier та GaussianNB, зокрема, знаходження причин розкиду значень метрики recall для його зниження і, відповідно, підвищення стабільності точності моделей. На основі даних трьох методів буде побудована асамблеяна класифікація розпізнавання хвороби серця, що може підвищити точність моделі.

Список використаної літератури:

1. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? / M.F. Delgado, E. Cernadas, S. Barro, D. Amorim // *Journal of Machine Learning Research*. – 2014. – V. 15. – P. 3133-3181.
2. Лабинцев, Е. Метрики в задачах машинного обучения [Електронний ресурс] / Е. Лабинцев. – Режим доступу: <https://habr.com/ru/company/ods/blog/328372/>
3. Narkhede, S. Understanding Confusion Matrix [Електронний ресурс] / S. Narkhede. – Режим доступу: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>
4. Афанасьев, С. Gini & ROC & Precision-Recall: проблемы метрик в банковском моделировании [Електронний ресурс] / С. Афанасьев, А. Смирнова. – Режим доступу: <http://futurebanking.ru/post/3761>
5. *Encyclopedia of Machine Learning* / C. Sammut and G.I. Webb, Eds. – New York: Springer, 2011. – 892 p.
6. Statlog (Heart) Data Set [Електронний ресурс]. – Режим доступу: [http://archive.ics.uci.edu/ml/datasets/statlog+\(heart\)](http://archive.ics.uci.edu/ml/datasets/statlog+(heart)).
7. Van der Maaten, L.J.P.. Visualizing High-Dimensional Data Using t-SNE / L.J.P. van der Maaten, G.E. Hinton // *Journal of Machine Learning Research*. – 2008. – № 9. – P. 2579 – 2605.
8. Supervised learning [Електронний ресурс]. – Режим доступу: https://scikit-learn.org/stable/supervised_learning.html
9. Піскун, О.В. Застосування методів машинного навчання для побудови моделі рішення задачі класифікації / О.В. Піскун // *Вісник Черкаського університету. Серія: Прикладна математика. Інформатика*. – 2019. - №1. – С. 41-52.

References:

1. Delgado, M.F., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15, 3133-3181.
2. Labintsev, E. (2017). *Metriki v zadachah mashinnogo obucheniya* [Machine Learning Metrics]. Retrieved from <https://habr.com/ru/company/ods/blog/328372/> [in Russian].
3. Narkhede, S. (2018). Understanding Confusion Matrix. Retrieved from <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

4. Afanasev, S., Smirnova, A. (2019). Gini & ROC & Precision-Recall: problemy metrik v bankovskom modelirovanii [Gini & ROC & Precision-Recall: problems of metrics in banking modeling]. Retrieved from <http://futurebanking.ru/post/3761> [in Russian].
5. Sammut, C. & Webb, G.I. (Eds.). (2011). Encyclopedia of Machine Learning. New York: Springer.
6. Statlog (Heart) Data Set. (n.d.). Retrieved from [http://archive.ics.uci.edu/ml/datasets/statlog+\(heart\)](http://archive.ics.uci.edu/ml/datasets/statlog+(heart)).
7. Van der Maaten, L.J.P., & Hinton, G.E. (2008) Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research*, 9(Nov), 2579–2605.
8. Supervised learning. (2019). Retrieved from https://scikit-learn.org/stable/supervised_learning.html
9. Piskun, O.V. (2019). Zastosuvannya metodiv mashinnogo navchannya dlya pobudovi modeli rishennya zadachi klasifikaciyi [Application of machine learning methods to build a model for solving the classification problem]. *Visnik Cherkaskogo universitetu. Seriya: Prikladna matematika. Informatika – Cherkasy University Bulletin: Applied Mathematics. Informatics*, 1, 41–52 [in Ukrainian].

PISKUN Oleksandr,

Candidate of Technical Sciences, Associate Professor, Chair of Department of Applied Mathematics and Informatics, Bohdan Khmelnytsky National University of Cherkasy

ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR THE BINARY CLASSIFICATION PROBLEM

Summary. Introduction. The analysis of existing common methods of classification applied to the problem of the heart disease diagnostics was provided in the present paper. Various quality metrics of binary classification models that can be used in deciding on the optimality of the developed model are considered. Modelling was conducted with and without parameter optimization. Parameter tuning was performed using validation curves, followed by grid search with cross-validation of each parameter combination. Decision Tree Classifier, Gradient Boosting Classifier and Gaussian Naive Bayes showed the best results for our particular problem.

Binary classification is one of the most common problems of applied statistics and machine learning, which is present in many applied fields - in medicine, biology, meteorology, analysis of mail messages, classification of texts, images, etc.

Assessment of the quality of classification models is an important aspect in all areas to which machine learning models are applied. Accuracy Score answers the question of how well a classifier separates the classes we are interested in over some sample.

Purpose. The purpose of the present paper is to find the optimal machine learning algorithm to solve the problem of classification for determining the presence or absence of heart disease on the basis of real data.

Results. A set of commonly used classifiers is investigated on real data of heart disease. Models accuracy scores (with 10 windows cross-validation) with default and optimized parameters were obtained. The accuracy metric was generally used to evaluate the performance of the model, and recall score - for our particular problem.

Conclusion. The analysis of common methods of classification applied to the problem of the heart disease diagnostics was carried out in the present work. Different quality metrics of binary classification models effectiveness that can be used in deciding on the optimality of the model were considered. Decision Tree Classifier, Gradient Boosting Classifier and Gaussian Naive Bayes showed the best results for our particular problem, but they are still not effective enough to be applied in the medical sector. Since we have a balanced dataset with no outliers, it is most likely that the sample is not large enough for the optimal model training.

Future studies will provide a detail analysis of the selected models, in particular, finding reasons for the volatility of recall metric values to reduce it and, consequently, improve the stability of model accuracy. Based on these three methods, an assembly classification of heart disease recognition will be built, which might improve the accuracy of the model.

Keywords: machine learning, accuracy metrics, binary classification, classification algorithms.

*Одержано редакцією 22.05.2019 р.
Прийнято до публікації 09.10.2019 р.*

УДК 004.421

DOI 10.31651/2076-5886-2019-2-96-103

УЖВА Дмитро Геннадійович
магістрант спеціальності «Прикладна
математика» Черкаського національного
університету імені Богдана
Хмельницького
e-mail: kraativ@gmail.com
ORCID 0000-0003-0073-3928

МЕТОД ОРГАНІЗАЦІЇ КРОСПЛАТФОРМЕННОГО ПРОЕКТУ

У статті розглядається метод створення кросплатформного проекту для подальшого створення програм для різних платформ з використанням спільного програмного коду. Велика кількість різноманітних платформ та операційних систем вимагають створення нових методів та технологій розробки коду. Кросплатформні рішення дозволяють швидше та менш затратно отримати готовий продукт, що буде доступний користувачам з різними цільовими машинами, що є важливим фактором для програм та ігор. Використовується середовище програмування IntelliJ IDEA, система автоматичного збирання Gradle та мова програмування Kotlin, а також середовище XCode для запуску на платформі iOS.

Ключові слова: кросплатформне програмування, kotlin, gradle.

Вступ. У наш час існує багато платформ для різних пристроїв, наслідком чого є створення нових методів та технологій створення коду, який можна використати на багатьох платформах, що значно прискорює розробку під велику кількість аудиторії та скорочує витрати.

Метою статті є ознайомлення з одним з підходів до створення кросплатформного проекту та перевикористанням коду, використовуючи мову програмування Kotlin та технологій Kotlin Native, Kotlin Multiplatform.

Виклад основного матеріалу

Для створення проекту використовується програмне середовище IntelliJ IDEA, що має вбудовану підтримку мови програмування Kotlin. Це середовище дозволяє автоматично створювати один з запропонованих варіантів кросплатформних “helloworld” проектів, що відрізняються платформами та призначенням. Такі проекти використовують систему автоматичного збирання Gradle, що має спеціальні плагіни для мови програмування Kotlin, технологій кросплатформного програмування Kotlin Multiplatform та нативної компіляції Kotlin Native.

Кросплатформне рішення за допомогою технологій Kotlin Multiplatform та Kotlin Native дозволяє зробити значну частину коду, у спільній частині, залишивши для реалізації на кожній платформі окремо специфічних частин. Прикладами задач, які потребують специфічних рішень є інтерфейси мобільних додатків для Android та iOS, де використовуються різні компоненти, або робота з графікою OpenGL, Direct3D.

Для створення потрібного Gradle проекту були розглянуті та розібрані стандартні проекти, що створює програмне середовище IntelliJ IDEA, серед них:

- Multiplatform Library – проект, що дозволяє перевикористовувати код котліна між трьома основними платформами: JVM, JS і Native;
- Mobile Shared Library – проект, що дозволяє будувати кросплатформну бібліотеку для мобільних платформ Android і iOS
- Mobile Android/iOS - проект, що дозволяє створювати додатки, перевикористовуючи код котліна між мобільними платформами Android і iOS;

Після дослідження проектів, що пропонує IntelliJ IDEA було створено проект, що містить два кросплатформних модулі library та app - бібліотеку та додаток, що її використовує. Кожен з цих модулів має під модулі для кожної платформи, а також модуль зі спільним кодом. Загальна структура проекту подана на рис. 1.

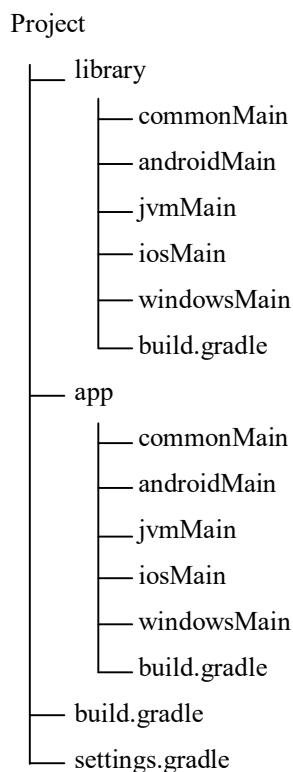


Рис. 1

Кореневий файл build.gradle містить репозиторії бібліотек та залежності інструментів та бібліотек, а settings.gradle назви модулів, ці файли використовують стандартні рішення при створенні Gradle проектів.

Файли build.gradle модулів core та run використовують плагін Kotlin Multiplatform. Плагін дозволяє розділяти налаштування та залежності для кожної платформи в замиканні (Closure) kotlin. Використовуючи замикання kotlin {} потрібно вказати цільові платформи та назви модулів, а також додати залежності бібліотек Kotlin та довільні потрібні вираховуючи замикання kotlin.sourceSets.

За замовчуванням Kotlin Multiplatform має модуль для спільного коду, а у замиканні kotlin.sourceSets вказана залежність від основної бібліотеки Kotlin:

```

[app,library]/build.gradle - Groove code
apply plugin: "org.jetbrains.kotlin.multiplatform"

kotlin {
    sourceSets {
        commonMain {
            dependencies {
                implementation kotlin("stdlib-common")
            }
        }
    }
}

```

Початкове налаштування скриптів `build.gradle` додатку та бібліотеки відрізняється лише залежністю додатка від бібліотеки, тому у файлі `app/build.gradle` потрібно вказувати залежність від бібліотеки для кожної платформи:

```
dependencies {
    implementation kotlin("stdlib-common")
    implementation project(":library")
}
```

Для додання наступної платформи необхідно додати потрібну платформу в замиканні `kotlin` та вказати залежності від основних бібліотек котліна, для того щоб мати доступ до реалізації основних функцій на конкретній платформі. Важливо підмітити те, що бібліотеки для конкретної платформи можуть мати різні реалізації, наприклад для платформи JVM використовують «Kotlin Standard Library», «Kotlin Standard Library JDK 7» або «Kotlin Standard Library JDK 8», останні з них відрізняються додатковими функціями та змінами для підтримки та сумісності з цільовими платформами.

Додамо платформу JVM та залежність Kotlin Standard Library JDK 8 до проекту:

```
[app,library]/build.gradle - Groove code
kotlin {
    jvm()

    sourceSets {
        commonMain {
            dependencies {
                implementation kotlin("stdlib-common")
            }
        }
        jvmMain {
            dependencies {
                implementation kotlin("stdlib-jdk8")
            }
        }
    }
}
```

Для написання JVM проектів зазвичай використовуються залежності у вигляді JAR архівів, що містять скомпільовані класи, тому для того щоб наша програма після збірки мала можливість запускатись без додаткових файлів, потрібно налаштувати Gradle додавати бібліотеки до архіву вихідної програми. Також для запуску JAR архіву як програми потрібно вказати головний клас. Виконаємо ці кроки у файлі `build.gradle` для додатку:

```
app/build.gradle - Groove code
kotlin {
    jvm()

    sourceSets {
        . . .
    }

    configure(jvm()) {
        jvmJar {
            withJava()
            manifest {
                attributes "Main-Class": "com.example.app.Main"
            }
        }
    }
}
```



```

dependsOn framework.linkTask
doLast {
    def srcFile = framework.outputFile
    def targetDir = getProperty("configuration.build.dir")
    copy {
        from srcFile.parent
        into targetDir
        include "app.framework/**"
        include "app.framework.dSYM"
    }
}
}

```

У середовищі XCode потрібно створити новий або відкрити вже існуючий проект, перейти до етапів збірки (Build Phases), та створити новий скрипт, що буде запускати Gradle task:

```

Run Script - Command line code
# move to the xcode-frameworks folder
cd "$SRCROOT/../shared/build/xcode-frameworks"

# run the gradle task
./gradlew :library: buildIOSFramework -
PXCODE_CONFIGURATION=${CONFIGURATION}

```

Тепер потрібно впевнитися, що цей етап збірки виконується раніше ніж компілюються вихідні файли проекту середовища XCode, після чого додаток готовий до запуску.

Перш ніж додавати платформу Android до плагіну Kotlin Multiplatform, необхідно підключити та налаштувати плагіни Android для додатку та для бібліотеки. В даному випадку їх налаштування відрізняється лише відсутністю ідентифікатора додатка у бібліотеки. Додамо підключення та налаштування плагіну перед налаштуванням плагіну Kotlin Multiplatform, вказуючи потрібні версії бібліотек та додатку:

```

[app,library]/build.gradle - Groove code
apply plugin: "org.jetbrains.kotlin.multiplatform"
// Для додатку та бібліотеки використовуємо різні плагіни
// apply plugin: "com.android.application"
apply plugin: "com.android.library"

android {
    compileSdkVersion 29

    defaultConfig {
        // Для додатку потрібно вказати ідентифікатор
        // applicationId "com.example.app"
        minSdkVersion 14
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"
    }
}

kotlin {
    . . .
}

```

Додаємо платформу Android до плагіну Kotlin Multiplatform:

```
[app,library]/build.gradle - Groove code
kotlin {
    android()
    iosX64("ios")
    jvm()
    mingwX64("windows")

    sourceSets {
        commonMain {
            dependencies {
                implementation kotlin("stdlib-common")
            }
        }
        androidMain {
            dependencies {
                implementation kotlin("stdlib")
            }
        }
        iosMain {
            dependencies {
                implementation kotlin("stdlib")
            }
        }
        jvmMain {
            dependencies {
                implementation kotlin("stdlib-jdk8")
            }
        }
        windowsMain {
            dependencies {
                implementation kotlin("stdlib")
            }
        }
    }
}
```

Плагіни налаштовані та готові до збірки проекту. Для перевірки результату напишемо спільний код, що використовуватиме код, що знаходиться в модулях платформ. Спільна частина бібліотеки:

library/src/commonMain/kotlin/CommonSample.kt - Kotlin code

```
expect object Platform {
    val name: String
}
```

```
fun hellotext(): String = "Hello from ${Platform.name}!"
```

Платформозалежна частина:

library/src/jvmMain/kotlin/JVMSample.kt - Kotlin code

```
actual object Platform {
    actual val name: String = "JVM"
}
```

library/src/androidMain/kotlin/AndroidSample.kt - Kotlin code

```
actual object Platform {
    actual val name: String = "Android"
```

```
}
```

```
library/src/iosMain/kotlin/IOSSample.kt - Kotlin code
```

```
actual object Platform {  
    actual val name: String = "iOS"  
}
```

```
library/src/windowsMain/kotlin/WindowsSample.kt - Kotlin code
```

```
actual object Platform {  
    actual val name: String = "Windows"  
}
```

У настільних платформах точкою входу в програму є функція `main()`, в той час як на мобільних платформах створюються специфічні екрани. Для цього ми створюємо потрібні речі у платформозалежних модулях програми. У випадку з настільними платформами запуск та використання кросплатформної бібліотеки виглядає так:

```
app/src/windowsMain/kotlin/WindowsApp.kt - Kotlin code
```

```
object Main {  
    fun main() {  
        println(hellotext())  
    }  
}
```

У потрібному місці, в даному випадку у функції `main`, викликається кросплатформна функція `hellotext`, яка, в залежності від платформи, працює з різними об'єктами `Platform`. В результаті виклику функції `hellotext` отримаємо рядок з текстом, частина якого відрізняється на кожній платформі:

Output:

```
JVM: Hello from JVM!  
Android: Hello from Android!  
iOS: Hello from iOS!  
Windows: Hello from Windows!
```

Отримано проект, що дозволяє будувати кросплатформну бібліотеку та додаток, що її використовує. Бібліотека та додаток можуть мати спільну та окрему, платформозалежну частину коду.

Висновки. Мова програмування Kotlin має плагіни Kotlin Multiplatform та Kotlin Native, які дозволяють створити кросплатформний проект, писати спільний код для всіх платформ та специфічний для окремих. Отримані додатки запускаються на популярних платформах Android, JVM, iOS, Windows, тощо.

Список використаної літератури:

1. Building Multiplatform Projects with Gradle - Kotlin Programming Language [Електронний ресурс] – Режим доступу: <https://kotlinlang.org/docs/reference/building-mpp-with-gradle.html>
2. Building your first Kotlin Multiplatform app—Getting Started [Електронний ресурс] – Режим доступу: <https://vivekc.xyz/building-your-first-kotlin-multiplatform-app-getting-started-8ad10d7d4e9f>
3. Which standard library to use in Kotlin [Електронний ресурс] – Режим доступу: <https://stackoverflow.com/questions/51858596/which-standard-library-to-use-in-kotlin/>
4. Configure your build | Android Developers [Електронний ресурс] – Режим доступу: <https://developer.android.com/studio/build>
5. The Build Process objc.io [Електронний ресурс] – Режим доступу: <https://www.objc.io/issues/6-build-tools/build-process/>

Bibliography:

1. Building Multiplatform Projects with Gradle - Kotlin Programming Language. Retrieved from: <https://kotlinlang.org/docs/reference/building-mpp-with-gradle.html>
2. Building your first Kotlin Multiplatform app—Getting Started. Retrieved from: <https://vivekc.xyz/building-your-first-kotlin-multiplatform-app-getting-started-8ad10d7d4e9f>
3. Which standard library to use in Kotlin. Retrieved from: <https://stackoverflow.com/questions/51858596/which-standard-library-to-use-in-kotlin/>
4. Configure your build | Android Developers. Retrieved from: <https://developer.android.com/studio/build>
5. The Build Process objc.io. Retrieved from: <https://www.objc.io/issues/6-build-tools/build-process/>

UZHVA Dmitry,

student, The Bohdan Khmelnytsky National University of Cherkasy

METHOD OF ORGANIZATION OF THE CROSS-PLATFORM PROJECT

Summary. Introduction. *A large number of different platforms and operating systems require the creation of new code creation techniques and technologies. Cross-platform solutions make it faster and less expensive to get a ready-made product that is accessible to users with different targeting machines, which is an important factor for applications and games.*

There is a way to create and organize a cross-platform project to create programs for different platforms using a common source code that uses the IntelliJ IDEA programming environment, Gradle auto-build, and Kotlin programming language, as well as XCode to run on the iOS platform.

This method allows you to split the settings and dependencies for each platform or use common ones for all, allowing you to write the most effective code for a specific platform and reuse as much as possible.

The purpose of this paper is to define the method that allows you to create and organize a project for writing cross-platform apps and libraries in Kotlin programming language, to explain how tools and plugins works with cross-platform code and controlling it.

Results. *We have a project that allows us to write cross-platform library and the application that uses it. The library and application has a code that can be shared and separated between platforms, architectures. Each compilation is accompanied by a default source set that stores sources and dependencies specific to that compilation. A Kotlin source set is a collection of Kotlin sources, along with their resources, dependencies, and language settings, which may take part in Kotlin compilations. some platforms require specific setup, use, or launch, but this is all automated in the process of setting up a project build.*

We also have a simple code for an example of how to write cross-platform code, use a common objects to access platform-specific code and try it.

Conclusion. *The Kotlin programming language has Kotlin Multiplatform and Kotlin Native plugins, which allow you to create a cross-platform project, write common code for all platforms, and specific for individual ones. This programming language and tools also have a powerful programming environment that supports building and debugging that combination of solutions.*

Obtained applications are self-sufficient and can run on the most popular platforms like Android, JVM, iOS, Windows, etc.

Keywords: *cross-platform programming, kotlin, gradle.*

Одержано редакцією 27.06.2019 р.
Прийнято до публікації 09.10.2019 р.

ЗМІСТ**СЕКЦІЯ «ПРИКЛАДНА МАТЕМАТИКА»**

Л. В. Батюк, Н. М. Кізілова

ДОСЛІДЖЕННЯ ТЕЧІЙ БІНГАМІВСЬКИХ НАНОРІДИН ПО
МІКРОКАНАЛАХ 4

Б. П. Головня

РАСЧЕТ ЛАМИНАРНО-ТУРБУЛЕНТНОГО ПЕРЕХОДА В
ПОГРАНИЧНОМ СЛОЕ НА ПЛОСКОЙ ПЛАСТИНЕ, ОБДУВАЕМОЙ
ТУРБУЛИЗИРОВАННЫМ ПОТОКОМ 11

В. М. Соловйов, О. А. Сердюк

МОДЕЛІ ЗАСТОСУВАННЯ ЕНТРОПІЇ ПЕРІОДУ РЕКУРЕНТНОСТІ ТА
ЕНТРОПІЇ РЕКУРЕННОСТІ ДО ДОСЛІДЖЕННЯ ДИНАМІКИ СКЛАДНИХ
СИСТЕМ 20

Д. В. Черкас, Н. О. Красношлик

ДОСЛІДЖЕННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ РОЗВ'ЯЗУВАННЯ ЗАДАЧ
ОПТИМІЗАЦІЇ 34

Б. П. Головня

К ВОПРОСУ О ПРЕПОДАВАНИИ МЕТОДА СОПРЯЖЕННЫХ
ГРАДИЕНТОВ 43

СЕКЦІЯ «ІНФОРМАТИКА»

Я. Д. Гонтаренко, Н. О. Красношлик

ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ДІЙ
ЛЮДИНИ ПО ВІДЕО 59

М. О. Пасічний

ДОСЛІДЖЕННЯ КРИВИХ ФАЗОВОЇ РІВНОВАГИ БІНАРНИХ ГЦК-
СПЛАВІВ З ОБМЕЖЕНОЮ РОЗЧИННІСТЮ КОМПОНЕНТІВ
РЕШТКОВИМИ МЕТОДАМИ МОНТЕ-КАРЛО 72

О. В. Піскун

АНАЛІЗ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ ЗАДАЧІ
БІНАРНОЇ КЛАСИФІКАЦІЇ

86

Д. Г. Ужва

МЕТОД ОРГАНІЗАЦІЇ КРОСПЛАТФОРМЕННОГО ПРОЕКТУ

96

CONTENTS**APPLIED MATHEMATICS SECTION**

L. V. Batyuk, N. M. Kizilova

INVESTIGATION OF FLOWS OF BINGHAM FLUIDS THROUGH THE
MICROCHANNELS 4

B. P. Golovnya

CALCULATION OF A LAMINAR-TURBULENT TRANSITION IN A BOUNDARY
LAYER ON A FLAT PLATE BLOWED BY A TURBULIZED FLOW 11

V. N. Soloviev, O. A. Serdiuk

THE MODELS OF APPLICATION THE RECURRENCE ENTROPY AND
RECURRENCE PERIOD DENSITY ENTROPY TO THE ANALYSIS OF COMPLEX
SYSTEMS DYNAMICS 20

D. V. Cherkas, N. O. Krasnoshlyk

RESEARCH OF GENETIC ALGORITHMS OF SOLUTION OF OPTIMIZATION
PROBLEMS 34

B. P. Golovnya

ON THE TEACHING OF THE CONJUGATE GRADIENTS METHOD 43

INFORMATICS SECTION

Y. D. Hontarenko, N. O. Krasnoshlyk

HUMAN ACTION RECOGNITION FROM VIDEOS USING NEURAL
NETWORKS 59

M. O. Pasychnyy

MONTE-CARLO INVESTIGATION OF PHASE EQUILIBRIUM CURVES OF
BINARY FCC ALLOYS WITH LIMITED SOLUBILITY 72

O. V. Piskun

ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR THE BINARY
CLASSIFICATION PROBLEM 86

D. G. Uzhva

METHOD OF ORGANIZATION OF THE CROSS-PLATFORM PROJECT

96

ВІСНИК ЧЕРКАСЬКОГО УНІВЕРСИТЕТУ

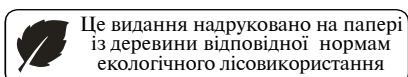
Серія Прикладна математика. Інформатика
№2.2019

Відповідальний за випуск
Пасічний М.О., Головня Б.П.

Відповідальний секретар
Сердюк О.А.

Комп'ютерне верстання
Сердюк О.А.

Підписано до друку 05.11.2019.
Формат 60x84/8. Папір офсет. Гарнітура Times.
Ум. др. арк 12,56. Наклад 300 прим.



Надруковано ФОП Гордієнко Є.І.

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виготовників і
розповсюджувачів видавничої продукції

Серія ДК № 4518 від 04.04.2013 р.

Україна, 18000, м. Черкаси

тел./факс: (0472) 56-56-12, (067) 444-28-94

e-mail: book.druk@gmail.com