

ІНТЕРАКТИВНИЙ ПРОГРАМНИЙ ЗАСІБ ВІЗУАЛІЗАЦІЇ ДАНИХ

Mykola Kolodnytsky ¹⁾, Андрій Ковальчук ²⁾

1) Natural Interactive Systems Laboratory (NISLab), University of Southern Denmark, Science Park 10, DK-5230 Odense M, Denmark. mykola@nis.sdu.dk

2) Кафедра Програмного Забезпечення Обчислювальної Техніки, Житомирський Інженерно-Технологічний Інститут, Черняхівського 103, Житомир 10005, Україна. sekito@ziet.zhitomir.ua

Анотація: В даній роботі представлено короткий огляд існуючих програмних засобів візуалізації даних, розглянуто певні особливості програм даного виду. Представлено інтерактивний та адаптивний інтерфейс користувача для програмного інструментального засобу візуалізації наукових даних. Запропонований до розгляду засіб був спроектований та розроблений авторами та має назву “Graph Server”. Представлено приклади інтерфейсу та деякі приклади 2D та 3D візуалізації.

Ключові слова: програмний засіб, візуалізація даних, адаптивний та інтерактивний інтерфейс користувача.

1. ВСТУП

В наш час існує велика кількість програмних засобів для наукових обчислень та візуалізації даних. Область використання таких програмних засобів дуже різноманітна: навчальний процес, наукові дослідження, промисловість тощо. Все існуюче в даній предметній області програмне забезпечення може бути розділене на універсальне: MathLab, Matematica, PV-WAVE, Statistica, TK Solver, SPSS тощо. [1–7] та спеціалізоване: Application Visualization System (AVS) [8], Khoros [9], IBM Data Explorer [10], IRIS Explorer [11], (Рис. 1–4). Перераховані універсальні програмні системи, що являються засобами чисельного аналізу та моделювання (MathLab, Matematica тощо), як правило мають вбудовані інструменти для візуалізації даних. Використання потужних спеціалізованих засобів ускладнюється через їх автономність та не пристосованість до інтеграції в інші програмні системи.

Однак, слід відзначити, що використання існуючих інструментальних засобів візуалізації даних вимагає від користувача великих та, іноді не потрібних, зусиль. Як правило, користувачеві доводиться вивчати спеціалізовану мову програмування, що використовується в програмному засобі, або досить оригінальний та не стандартний інтерфейс користувача. З нашої точки зору підхід до створення програмного забезпечення, а саме системи візуалізації даних, що вимагає від користувача непотрібних зусиль, є не раціональним та не життєздатним.

Сучасні комп'ютерні інформаційні технології висувають також нові вимоги до програмних засобів, в яких інтерфейс користувача повинен буде інтелектуальним та дружнім. Серед таких вимог в першу чергу потрібно відзначити необхідність використання: MS Active Technologies, методів динамічного обміну даними DDE та Automation, багатопоточних обчислень, сучасних графічних бібліотек тощо.

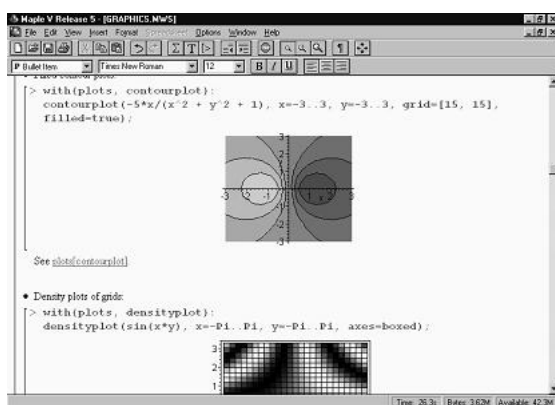


Рис. 1. Maple V

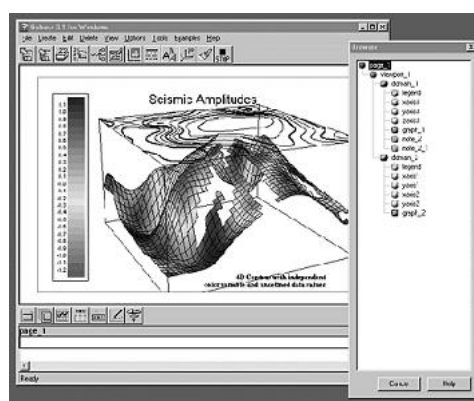


Рис. 2. AVS

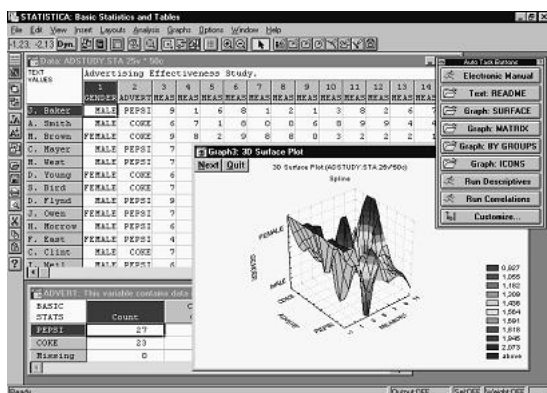


Рис. 3. STATISTICA

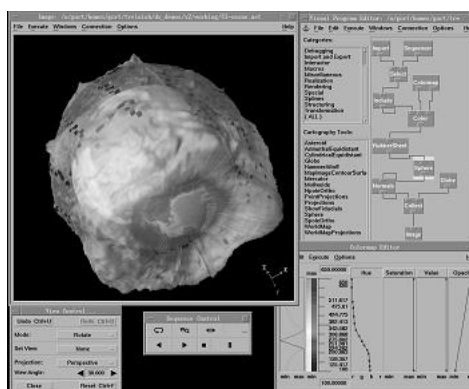


Рис. 4. IBM Data Explorer

Отже, очевидно є необхідність проектування та створення простого у використанні інструментального засобу, що буде здійснювати попередню обробку та візуалізацію даних. Даний засіб, що має уніфікований та адаптивний інтерфейс користувача, був розроблений в рамках програмного комплексу чисельного аналізу математичних задач “DSR OpenLab 1.0” [12, 13]. Далі ми пропонуємо розглянути організацію інтерактивного інтерфейсу користувача програмного засобу “Graph Server”.

2. ВІЗУАЛІЗАЦІЯ: ОБ’ЄКТНО-ОРІЄНТОВАНИЙ ПІДХІД (ООП)

Візуальне представлення наукових даних залежить від їх певних характеристик, а саме: кількісних характеристик, відомостей про тип даних або спосіб їх обчислення. Розглянемо об’єктно-орієнтовану модель процесу візуалізації результатів чисельного аналізу функцій, для чого спочатку розглянемо перелік та властивості об’єктів, сутностями яких є дані для візуалізації. В результаті чисельного аналізу функцій, як правило, отримуємо впорядковану сукупність елементів (множину відношень), а саме унарні відношення ($n = 1$), бінарні відношення ($n = 2$), тернарні відношення ($n = 3$) та відношення, де кількість елементів $n > 3$. Візуально можуть бути представлені відношення, які можуть бути відображені в одновимірному (1D), двовимірному (2D) або тривимірному (3D) просторі, тобто унарні, бінарні або тернарні. Іноді певний сенс має відображення 4D та 5D даних, в даному випадку використовуються модифікації 3D механізмів відображення.

В основі об’єктно-орієнтованого процесу візуалізації знаходиться абстрактний об’єкт, сутністю якого є впорядковані сукупності елементів: унарні, бінарні або тернарні відно-

шення) Пропонується називати його базовим об’єктом (BaseObject, BO). Створення такого BO є привабливим, в силу можливості узагальнення певних властивостей даних та досягнення інваріантності відносно програмної реалізації конкретних об’єктів. Базовий об’єкт є батьківським для всіх інших об’єктів та вміщує основні віртуальні механізми (методи) створення, знищення, зберігання, зчитування та візуалізації даних, $BOMethod = \{ bod_{cr}, aod_d, bod_{sr}, aod_{ld}, aod_p, aod_{vs} \}$, де cr – створення, d – знищення, sr – збереження, ld – зчитування, t – пертворення, p – обробка, vs – візуалізація об’єкту. Спроекувати реальні об’єкти для візуалізації конкретних видів даних можна завдяки використанню механізмів наслідування та поліморфізму в рамках ООП.

На наступному рівні абстракції даних та властивостей знаходиться об’єкт, що породжується від базового об’єкту і сутностями якого є атрибути візуального відображення; його умовна назва – об’єкт атрибутів (AttributesObject, AO). Об’єкт атрибутів містить множину властивостей візуалізації $AOPropertiers = \{ aop_1, aop_2, \dots, aop_n \}$ та множину віртуальних методів $AOMethod = \{ aod_1, aod_2, \dots, aod_n \}$. Існує декілька різних типів об’єктів породжених від BO, кожен з яких відповідає за візуальне представлення у одновимірному, двовимірному або тривимірному просторі. Використання поліморфних властивостей об’єктів, що породжені від об’єкту атрибутів, дає можливість модифікувати множину методів AOMethod, що певним чином дозволяє адаптувати об’єкти для візуального відображення даних різних типів.

Наступним у рівні наслідування є об’єкт основною функціональністю якого є візуалізація даних, його умовна назва – візуальний об’єкт (Visual Object, VO). Об’єкти даного типу реалізують основні методи візуалізації для різних

видів даних. Тобто, візуальний об'єкт на основі множини віртуальних методів {BOMethod, AOMethod}, успадкованих від батьківського об'єкту, реалізує конкретні методи візуалізації. Основною особливістю візуальних об'єктів є необхідність зберігання не тільки вхідних (оригінальних) даних та їх атрибутів, але й проміжних (додаткових), попередньо оброблених та підготовлених для візуалізації множини даних.

Кожен візуальний об'єкт має множину властивостей та методів {BOMethod, AOMethod}, що використовуються під час візуалізації даних. Для зміни таких властивостей та, як результат зміни візуального представлення об'єкту, існуючий набір віртуальних методів перевизначається для кожного конкретного об'єкту, породженого від об'єкту атрибутів, таблиця 1.

Крім вище згаданих дій над візуальними об'єктами, завдяки використанню ООП, досягнуто можливості трансформувати один тип візуального об'єкту в інший. В результаті зазначеного процесу отримуємо об'єкт з таким самим набором даних, як і до трансформації, але з іншими властивостями, діями та алгоритмами візуалізації. Існують певні обмеження на трансформацію об'єктів, наприклад неможливо трансформувати двовимірний візуальний об'єкт у тривимірний, поверхню в годограф тощо.

3. ОРГАНІЗАЦІЯ УНІФІКОВАНОГО ТА АДАПТИВНОГО ІНТЕРФЕЙСУ КОРИСТУВАЧА

Створення інтерфейсу користувача вимагає проектування та розробки, певної кількості

множин елементів управління, за допомогою якого користувач змінюватиме властивості візуальних об'єктів та впливатиме на процес візуалізації даних. Множини таких елементів управління об'єднуються в групи, які умовно будемо називати панелями управління. Деякі з означених панелей управління призначені для індивідуального використання з об'єктами певного виду, а деякі містять множини елементів управління загального використання та призначені для двох або більше типів об'єктів. У таблиці 2 перераховано панелі управління для множини візуальних об'єктів, що визначені у програмній підсистемі "Graph Server".

Алгоритми управління інтерфейсом підсистеми "Graph Server" динамічно змінюють множину активних панелей інструментів (тобто адаптують інтерфейс) в залежності від видів візуальних об'єктів, з якими вданий момент працює користувач, у відповідності з таблицею 2. Також потрібно відзначити, що в програмній підсистемі, завдяки фіксованому набору інструментів, реалізовано уніфікованість елементів управління процесом візуалізації.

4. ВИКОРИСТАННЯ СУЧАСНИХ ПРОГРАМНИХ ТЕХНОЛОГІЙ

Система візуалізації результатів чисельного аналізу повинна виконувати функції серверу активного документа (Active Document Server), що вимагає використання Document/View архітектури. Об'єкт Document відповідальний за зчитування, зберігання, модифікацію та обробку даних, структура зберігання та обробки

Таблиця 1. Віртуальні методи візуальних об'єктів

Віртуальні методи об'єктів	Види візуального представлення даних							
	1 ⁺	2	3	4	5	6	7	8
Активізація об'єкту	√	√	√	√	√	√	√	√
Зміна властивостей об'єкту	*	*	*	√	√	√	*	√
Створення проєкцій на осі координат	√					√		
Створення проєкцій на координатні площини		√					√	√
Зміна властивостей проєкцій об'єктів	*	*	√	√		*	*	√
Зміна масштабу (відображення об'єкту в індивідуальному або спільному масштабі)	√	√	√	√		√	√	√
Позиціонування об'єкту вздовж осей координат	√	√	√	√		√	√	√
Обертання об'єкту навколо координатних осей		√		√			√	√
Анімаційний режим відображення об'єкту	√	√	√	√		√	√	√
Проріджування великих вхідних множин даних	√	√	√	√	√	√	√	√
Створення копії об'єкту	√	√	√	√	√	√	√	√

* – не всі з перерахованих під означеними номерами властивостей можуть бути змінені, для деяких з цих властивостей існує залежність від математичної природи даних.

⁺ – 1 – 2D годограф; 2 – 3D годограф; 3 – 2D гістограма; 4 – 3D гістограма; 5 – 2D область;

6 – Векторне поле 2D; 7 – Векторне поле 3D; 8 – 3D поверхня.

Таблиця 2. Панелі управління множини візуальних об’єктів.

Панелі управління	Види візуального представлення даних							
	1	2	3	4	5	6	7	8
Загальні властивості об’єкту, рис 5.	√	√	√	√	√	√	√	√
Властивості ліній (2D, 3D), рис. 5.	√	√	√	√	√	√	√	√
Властивості маркерів (2D, 3D), рис. 6	√	√	√	√		√		√
Властивості заливки (2D, 3D), рис 6.					√			√
Обертання, рис 7.		√		√		√	√	√
Масштабування (2D)	√		√		√			
Масштабування (3D), рис 7.		√		√		√	√	√
Позиціювання (2D)	√		√		√			
Позиціювання (3D), рис 8.		√		√		√		√

представлена на рис.7. Візуалізація результатів чисельного аналізу відбувається за наступним сценарієм: множини даних передаються до документу за допомогою методів автоматизації (OLE Automation, реалізовано близько 40 методів автоматизації) і потрапляють до вікна

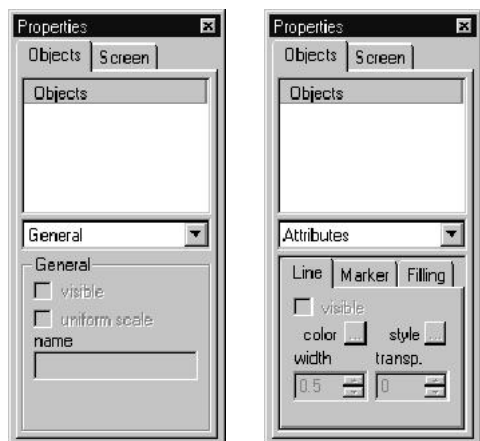


Рис. 5. Панелі інструментів “General” та “Attributes–Lines”

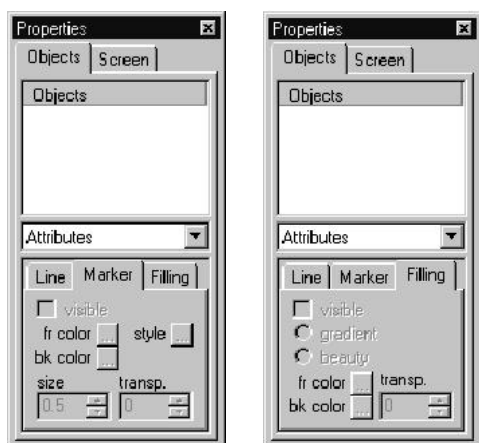


Рис. 6. Панелі інструментів “Attributes–Marker” та “Attributes–Filling”

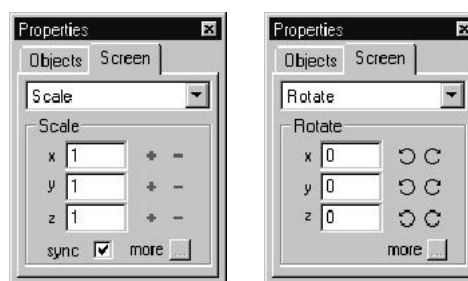


Рис. 6. Панелі інструментів “Screen–Scale” та “Screen–Rotate”

представлення (View), яке в свою чергу, передає їх до вікна екрану (Screen). Вікно-екран за допомогою спеціальної системи повідомлень передає данні до відповідного потоку, який, власне, і відповідає за їх візуалізацію. Кожне вікно представлення може мати кілька вікон екранів (Рис. 7), тобто в одному вікні представлення можуть одночасно відобразитися декілька принципово різних множин даних. Кожен екран (Screen) є контейнером для додаткового потоку та несе відповідальність за його створення, знищення та взаємодію з ним.

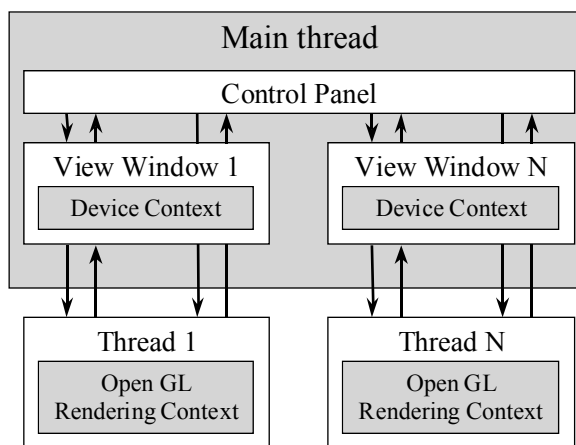


Рис. 7. Реалізація багатопоточності в інструментальному засобі “Graph Server”

Програмний засіб “Graph Server” як інструмент, що використовується для візуалізації даних, має забезпечити користувача максимально повним набором методів модифікації зображення. Однак існують методи, реалізувати які без використання сучасних програмних технологій практично неможливо. До таких методів потрібно віднести збільшення або зменшення масштабу зображення у автоматичному режимі, обертання зображення навколо координатних осей у автоматичному режимі, режим осцилографа, анімація, одночасний процес візуалізації декількох візуальних об’єктів тощо. Вислів “багатопоточна візуалізація даних” у програмній підсистемі “Graph Server” означає, що процес візуалізації різнотипних множин даних відбувається одночасно. Такого результату було досягнуто завдяки використанню додаткових потоків (Thread), які працюють паралельно з головним потоком підсистеми (Main Thread). Для взаємодії між об’єктами, що знаходяться в головному потоці підсистеми та об’єктами з додаткових потоків було розроблено систему обробки міжпотоківих повідомлень, яка має дві схеми роботи з повідомленнями: PostMessage та SendMessage.

Методи візуалізації підсистеми створені на основі використання алгоритмів бібліотеки OpenGL Graphics System, version 1.1., що дає можливість якісно вирішити проблему візуалізації даних (рис. 8).

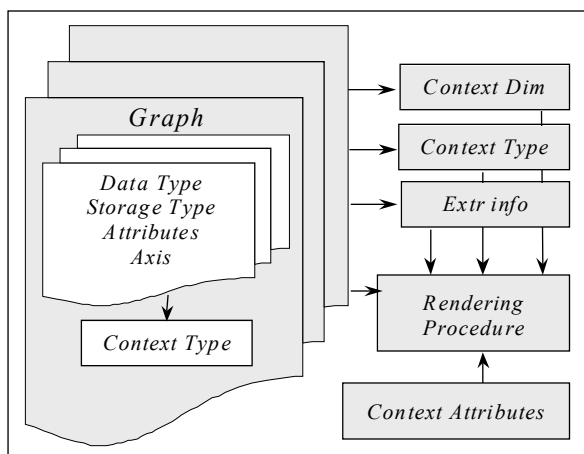


Рис. 8 Схема візуалізації даних

5. ПРИКЛАДИ ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАСОБУ

Далі приведено декілька прикладів інтерфейсу користувача програмного засобу візуалізації даних “Graph Server” (рис. 9–12).

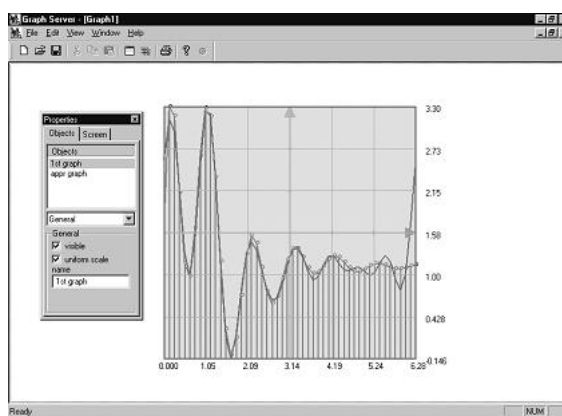


Рис. 9. Відображення прямого та зворотного перетворення Фур’є

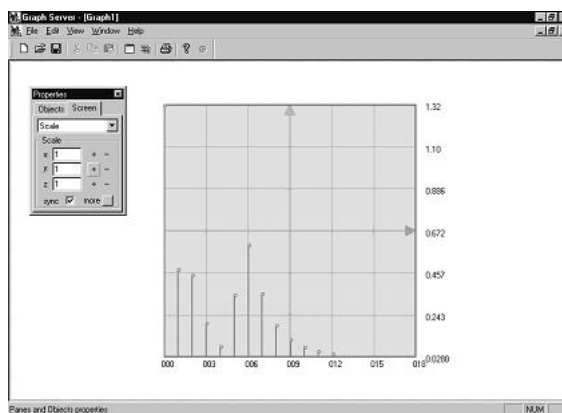


Рис. 10. Фур’є спектр

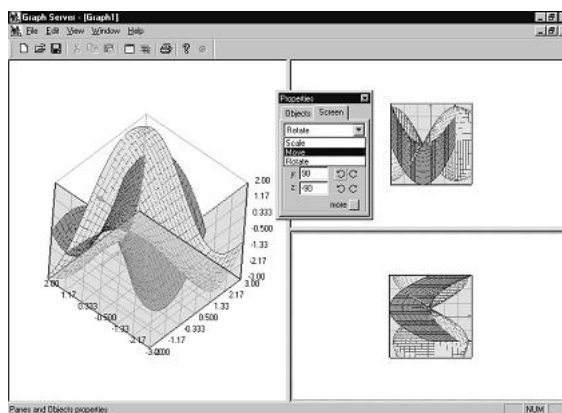


Рис. 11. 3D-візуалізація

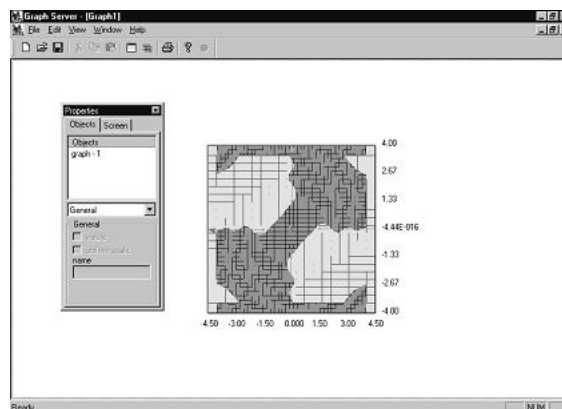


Рис. 12. 2D Область

6. ВИСНОВКИ

Програмний інструментальний засіб “Graph Server” забезпечує досить просте, гнучке та зручне управління процесом візуалізації даних завдяки запропонованій реалізації інтерфейсу користувача. Використання об’єктно-орієнтованої парадигми та сучасних технологій програмування дозволяють розробити зручний адаптивний та уніфікований інтерфейс в програмній системі. Запропоновані приклади роботи інструментального засобу дають можливість оцінити перспективи використання даної системи.

7. ЛІТЕРАТУРА

[1] *MATLAB User's Manual, MATHWORKS Inc, MA, USA, 1999.*

[2] *Gaylord R.J., Kamin S.N. and Wellin P.R. Programming with Mathematica. 2nd Edition, Springer-Verlag, TELOS, Santa Clara, 1996.*

[3] *M.B. Monagan, K.O. Geddes, K. M. Heal, G. Labahn, S.M. Vorkoetter, Maple V Release 5. Learning Guide. Springer-Verlag, 1998, 284 p.*

[4] <http://www.spss.com/>

[5] <http://www.statsoft.com/>

[6] *Melissa Wiggins, An overview of program visualisation tools and systems // Proceedings of the 36th annual conference on Southeast regional conference, 1998, p 194 - 200*

[7] *Kolodnytsky M., Levitsky V. The review of the main mathematical software systems. // Computer-aided Design and Graphics. 1999. Vol. 10. p. 56 - 65.*

[8] <http://www.av.s.com/>

[9] <http://www.khoral.com/>

[10] <http://www.roe.ac.uk/acdwww/vissys/dx.html>

[11] http://www.nag.co.uk/Welcome_IEC.html

[12] *Kolodnytsky M., Kovalchuk A., Kuryata S., Levitsky V. “A Theory of Approximation of Functions” Courseware // Proceedings of the 5th Asian Technology Conference in Mathematics, December 17-21, Chiang Mai, Thailand, 2000. p. 86–95.*

[13] *Kolodnytsky M., Ivanitsky I., Kovalchuk A., Kuryata S., Levitsky V. “DSR Open Lab 1.0” – software system for simulation // The 21st International Conference on Information Technology Interfaces, Pula, Croatia, 1999. p. 31.*



Колодницький Микола Михайлович – професор кафедри комп’ютерних інтегрованих технологій, Житомирського Інженерно-Технологічного Інституту (ЖІТІ), кандидат технічних наук. Розпочав наукову діяльність, навчаючись в Київському політехнічному інституті

(КПІ), Житомирський філіал за спеціальністю “Автоматика та Телемеханіка”. В 1989 р. закінчив аспірантуру КПІ за спеціальністю “Системи Автоматизованного Проектування (САПР) в електроніці”. Дисертаційна робота була присвячена проблемі розробки адаптивної процедури чисельного аналізу динамічних моделей електронних схем. Працюючи потім в ЖІТІ науковим співробітником, старшим викладачем, доцентом кафедри програмного забезпечення, та докторантом, опублікував більше 100 наукових праць, серед яких три підручники (“Технічне та програмне забезпе-

чення КІТ”, 1995; “Елементи теорії САПР складних систем”, 1999; “Основи теорії математичного моделювання систем”, 2001). Приймав участь у ряді міжнародних конференцій: УкрПрог’98, АТСМ’99, Mechanics’2000, ІТІ’21, ASCM’2000 та інших. Під його керівництвом три аспіранти виконали дисертації в області математичного моделювання систем та комп’ютерних інформаційних технологій. Конт акт на адреса: kolod@ziet.zhitomir.ua.